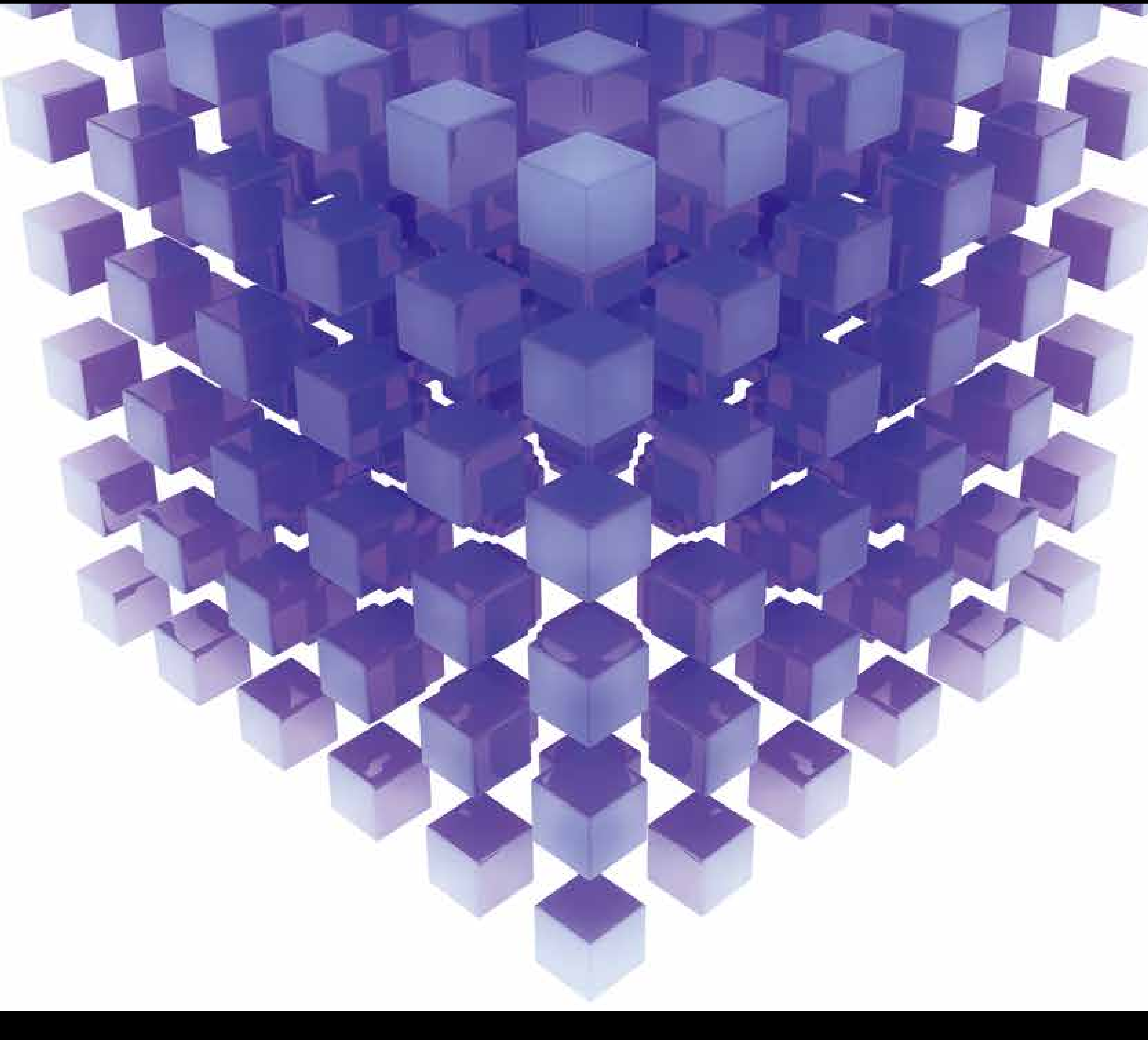
Mathematical Problems in Engineering

Mathematical Tools of Soft Computing 2014

Guest Editors: Ker-Wei Yu, Yang Xu, and Jer-Guang Hsieh



Mathematical Tools of Soft Computing 2014

Mathematical Problems in Engineering

Mathematical Tools of Soft Computing 2014

Guest Editors: Ker-Wei Yu, Yang Xu, and Jer-Guang Hsieh

Editorial Board

Contents

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 234176, 3 pages
http://dx.doi.org/10.1155/2015/234176

Editorial

Mathematical Tools of Soft Computing 2014

Ker-Wei Yu,[1] Yang Xu,[2] and Jer-Guang Hsieh[3]

[1] *Department of Marine Engineering, National Kaohsiung Marine University, Kaohsiung 81157, Taiwan*
[2] *Department of Applied Mathematics, Southwest Jiaotong University, Chengdu, Sichuan 610031, China*
[3] *Department of Electrical Engineering, I-Shou University, Kaohsiung 84001, Taiwan*

Correspondence should be addressed to Ker-Wei Yu; kwyu@webmail.nkmu.edu.tw

Received 27 October 2014; Accepted 27 October 2014

It is an important and promising topic that applies the soft computing techniques on engineering problems. Many successful applications in diverse fields using the soft computing techniques have been reported. This special issue represents the recent developments of the mathematical tools and algorithms of soft computing to solve formal and practical problems arising from engineering areas. We have collected thirty papers in this special issue, which are summarized as follows.

In the paper "*A hybrid metaheuristic-based approach for the aerodynamic optimization of small hybrid wind turbine rotors*" by J. F. Herbert-Acero et al., the authors present a novel framework for the aerodynamic design and optimization of blades for small horizontal axis wind turbines. Based on the experimental results, improvements in terms of the aerodynamic efficiency, the cut-in wind speed, and the amount of material used during the manufacturing process can be achieved.

The opposition-based learning strategy and free search operator are embedded into the krill herd optimization algorithm to form a new opposition-based free search krill herd optimization algorithm for function learning problems in the paper "*A free search krill herd algorithm for functions optimization*" by L. Li et al. Fourteen benchmark functions were tested. It is claimed that, compared with some other evolutionary computation algorithms, the proposed algorithm shows better performance and robustness.

In the paper "*A hybrid optimized weighted minimum spanning tree for the shortest intrapath selection in wireless sensor network*" by M. Saravanan and M. Madheswaran, a novel hybrid evolutionary algorithm, called bee algorithm-simulated annealing weighted minimal spanning tree routing, is proposed in which randomly deployed sensor nodes are split into the best possible number of independent clusters with cluster head and optimal route. Some numerical simulations are given to illustrate the proposed method.

A novel adaptive elite-based PSO approach is proposed in "*A novel adaptive elite-based particle swarm optimization applied to VAR optimization in electric power systems*" by Y.-Y. Hong et al. A realistic loss minimization problem in an electric power system is studied to show the robustness of the proposed method.

A new method for rigid video object recognition is presented in the paper "*An innovative SIFT-based method for rigid video object recognition*" by J. Yu et al. It is claimed that the new method can improve greatly the completeness of the feature database of the target video object automatically, which in turn increases drastically the ratio of correct recognition, at the expense of more affordable computation time in millisecond level.

In the paper "*Attribute extended algorithm of lattice-valued concept lattice based on congener formal context*" by L. Yang and Y. Xu, the definitions of attribute extended formal context and congener formal context are given. Under some mild condition, necessary and sufficient conditions of forming attribute values are derived. Based on these conditions, the algorithms of generating lattice-valued congener formal context and establishing concept lattice are developed, by which one can construct algorithms for lattice-valued concept lattices in distributed and parallel systems.

The paper entitled "*Comparing the selected transfer functions and local optimization methods for neural network flood runoff forecast*" by P. Maca et al. analyzes the influence of the selection of the transfer functions and the training algorithms on neural network for flood runoff forecast. Nine of the most significant flood events, caused by the extreme rainfall, were selected from 10 years of measurements on small headwater catchment in the Czech Republic, and flood runoff forecast was investigated using the extensive set of multilayer perceptrons with one hidden layer of neurons.

A two-dimensional continuous genetic algorithm is proposed to numerically solve the Stokes and nonlinear Navier Stokes problems in "*Continuous genetic algorithm as a novel solver for Stokes and nonlinear Navier Stokes problems*" by Z. S. Abo-Hammour et al. The simulation results showed that the proposed continuous genetic algorithm has an excellent accuracy as compared to some other numerical solvers.

The paper "*Dealing with nonregular shapes packing*" by B. A. Júnior et al. addresses the irregular strip packing problem, a particular two-dimensional cutting and packing problem in which convex and nonconvex shapes (polygons) have to be packed onto a single rectangular object. An approach is proposed that prescribes the integration of a metaheuristic engine (i.e., genetic algorithm) and a placement rule (i.e., greedy bottom-left rule). Computational experiments performed on standard benchmark problems, as well as practical case studies developed in the ambit of a large textile industry, are reported and discussed.

A new derivative-based trapezoid rule for the Riemann-Stieltjes integral is proposed in the paper "*Derivative-based trapezoid rule for the Riemann-Stieltjes integral*" by W. Zhao and Z. Zhang. This new rule may increase two orders of precision over the trapezoid rule for the Riemann-Stieltjes integral. The rationality of the generalization of derivative-based trapezoid rule for Riemann-Stieltjes integral is also demonstrated.

Network-based control systems have become emerging technologies in the control of nonlinear systems over the past few years. The paper entitled "*Discrete globalised dual heuristic dynamic programming in control of the two-wheeled mobile robot*" by M. Szuster and Z. Hendzel focuses on the implementation of the approximate dynamic programming algorithm in the network-based tracking control system of the two-wheeled mobile robot, Pioneer 2-DX. The proposed discrete tracking control system consists of the globalised dual heuristic dynamic programming algorithm, the PD controller, the supervisory term, and an additional control signal. The performance of the proposed control system was verified by a series of computer simulations and experiments realised using the wheeled mobile robot Pioneer 2-DX.

Energy consumption has become a more and more important issue in computer systems. In the paper "*Energy-aware real-time task scheduling for heterogeneous multiprocessors with particle swarm optimization algorithm*" by W. Zhang et al., the authors first formulate and describe the energy-aware real-time task scheduling problem in heterogeneous multiprocessors. Then they propose a particle swarm optimization based algorithm, which can successfully reduce the energy cost and the time for searching for feasible solutions. Some experimental results are provided.

In the paper "*Expert system for competences evaluation 360° feedback using fuzzy logic*" by A. A. A. Lasserre et al., the authors propose an expert performance evaluation system based on a fuzzy logic model, with competences 360° feedback oriented to human behavior. The model was validated in the administrative department of a real Mexican manufacturing company. The experimental results showed that the proposed method has better performance over some other traditional 360° performance evaluation methods.

In the paper "*Extracting credible dependencies for averaged one-dependence estimator analysis*" by L. Wang et al., the authors present a new type of semi-Naive Bayesian operation, which selects superparent attributes by building maximum weighted spanning tree and removes highly correlated children attributes by functional dependency and canonical cover analysis. Some experimental comparisons are made on UCI datasets.

Surrogate models can be used, often in conjunction with evolutionary algorithms, in optimization when it is expensive to test candidate solutions. In the paper "*Geometric generalisation of surrogate model-based optimisation to combinatorial and program spaces*" by Y.-H. Kim et al., it is shown how radial basis functions can provide a generalised surrogate model for combinatorial problems which have a geometric solution representation, through the conversion of that representation to a different metric space.

In the paper "*Hybrid soft computing schemes for the prediction of import demand of crude oil in Taiwan*" by Y. E. Shao et al., single-stage and two-stage hybrid forecasting models are proposed for the prediction of imported crude oil in Taiwan. Real data set of crude oil in Taiwan over the period of 1993–2010 was tested.

In recent years, grammatical evolution has been used as a representation of genetic programming, which has successfully been applied in many optimization problems. In the paper "*Improving the bin packing heuristic through grammatical evolution based on swarm intelligence*" by M. A. Sotelo-Figueroa et al., the authors propose using particle evolutionary swarm optimization (PESO) and particle swarm optimization (PSO) in the frame of grammatical evolution as strategies to generate heuristics that solve the bin packing problems.

The supply chain is a major issue in a global market. In the paper "*Methodology for supply chain integration: a case study in the artisan industry of footwear*" by J. R. Pérez-Gallardo et al., a methodology is proposed for integrating the supply chain that can be applied in any type of business. To demonstrate its relevance, a case study is performed at a handmade shoe company.

Due to the fact that the bit reversal process requires considerable processing time of the fast Fourier transformation (FFT) algorithm, it is vital to optimize the bit reversal algorithm (BRA). An efficient BRA with multiple memory structures is proposed in the paper "*Multiple memory structure bit reversal algorithm based on recursive patterns of bit reversal permutation*" by K. K. L. B. Adikaram et al. Some numerical examples are given to illustrate the proposed method.

The decision support systems are usually created to help decision makers to make complex decisions. In the paper "*OrclassWeb: a tool based on the classification methodology ORCLASS from verbal decision analysis framework*" by T. C. S. Machado et al., a new tool, called OrclassWeb, is developed to reproduce the procedure to apply the verbal decision analysis methodology ORCLASS.

A hesitant fuzzy linguistic term set (HFLTS) is very useful to express people's hesitancy in practical decision making problems. In the paper "*Pairwise comparison and distance measure of hesitant fuzzy linguistic term sets*" by H.-C. Huang and X. Yang, the authors present a comparison method for HFLTSs based on pairwise comparisons of each linguistic term in the two HFLTSs. Then, a distance measure method based on the pairwise comparison matrix of HFLTSs is proposed, and this distance is proved to be equal to the distance of the average values of HFLTSs, which makes the distance measure much more simple. Finally, the pairwise comparison and distance measure methods are utilized to develop two multicriteria decision making approaches under hesitant fuzzy linguistic environments.

Tourist distribution, a vector reflecting the tourist number of scenic spots in a certain period of time, serves as the foundation for a scenic spot manager to make a schedule scheme. In the paper "*Real-time forecast of tourists distribution based on the improved k-means method*" by P. Ren et al., a forecast model based on improved k-means method is provided to forecast tourist distribution. A case study of Jiuzhai Valley is adopted to illustrate the effectiveness of the proposed forecast model.

The trajectory optimization problem subject to terminal impact time and angle specifications is investigated in the paper "*Receding horizon trajectory optimization with terminal impact specifications*" by L. Zhang et al. A receding horizon optimization strategy is implemented to reject the errors caused by the motion of a surface target. Several simulations were performed to validate the proposed method.

In the paper "*The lattice-valued Turing machines and the lattice-valued type 0 grammars*" by J. Tang et al., the authors introduce a new concept of the l-valued Turing machine. It can be used to characterize recognition, natural language processing, and dynamic characteristics.

The global warming will lead to rising temperature in Tibetan plateau, and it has severe effects to the long-term stability of frozen soil roadbed. In the paper "*Simulation analysis of the long-term stability for frozen soil roadbed*" by X. Liu et al., a numerical simulation model is established for the frozen soil roadbed in Tibetan plateau.

The Triple I method for the model of intuitionistic fuzzy modus tollens (IFMT) satisfies the property of local reductivity but not the reductivity. In order to improve the quality of the Triple I method for the lack of reductivity, a new approximate reasoning method is proposed for IFMT problem in the paper "*The dual Triple I methods of FMT and IFMT*" by L. Yan and Z. Mucong.

In the paper "*A MOLP method for solving fully fuzzy linear programming with LR fuzzy parameters*" by X.-P. Yang et al., a new class of fully fuzzy linear programming (FFLP) problems with LR fuzzy parameters are studied. The FFLP is converted into a multiobjective linear programming (MOLP) according to the order relation for comparing the LR at fuzzy numbers. Moreover, the classical fuzzy programming method is modified and used to solve the MOLP problems. A numerical example is given to illustrate the feasibility of the proposed method.

In the paper "*A multiobjective optimization algorithm based on discrete bacterial colony chemotaxis*" by Z. Lu et al., a discrete bacterial colony chemotaxis algorithm is proposed to solve multiobjective optimization problems. Some numerical results are given to demonstrate the effectiveness of the proposed method.

A prime objective in constructing data streaming mining models is to achieve good accuracy, fast learning, robustness to noise, and a compact tree size. In the paper "*A lightweight data preprocessing strategy with fast contradiction analysis for incremental classifier learning*" by S. Fong et al., a new data preprocessing strategy is proposed which is suitable for the progressive purging of noisy data from the training dataset without the need to process the whole dataset at one time. This strategy is tested via a computer simulation to show the significant benefit of allowing for the dynamic removal of bad records in the incremental classifier learning process.

In the paper "*Multisegment scheme applications to modified Chebyshev Picard Iteration method for highly elliptical orbits*" by D. Kim et al., a modified Cheyshev Picard iteration method is proposed for solving orbit propagation initial/boundary value problems. Cosine sampling techniques are used to reduce Runge's phenomenon that plagues many series approximations. The authors suggest a multisegment method to obtain accurate solutions overall regardless of initial positions and eccentricity by dividing the given orbit into two or more segments.

Acknowledgments

We wish to express our sincere appreciation to the authors for their excellent contributions. The hard work of all reviewers is greatly acknowledged. Without them, this special issue can never become possible.

Ker-Wei Yu
Yang Xu
Jer-Guang Hsieh

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 125781, 11 pages
http://dx.doi.org/10.1155/2015/125781

Research Article

A Lightweight Data Preprocessing Strategy with Fast Contradiction Analysis for Incremental Classifier Learning

Simon Fong,[1] Robert P. Biuk-Aghai,[1] Yain-whar Si,[1] and Bee Wah Yap[2]

[1] *Department of Computer and Information Science, University of Macau, Macau*
[2] *Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia*

Correspondence should be addressed to Simon Fong; ccfong@umac.mo

Received 4 February 2014; Accepted 26 July 2014

Academic Editor: Yang Xu

A prime objective in constructing data streaming mining models is to achieve good accuracy, fast learning, and robustness to noise. Although many techniques have been proposed in the past, efforts to improve the accuracy of classification models have been somewhat disparate. These techniques include, but are not limited to, feature selection, dimensionality reduction, and the removal of noise from training data. One limitation common to all of these techniques is the assumption that the full training dataset must be applied. Although this has been effective for traditional batch training, it may not be practical for incremental classifier learning, also known as data stream mining, where only a single pass of the data stream is seen at a time. Because data streams can amount to infinity and the so-called big data phenomenon, the data preprocessing time must be kept to a minimum. This paper introduces a new data preprocessing strategy suitable for the progressive purging of noisy data from the training dataset without the need to process the whole dataset at one time. This strategy is shown via a computer simulation to provide the significant benefit of allowing for the dynamic removal of bad records from the incremental classifier learning process.

1. Introduction

Data preprocessing has traditionally referred to cleaning up the training dataset prior to sending it to the classifier construction learning process. Many researchers advocate it as an indispensable step in the data mining process because it helps to get rid of "bad" data and/or attributes (features), thus supporting a compact degree of dimensionality and enhancing the accuracy of the learned model. While feature selection is undertaken to prune redundant attributes (from the columns of a 2D dataset matrix), removing records that do not contribute to model learning is another common approach taken to maintain the quality of the learned model in terms of classification accuracy. Though it is known that the so-called "bad" records in the training dataset will subsequently affect the performance of the learned model, it is sometimes impossible to tell which records contain "bad" data until they are later reflected by a decline in the performance of the learned model, which will clearly become apparent at too late a stage, especially in critical applications.

Most engineering applications nowadays require harvesting useful insights from the so-called big data which are not only large in volume but fresh data are generated and accumulated continuously. Some typical engineering applications are vital health signs monitoring, genes analysis in biomedical engineering, real-time traffic monitoring in civil engineering, real-time fraud detection in security and data engineering, and so forth. A learnt data mining model needs to be continuously updated and refreshed whenever new records arrive. Typically, data stream mining methods that demand for both accuracy and speed in quick model induction would be useful in these applications. Although many papers can be found in literature about the learning mechanisms of incremental learning in various data stream mining models, the preprocessing techniques for incremental learning are relatively less explored. This paper aims at shedding some light on a lightweight preprocessing method for incremental classifier learning.

In data preprocessing with an objective of cleaning the training data from noise prior to model induction, it is not easy to identify data instances as bad or good without any external judgment or intervention. This is analogous to the paradox that arises when a majority of data is wrong: the classifier model will take them as "normal." A pressing challenge facing the data mining community is how to pinpoint such noisy data when they are part of a normal statistical distribution and do not clearly fall into the categories of extreme values or outliers. In comparison, handling missing values is a relatively trivial preprocessing issue as the rows that contain any data on blank fields can be deleted or blank field scan can be filled artificially with guessed data [1].

Tagging data as noisy is a challenge when *a priori* knowledge is unavailable. One way of doing so is to compare a given dataset with the normal distribution of values for such data to determine whether it is normal or rare. This may not be a favorable approach: as it normalizes all of the data, the learning model resulting from the training process will lose the ability to make specific predictions and will be blunted by generality. The other technique commonly applied is to conduct a trial classification/test on data taken from the training dataset and check whether they lead to a satisfactory level of classification accuracy. If not, the data are deemed to be bad as they have failed the trait test; retaining them in the training dataset would only lead to a subsequent deterioration in training accuracy. Hence, the data should be purged, based on the principle that good data contribute to constructing a reliable classifier, whereas bad data only confuse the underlying cognition of the classifier model as the model is learnt in a manner similar to pattern recognition among samples from the training dataset.

In Weka, a popular collection of machine learning algorithms used for data mining with open source machine learning software, a filter function called *RemoveMisclassified()* (http://weka.wikispaces.com/Removing+misclassified+instances+from+dataset) has been made available to preemptively remove from the training dataset a subset of data that could reduce the accuracy of the training model. The function however has been developed years ago by the Weka research team and made into a standard function. Recently, it has been used by a data filtering model called PRISM (preprocessing instances that should be misclassified) [2], which eliminates certain instances from the original dataset. They are called "should be misclassified" because the filter function predicts in advance that these instances will likely be misclassified by inducing a preliminary classification model prior to the actual model induction. However, the function doubles the total model learning time at worst (involving a test run of one round of classification to filter out instances that have been misclassified + the actual building of the model based only on correctly classified instances). Therefore, it may not be an elegant solution if a user opts for a lightweight data preprocessing step that does not have to take as much time as building one dummy classifier in advance to distinguish between good and bad data. This may work for batch type data mining, but not for incremental data mining.

Section 2 outlines the contributions other researchers have made for the detection of noise in training data; few techniques have been put forward to distinguish bad data from good data. However, those techniques are all principle-based solutions, and they are likely to share the same shortcoming when it comes to data stream mining. All of these techniques need a reference model to distinguish between two groups of data; the first comprises instances useful for building a good model, and the second consists of noisy data that would not contribute to the construction of a sound model. The problem common to all of these proposed techniques is the need to use the full dataset to establish such a reference model. The decoyed classifier generated by the *RemoveMisclassified()* function, which requires the full data cache and enough time to divide up the samples, is exactly this type of reference model.

This paper proposes a novel lightweight preprocessing strategy that takes the optimal amount of time to produce a trimmed/cleansed training dataset enabling the training model to yield the highest possible degree of accuracy.

Holistically, this preprocessing strategy allows a training model to achieve a good level of accuracy with a short preprocessing time and a compact decision tree (or the minimal rules that are sufficiently significant). In practical terms, these benefits translate into a very fast process for filtering noisy instances and training a relatively accurate model at the same time. This is important in enabling incremental learning where the preprocessing component is quick and effective. The compact decision tree means that only useful classification rules are generated; technically, this means that stringent memory constraints could be met in real-time applications.

The rest of the paper is structured as follows. Section 2 highlights some commonly adopted techniques that other researchers have proposed for the detection and removal of noise from training datasets. Section 3 describes our new lightweight methodology, which includes a workflow, a calibration function for finding the optimal settings, and the details of the "contradiction analysis" mechanism used for removing misclassified instances. Section 4 reports an experiment conducted to validate our strategy. Section 5 concludes the paper.

2. Related Work

Numerous researchers have attempted to apply different techniques in detecting and removing noisy data, which are generally referred to as incorrect or erroneous data instances. These techniques all center on the common observation of how such data instances disrupt training data and thwart the pursuit of classification accuracy. In other words, they generally focus on the observation of data irregularities, how each data instance relates to the others (the majority), and how such data instances relate to classification performance. Most of these techniques can be grouped into the following three categories: statistics-based, similarity-based, and classification-based methods.

2.1. Statistics-Based Noise Detection Methods. Outliers, or data with extraordinary values, are interpreted as noise in this kind of method. Detection techniques proposed in the literature range from finding extreme values beyond a certain number of standard deviations to complex normality tests. Comprehensive surveys of outlier detection methods used to identify noise in preprocessing can be found in [3, 4]. In [5], the authors adopted a special outlier detection approach in which the behavior projected by the dataset is checked. If a point is sparse in a lower low-dimensional projection, the data it represents are deemed abnormal and are removed. Brute force, or, at best, some form of heuristics, is used to determine the projections.

A similar method outlined by [6] builds a height-balanced tree containing clustering features on nonleaf nodes and leaf nodes. Leaf nodes with a low density are then considered outliers and are filtered out.

Recently, there are some new developments founded on sophisticated statistics models too, such as outlier detection in adaptive functional-coefficient autoregressive models based on Extreme Value Theory [7]. Extending from such statistics computation, the outlier detection evolves into detecting outliers in time-series of univariate data. It has a wide range of applications like detecting outliers from astronautical time series data [8], recovering outliers from traffic volume data using Tensor model [9], and finding outliers from a time series records of available parking spaces [10].

2.2. Similarity-Based Noise Detection Methods. This group of methods generally requires a reference by which data are compared to measure how similar or dissimilar they are to the reference.

In [11], the researchers first divided data into many subsets before searching for the subset that would cause the greatest reduction in dissimilarity within the training dataset if removed. The dissimilarity function can be any function returning a low value between similar elements and a high value between dissimilar elements, such as variance. However, the authors remarked that it is difficult to find a universal dissimilarity function.

Xiong et al. [12] proposed the Hcleaner technique applied through a hyperclique-based data cleaner. Every pair of objects in a hyperclique pattern has a high level of similarity related to the strength of the relationship between two instances. The Hcleaner filters out instances excluded from any hyperclique pattern as noise.

Another team of researchers [13] applied a k-NN algorithm, which essentially compares test data with neighboring data to determine whether they are outliers by reference to their neighbors. By using their nearest neighbors as references, different data are treated as incorrectly classified instances and are removed. The authors studied patterns of behavior among data to formulate Wilson's editing approach, a set of rules that automatically select the data to be purged.

This group of methods that are based on similarity measure is sometimes known as distance-based approach. They are also sometimes known as noise reduction algorithms [14] as they aim at detecting outliers from normal data. There are several classical contributions such as local outlier factor (LOF) [15]; a recent work on using LOF for detecting outliers in data stream mining is presented in [16]. Some variants of outlier distance based algorithm are called Enhanced Class Outlier Distance Based Algorithm (ECODB) [17] and repeated edited nearest neighbor (RENN) [18]. Some latest works reported in 2014 include angle based outlier detection method in preprocessing [19], automatic outlier detection using inter-quartile-range [20], and a deviation-based approach for real-time wireless sensor network data [21], just to name a few.

2.3. Classification-Based Noise Detection Methods. Classification-based methods are those that rely on one or more preliminary classifiers built as references for deciding which data instances are incorrectly classified and should be removed.

In [22], the authors used an n-fold cross-validation approach to identify mislabeled instances. In this technique, the dataset is partitioned into n subsets. For each of the n subsets, m classifiers are trained on the instances in the other $n-1$ subsets and the instances in the excluded subset are classified. Each classifier tags an instance as misclassified if it is classified incorrectly. Majority voting or a consensus approach can be used in the filtering process.

Another team of researchers [23] presented a robust decision tree method for the removal of outliers. In this method, a pruning tree is built on the training dataset and is used to classify the training data. Instances the pruned tree classifies incorrectly are removed from the training dataset. These processes are repeated until the pruned tree correctly classifies all instances in the training dataset.

In the study reported in [24], the researchers innovatively used a genetic algorithm (GA) to create a set of suspicious noisy instances and select a prototype to identify the set of actual noisy instances. The fitness function of the GA is a generic classifier built in advance, and the GA uses it to search heuristically for misclassified instances.

There are recently other similar contributions which centered on the idea of building a preliminary classifier in advance; from the performance of the early model, the user judges which data instances are deemed to be removed prior to constructing the main classification model. These methods have this principle in common, though the underlying algorithms that they use may vary, such as [2, 25]. Nevertheless, they were designed for traditional data mining where a full dataset needs to be loaded in for batch learning, rather than for incremental learning.

3. Our Proposed Incremental Preprocessing Model (IPPM)

All of the techniques reviewed above were designed for preprocessing in batch mode, which requires a full set of data to determine which instances are to be deleted.

We argue that, for real-time data stream mining, traditional preprocessing approaches may cause a reduction in performance simply due to the different nature of operations and the requirements of a lightweight mechanism. The unique data preprocessing and model learning approach proposed here are different from all those outlined in Section 2.

Preprocessing has traditionally been seen as a standalone step which takes place before model learning starts. The dataset is fully screened at least once to determine which instances should be removed because they would cause misclassification at a later stage. The enhanced training set, which is usually a subset of the original dataset, is then fed

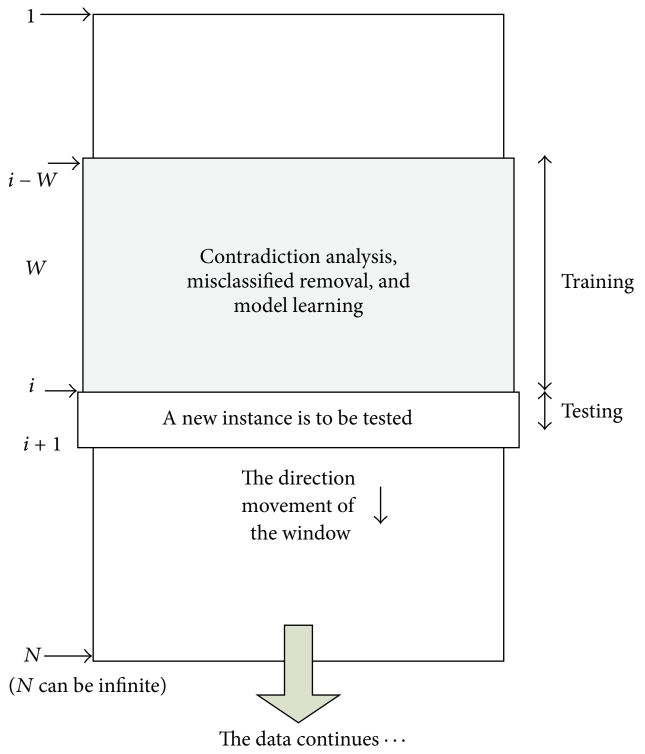

1
i − W
W
Contradiction analysis, misclassified removal, and model learning
Training
i
A new instance is to be tested
Testing
i + 1
The direction movement of the window
N
(N can be infinite)
The data continues · · ·

Figure 1: Illustration of how the IPPM works.

into the learning process in the hope that it will facilitate noise-free learning.

In contrast, our incremental preprocessing model (IPPM) is embedded in the incremental learning process, and all of the steps—noise detection, misclassified data removal, and learning—occur within the same timeframe. In this dual approach, preprocessing and training are followed by testing work as the data stream flows in. By way of illustration, Figure 1 shows a window of size W rolling along the data stream. Within the window, the data are first subject to contradiction analysis (for noise detection), then to misclassified data removal and training (model building).

After the model is duly trained, incoming instances are tested. Because this allows for intermediate performance results to be obtained, the average performance level can also be calculated at the end of the process based on the overall performance results.

3.1. Workflow of the Preprocessing and Incremental Learning Model. The full operational workflow of the IPPM is shown in Figure 2.

Both preprocessing and training occur within the same window, which slides along the data stream from the beginning and is unlikely to require all available data. In data mining jargon, this is an anytime method, which means the model is ready to use (for testing) at any time. Whenever new data come in, the window progressively covers the new instances and fades out the old (outdated) instances, and as the analysis kicks in again, the model is updated incrementally in real time.

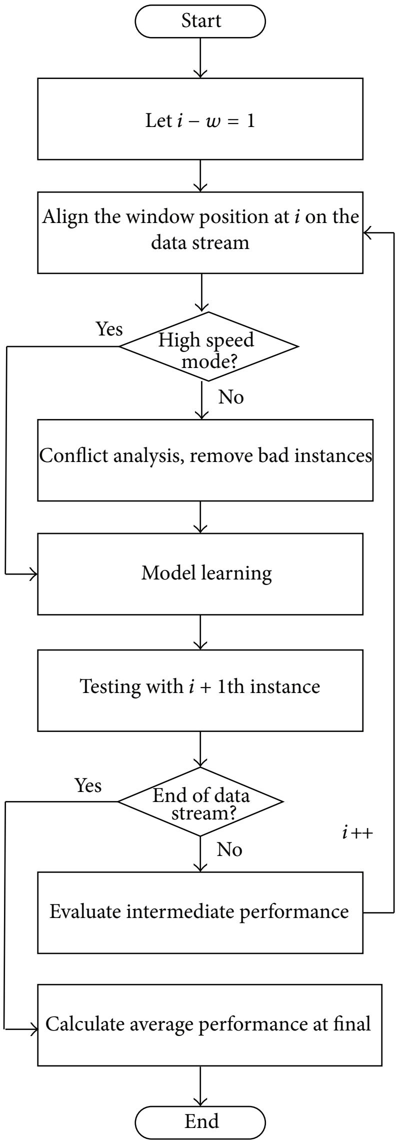

Start
Let i − w = 1
Align the window position at i on the data stream
Yes
High speed mode?
No
Conflict analysis, remove bad instances
Model learning
Testing with i + 1th instance
Yes
End of data stream?
No
i++
Evaluate intermediate performance
Calculate average performance at final
End

Figure 2: Workflow of the incremental learning strategy.

Through this approach, there is no need to assume the dataset is static and bounded, and the advantages of removing misclassified instances persist. Each time the window moves on to fresh data, the training dataset framed in the window W is enhanced and the model incrementally learns from the inclusion of fresh data within W. Another benefit of the proposed approach is that the statistics retained by the rolling window W can be cumulative. By accumulating statistics on the contradiction analysis undertaken within each frame of the window as it rolls forward, the characteristics of the data are subtly captured from a long-run global perspective. Contradiction analysis can be improved by employing such global information, and it can possibly become more accurate in recognizing noisy data. In other words, the noise detection function becomes more experienced (by tapping into cumulative statistics) and refined in picking up noise. Noise is, of course, a relative concept, the identification of which requires an established reference.

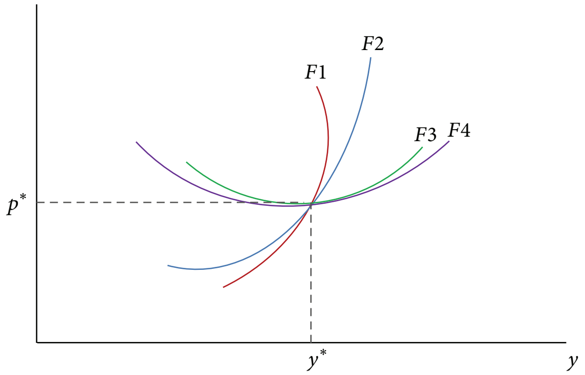


Figure 3: Illustration of optimal p^* and y^* that can be derived among several performance variables.

3.2. Calibration for Optimal Settings. Crucial factors for the IPPM to work well include the choice of window size W and the algorithms used for contradiction analysis and incremental model learning.

There are more than a dozen choices of algorithms, and there is no universal algorithm that fits datasets of all sizes and natures and the theme of the application. For example, a support vector machine, which is relatively fast and has a lower false alarm rate, would be recommended or the rapid detection of malicious attacks in cyber security. For applications where accuracy is more important than speed and relations between attributes and classes are of a complex nonlinear nature, a neural network would be ideal.

For the choice of W, there is always a compromise between time taken and accuracy. In general, the larger the window, the greater the amount of data included in the contradiction analysis and for subsequent enhancement of the training data used for model learning. Hence, accuracy is slightly increased because sufficient data are available. However, what degree of sufficiency is required? How large should the window be to ensure an optimum balance between accuracy and time taken, which is proportional to the amount of data included in W?

Therefore, it is suggested that, before the actual work begins, IPPM operations be initialized with respect to different data and applications. This is why a calibration step is advocated, an exercise that could be used to find the optimal W that produces the performance variables desired. Different algorithms can also be test driven in this step so that the user can gauge the goodness or suitability of each candidate algorithm. Figure 3 shows an example of the optimal point that projects the right W size where all performance variables can be maximized.

The curves $F1, F2, \ldots$, and $F3$ represent factors that influence the values of p^* and y^* simultaneously. The intersection of these curves is an optimum balance by which the corresponding values on the x-axis and y-axis could be mutually maximized without jeopardizing one from another. In our case here, p^* may represent the percentage of performance that could be achieved by our learning model given y^* which is the percentage of the amount of training data. Intuitively, the more the data available for model training, the better the performance that can be yielded. Two factors, time and accuracy, however, are often in opposition—high accuracy usually comes with more data and longer processing time.

In this paper, we adopt a theory of economic order quantity (EOQ), which is also known as the Barabas EOQ model [26], for finding the optimal point that produces a good balance of accuracy and time such that y^* can be deduced from the intersect; hence, the optimal W is obtained.

Given that W is the window size that holds a certain quantity of training samples at a time, W^* is the optimal window size we want to find. N is the total number of samples or instances in the whole dataset. The efficiency gain, E, is defined as the degree of efficiency which is the inverse of the required processing time for processing each instance in average. A, then, is the accuracy cost (loss) that drops per piece of instance that is left out from the training set. Hence, we want to optimize the following:

$$\begin{aligned}\text{Total cost (TC)} &= \text{Data size cost} + \text{time consumption cost}\\ &\quad + \text{accuracy cost},\end{aligned} \tag{1}$$

where data size cost is the variable cost of data which is the data unit cost × the total quantity in the dataset. Time consumption cost is the cost of preprocessing the data; each data has an assumed fixed cost E, and we need to spend N/W rounds of iteration; the time consumption cost is then $E \times N/W$. For the average window size is about $W/2$, the accuracy cost is thus $A \times N/W$. Consider

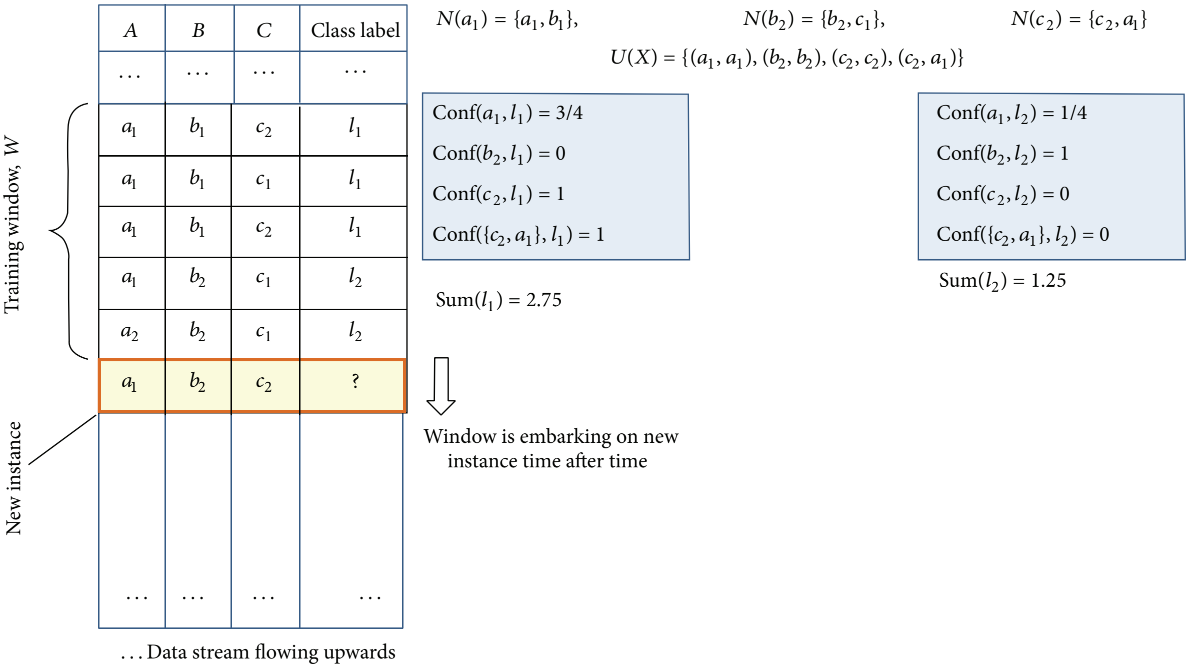
$$\begin{gathered}\text{Hence TC} = N \times D + \frac{N \times E}{W^2} + \frac{A \times W}{2}\\ D = -\frac{N \times E}{W^2} + \frac{A}{2}.\end{gathered} \tag{2}$$

Solving for W gives W^* which is the optimal amount of sample in the window. One has

$$W^2 = \frac{2 \times N \times E}{A}; \quad \text{therefore } W^* = \sqrt{\frac{2 \times N \times E}{A}}. \tag{3}$$

The optimization is not limited to two factors; the equation can be extended to multiple factors such as ROC, precision and recall, $F1$ measure, Kappa's statistic, tree size, and model complexity for a multidimensional optimization.

3.3. Contradiction Analysis. For contradiction analysis, a modified pair-wise based classifier (PWC) is used that is based on the dependencies of the attribute values and the class labels. PWC is similar to instance-based classifier or lazy classifier which only gets activated for testing an instance and incrementally trains at most in one round a classifier when the instance arrives. PWC has several incentives over other methods despite the fact that it is very fast in processing which is a prerequisite for lightweight preprocessing. The advantages include simplicity in merely computing the support and confidence values for estimating which target label one instance should be classified into, no persistent tree structure or trained model needs to be retained except small registers



A	B	C	Class label
...	...	...	...
a_1	b_1	c_2	l_1
a_1	b_1	c_1	l_1
a_1	b_1	c_2	l_1
a_1	b_2	c_1	l_2
a_2	b_2	c_1	l_2
a_1	b_2	c_2	?
...	...	...	...

Training window, W
New instance
$N(a_1) = \{a_1, b_1\}$,
$N(b_2) = \{b_2, c_1\}$,
$N(c_2) = \{c_2, a_1\}$
$U(X) = \{(a_1, a_1), (b_2, b_2), (c_2, c_2), (c_2, a_1)\}$
$\text{Conf}(a_1, l_1) = 3/4$
$\text{Conf}(b_2, l_1) = 0$
$\text{Conf}(c_2, l_1) = 1$
$\text{Conf}(\{c_2, a_1\}, l_1) = 1$
$\text{Sum}(l_1) = 2.75$
$\text{Conf}(a_1, l_2) = 1/4$
$\text{Conf}(b_2, l_2) = 1$
$\text{Conf}(c_2, l_2) = 0$
$\text{Conf}(\{c_2, a_1\}, l_2) = 0$
$\text{Sum}(l_2) = 1.25$
Window is embarking on new instance time after time
...Data stream flowing upwards

Figure 4: Illustration of a rolling window from which contradiction analysis takes place.

for statistics, and the samples (reference) required for noise detection can scale flexibly to any amount ($\leq W$).

One example that is inspired by [27] about a weighted PWC is shown in Figure 4.

In each time, where current position is i, the sliding window contains W potential training samples over three attributes (A, B, and C) and one target class. X is the new instance at the position $i + 1$ which is just ahead of the preceding end of the window, and it has a vector of values $\{a_1, b_2, c_2\}$ as an example. Assuming $k = W/2$ which rounds up to 2, the neighborhood sets for each attribute value of X are shown in the upper-right area of the figure. For example, $N(a_1) = \{a_1, b_1\}$ because $\text{conf}(a_1, a_1) = 1$ and $\text{conf}(a_1, b_1) = 0.75$ are the highest two for a_1. The resulting $U(X)$ set is found below. For instance, associated with a_1 in $U(X)$ is only a_1 itself, forming the pair (a_1, a_1). It does not belong to X and will be excluded from $U(x)$ although $b_1 \in N(a_1)$. The same applies to c_1 with respect to $N(b_2)$, whereas both c_2 and a_1, which belong to $N(c_2)$, are included in $U(X)$ associated with c_1. For each member of $U(X)$, PWC examines the confidence values against the two target classes l_1 and l_2. For instance, for (a_1, l_1), we calculate a $\text{conf}(a_1, l_1) = \text{support}(\{a_1, l_1\})/\text{support}(a_1) = 3/4 = 0.75$, which checks a first-order dependency between a_1 and l_1. In contrast, for pair (c_2, a_1), we examine a second-order dependency by calculating $\text{conf}(\{c_2, a_1\}, l_1) = \text{support}(\{a_1, c_2, l_1\})/\text{support}(a_1, c_2) = 2/2 = 1$. Taking the sum of confidence values for each class, we obtain $\text{Sum}(l_1) = 2.75$ and $\text{Sum}(l_2) = 1.25$; therefore, the new instance should belong to class l_1. In this way, contradiction is determined by observing whether the calculated class membership matches the actual class label of each new instance. If the new instance is in agreement with the PWC calculation, no contradiction is assumed and the window proceeds forward by one row, leaving out the last row and including the new instance in the training set. If the class label of the new instance contradicts the result of the calculated class label, the new instance is purged.

One modification we made in our process is the use of the other version of neighbour sets. The current neighbor sets store only the most updated confidence values of the pairs within the current window frame, which are called local sets. The information in the local sets gets replaced (recomputed) by the new results every time when the window moves to a new position with inclusion of a new instance. In our design, a similar buffer called global sets is used that do not replace but accumulate the newly computed confidence values corresponding to each pair in the window frame.

Of course, the contradiction analysis can be implemented by similar algorithms. It can be seen that, by using PWC, the required information and calculation are kept as minimal as possible, which implies fast operation.

4. Experiment

The objective of the experiment is to verify the efficacy of the proposed IPPM preprocessing strategy. In particular, we want to see how IPPM works when coupled with different classification algorithms which we briefly divided into two groups—those that were founded on greedy search method and instance-based method. A total of eight algorithms were put under test of IPPM preprocessing strategy. For greedy search, representative algorithms include decision tree (J48), naive Bayesian (NB), neural network (NN), and support vector machine (SVM). Instance-based classifiers include K*—an instance-based learner using an entropic distance measure (KStar), K-nearest neighbours classifier

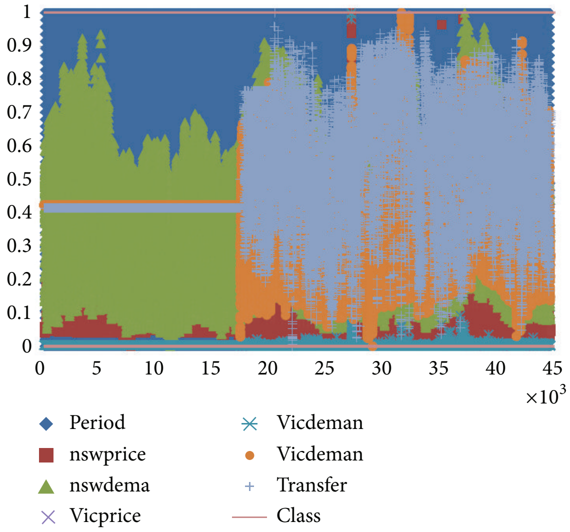


FIGURE 5: Visualization of the electricity dataset.

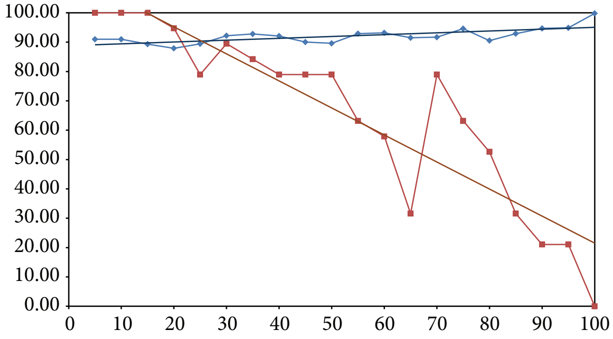


FIGURE 6: Performance curves for J48.

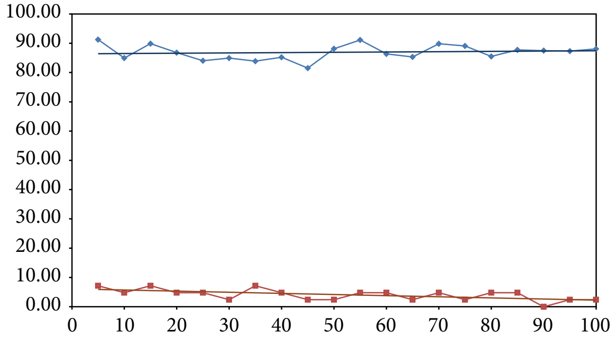


FIGURE 7: Performance curves for NB.

(IBK), locally weighted learning (LWL), and decision table (DT). The experiment is conducted in a Java-based open source platform called Weka which is a popular software tool for machine learning experiments from University of Waikato. All the aforementioned algorithms are available as either standard or plug-in functions on Weka which have been well documented in the Weka documentation file (which is available at http://www.cs.waikato.ac.nz/ml/weka). Hence, their details are not repeated here. The hardware used is Lenovo laptop with Intel Pentium Dual-Core T3200 2 GHz processor, 8 Gb RAM and 64-bits Windows 7.

The test dataset used is called electricity (http://moa.cs.waikato.ac.nz/datasets) which is popularly used for testing data stream mining algorithms. The data was collected from the Australian New South Wales Electricity Market. In this market, prices are not fixed and are affected by demand and supply of the market. They are set every five minutes. The dataset contains 45,312 instances in 7 attributes. The class label identifies the change of the price relative to a moving average of the last 24 hours, whether it goes up or down as class label. Besides, it was widely used in data stream mining; this particular dataset has a property of slight concept drift. It means the statistical properties of the target class the model is trying to predict change over time without any prior clue. This usually deters the accuracy of a classification model; the predictions that are based on previously trained data become less accurate as time passes. This feature is useful for testing the two modes of the local memory and the global memory in our contradiction analysis. A visualization of the electricity dataset is shown in Figure 5; one can see that the variables *vicprice* and *vicdemand* suddenly vary their range and dominate the data distribution at approximately instance 17,000 onwards. To make the problem more challenging for stress testing the efficiency of the preprocessing method, additional random noises are perturbed at 30% of the dataset.

To start the experiment, the dataset is first subject to a collection of eight algorithms in the calibration stage for testing out the optimal size of W. In practice, only a relatively small sample will be used, and calibration could repeat periodically or whenever the performance of the incremental learning drops, for fine-tuning the window size.

For each of the eight algorithms, the accuracy curve is plotted versus the time efficiency, in the same range between 0 and 100, as described in Section 3.2. It is assumed that the optimal point between accuracy that is defined as the percentage of correctly classified instances and time efficiency is wanted. An averaged trend line is regressed on each curve in order to rectify the fluctuations. The line in blue is for accuracy, and the one in red is for time efficiency. They are obtained from running the algorithm over a sample of training data, which is being scaled down from 100% to 5% with a decrement of 5% each time. Reservoir sampling technique is applied here in extracting a certain percentage of the sample so that the selection of data for exclusion would be fair among the data. Being on the same scale at the y-axis, the lines may intersect at a Pareto optimum (Pareto optimal) that reveals the appropriate training size on the x-axis that will just be enough for producing an optimal level of accuracy at the minimal cost of time efficiency. The optimization charts are shown in Figures 6, 7, 8, 9, 10, 11, 12, and 13, respectively, for the eight algorithms.

It can be observed that, interestingly, optimization of the two performance characters (accuracy versus time) over some algorithms produces a cross-point. For instance, the intersection points for J48 and DT fall at the range between 20% and 30% of sampling data. IBK interests at approximately

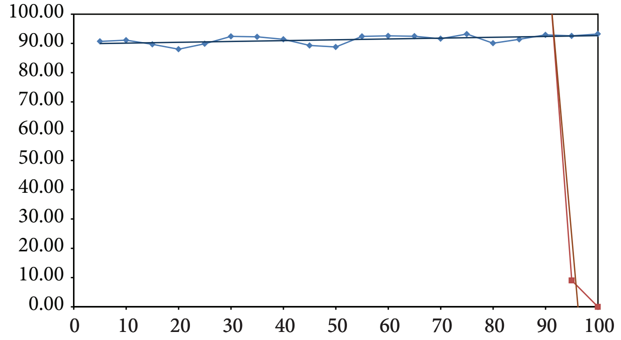


Figure 8: Performance curves for NN.

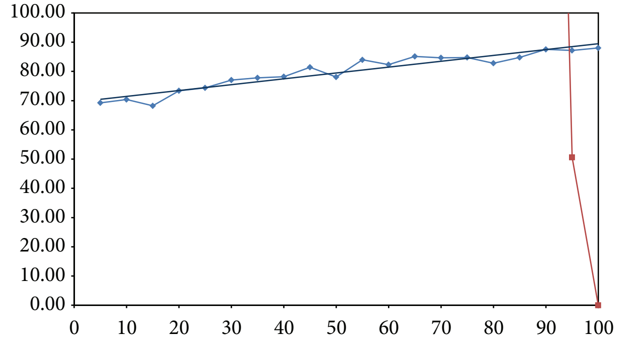


Figure 9: Performance curves for SVM.

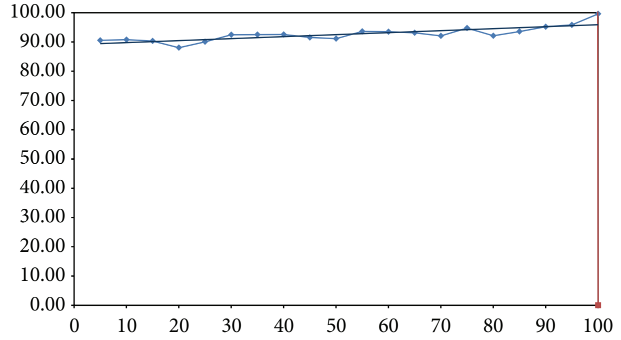


Figure 10: Performance curves for KStar.

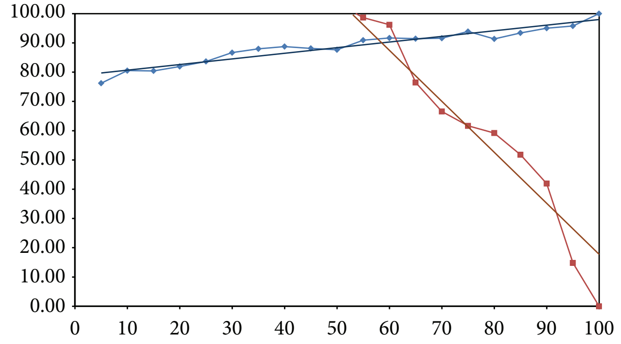


Figure 11: Performance curves for IBK.

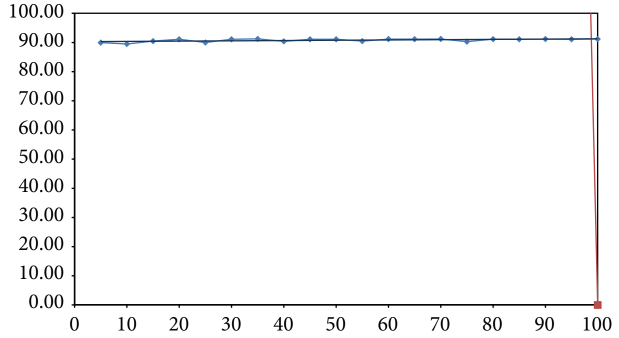


Figure 12: Performance curves for LWL.

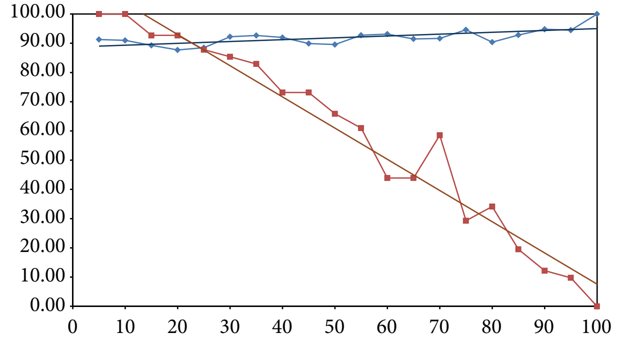


Figure 13: Performance curves for DT.

60%. That means commonly J48 and DT algorithms require about 25% of sampling data in order to achieve an optimal balance or compromise between good accuracy and time efficiency. IBK needs 2.4 times larger the size of the sample. All the achievable accuracies rest on approximately 90% which is not too bad. In contrast, the rest of the algorithms like neural network, SVM, KStar, and LWL require almost the full set of data for reaching an acceptable level of accuracy and speed. This can be explained by the fact that they generally require hundreds times longer the time than those others (e.g., LWL trains and tests for 5052 seconds; compared to IBK that only needs 72 seconds for processing 100% amount of the data). NB is an outlier which is executed extremely fast, but the accuracy fluctuates to the greatest extent among all.

After all, the choice of W and the algorithm could be subjective by the users. From the results, it seems like 25% of window size W is a logical choice. One may argue that if the data stream is potential infinite, how do we know relatively the ideal percentage of the required window size? In such a case, an absolute number of instances would be used in lieu of a relative percentage. The number of training instances would be reduced gradually while observing the degeneration of accuracy, and it will stop when the accuracy falls onto an acceptable minimum threshold.

Next, the whole dataset is subject to the dual process of IPPM preprocessing and incremental learning, after an appropriate W is derived. The dataset now serves a data stream that is being tested and model-trained segment by segment via a sliding window. Again, the eight algorithms

Table 1: Performance comparison of algorithms under batch learning mode.

Algorithm	Batch learning					
	No preprocessing			Preprocessing		
	Accuracy	Time	Gain ratio	Accuracy	Time	Gain ratio
J48	62.5110	33.0000	1.8943	99.7983	9.0000	11.0887
NB	55.0207	3.0000	18.3402	88.0629	4.0000	22.0157
NN	60.0702	1083.0000	0.0555	93.2245	613.0000	0.1521
SVM	58.1060	3607.0000	0.0161	88.0324	803.0000	0.1096
Kstar	91.7483	7970.0000	0.0115	99.6041	7860.0000	0.0127
IBK	100.0000	258.0000	0.3876	100.0000	72.0000	1.3889
LWL	60.5204	16858.0000	0.0036	91.1804	5082.0000	0.0179
DT	61.5731	37.0000	1.6641	100.0000	42.0000	2.3810

Table 2: Performance comparison of algorithms under incremental learning mode.

Algorithm	Incremental learning								
	No preprocessing			PP. with respect to local reference			PP. with respect to global reference		
	Accuracy	Time	Gain ratio	Accuracy	Time	Gain ratio	Accuracy	Time	Gain ratio
J48	56.7105	12.0000	4.7259	71.6697	5.3333	13.4381	69.2177	12.5000	5.5374
NB	55.7100	2.6667	20.8913	68.5534	4.6667	14.6900	84.5540	2.0000	42.2770
NN	53.7179	834.0000	0.0644	66.2704	521.3333	0.1271	73.4672	984.0000	0.0747
SVM	55.8189	844.0000	0.0661	69.6478	238.6667	0.2918	76.5615	1040.0000	0.0736
Kstar	55.9808	3485.3333	0.0161	74.2809	1300.6667	0.0571	78.4271	3262.5000	0.0240
IBK	52.8808	63.3333	0.8350	69.1130	24.0000	2.8797	66.2719	61.5000	1.0776
LWL	59.5147	2562.6667	0.0232	73.4821	911.3333	0.0806	83.6198	4102.0000	0.0204
DT	56.9724	9.3333	6.1042	71.4792	4.6667	15.3170	69.0701	9.0000	7.6745

are used in enabling the model learning. The models by the eight algorithms are updated and refreshed every time when the window progresses by one instance. Specifically, the two modes of contradiction analysis, local reference and global reference, are tested with the given data stream. In data streaming, a significant difference is that the temporal pattern (a.k.a. data evolution pattern or concept drift) can be taken into effect. That is contrary to batch training where the whole data is lumped up and the overall data distributions could only be observed—the change of data distribution in time cannot be seen.

The batch mode learning is still conducted as a comparison benchmark to the incremental learning. Supposedly, the batch mode learning has the luxury of accessing the full set of data; thereby, accuracy can be possibly made to the highest with the full provision. The ultimate objective of this experiment is to compare and judge if incremental learning can be on par with traditional batch learning. In both cases, they are done with and without preprocessing for all the eight algorithms. In batch mode, the training and testing of the data by a learning model are done over the full length of data; the testing results are validated by 10-fold validation; in incremental mode, the process is carried out according to IPPM as specified in Section 3.1.

The performance results in terms of accuracy and total processing time in seconds are shown in Tables 1 and 2 for batch learning and incremental learning, respectively. A gain ratio is added that is simply the accuracy value divided by the time taken value, for easy comparison. The higher the accuracy, the shorter the time taken, and the better the gain ratio. The value of gain ratio, if close to zero, means this algorithm under this configuration of learning mode and preprocessing method is not favorable. Otherwise, it is a good candidate for the combo of algorithm and learning environment when the gain ratio is large.

In the batch learning plus no preprocessing, in Table 1, the gain ratios for those algorithms (NN, SVM, KStar, IBK, and LWL) are consistent with the optimization results in Figures 6 to 13 carry a small gain ratio. Those algorithms that require long training are unfavorable and their accuracies are poor (with exceptions of KStar and IBK). NB has the highest gain ratio; J48 and DT achieve a gain ratio of greater than 1. With preprocessing in batch mode, the trend persists but the gain ratio scaled up significantly. The same PWC was used in all preprocessing methods for consistency.

Some interesting phenomenon is observed when comparing batch learning with no preprocessing and incremental learning with no preprocessing. While the gain ratios of most of the algorithms remain about the same, the gain ratios for J48 and DT increased severalfold in the incremental learning mode. The gain ratios increase even more remarkably when preprocessing applies in incremental learning. The gain ratio for J48 increases from 4.7 (no preprocessing) to 13.4 (preprocessing with local reference) and to 5.5 (preprocessing with global reference). The gain ratio for DT escalates from 6.1 (no preprocessing) to 15.3 which is more than double

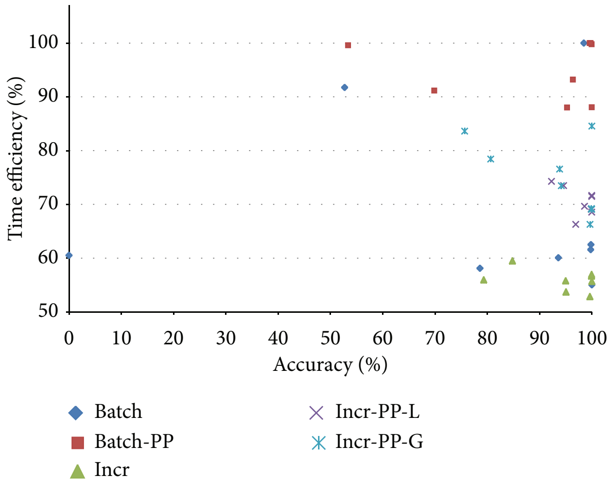


FIGURE 14: Performance comparison of groups of learning and preprocessing modes at a glance.

the gain (preprocessing with local reference) and to 7.7 (preprocessing with global reference). The other algorithms have their share of increase in gain ratio in varying degrees too, when preprocessing applies in incremental learning.

In general, it can be shown that incremental learning for the eight popular algorithms has an edge over batch learning especially when the lightweight preprocessing methods applied. It may provide an alternative methodology of data mining over the traditional preprocessing and then training followed by testing approach, in batch mode.

In incremental learning, almost all the algorithms perform better in preprocessing with local reference than global preprocessing with local reference, except only NB. In fact, the gain ratio of NB in preprocessing with local reference is even worse than no preprocessing in incremental learning. The increase in local reference over global reference in preprocessing attributes to the effect of concept drift experienced from the given dataset. Keeping check of the information in local context can effectively adapt to the new data distribution caused by the concept drift. New (local) data are more relevant than the aged data, so making use of which maintains good accuracy. NB in opposite, works well in preprocessing with global reference as NB is purely probability-based and it is known to provide accurate result by digesting all the instances of data and enumerating all the probabilities of outcomes and variables. Constraining NB by a local window may thwart its pursuit of counting sufficient instances for probabilities estimation, hence it may result in poor accuracy.

The performance of all the algorithms in terms of accuracy and time is visualized in several groups. They are visualized in Figure 14. The time efficiency is scaled into a range of 0% to 100%. The best performance is at the upper-right corner of the chart. We can easily see a general pattern that the performance of batch learning is leading. Incremental learning without preprocessing ranks at the bottom. However, incremental learning equipped with preprocessing methods follows the performance of batch learning. Out of the two preprocessing methods, the one with local reference is superior to global reference except for the case of NB.

5. Conclusion

Noise is known to be a cause of confusion in the construction of classification models and thus as a factor leading to a deterioration in accuracy. We regard noise as simply a contradicting instance that does not agree with the majority of data; this disagreement causes the establishment of erroneous rules in classification models and disrupts homogenous meta-knowledge or statistical patterns by distorting the training dataset. Other authors refer to noise as outliers, misclassified instances, or misfits, all of which are data types, the removal of which will improve the accuracy of the classification model. Though this research topic has been studied for over two decades, techniques previously proposed for removing such noise assume batch operations requiring the full dataset to be used in noise detection.

To the best of our knowledge, this paper is the first to propose the novel preprocessing strategy incorporated into our model, the IPPM, which is of particular relevance to incremental classification learning. The chief advantage of the IPPM lies in its lightweight mechanism, which provides optimal speed and accuracy. Its design is also suitable for mining moving data streams. The IPPM is extremely simple to use in comparison with other more complex techniques such as those outlined in Section 2. Our experiment validates its benefits in terms of its very high speed and its efficacy in providing a noise-free streamlined training dataset for incremental learning. In future work, the IPPM will be tested with datasets that have different types and numbers of attributes. More importantly, sample datasets with concept-drift characteristics will be used to test our model. Concept drift is a phenomenon whereby the underlying meaning of the dataset changes over time. Because the contradiction analysis statistics retained by the model are of two types—those obtained from a window frame only and those obtained by accumulating the values calculated during the run as a whole—they represent local optima and global optima, respectively. Whereas local statistics are akin to local memory, global statistics are like global memory. It would be interesting to see whether the latter adapt better to the evolving dataset than local see-and-forget statistics. With its unique lightweight capability of empowering incremental learning with the benefit of fast data preprocessing, it is anticipated that the IPPM will contribute to a wide range of data stream applications [28] in which speed and accuracy are equally important.

Conflict of Interests

The authors of this paper do not have a direct financial relationship with the commercial identities mentioned in this paper that might lead to a conflict of interests.

Acknowledgments

The authors are thankful for the financial support from the Research Grant “Adaptive OVFDT with Incremental Pruning and ROC Corrective Learning for Data Stream Mining,” Grant no. MYRG073(Y3-L2)-FST12-FCC, offered by the University of Macau, FST, and RDAO.

References

[1] H. Yang and S. Fong, "Aerial root classifiers for predicting missing values in data stream decision tree classification," in *Proceedings of the SIAM International Conference on Data Mining (SDM '11)*, pp. 1–10, Mesa, Ariz, USA, April 2011.

[2] M. R. Smith and T. Martinez, "Improving classification accuracy by identifying and removing instances that should be misclassified," in *Proceedings of the International Joint Conference on Neural Network (IJCNN '11)*, pp. 2690–2697, San Jose, Calif, USA, July-August 2011.

[3] X. Zhu and X. Wu, "Class noise vs. Attribute noise: a quantitative study," *Artificial Intelligence Review*, vol. 22, no. 3, pp. 177–210, 2004.

[4] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[5] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 37–46, May 2001.

[6] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Procceedings of the Conference of Management of Data (ACM SIGMOD '96)*, pp. 103–114, 1996.

[7] P. Chen, L. Dong, W. Chen, and J.-G. Lin, "Outlier detection in adaptive functional-coefficient autoregressive models based on extreme value theory," *Mathematical Problems in Engineering*, vol. 2013, Article ID 910828, 9 pages, 2013.

[8] W. Hu and J. Bao, "The outlier interval detection algorithms on astronautical time series data," *Mathematical Problems in Engineering*, vol. 2013, Article ID 979035, 6 pages, 2013.

[9] H. Tan, J. Feng, G. Feng, W. Wang, and Y. Zhang, "Traffic volume data outlier recovery via tensor model," *Mathematical Problems in Engineering*, vol. 2013, 8 pages, 2013.

[10] Y. Ji, D. Tang, W. Guo, P. T. Blythe, and G. Ren, "Detection of outliers in a time series of available parking spaces," *Mathematical Problems in Engineering*, vol. 2013, Article ID 416267, 12 pages, 2013.

[11] A. Arning, R. Agrawal, and P. Raghavan, "A linear method for deviation detection in large databases," in *Proceedings of the International Conference on Data Mining and Knowledge Discovery (KDD '96)*, pp. 164–169, Portland, Ore, USA, 1996.

[12] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar, "Enhancing data analysis with noise removal," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 3, pp. 304–319, 2006.

[13] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," *Data Mining and Knowledge Discovery*, vol. 6, no. 2, pp. 153–172, 2002.

[14] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 427–438, 2000.

[15] M. M. Breuniq, H. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," *SIGMOD Record*, vol. 29, no. 2, pp. 93–104, 2000.

[16] S. Fong, Z. Luo, B. W. Yap, and S. Deb, "Incremental methods for detecting outliers from multivariate data stream," in *Proceedings of the 13th IASTED International Conference on Artificial Intelligence and Applications (AIA '14)*, Innsbruck, Austria, February 2014.

[17] M. K. Saad and N. M. Hewahi, "A comparative study of outlier mining and class outlier mining," *Computer Science Letters*, vol. 1, no. 1, 2009.

[18] I. Tomek, "An experiment with the edited nearest-neighbor rule," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, no. 6, pp. 448–452, 1976.

[19] M. M. Tavakoli and A. Sami, "Robust preprocessing for improving angle based outlier detection technique," in *Proceedings of the Iranian Conference on Intelligent Systems (ICIS '14)*, pp. 1–6, February 2014.

[20] L. Sunitha, M. BalRaju, J. Sasikiran, and E. Venkat Ramana, "Automatic outlier identification in data mining using IQR in real-time data," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 6, pp. 7255–7257, 2014.

[21] M. Govindarajan and V. Abinaya, "An outlier detection approach with data mining in wireless sensor network," *International Journal of Current Engineering and Technology*, vol. 4, no. 2, pp. 929–932, 2014.

[22] C. E. Brodley and M. A. Friedl, "Identifying and eliminating mislabeled training instances," in *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 799–805, AAI Press, Portland, Ore, USA, August 1996.

[23] G. H. John, "Robust decision tree: removing outliers from databases," in *Proceedings of the 1st International Conference Knowledge Discovery and Data Mining*, pp. 174–179, 1995.

[24] B. Byeon, K. Rasheed, and P. Doshi, "Enhancing the quality of noisy training data using a genetic algorithm and prototype selection," in *Proceedings of the International Conference on Artificial Intelligence (ICAI '08)*, pp. 821–827, Las Vegas, Nev, USA, July 2008.

[25] P. Chen, L. Dong, W. Chen, and L. Jin-Guan, "Outlier detection in adaptive functional-coefficient autoregressive models based on extreme value theory," *Mathematical Problems in Engineering*, vol. 2013, Article ID 910828, 9 pages, 2013.

[26] A. Hax and D. Candea, *Production and Operations Management*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1984.

[27] A. Nanopoulos, A. N. Papadopoulos, Y. Manolopoulos, and T. Welzer-Druzovec, "Robust classification based on correlations between attributes," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, 14 pages, 2007.

[28] S. Fong and H. Yang, "The six technical gaps between intelligent applications and real-time data mining: a critical review," *Journal of Emerging Technologies in Web Intelligence*, vol. 3, no. 2, pp. 63–73, 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 290781, 7 pages
http://dx.doi.org/10.1155/2015/290781

Research Article

Multisegment Scheme Applications to Modified Chebyshev Picard Iteration Method for Highly Elliptical Orbits

Donghoon Kim,[1] John L. Junkins,[2] and James D. Turner[1]

[1] *Aerospace Engineering, Texas A&M University, College Station, TX 77843-3141, USA*
[2] *Royce Wisenbaker Chair in Engineering, Aerospace Engineering, Texas A&M University, College Station, TX 77843-3141, USA*

Correspondence should be addressed to Donghoon Kim; aerospace38@gmail.com

Received 10 April 2014; Accepted 22 July 2014

Academic Editor: Ker-Wei Yu

A modified Chebyshev Picard iteration method is proposed for solving orbit propagation initial/boundary value problems. Cosine sampling techniques, known as Chebyshev-Gauss-Lobatto (CGL) nodes, are used to reduce Runge's phenomenon that plagues many series approximations. The key benefit of using the CGL data sampling is that the nodal points are distributed nonuniformly, with dense sampling at the beginning and ending times. This problem can be addressed by a nonlinear time transformation and/or by utilizing multiple time segments over an orbit. This paper suggests a method, called a multisegment method, to obtain accurate solutions overall regardless of initial states and albeit eccentricity by dividing the given orbit into two or more segments based on the true anomaly.

1. Introduction

A modified Chebyshev Picard iteration (MCPI) is an iterative numerical method for approximating solutions of linear or nonlinear ordinary differential equations to obtain time histories of system state trajectories [1, 2]. In contrast to many step-by-step integrators, the MCPI algorithm approximates long arcs of the state trajectory with an iterative path approximation approach and is ideally suited to parallel computation [3]. It is well known that Picard iteration has theoretical guarantees for converging to the solution assuming the forces are continuous, once differentiable, and the solution of the differential equation is unique [4]. The rate of convergence of Picard iteration is geometric rather than quadratic for Jacobian based methods. However, given a good starting approximation, excellent efficiency is possible, and the case for parallelization provides a significant advantage [5, 6].

Orthogonal Chebyshev polynomials are used as basis functions during each path iteration, and the integrations of Picard iteration are then performed analytically. The orthogonality of the Chebyshev basis functions implies that the least-square approximations can be computed to arbitrary precision without a matrix inversion; the coefficients are conveniently and robustly computed from discrete inner products [7]. Similar approximation approaches that use Legendre polynomials can be utilized, but the authors obtain slightly better results because the starting and ending points of the fits are not sampled as densely as the MCPI algorithm, and importantly the location of the nodes for the Chebyshev basis functions is computed exactly without iterations. The MCPI algorithm utilizes a vector-matrix framework for computational efficiency. Additionally, all Chebyshev coefficients and integrand function evaluations are independent, meaning that they can be simultaneously computed in parallel for further decreased computational costs [3].

For the MCPI algorithm, the cosine sampling techniques, known as Chebyshev-Gauss-Lobatto (CGL) nodes [8], are utilized to reduce Runge's phenomenon. The Runge phenomenon is a problem of oscillation at the edges of an interval that occurs when using polynomial interpolation with polynomials of high degree [9]. Since dense sample points are distributed at the beginning and ending locations, less accurate solutions are usually obtained where sample points are more uniformly distributed [10].

For the most extreme counterexample, let us consider an unperturbed two-body problem, where the initial position

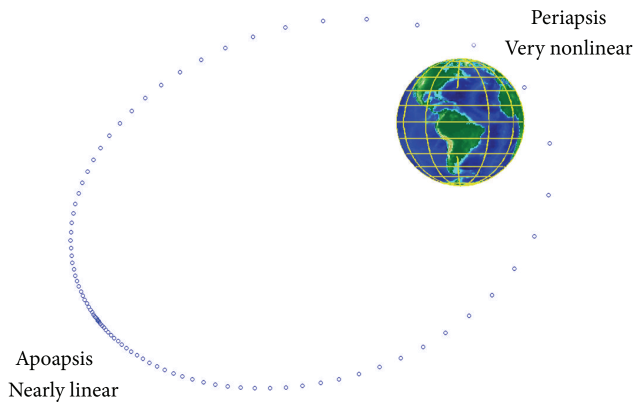


Figure 1: Sparse sample point distribution description at periapsis.

is not located near the periapsis. Obviously, large errors can be observed near the periapsis due to sparse sample points where the dynamics are most nonlinear, yet we waste the dense sample points at apoapsis when the problem is most linear as shown in Figure 1.

This problem is overcome by introducing a multisegment method and the results are compared with the basic MCPI algorithm. This paper only considers two and three segments per one orbit. The performance of the proposed approach is established by numerical examples of the two-body problem.

2. Modified Chebyshev Picard Iteration

The MCPI algorithm combines the discoveries of two great mathematicians: Émile Picard (Picard iteration) and Rafnuty Chebyshev (Chebyshev polynomials). Combing these techniques was first proposed by Clenshaw and Norton in 1963 [11].

Picard stated that any first-order differential equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}\left(t, \mathbf{x}\left(t\right)\right) \tag{1}$$

with an initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$ can be rearranged without approximation as follows:

$$\mathbf{x}\left(t\right) = \mathbf{x}\left(t_0\right) + \int_{t_0}^{t} \mathbf{f}\left(\tau, \mathbf{x}\left(\tau\right)\right) d\tau. \tag{2}$$

In the MCPI algorithm, orthogonal Chebyshev polynomials are used as basis functions to approximate the integrand in the Picard integral. Chebyshev polynomials reside in the domain $\tau = [-1, 1]$ and are defined recursively as

$$\mathbf{x}^{i}\left(t\right) = \mathbf{x}\left(t_0\right) + \int_{t_0}^{t} \mathbf{f}\left(\tau, \mathbf{x}^{i-1}\left(\tau\right)\right) d\tau, \quad i = 1, 2, \ldots. \tag{3}$$

The system dynamics are normalized such that the time span of integration is projected onto the domain of the Chebyshev polynomials. Thus, the system states are approximated using the Chebyshev polynomial basis functions. The orthogonal nature of the basis functions means the coefficients that linearly scale the basis functions are computed independently as simple ratios of inner products without requiring matrix inversions.

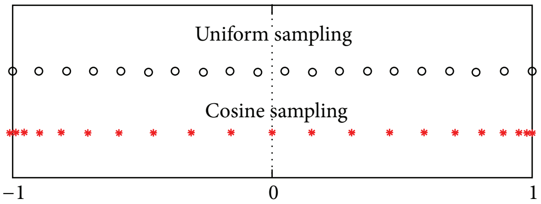


Figure 2: Uniform and cosine sampling descriptions.

A key feature of the MCPI algorithm is a nonuniform cosine density sampling of the domain of the Chebyshev basis functions, called CGL nodes, defined as follows:

$$\begin{gathered} T_0\left(\tau\right) = 1, \\ T_1\left(\tau\right) = \tau, \\ T_{k+1}\left(\tau\right) = 2\tau T_k\left(\tau\right) - T_{k-1}\left(\tau\right). \end{gathered} \tag{4}$$

This sampling scheme provides much higher density towards the edges (beginning and ending points), which enables high accuracy solutions near the boundaries of the state trajectory. This scheme eliminates the Runge phenomenon, a common issue in function approximations, whereby noisy estimates are returned near the edges due to lack of knowledge of the states on the other sides of the boundaries (see Figure 2). The coefficients multiplying the Chebyshev basis functions are approximated by the method of least squares, which generally requires a matrix inversion. A wonderful side effect of the cosine sampling scheme is that the matrix required to be inverted in the normal equations of least squares is diagonal; thus the inverse is trivial.

A full derivation of the MCPI algorithm is not included in this work (refer to Bai [3]). Instead, the authors present a flowchart in Figure 3 briefly summarizing the mathematics underlying the MCPI algorithm for solution of initial value problems.

3. Multisegment Approach for MCPI Algorithm

This work considers an unperturbed two-body problem, where the initial position is not located near the periapsis. As expected, large errors are observed near the periapsis where dense sample points are required, but sparse sample points are distributed. In addition, even though initial positions are located near the periapsis, accurate solutions cannot be obtained for highly elliptical orbits. To obtain accurate solutions for the above cases using the MCPI algorithm, the multisegment approach is proposed.

Given the initial true anomaly (f_0), two and three segmented orbits are considered as shown in Figure 4. These two cases require patch times to link the divided segments. To distribute dense sample points near the periapsis, several strategies are presented.

For two segmented orbits, the time for the patch point is selected as the time at the perigee, where the true anomaly (f)

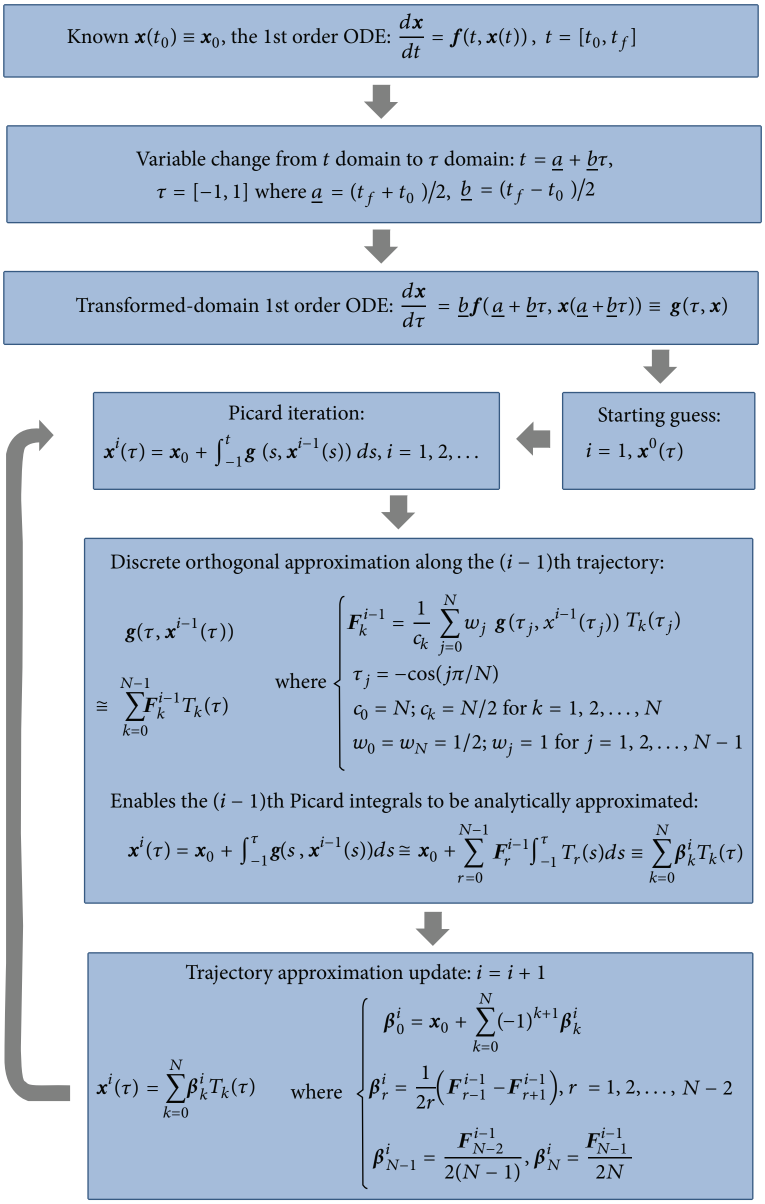

Known $\boldsymbol{x}(t_0) \equiv \boldsymbol{x}_0$, the 1st order ODE: $\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(t, \boldsymbol{x}(t))$, $t = [t_0, t_f]$
Variable change from t domain to τ domain: $t = \underline{a} + \underline{b}\tau$,
$\tau = [-1, 1]$ where $\underline{a} = (t_f + t_0)/2$, $\underline{b} = (t_f - t_0)/2$
Transformed-domain 1st order ODE: $\frac{d\boldsymbol{x}}{d\tau} = \underline{b}\boldsymbol{f}(\underline{a} + \underline{b}\tau, \boldsymbol{x}(\underline{a} + \underline{b}\tau)) \equiv \boldsymbol{g}(\tau, \boldsymbol{x})$
Starting guess:
$i = 1, \boldsymbol{x}^0(\tau)$
Picard iteration:
$\boldsymbol{x}^i(\tau) = \boldsymbol{x}_0 + \int_{-1}^{t} \boldsymbol{g}(s, \boldsymbol{x}^{i-1}(s))\, ds$, $i = 1, 2, \ldots$
Discrete orthogonal approximation along the $(i-1)$th trajectory:
$\boldsymbol{g}(\tau, \boldsymbol{x}^{i-1}(\tau)) \cong \sum_{k=0}^{N-1} \boldsymbol{F}_k^{i-1} T_k(\tau)$
where
$\boldsymbol{F}_k^{i-1} = \frac{1}{c_k} \sum_{j=0}^{N} w_j\, \boldsymbol{g}(\tau_j, x^{i-1}(\tau_j))\, T_k(\tau_j)$
$\tau_j = -\cos(j\pi/N)$
$c_0 = N$; $c_k = N/2$ for $k = 1, 2, \ldots, N$
$w_0 = w_N = 1/2$; $w_j = 1$ for $j = 1, 2, \ldots, N-1$
Enables the $(i-1)$th Picard integrals to be analytically approximated:
$\boldsymbol{x}^i(\tau) = \boldsymbol{x}_0 + \int_{-1}^{\tau} \boldsymbol{g}(s, \boldsymbol{x}^{i-1}(s))ds \cong \boldsymbol{x}_0 + \sum_{r=0}^{N-1} \boldsymbol{F}_r^{i-1} \int_{-1}^{\tau} T_r(s)ds \equiv \sum_{k=0}^{N} \boldsymbol{\beta}_k^i T_k(\tau)$
Trajectory approximation update: $i = i + 1$
$\boldsymbol{x}^i(\tau) = \sum_{k=0}^{N} \boldsymbol{\beta}_k^i T_k(\tau)$
where
$\boldsymbol{\beta}_0^i = \boldsymbol{x}_0 + \sum_{k=0}^{N} (-1)^{k+1} \boldsymbol{\beta}_k^i$
$\boldsymbol{\beta}_r^i = \frac{1}{2r}\left(\boldsymbol{F}_{r-1}^{i-1} - \boldsymbol{F}_{r+1}^{i-1}\right)$, $r = 1, 2, \ldots, N-2$
$\boldsymbol{\beta}_{N-1}^i = \frac{\boldsymbol{F}_{N-2}^{i-1}}{2(N-1)}$, $\boldsymbol{\beta}_N^i = \frac{\boldsymbol{F}_{N-1}^{i-1}}{2N}$

FIGURE 3: Flowchart for the MCPI algorithm for solution of initial value problems.

is 0 degrees. For three segmented orbits, the time for the first patch point is selected where $f = -f_0$ degree for symmetry, and the time for the second patch point is selected where $f = 0$ degrees. To find propagation times for each segment, the following calculation needs to be performed. First, given the initial position and velocity vectors, prescribe the break point f_0 and one orbit period time (T_P) as follows [12]:

$$T_P = 2\pi \sqrt{\frac{a^3}{\mu}}. \tag{5}$$

Second, calculate the initial mean anomaly (M_0) as follows [13]:

$$M_0 = E_0 - e \sin E_0, \tag{6}$$

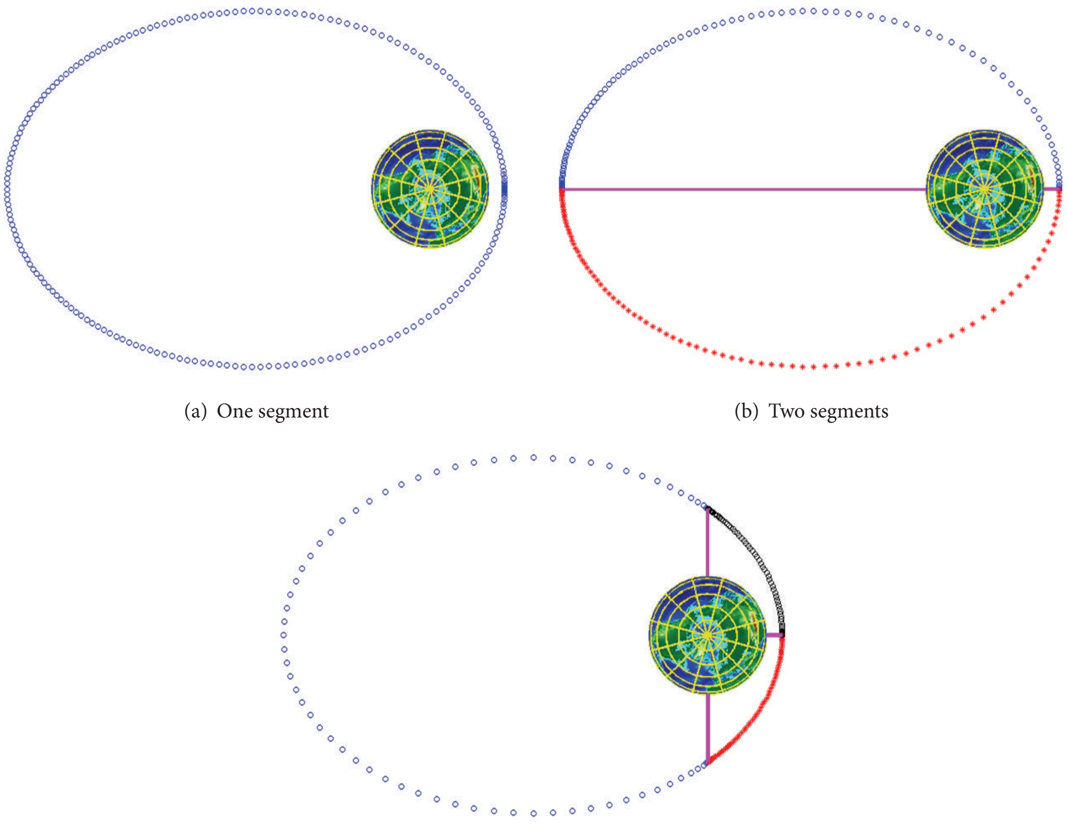
where e is the eccentricity and the eccentric anomaly (E_0) is defined as [13]

$$E_0 = 2\tan^{-1}\left[\sqrt{\frac{1-e}{1+e}} \tan\left(\frac{f_0}{2}\right)\right]. \tag{7}$$

Finally, the propagation times for each segment are calculated as follows:

$$\begin{aligned} T_{P_1} &= T_P - (S-1)\, T_{P_2}, \\ T_{P_2} &= M_0 \sqrt{\frac{a^3}{\mu}}, \end{aligned} \tag{8}$$

where $S > 1$ is the number of segments for the orbit.



(a) One segment

(b) Two segments

(c) Three segments

Figure 4: Segmented orbit descriptions.

The sets of propagation times are determined as follows:

$$\begin{aligned} &\text{two segments: } \mathbb{T}_{\mathbb{P}} = \left[T_{P_1}, T_{P_2}\right], \\ &\text{three segments: } \mathbb{T}_{\mathbb{P}} = \left[T_{P_1}, T_{P_2}, T_{P_2}\right]. \end{aligned} \tag{9}$$

For more than three segments and the associated break points, the above logic is readily extended. The optimization of the break points to achieve efficiency and accuracy is not addressed in this paper but is an easy-to-pose optimization problem research for a future study.

4. Numerical Examples

A satellite motion integration problem, idealized for the case with only the inverse square gravitational force from the Earth, is considered. The three-dimensional dynamical equations are given by [12]

$$\begin{aligned} \ddot{x} &= -\frac{\mu}{r^3} x, \\ \ddot{y} &= -\frac{\mu}{r^3} y, \\ \ddot{z} &= -\frac{\mu}{r^3} z, \end{aligned} \tag{10}$$

where x, y, and z are the three coordinates in Earth-centered inertial reference frame; r is the distance of the satellite from the Earth; μ is the Earth gravitational constant and is chosen as $3.98600433 \times 10^{14}\ \mathrm{m}^3/\mathrm{s}^2$.

To verify the results, the following normalized energy error check is utilized:

$$\mathscr{E}_{\text{error}} = \frac{\left|\mathscr{E} - \mathscr{E}_0\right|}{\left|\mathscr{E}_0\right|}, \tag{11}$$

where $\mathscr{E}_0$ is the initial energy and the energy is calculated as follows:

$$\mathscr{E} = \frac{1}{2}\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right) - \frac{\mu}{r}. \tag{12}$$

Note that the goal for the demonstration example in this paper is to obtain solutions where $\mathscr{E}_{\text{err}} < 10^{-13}$. Moreover, for the unperturbed two-body problem, the analytical solution [12] for the F&G function can also be used to confirm the accuracy of the solution.

Table 1: Classical orbital elements.

Parameter	Symbol	Value	Unit
Semimajor axis	a	2.7×10^7	m
Eccentricity	e	0.7	—
Inclination	i	60	Degree
Right ascension of the ascending node	Ω	45	Degree
Argument of periapsis	w	30	Degree
Orbit period	T_P	4.4153×10^4	s

Table 2: Tuning parameters.

Parameter	Symbol	Value	Unit
Maximum iteration number	I_M	50	—
Error tolerance	T_E	10^{-13}	—
Degree of polynomial	N	200	—
Number of sample points (one segment)	M	200	—
Number of sample points (two segments)	M_1, M_2	100, 100	—
Number of sample points (three segments)	M_1, M_2, M_3	68, 66, 66	—

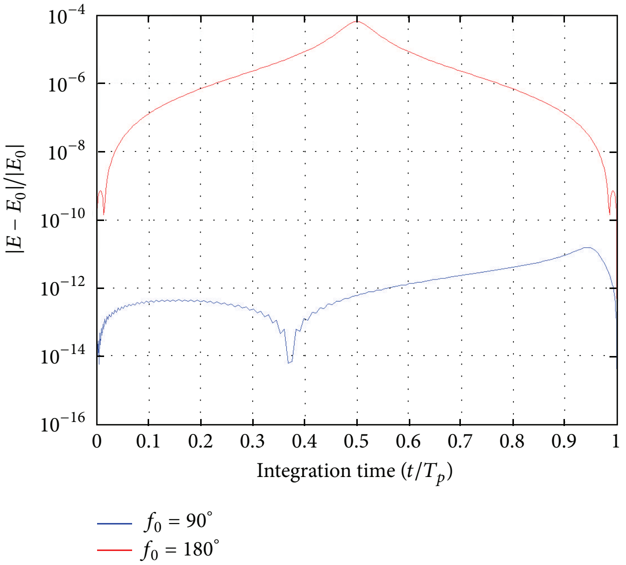


Figure 5: Time trajectories of the energy error; one segment.

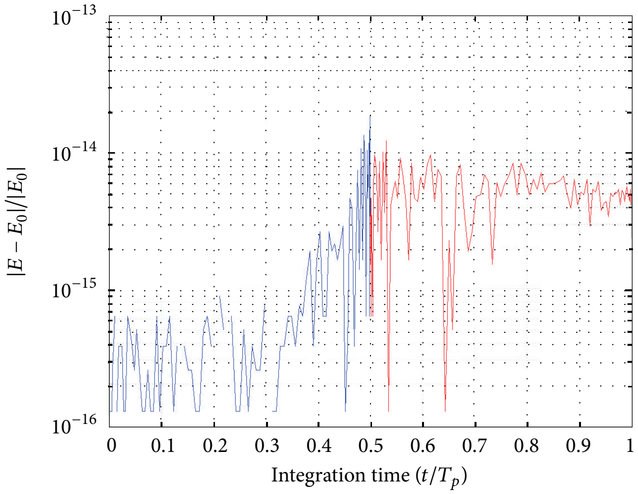


Figure 6: Time trajectories of the energy error and its comparison; two segments (f_0 = 180 degrees).

Two sets of initial position and velocity vectors are given as follows:

$$\begin{aligned}\mathbf{r}(t_0) &= [-0.9085, -0.0652, 1.0328]^{\mathrm{T}} \times 10^7\ \mathrm{m}, \\ \mathbf{v}(t_0) &= [-4.8283, -4.4242, 0.4949]^{\mathrm{T}} \times 10^3\ \mathrm{m/s}, \\ \mathbf{r}(t_0) &= [-1.9994, -3.6222, -1.9875]^{\mathrm{T}} \times 10^7\ \mathrm{m}, \\ \mathbf{v}(t_0) &= [1.0649, 0.0765, -1.2106]^{\mathrm{T}} \times 10^3\ \mathrm{m/s}.\end{aligned} \tag{13}$$

The given initial states lead to $f_0 = 90$ and 180 degrees, respectively, and the classical orbital elements are listed in Table 1.

For the MCPI algorithm implementation to solve this problem, various tuning parameters are determined in a prior calculation: (1) maximum iteration number (I_M), (2) error tolerance (T_E), (3) degree of polynomial (N), and (4) number of sample points (M). This work focuses on finding a methodology to improve MCPI accuracy and reduce computational burden given the described factors in Table 2 and initial conditions listed in Table 1.

Numerical simulations are performed, and the normalized energy error results are shown in Figures 5–8. Figure 5 shows that the normalized energy errors are much larger than the requirement ($\mathscr{E}_{\mathrm{err}} < 10^{-13}$). Obviously, the largest error is observed at the periapsis when $f_0 = 180$ degrees because of sparse sample point distributions at the periapsis.

Figure 6 shows that the solution satisfies the requirement when $f_0 = 180$ degrees using the two-segment scheme. The same number of sample points is distributed for each segment, and the total number of the sample points is equal to the number of sample points for the basic (one-segment) MCPI algorithm. *Note that only the two-segment orbit approach is used when the initial position is located at apoapsis for symmetry.*

Figure 7 shows that the solution satisfies the requirement when $f_0 = 90$ degrees using the three-segmentation scheme. The same number of sample points is distributed for the second and third segments, and the total number of the sample points is equal to the number of sample points for the basic MCPI algorithm.

For the case where $f_0 = 90$ degrees, both approaches such as the two- and three-segment schemes are applicable. As shown in Figure 8, both approaches satisfy the requirement, but the three-segment orbit approach outperforms the other methods. The number of nodes for each approach is determined by a heuristic method for this paper (and tuned

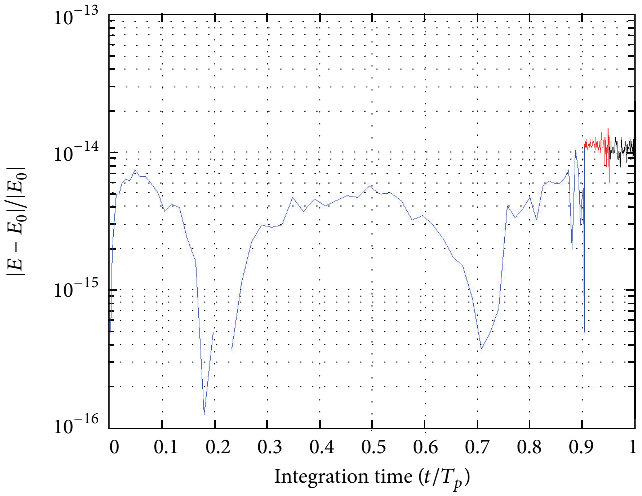


Figure 7: Time trajectories of the energy error and its comparison; three segments (f_0 = 90 degrees).

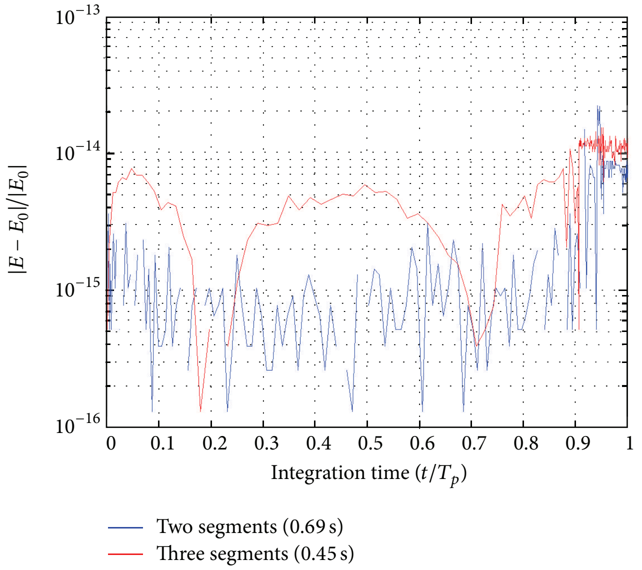


Figure 8: Time trajectories of the energy error and its comparison; two and three segments (f_0 = 90 degrees).

numerically); and a methodology to select optimal number of nodes is under development.

5. Conclusion

The modified Chebyshev Picard iteration (MCPI) algorithm uses Chebyshev-Gauss-Lobatto (CGL) nodes to reduce the Runge phenomenon. By using the CGL nodes, however, less accurate solutions may be obtained where sparse sample points are distributed. Physical insights indicate that the dense nodes should be located where the orbit is most nonlinear. However, the stating epoch state can be at a random point in the orbit. For the unperturbed two-body problem, where the initial state is not located near the periapsis and the eccentricity is high, the multisegment approach is utilized to obtain an accurate solution. The final perigee passage can be used to make all subsequent segment breaks symmetrical about the major axis. As a result, the multisegment approach provides much more accurate solutions when compared to the solution from the basic MCPI algorithm with random user-specified segmentation logic. Moreover, it is shown that the three-segment orbit approach outperforms others in terms of computational efficiency. To improve the performance of the MCPI algorithm, this approach will be very useful, especially when the initial position is not located near the periapsis and high eccentric orbits are given.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported by the Air Force Office of Scientific Research, USA.

References

[1] X. Bai and J. L. Junkins, “Modified chebyshev-picard iteration methods for orbit propagation,” *Journal of the Astronautical Sciences*, vol. 58, no. 4, pp. 583–613, 2011.

[2] X. Bai and J. L. Junkins, “Modified Chebyshev-Picard iteration methods for solution of boundary value problems,” *Journal of the Astronautical Sciences*, vol. 58, no. 4, pp. 615–642, 2011.

[3] X. Bai, *Modified Chebyshev-Picard iteration methods for solution of initial value and boundary value problems [Ph.D. thesis]*, Department of Aerospace Engineering, Texas A&M University, College Station, Tex, USA, 2010.

[4] L. Fox and I. B. Parker, *Chebyshev Polynomials in Numerical Analyis*, Oxford University Press, 1st edition, 1968.

[5] X. Bai and J. L. Junkins, “Modified Chebyshev-Picard iteration methods for solution of initial value problems,” *Journal of the Astronautical Sciences*, vol. 59, no. 1-2, pp. 335–359, 2012.

[6] X. Bai and J. L. Junkins, “Modified Chebyshev-Picard iteration methods for station-keeping of translunar halo orbits,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 926158, 18 pages, 2012.

[7] A. Bani-Younes, *Orthogonal polynomial approximation in higher dimensions: applications in astrodynamics [Ph.D. thesis]*, Department of Aerospace Engineering, Texas A&M University, College Station, Richardson, Tex, USA, 2013.

[8] S. A. Sarra, “Chebyshev interpolation: an interactive tour,” *Journal of Online Mathematics and its Applications*, pp. 1–13, 2006.

[9] P. O. Runck and K. Mahler, “Uber Konvergenzfragen bei Polynomialinterpolation mitäquidistanten Knoten. I,” *Journal fur die reine und angewandte Mathematik (Crelles Journal)*, vol. 208, pp. 51–69, 1961.

[10] D. Kim, J. L. Junkins, J. D. Turner, and A. Bani-Younes, “Multisegment adaptive modified chebyshev picard iteration method,” in *Proceedings of the 24th AAS/AIAA Space Flight Mechanics Meeting*, Santa Fe, NM, USA, January 2014, AAS Paper 14-232.

[11] C. W. Clenshaw and H. J. Norton, “The solution of nonlinear ordinary differential equations in Chebyshev series,” *The Computer Journal*, vol. 6, no. 1, pp. 88–92, 1963.

[12] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*, AIAA, Reston, Va, USA, 2nd edition, 2009.

[13] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Education Series, AIAA: American Institute of Aeronautics and Astronautics, 1999.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 782376, 10 pages
http://dx.doi.org/10.1155/2014/782376

Research Article

A MOLP Method for Solving Fully Fuzzy Linear Programming with *LR* Fuzzy Parameters

Xiao-Peng Yang,[1,2] Xue-Gang Zhou,[1,3] Bing-Yuan Cao,[1] and S. H. Nasseri[4]

[1] *School of Mathematics and Information Science, Key Laboratory of Mathematics and Interdisciplinary Sciences of Guangdong, Higher Education Institutes, Guangzhou University, Guangzhou 510006, China*
[2] *Department of Mathematics and Statistics, Hanshan Normal University, Chaozhou 521041, China*
[3] *Department of Applied Mathematics, Guangdong University of Finance, Guangzhou 510521, China*
[4] *Department of Mathematics, Mazandaran University, Babolsar 47416-95447, Iran*

Correspondence should be addressed to Bing-Yuan Cao; caobingy@163.com

Received 22 March 2014; Accepted 15 September 2014; Published 29 September 2014

Academic Editor: Yang Xu

Kaur and Kumar, 2013, use Mehar's method to solve a kind of fully fuzzy linear programming (FFLP) problems with *LR* fuzzy parameters. In this paper, a new kind of FFLP problems is introduced with a solution method proposed. The FFLP is converted into a multiobjective linear programming (MOLP) according to the order relation for comparing the *LR* flat fuzzy numbers. Besides, the classical fuzzy programming method is modified and then used to solve the MOLP problem. Based on the compromised optimal solution to the MOLP problem, the compromised optimal solution to the FFLP problem is obtained. At last, a numerical example is given to illustrate the feasibility of the proposed method.

1. Introduction

The research on fuzzy linear programming (FLP) has risen highly since Bellman and Zadeh [1] proposed the concept of decision making in fuzzy environment. The FLP problem is said to be a fully fuzzy linear programming (FFLP) problem if all the parameters and variables are considered as fuzzy numbers. In recent years, some researchers such as Lofti and Kumar were interested in the FFLP problems, and some solution methods have been obtained to the fully fuzzy systems [2–4] and the FFLP problems [5–13]. FFLP problems can be divided in two categories: (1) FFLP problems with inequality constraints; (2) FFLP problems with equality constraints. If the FFLP problems are classified by the types of the fuzzy numbers, they will include the next three classes: (1) FFLP problems with all the parameters and variables represented by triangular fuzzy numbers; (2) FFLP problems with all the parameters and variables represented by trapezoidal fuzzy numbers; (3) FFLP problems with all the parameters and variables expressed by *LR* fuzzy numbers (or *LR* flat fuzzy numbers).

Fuzzy programming method is a classical method to solve multiobjective linear programming (MOLP) [14, 15]. In this paper, the fuzzy programming method is modified and then used to obtain a compromised optimal solution of the MOLP. The modified fuzzy programming method is shown in Steps 4–10 of the proposed method in Section 3.

Dehghan et al. [2–4] employed several methods to find solutions of the fully fuzzy linear systems. Hosseinzadeh Lotfi et al. [6] used the lexicography method to obtain the fuzzy approximate solutions of the FFLP problems. Allahviranloo et al. [7] and Kumar et al. [5, 8] solved the FFLP problem by use of a ranking function.

Fan et al. [12] adopted the α-cut level to deal with a generalized fuzzy linear programming (GFLP) probelm. The feasibility of fuzzy solutions to the GFLP was investigated and a stepwise interactive algorithm based on the idea of design of experiment was advanced to solve the GFLP problem.

Kaur and Kumar [9] introduced Mehar's method to the FFLP problems with LR fuzzy parameters. They consider the following model:

$$\begin{aligned}
&\text{Maximize (or Minimize)}\\
&\sum_{j=1}^{n}\left(\left(p_j,q_j,\alpha'_j,\beta'_j\right)_{LR}\odot\left(x_j,y_j,\alpha''_j,\beta''_j\right)_{LR}\right)\\
&\text{subject to}\\
&\sum_{j=1}^{n}\left(\left(a_{ij},b_{ij},\alpha_{ij},\beta_{ij}\right)_{LR}\right.\\
&\qquad\left.\odot\left(x_j,y_j,\alpha''_j,\beta''_j\right)_{LR}\right)\preceq,\approx,\succeq\left(b_i,g_i,\gamma_i,\delta_i\right)_{LR},\\
&\qquad\qquad i=1,2,\ldots,m,
\end{aligned}\tag{1}$$

where the parameters and variables are LR flat fuzzy numbers and the order relation for comparing the numbers is defined as follows.

(i) $\tilde{u}\preceq\tilde{v}$ if and only if $\Re(\tilde{u})\le\Re(\tilde{v})$,

(ii) $\tilde{u}\succeq\tilde{v}$ if and only if $\Re(\tilde{u})\ge\Re(\tilde{v})$,

(iii) $\tilde{u}\approx\tilde{v}$ if and only if $\Re(\tilde{u})=\Re(\tilde{v})$.

Here $\tilde{u}$ and $\tilde{v}$ are two arbitrary LR flat fuzzy numbers.

In our study, we consider a new kind of FFLP problems with LR flat fuzzy parameters as follows:

$$\begin{aligned}
&\min\ (\text{or}\ \max)\\
&\quad z(\tilde{x})=\tilde{c}_1\otimes\tilde{x}_1\oplus\tilde{c}_2\otimes\tilde{x}_2\oplus\cdots\oplus\tilde{c}_n\otimes\tilde{x}_n\\
&\text{subject to}\\
&\quad\tilde{a}_{i1}\otimes\tilde{x}_1\oplus\tilde{a}_{i2}\otimes\tilde{x}_2\oplus\cdots\oplus\tilde{a}_{in}\otimes\tilde{x}_n\le\tilde{b}_i,\\
&\qquad\qquad i=1,2,\ldots,m,\\
&\quad\tilde{x}_j\ge\tilde{0},\quad j=1,2,\ldots,n,
\end{aligned}\tag{2}$$

where the parameters and variables are LR flat fuzzy numbers and the order relation shown in Definition 4 is different from the one above.

In this paper, we modify the classical fuzzy programming method. The FFLP is changed into a MOLP problem solved by the modified fuzzy programming method. We get the compromised optimal solution to the MOLP and generate the corresponding compromised optimal solution to the FFLP.

The rest of the paper is organized as follows. In Section 2, the basic definitions and the FFLP model are introduced. In Section 3, we propose a MOLP method to solve the FFLP problems. Some results are discussed from the solutions obtained by the proposed method. In Section 4, a numerical example is given to illustrate the feasibility of the proposed method. In Section 5, we show some short concluding remarks.

2. Preliminaries

2.1. Basic Notations

Definition 1 (LR fuzzy number, see [2]). A fuzzy number $\tilde{u}$ is said to be an LR fuzzy number if

$$\tilde{u}(x)=\begin{cases}L\left(\dfrac{m-x}{\alpha}\right), & x\le m,\quad \alpha>0,\\[2ex] R\left(\dfrac{x-m}{\beta}\right), & x\ge m,\quad \beta>0,\end{cases}\tag{3}$$

where m is the mean value of $\tilde{u}$ and α and β are left and right spreads, respectively, and function $L(\cdot)$ means the left shape function satisfying

(1) $L(x)=L(-x)$;

(2) $L(0)=1$ and $L(1)=0$;

(3) $L(x)$ is nonincreasing on $[0,\infty)$.

Naturally, a right shape function $R(\cdot)$ is similarly defined as $L(\cdot)$.

Definition 2 (LR flat fuzzy number, see [9, 16]). A fuzzy number $\tilde{u}$, denoted as $(m,n,\alpha,\beta)_{LR}$, is said to be an LR flat fuzzy number if its membership function $\tilde{u}(x)$ is given by

$$\tilde{u}(x)=\begin{cases}L\left(\dfrac{m-x}{\alpha}\right), & x\le m,\ \alpha>0,\\[2ex] R\left(\dfrac{x-n}{\beta}\right), & x\ge n,\ \beta>0,\\[2ex] 1, & m\le x\le n.\end{cases}\tag{4}$$

Definition 3 (see [5, 9]). An LR flat fuzzy number $\tilde{u}=(m,n,\alpha,\beta)_{LR}$ is said to be nonnegative LR flat fuzzy number if $m-\alpha\ge 0$ and is said to be nonpositive LR flat number if $n+\beta\le 0$.

We define $\tilde{u}=(m,n,0,0)_{LR}$ as an LR fuzzy number with membership function

$$\tilde{u}(x)=\begin{cases}1, & m\le x\le n,\\ 0, & \text{otherwise},\end{cases}\tag{5}$$

and denote $(0,0,0,0)_{LR}$ as $\tilde{0}$.

2.2. Arithmetic Operations. Let $\tilde{u} = (m_1, n_1, \alpha_1, \beta_1)_{LR}$ and $\tilde{v} = (m_2, n_2, \alpha_2, \beta_2)_{LR}$ be two *LR* flat fuzzy numbers, $k \in R$. Then the arithmetic operations are given as follows [9, 16]:

$$\tilde{u} \oplus \tilde{v} = (m_1 + m_2, n_1 + n_2, \alpha_1 + \alpha_2, \beta_1 + \beta_2)_{LR},$$

$$\tilde{u} \ominus \tilde{v} = (m_1 - n_2, n_1 - m_2, \alpha_1 + \beta_2, \alpha_2 + \beta_1)_{LR},$$

$$k\tilde{u} = \begin{cases} (km_1, kn_1, k\alpha_1, k\beta_1)_{LR}, & k \geq 0, \\ (kn_1, km_1, -k\beta_1, -k\alpha_1)_{RL}, & k < 0, \end{cases}$$

$$\tilde{u} \otimes \tilde{v} = \begin{cases} (m_1 m_2, n_1 n_2, m_1 \alpha_2 + \alpha_1 m_2, n_1 \beta_2 + \beta_1 n_2)_{LR}, \\ \qquad \tilde{u} \geq \tilde{0},\ \tilde{v} \geq \tilde{0}, \\ (m_1 n_2, n_1 m_2, \alpha_1 n_2 - m_1 \beta_2, \beta_1 m_2 - n_1 \alpha_2)_{LR}, \\ \qquad \tilde{u} \leq \tilde{0},\ \tilde{v} \geq \tilde{0}, \\ (n_1 m_2, m_1 n_2, n_1 \alpha_2 - \beta_1 m_2, m_1 \beta_2 - \alpha_1 n_2)_{LR}, \\ \qquad \tilde{u} \geq \tilde{0},\ \tilde{v} \leq \tilde{0}, \\ (n_1 n_2, m_1 m_2, -n_1 \beta_2 - \beta_1 n_2, -m_1 \alpha_2 - \alpha_1 m_2)_{LR}, \\ \qquad \tilde{u} \leq \tilde{0},\ \tilde{v} \leq \tilde{0}. \end{cases} \tag{6}$$

It is easy to verify that the operator $\oplus$ satisfies associative law. Hence, the formula $\sum_{j=1}^{n} \tilde{u}_j = \tilde{u}_1 \oplus \tilde{u}_2 \oplus \cdots \oplus \tilde{u}_n$ is reasonable, where $\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_n$ are *LR* flat fuzzy numbers.

2.3. Order Relation for Comparing the LR Flat Fuzzy Numbers. For comparing the *LR* flat fuzzy numbers, we introduce the order relation as follows.

Definition 4. Let $\tilde{u} = (m_1, n_1, \alpha_1, \beta_1)_{LR}$ and $\tilde{v} = (m_2, n_2, \alpha_2, \beta_2)_{LR}$ be any *LR* flat fuzzy numbers. Then

(i) $\tilde{u} = \tilde{v}$ if and only if $m_1 = m_2$, $n_1 = n_2$, $\alpha_1 = \alpha_2$, $\beta_1 = \beta_2$;

(ii) $\tilde{u} \leq \tilde{v}$ if and only if $m_1 \leq m_2$, $n_1 \leq n_2$, $m_1 - \alpha_1 \leq m_2 - \alpha_2$, $n_1 + \beta_1 \leq n_2 + \beta_2$;

(iii) $\tilde{u} \geq \tilde{v}$ if and only if $m_1 \geq m_2$, $n_1 \geq n_2$, $m_1 - \alpha_1 \geq m_2 - \alpha_2$, $n_1 + \beta_1 \geq n_2 + \beta_2$.

Based on the definition of order $\leq$, we may obtain that (i) $\tilde{u}$ is nonnegative if and only if $\tilde{u} \geq \tilde{0}$; (ii) $\tilde{u}$ is nonpositive if and only if $\tilde{u} \leq \tilde{0}$.

The following propositions are given to show the properties of the order relation defined above.

Proposition 5. *Let $\tilde{u}, \tilde{v}, \tilde{w}, \tilde{\sigma}$ be four arbitrary LR flat fuzzy numbers and k an arbitrary real number. Then*

$$(1)\ \ \tilde{u} \leq \tilde{v}, \quad \tilde{w} \leq \tilde{\sigma} \Longrightarrow \tilde{u} \oplus \tilde{w} \leq \tilde{v} \oplus \tilde{\sigma};$$

$$(2)\ \ \tilde{u} \leq \tilde{v} \Longrightarrow \begin{cases} k\tilde{u} \leq k\tilde{v}, & k \geq 0, \\ k\tilde{u} \geq k\tilde{v}, & k \leq 0. \end{cases} \tag{7}$$

Proof. Suppose $\tilde{u} = (m_1, n_1, \alpha_1, \beta_1)_{LR}$, $\tilde{v} = (m_2, n_2, \alpha_2, \beta_2)_{LR}$, $\tilde{w} = (m_3, n_3, \alpha_3, \beta_3)_{LR}$, and $\tilde{\sigma} = (m_4, n_4, \alpha_4, \beta_4)_{LR}$.

(1) It is obvious that $\tilde{u} \oplus \tilde{w} = (m_1 + m_3, n_1 + n_3, \alpha_1 + \alpha_3, \beta_1 + \beta_3)_{LR}$, $\tilde{v} \oplus \tilde{\sigma} = (m_2 + m_4, n_2 + n_4, \alpha_2 + \alpha_4, \beta_2 + \beta_4)_{LR}$. Since $\tilde{u} \leq \tilde{v}, \tilde{w} \leq \tilde{\sigma}$, we get

$$\begin{gathered} m_1 \leq m_2, \qquad n_1 \leq n_2, \qquad m_1 - \alpha_1 \leq m_2 - \alpha_2, \\ n_1 + \beta_1 \leq n_2 + \beta_2, \\ m_3 \leq m_4, \qquad n_3 \leq n_4, \qquad m_3 - \alpha_3 \leq m_4 - \alpha_4, \\ n_3 + \beta_3 \leq n_4 + \beta_4. \end{gathered} \tag{8}$$

So

$$\begin{gathered} m_1 + m_3 \leq m_2 + m_4, \\ n_1 + n_3 \leq n_2 + n_4, \\ (m_1 + m_3) - (\alpha_1 + \alpha_3) \leq (m_2 + m_4) - (\alpha_2 + \alpha_4), \\ (n_1 + n_3) + (\beta_1 + \beta_3) \leq (n_2 + n_4) - (\beta_2 + \beta_4). \end{gathered} \tag{9}$$

This indicates that $\tilde{u} \oplus \tilde{w} \leq \tilde{v} \oplus \tilde{\sigma}$.

(2) It is clear that $k\tilde{u} = (km_1, kn_1, k\alpha_1, k\beta_1)_{LR}$, $k\tilde{v} = (km_2, kn_2, k\alpha_2, k\beta_2)_{LR}$. From $\tilde{u} \leq \tilde{v}$, we get

$$\begin{gathered} m_1 \leq m_2, \qquad n_1 \leq n_2, \qquad m_1 - \alpha_1 \leq m_2 - \alpha_2, \\ n_1 + \beta_1 \leq n_2 + \beta_2. \end{gathered} \tag{10}$$

Therefore,

$$\begin{gathered} km_1 \leq km_2, \qquad kn_1 \leq kn_2, \qquad km_1 - k\alpha_1 \leq km_2 - k\alpha_2, \\ kn_1 + k\beta_1 \leq kn_2 + k\beta_2, \end{gathered} \tag{11}$$

for $k \geq 0$, and

$$\begin{gathered} km_1 \geq km_2, \qquad kn_1 \geq kn_2, \qquad km_1 - k\alpha_1 \geq km_2 - k\alpha_2, \\ kn_1 + k\beta_1 \geq kn_2 + k\beta_2, \end{gathered} \tag{12}$$

for $k \leq 0$. This indicates that

$$\begin{gathered} k\tilde{u} \leq k\tilde{v}, \quad k \geq 0, \\ k\tilde{u} \geq k\tilde{v}, \quad k \leq 0. \end{gathered} \tag{13}$$

□

Proposition 6. *Let $\tilde{u}, \tilde{v}, \tilde{w}$ be three arbitrary LR flat fuzzy numbers. Then*

(1) $\tilde{u} \leq \tilde{u}$;

(2) $\tilde{u} \leq \tilde{v}, \tilde{v} \leq \tilde{u} \Rightarrow \tilde{u} = \tilde{v}$;

(3) $\tilde{u} \leq \tilde{v}, \tilde{v} \leq \tilde{w} \Rightarrow \tilde{u} \leq \tilde{w}$.

Proof. Suppose $\tilde{u} = (m_1, n_1, \alpha_1, \beta_1)_{LR}$, $\tilde{v} = (m_2, n_2, \alpha_2, \beta_2)_{LR}$, and $\tilde{w} = (m_3, n_3, \alpha_3, \beta_3)_{LR}$.

(1) Obviously, $\tilde{u} = \tilde{u}$; hence, we have $\tilde{u} \leq \tilde{u}$.

(2) Since $\widetilde{u} \leq \widetilde{v}, \widetilde{v} \leq \widetilde{u}$, we get

$$\begin{aligned} &m_1 \leq m_2, \quad n_1 \leq n_2, \quad m_1 - \alpha_1 \leq m_2 - \alpha_2, \\ &n_1 + \beta_1 \leq n_2 + \beta_2, \\ &m_2 \leq m_1, \quad n_2 \leq n_1, \quad m_2 - \alpha_2 \leq m_1 - \alpha_1, \\ &n_2 + \beta_2 \leq n_1 + \beta_1. \end{aligned} \tag{14}$$

This means

$$\begin{aligned} &m_1 = m_2, \quad n_1 = n_2, \quad m_1 - \alpha_1 = m_2 - \alpha_2, \\ &n_1 + \beta_1 = n_2 + \beta_2. \end{aligned} \tag{15}$$

That is

$$m_1 = m_2, \quad n_1 = n_2, \quad \alpha_1 = \alpha_2, \quad \beta_1 = \beta_2. \tag{16}$$

Therefore, we have $\widetilde{u} = \widetilde{v}$.

(3) From $\widetilde{u} \leq \widetilde{v}, \widetilde{v} \leq \widetilde{w}$, we get

$$\begin{aligned} &m_1 \leq m_2, \quad n_1 \leq n_2, \quad m_1 - \alpha_1 \leq m_2 - \alpha_2, \\ &n_1 + \beta_1 \leq n_2 + \beta_2, \\ &m_2 \leq m_3, \quad n_2 \leq n_3, \quad m_2 - \alpha_2 \leq m_3 - \alpha_3, \\ &n_2 + \beta_2 \leq n_3 + \beta_3. \end{aligned} \tag{17}$$

This indicates

$$\begin{aligned} &m_1 \leq m_3, \quad n_1 \leq n_3, \quad m_1 - \alpha_1 \leq m_3 - \alpha_3, \\ &n_1 + \beta_1 \leq n_3 + \beta_3. \end{aligned} \tag{18}$$

Therefore, we have $\widetilde{u} \leq \widetilde{w}$. □

From Proposition 6, we know that the order relation ≤ is a partial order on the set of all *LR* fuzzy numbers.

2.4. Fully Fuzzy Linear Programming with LR Fuzzy Parameters. In this paper, we will consider the following model; that is,

$$\begin{aligned} \min \quad & z(\widetilde{x}) = \widetilde{c} \otimes \widetilde{x} \\ \text{s.t.} \quad & \widetilde{A} \otimes \widetilde{x} \leq \widetilde{b} \\ & \widetilde{x} \geq \widetilde{0}, \end{aligned} \tag{19}$$

or

$$\begin{aligned} \min \quad & z(\widetilde{x}) = \widetilde{c}_1 \otimes \widetilde{x}_1 \oplus \widetilde{c}_2 \otimes \widetilde{x}_2 \oplus \cdots \oplus \widetilde{c}_n \otimes \widetilde{x}_n \\ \text{s.t.} \quad & \widetilde{a}_{i1} \otimes \widetilde{x}_1 \oplus \widetilde{a}_{i2} \otimes \widetilde{x}_2 \oplus \cdots \oplus \widetilde{a}_{in} \otimes \widetilde{x}_n \leq \widetilde{b}_i, \\ & \qquad i = 1, 2, \ldots, m, \\ & \widetilde{x}_j \geq \widetilde{0}, \quad j = 1, 2, \ldots, n, \end{aligned} \tag{20}$$

where $\widetilde{c} = [\widetilde{c}_j]_{1\times n}$, $\widetilde{b} = [\widetilde{b}_i]_{m\times 1}$, $\widetilde{A} = [\widetilde{a}_{ij}]_{m\times n}$, and $\widetilde{x} = [\widetilde{x}_j]_{n\times 1}$ represent *LR* fuzzy matrices and vectors and $\widetilde{c}_j, \widetilde{b}_i, \widetilde{a}_{ij}$, and $\widetilde{x}_j$ are *LR* flat fuzzy numbers. The order relations for comparing the *LR* flat fuzzy numbers both in the objective function and the constraint inequalities are as shown in Definition 4.

3. Proposed Method

Steps of the proposed method are given to solve problem (20) as follows. This method is applicable to minimization of FFLP problems, and the solution method of maximization problems is similar to that of minimization ones.

Step 1. If all the parameters $\widetilde{c}_j, \widetilde{b}_i, \widetilde{a}_{ij}, \widetilde{x}_j$ are represented by *LR* flat fuzzy numbers $(c_{j1}, c_{j2}, \alpha_{c_j}, \beta_{c_j})_{LR}$, $(b_{i1}, b_{i2}, \alpha_{b_i}, \beta_{b_i})_{LR}$, $(a_{ij1}, a_{ij2}, \alpha_{a_{ij}}, \beta_{a_{ij}})_{LR}$, and $(x_{j1}, x_{j2}, \alpha_{x_j}, \beta_{x_j})_{LR}$, then the FFLP (20) can be written as

$$\begin{aligned} \min \quad & z(\widetilde{x}) = \sum_{j=1}^{n} \Big(\big(c_{j1}, c_{j2}, \alpha_{c_j}, \beta_{c_j}\big)_{LR} \\ & \qquad \otimes \big(x_{j1}, x_{j2}, \alpha_{x_j}, \beta_{x_j}\big)_{LR} \Big) \\ \text{s.t.} \quad & \sum_{j=1}^{n} \Big(\big(a_{ij1}, a_{ij2}, \alpha_{a_{ij}}, \beta_{a_{ij}}\big)_{LR} \otimes \big(x_{j1}, x_{j2}, \alpha_{x_j}, \beta_{x_j}\big)_{LR} \Big) \\ & \quad \leq \big(b_{i1}, b_{i2}, \alpha_{b_i}, \beta_{b_i}\big)_{LR}, \quad i = 1, 2, \ldots, m, \\ & \big(x_{j1}, x_{j2}, \alpha_{x_j}, \beta_{x_j}\big)_{LR} \geq (0, 0, 0, 0,)_{LR}, \\ & \qquad j = 1, 2, \ldots, n. \end{aligned} \tag{21}$$

Step 2. Calculate $(c_{j1}, c_{j2}, \alpha_{c_j}, \beta_{c_j})_{LR} \otimes (x_{j1}, x_{j2}, \alpha_{x_j}, \beta_{x_j})_{LR}$ and $(a_{ij1}, a_{ij2}, \alpha_{a_{ij}}, \beta_{a_{ij}})_{LR} \otimes (x_{j1}, x_{j2}, \alpha_{x_j}, \beta_{x_j})_{LR}$, respectively, and suppose that $(c_{j1}, c_{j2}, \alpha_{c_j}, \beta_{c_j})_{LR} \otimes (x_{j1}, x_{j2}, \alpha_{x_j}, \beta_{x_j})_{LR} = (x'_{j1}, x'_{j2}, \alpha'_{x_j}, \beta'_{x_j})_{LR}$ and $(a_{ij1}, a_{ij2}, \alpha_{a_{ij}}, \beta_{a_{ij}})_{LR} \otimes (x_{j1}, x_{j2}, \alpha_{x_j}, \beta_{x_j})_{LR} = (p_{ij}, q_{ij}, \gamma_{ij}, \delta_{ij})_{LR}$; then the FFLP problem obtained in Step 1 can be written as

$$\begin{aligned} \min \quad & z(\widetilde{x}) = \left(\sum_{j=1}^{n} x'_{j1}, \sum_{j=1}^{n} x'_{j2}, \sum_{j=1}^{n} \alpha'_{x_j}, \sum_{j=1}^{n} \beta'_{x_j} \right)_{LR} \\ \text{s.t.} \quad & \left(\sum_{j=1}^{n} p_{ij}, \sum_{j=1}^{n} q_{ij}, \sum_{j=1}^{n} \gamma_{ij}, \sum_{j=1}^{n} \delta_{ij} \right)_{LR} \leq \big(b_{i1}, b_{i2}, \alpha_{b_i}, \beta_{b_i}\big)_{LR}, \\ & \qquad i = 1, 2, \ldots, m, \\ & \big(x_{j1}, x_{j2}, \alpha_{x_j}, \beta_{x_j}\big)_{LR} \geq (0, 0, 0, 0,)_{LR}, \\ & \qquad j = 1, 2, \ldots, n. \end{aligned} \tag{22}$$

Step 3. According to the order relation defined above, the problem obtained in Step 2 is equivalent to

$$\begin{aligned}
\min \quad & \sum_{j=1}^{n} x'_{j1} \\
& \sum_{j=1}^{n} x'_{j2} \\
& \sum_{j=1}^{n} \left(x'_{j1} - \alpha'_{x_j} \right) \\
& \sum_{j=1}^{n} \left(x'_{j2} + \beta'_{x_j} \right) \\
\text{s.t.} \quad & \sum_{j=1}^{n} p_{ij} \leq b_{i1}, \quad i = 1, 2, \ldots, m, \\
& \sum_{j=1}^{n} q_{ij} \leq b_{i2}, \quad i = 1, 2, \ldots, m, \\
& \sum_{j=1}^{n} \left(p_{ij} - \gamma_{ij} \right) \leq \left(b_{i1} - \alpha_{b_i} \right), \quad i = 1, 2, \ldots, m, \\
& \sum_{j=1}^{n} \left(q_{ij} + \delta_{ij} \right) \leq \left(b_{i2} + \beta_{b_i} \right), \quad i = 1, 2, \ldots, m, \\
& x_{j1} \leq x_{j2}, \qquad \alpha_{x_j} \geq 0, \qquad \beta_{x_j} \geq 0, \\
& x_{j1} - \alpha_{x_j} \geq 0, \quad j = 1, 2, \ldots, n.
\end{aligned} \tag{23}$$

We denote $X = (x_{11}, x_{12}, \alpha_{x_1}, \beta_{x_1}, x_{21}, x_{22}, \alpha_{x_2}, \beta_{x_2}, \ldots, x_{n1}, x_{n2}, \alpha_{x_n}, \beta_{x_n})^T$, $z_1(X) = \sum_{j=1}^{n} x'_{j1}$, $z_2(X) = \sum_{j=1}^{n} x'_{j2}$, $z_3(X) = \sum_{j=1}^{n} (x'_{j1} - \alpha'_{x_j})$, $z_4(X) = \sum_{j=1}^{n} (x'_{j2} + \beta'_{x_j})$ and $D = \{X \mid X$ satisfies the constraints of programming (23)$\}$. Programming (23) may be written as the programming (24), below for short, as follows:

$$\begin{aligned}
\min \quad & z_1(X) \\
& z_2(X) \\
& z_3(X) \\
& z_4(X) \\
\text{s.t.} \quad & X \in D.
\end{aligned} \tag{24}$$

Obviously, programming (24) is a crisp multiobjective linear programming problem. In fact we have $z(\widetilde{x}) = (z_1(X), z_2(X), z_1(X) - z_3(X), z_4(X) - z_2(X))$.

Step 4. Solve the subproblems

$$\begin{aligned}
\min \quad & z_t(X) \\
\text{s.t.} \quad & X \in D,
\end{aligned} \tag{25}$$

where $t = 1, 2, 3, 4$. We find optimal solutions X_1, X_2, X_3, and X_4, respectively. And the corresponding optimal values will be $z_1^{\min} = z_1(X_1)$, $z_2^{\min} = z_2(X_2)$, $z_3^{\min} = z_3(X_3)$, and $z_4^{\min} = z_4(X_4)$.

Step 5. Let $z_t^{\max} = \max\{z_t(X_1), z_t(X_2), z_t(X_3), z_t(X_4)\}$, $t = 1, 2, 3, 4$, and the membership function of $z_t(X)$ is given by

$$\mu_{z_t}(z_t(X)) = \begin{cases} 1, & z_t(X) < z_t^{\min}, \\ \dfrac{z_t^{\max} - z_t(X)}{z_t^{\max} - z_t^{\min}}, & z_t^{\min} \leq z_t(X) \leq z_t^{\max}, \\ 0, & z_t(X) > z_t^{\max}, \end{cases} \tag{26}$$

where $t = 1, 2, 3, 4$.

Step 6. Let $I_0 = \{1, 2, 3, 4\}$; the MOLP problem obtained in Step 3 can be equivalently written as

$$\begin{aligned}
\max \quad & \lambda \\
\text{s.t.} \quad & \mu_{z_t}(z_t(X)) \geq \lambda, \quad t \in I_0, \\
& X \in D.
\end{aligned} \tag{27}$$

Suppose X^1 is one of the optimal solutions (if there exits only one optimal solution, X^1 is the unique one), and λ^{1*} is the optimal objective value (in fact, the optimal solution should be written as (X^1, λ^{1*}). Since λ is an auxiliary variable, we denote (X^1, λ^{1*}) as X^1 for simplicity). Then $\mu_{z_{s_1}}(z_{s_1}(X^1)) = \lambda^{1*}$ for at least one s_1 in I_0. (s_1 is an arbitrary element in the set $J = \{j \mid \mu_{z_j}(z_j(X^1)) = \lambda^{1*}\}$).

Step 7. Let $I_1 = I_0 - \{s_1\}$, and solve the following crisp programming:

$$\begin{aligned}
\max \quad & \lambda \\
\text{s.t.} \quad & \mu_{z_t}(z_t(X)) \geq \lambda, \quad t \in I_1, \\
& \mu_{z_{s_1}}\left(z_{s_1}(X)\right) = \lambda^{1*}, \\
& X \in D.
\end{aligned} \tag{28}$$

If X^2 is one of the optimal solutions and λ^{2*} is the optimal objective value, then $\mu_{z_{s_2}}(z_{s_2}(X^2)) = \lambda^{2*}$ for at least one s_2 in I_1.

Step 8. Let $I_2 = I_0 - \{s_1, s_2\}$, and solve the following crisp programming:

$$\begin{aligned}
\max \quad & \lambda \\
\text{s.t.} \quad & \mu_{z_t}(z_t(X)) \geq \lambda, \quad t \in I_2, \\
& \mu_{z_{s_1}}\left(z_{s_1}(X)\right) = \lambda^{1*}, \\
& \mu_{z_{s_2}}\left(z_{s_2}(X)\right) = \lambda^{2*}, \\
& X \in D.
\end{aligned} \tag{29}$$

Suppose X^3 is one of the optimal solutions and λ^{3*} is the optimal objective value. Then $\mu_{z_{s_3}}(z_{s_3}(X^3)) = \lambda^{3*}$ for at least one s_3 in I_2.

Step 9. Let $I_3 = I_0 - \{s_1, s_2, s_3\}$, and solve the following crisp programming:

$$\begin{aligned}
\max \quad & \lambda \\
\text{s.t.} \quad & \mu_{z_t}(z_t(X)) \geq \lambda, \quad t \in I_3, \\
& \mu_{z_{s_1}}\left(z_{s_1}(X)\right) = \lambda^{1*}, \\
& \mu_{z_{s_2}}\left(z_{s_2}(X)\right) = \lambda^{2*}, \\
& \mu_{z_{s_3}}\left(z_{s_3}(X)\right) = \lambda^{3*}, \\
& X \in D.
\end{aligned} \tag{30}$$

Suppose X^4 is one of the optimal solutions, and λ^{4*} is the optimal objective value. Then $\mu_{z_{s_4}}(z_{s_4}(X^4)) = \lambda^{4*}$ with s_4 in I_3.

Step 10. Take $X^* = X^4$ as the compromised optimal solution to programming (23) and generate the compromised optimal solution $\tilde{x}^*$ to programming (21) by X^*. Assuming

$$\begin{aligned}
X^* = & \left(x_{11}^*, x_{12}^*, \alpha_{x_1}^*, \beta_{x_1}^*, x_{21}, x_{22}^*, \alpha_{x_2}^*,\right. \\
& \left.\beta_{x_2}^*, \ldots, x_{n1}, x_{n2}^*, \alpha_{x_n}^*, \beta_{x_n}^*\right)^T,
\end{aligned} \tag{31}$$

we may obtain

$$\begin{aligned}
\tilde{x}^* &= (\tilde{x}_1^*, \tilde{x}_2^*, \ldots, \tilde{x}_n^*)^T \\
&= \left(\left(x_{11}^*, x_{12}^*, \alpha_{x_1}^*, \beta_{x_1}^*\right)_{LR}, \left(x_{21}^*, x_{22}^*, \alpha_{x_2}^*, \beta_{x_2}^*\right)_{LR},\right. \\
&\quad \left.\ldots, \left(x_{n1}^*, x_{n2}^*, \alpha_{x_n}^*, \beta_{x_n}^*\right)_{LR}\right)^T,
\end{aligned} \tag{32}$$

and the corresponding objective value $z^* = z(\tilde{x}^*)$.

Remark 7 ($\{s_1, s_2, s_3, s_4\} = I = \{1, 2, 3, 4\}$). Some properties of the solutions obtained in Steps 6–10 are shown in the following proposition.

Proposition 8. *Suppose* $\mu_{z_{s_j}}(z_{s_j}(X^*))$, X^j, λ^{j*} $(j = 1, 2, 3, 4)$, *and* X^* *are the notations described in Steps 1–10; then*

$$\begin{aligned}
(1)\ & \mu_{z_{s_1}}\left(z_{s_1}\left(X^*\right)\right) = \lambda^{1*} = \mu_{z_{s_1}}\left(z_{s_1}\left(X^1\right)\right), \\
& \mu_{z_{s_2}}\left(z_{s_2}\left(X^*\right)\right) = \lambda^{2*} = \mu_{z_{s_2}}\left(z_{s_2}\left(X^2\right)\right), \\
& \mu_{z_{s_3}}\left(z_{s_3}\left(X^*\right)\right) = \lambda^{3*} = \mu_{z_{s_3}}\left(z_{s_3}\left(X^3\right)\right), \\
& \mu_{z_{s_4}}\left(z_{s_4}\left(X^*\right)\right) = \lambda^{4*} = \mu_{z_{s_4}}\left(z_{s_4}\left(X^4\right)\right),
\end{aligned} \tag{33}$$

(2) $\lambda^{1*} \leq \lambda^{2*} \leq \lambda^{3*} \leq \lambda^{4*}$.

Proof. (1) From the results of Steps 6–9, it is obviously clear that

$$\begin{aligned}
\lambda^{1*} &= \mu_{z_{s_1}}\left(z_{s_1}\left(X^1\right)\right), \\
\lambda^{2*} &= \mu_{z_{s_2}}\left(z_{s_2}\left(X^2\right)\right), \\
\lambda^{3*} &= \mu_{z_{s_3}}\left(z_{s_3}\left(X^3\right)\right), \\
\lambda^{4*} &= \mu_{z_{s_4}}\left(z_{s_4}\left(X^4\right)\right),
\end{aligned} \tag{34}$$

and $\mu_{z_{s_4}}(z_{s_4}(X^*)) = \lambda^{4*}$ with $X^* = X^4$. Since $X^* = X^4$ is an optimal solution to programming (30), we know that X^* satisfies the constraints of programming (30), and so $\mu_{z_{s_1}}(z_{s_1}(X^*)) = \lambda^{1*}$, $\mu_{z_{s_2}}(z_{s_2}(X^*)) = \lambda^{2*}$, and $\mu_{z_{s_3}}(z_{s_3}(X^*)) = \lambda^{3*}$.

(2) In fact, (X^1, λ^{1*}) is an optimal solution to programming (27); therefore, it is a feasible solution. We have

$$\begin{aligned}
& \mu_{z_t}\left(z_t\left(X^1\right)\right) \geq \lambda^{1*}, \quad t \in I_1 \subseteq I_0, \\
& X^1 \in D,
\end{aligned} \tag{35}$$

and it is obvious that $\mu_{z_{s_1}}(z_{s_1}(X^*)) = \lambda^{1*}$ from the result of Step 6. Hence, (X^1, λ^{1*}) is a feasible solution to programming (28). The objective value of (X^1, λ^{1*}) is λ^{1*}, and the optimal objective value of programming (28) is λ^{2*}; so we get $\lambda^{1*} \leq \lambda^{2*}$. It is similar to prove $\lambda^{2*} \leq \lambda^{3*}$ and $\lambda^{3*} \leq \lambda^{4*}$. □

4. Numerical Example

In this section, we present a numerical example to illustrate the feasibility of the solution method proposed in Section 3.

We aim to find the compromised optimal solution and corresponding objective value of the following fully fuzzy linear programming problem:

$$\begin{aligned}
\max \quad & z(\tilde{x}) = z(\tilde{x}_1, \tilde{x}_2) \\
& \qquad = (6, 7, 1, 2)_{LR} \otimes \tilde{x}_1 \oplus (7, 9, 1, 2)_{LR} \otimes \tilde{x}_2 \\
\text{s.t.} \quad & (9, 10, 2, 1)_{LR} \otimes \tilde{x}_1 \oplus (1, 1, 1, 1)_{LR} \otimes \tilde{x}_2 \\
& \quad \leq (50, 55, 4, 3)_{LR}, \\
& (2, 3, 1, 1)_{LR} \otimes \tilde{x}_1 \oplus (4, 5, 1, 2)_{LR} \otimes \tilde{x}_2 \\
& \quad \leq (66, 70, 3, 5)_{LR}, \\
& \tilde{x}_1 \geq \tilde{0}, \qquad \tilde{x}_2 \geq \tilde{0},
\end{aligned} \tag{36}$$

where $\tilde{x}_1 = (x_{11}, x_{12}, \alpha_1, \beta_1)_{LR}$ and $\tilde{x}_2 = (x_{21}, x_{22}, \alpha_2, \beta_2)_{LR}$.

Table 1: The optimal values and solutions of the four subproblems.

	The optimal objective value	The optimal solution
(a)	$z_1^{\max} = 70$	$X_1 = (0, 0.3223, 0, 0.2887, 10, 10.1163, 7.0903, 0.1898)^T$
(b)	$z_2^{\max} = 103.3188$	$X_2 = (1.6015, 3.7101, 1.6039, 0, 4.2500, 8.5942, 3.1084, 0)^T$
(c)	$z_3^{\max} = 50$	$X_3 = (0, 0.2544, 0, 0.2973, 10, 10.0893, 0, 0.2179)^T$
(d)	$z_4^{\max} = 142.5882$	$X_4 = (1.4272, 4.2157, 1.2516, 0, 0, 0, 0, 11.6275)^T$

According to Steps 1 and 2 in the proposed method, we obtain the following programming:

$$\begin{aligned}
\max \quad & z = \big(6x_{11} + 7x_{21}, 7x_{12} + 9x_{22}, 6\alpha_1 + 7\alpha_2 \\
& \quad + x_{11} + 2x_{21}, 7\beta_1 + 9\beta_2 + 2x_{12} + x_{22}\big)_{LR} \\
\text{s.t.} \quad & \big(9x_{11} + x_{21}, 10x_{12} + x_{22}, 9\alpha_1 + \alpha_2 + 2x_{11} \\
& \quad + x_{21}, 10\beta_1 + \beta_2 + x_{12} + x_{22}\big)_{LR} \\
& \quad \le (50, 55, 4, 3)_{LR}, \\
& \big(2x_{11} + 4x_{21}, 3x_{12} + 5x_{22}, 2\alpha_1 + 4\alpha_2 + x_{11} \\
& \quad + x_{21}, 3\beta_1 + 5\beta_2 + x_{12} + 2x_{22}\big)_{LR} \\
& \quad \le (66, 70, 3, 5)_{LR}, \\
& \big(x_{11}, x_{12}, \alpha_1, \beta_1\big)_{LR} \ge (0, 0, 0, 0)_{LR}, \\
& \big(x_{21}, x_{22}, \alpha_2, \beta_2\big)_{LR} \ge (0, 0, 0, 0)_{LR}.
\end{aligned} \tag{37}$$

By Step 3, the programming above is transformed into the following programming:

$$\begin{aligned}
\max \quad & z_1 = 6x_{11} + 7x_{21} \\
& z_2 = 7x_{12} + 9x_{22} \\
& z_3 = 5x_{11} + 5x_{21} - 6\alpha_1 - 7\alpha_2 \\
& z_4 = 9x_{12} + 10x_{22} + 7\beta_1 + 9\beta_2 \\
\text{s.t.} \quad & 9x_{11} + x_{21} \le 50, \\
& 10x_{12} + x_{22} \le 55, \\
& 7x_{11} - 9\alpha_1 - \alpha_2 \le 46, \\
& 11x_{12} + 2x_{22} + 10\beta_1 + \beta_2 \le 58, \\
& 2x_{11} + 4x_{21} \le 66, \\
& 3x_{12} + 5x_{22} \le 70, \\
& x_{11} + 3x_{21} - 2\alpha_1 - 4\alpha_2 \le 63, \\
& 4x_{12} + 7x_{22} + 3\beta_1 + 5\beta_2 \le 75, \\
& x_{11} - \alpha_1 \ge 0, \qquad x_{21} - \alpha_2 \ge 0, \\
& x_{11} \le x_{12}, \qquad x_{21} \le x_{22}, \\
& \alpha_1, \alpha_2, \beta_1, \beta_2 \ge 0.
\end{aligned} \tag{38}$$

Programming (38) can be abbreviated to the following programming:

$$\begin{aligned}
\max \quad & z_1(X) \\
& z_2(X) \\
& z_3(X) \\
& z_4(X) \\
\text{s.t.} \quad & X \in D,
\end{aligned} \tag{39}$$

where $X = (x_{11}, x_{12}, \alpha_1, \beta_1, x_{21}, x_{22}, \alpha_2, \beta_2)^T$.

Solve the following subproblems:

$$\begin{aligned}
\text{(a)} \ \max \quad & z_1(X) \\
\text{s.t.} \quad & X \in D, \\
\text{(b)} \ \max \quad & z_2(X) \\
\text{s.t.} \quad & X \in D,
\end{aligned} \tag{40}$$

$$\begin{aligned}
\text{(c)} \ \max \quad & z_3(X) \\
\text{s.t.} \quad & X \in D, \\
\text{(d)} \ \max \quad & z_4(X) \\
\text{s.t.} \quad & X \in D,
\end{aligned} \tag{41}$$

respectively, and we obtain the optimal objective value and one of the optimal solutions as shown in Table 1.

According to $z_t^{\min} = \min\{z_t(X_1), z_t(X_2), z_t(X_3), z_t(X_4)\}$, we acquire the lower objective values $z_1^{\min} = 8.5631$, $z_2^{\min} = 29.5098$, $z_3^{\min} = 0.3681$, and $z_4^{\min} = 107.2245$, with corresponding membership functions given below. Consider

$$\mu_{z_1}(z_1(X)) = \begin{cases} 1, & z_1(X) > 70; \\ \dfrac{z_1(X) - 8.5631}{70 - 8.5631}, & 8.5631 \le z_1(X) \le 70; \\ 0, & z_1(X) < 8.5631, \end{cases}$$

$$\begin{aligned}
\mu_{z_2}(z_2(X)) &= \begin{cases} 1, & \\ \quad z_2(X) > 103.3188; & \\ \dfrac{z_2(X) - 29.5098}{103.3188 - 29.5098}, & \\ \quad 29.5098 \le z_2(X) \le 103.3188; & \\ 0, & \\ \quad z_2(X) < 29.5098, & \end{cases} \\
\mu_{z_3}(z_3(X)) &= \begin{cases} 1, & \\ \quad z_3(X) > 50; & \\ \dfrac{z_3(X) - 0.3681}{50 - 0.3681}, & \\ \quad 0.3681 \le z_3(X) \le 50; & \\ 0, & \\ \quad z_3(X) < 0.3681, & \end{cases} \\
\mu_{z_4}(z_4(X)) &= \begin{cases} 1, & \\ \quad z_4(X) > 142.5882; & \\ \dfrac{z_4(X) - 107.2245}{142.5882 - 107.2245}, & \\ \quad 107.2245 \le z_4(X) \le 142.5882; & \\ 0, & \\ \quad z_4(X) < 107.2245. & \end{cases}
\end{aligned} \tag{42}$$

By Steps 4–6, we get

$$\begin{aligned}
\max \quad & \lambda \\
\text{s.t.} \quad & \mu_{z_1}(z_1(X)) = \frac{6x_{11} + 7x_{21} - 8.5631}{70 - 8.5631} \ge \lambda, \\
& \mu_{z_2}(z_2(X)) = \frac{7x_{12} + 9x_{22} - 29.5098}{103.3188 - 29.5098} \ge \lambda, \\
& \mu_{z_3}(z_3(X)) \\
& \quad = \frac{5x_{11} + 5x_{21} - 6\alpha_1 - 7\alpha_2 - 0.3681}{50 - 0.3681} \ge \lambda, \\
& \mu_{z_4}(z_4(X)) \\
& \quad = \frac{9x_{12} + 10x_{22} + 7\beta_1 + 9\beta_2 - 107.2245}{142.5882 - 107.2245} \ge \lambda, \\
& X \in D.
\end{aligned} \tag{43}$$

The optimal objective value is $\lambda^{1*} = 0.6033$, and one of the optimal solutions is $X^1 = (2.5901, 3.9107, 0.4927, 0, 4.7997, 5.1848, 0.3109, 4.6127)^T$.

Calculate the value of the membership function of $z_t(X)$ $(t = 1, 2, 3, 4)$ at $X = X^1$, and we get $\mu_{z_1}(z_1(X^1)) = 0.6604$, $\mu_{z_2}(z_2(X^1)) = 0.6033$, $\mu_{z_3}(z_3(X^1)) = 0.6336$, and $\mu_{z_4}(z_4(X^1)) = 0.6033$.

Solve the following problem:

$$\begin{aligned}
\max \quad & \lambda \\
\text{s.t.} \quad & \mu_{z_1}(z_1(X)) = \frac{6x_{11} + 7x_{21} - 8.5631}{70 - 8.5631} \ge \lambda, \\
& \mu_{z_2}(z_2(X)) = \frac{7x_{12} + 9x_{22} - 29.5098}{103.3188 - 29.5098} \ge \lambda, \\
& \mu_{z_3}(z_3(X)) \\
& \quad = \frac{5x_{11} + 5x_{21} - 6\alpha_1 - 7\alpha_2 - 0.3681}{50 - 0.3681} \ge \lambda, \\
& \mu_{z_4}(z_4(X)) \\
& \quad = \frac{9x_{12} + 10x_{22} + 7\beta_1 + 9\beta_2 - 107.2245}{142.5882 - 107.2245} = 0.6033, \\
& X \in D.
\end{aligned} \tag{44}$$

The optimal objective value is $\lambda^{2*} = 0.6033$, and one of the optimal solutions is $X^2 = (2.8, 3.9107, 0.3135, 0, 4.7161, 5.1850, 0.4310, 4.6124)^T$.

Calculate the value of the membership function of $z_t(X)$ $(t = 1, 2, 3, 4)$ at $X = X^2$, and we get $\mu_{z_1}(z_1(X^2)) = 0.6714$, $\mu_{z_2}(z_2(X^2)) = 0.6033$, $\mu_{z_3}(z_3(X^2)) = 0.6511$, and $\mu_{z_4}(z_4(X^2)) = 0.6033$.

Solve the following problem:

$$\begin{aligned}
\max \quad & \lambda \\
\text{s.t.} \quad & \mu_{z_1}(z_1(X)) = \frac{6x_{11} + 7x_{21} - 8.5631}{70 - 8.5631} \ge \lambda, \\
& \mu_{z_3}(z_3(X)) \\
& \quad = \frac{5x_{11} + 5x_{21} - 6\alpha_1 - 7\alpha_2 - 0.3681}{50 - 0.3681} \ge \lambda, \\
& \mu_{z_2}(z_2(X)) = \frac{7x_{12} + 9x_{22} - 29.5098}{103.3188 - 29.5098} = 0.6033, \\
& \mu_{z_4}(z_4(X)) \\
& \quad = \frac{9x_{12} + 10x_{22} + 7\beta_1 + 9\beta_2 - 107.2245}{142.5882 - 107.2245} = 0.6033, \\
& X \in D.
\end{aligned} \tag{45}$$

The optimal objective value is $\lambda^{3*} = 0.7126$, and one of the optimal solutions is $X^3 = (2.6750, 3.9107, 0.1158, 0.0005, 5.1850, 5.1850, 0.2608, 4.6125)^T$.

Calculate the value of the membership function of $z_t(X)$ $(t = 1, 2, 3, 4)$ at $X = X^3$, and we get $\mu_{z_1}(z_1(X^3)) = 0.7126$, $\mu_{z_2}(z_2(X^3)) = 0.6033$, $\mu_{z_3}(z_3(X^3)) = 0.7274$, and $\mu_{z_4}(z_4(X^3)) = 0.6033$.

Table 2: Values of the four membership functions at X^j.

	$\mu_{z_1}(z_1(X))$	$\mu_{z_2}(z_2(X))$	$\mu_{z_3}(z_3(X))$	$\mu_{z_4}(z_4(X))$
$X = X^1$	0.6604	0.6033	0.6336	0.6033
$X = X^2$	0.6714	0.6033	0.6511	0.6033
$X = X^3$	0.7126	0.6033	0.7274	0.6033
$X = X^4$	0.7126	0.6033	0.7844	0.6033

Table 3: Values of the objective function $z(\tilde{x})$ at $\tilde{x}^j$.

	$z(\tilde{x})$
$\tilde{x} = \tilde{x}^1$	$(49.13, 74.03, 17.32, 38.96)_{LR}$
$\tilde{x} = \tilde{x}^2$	$(49.81, 74.03, 17.13, 38.96)_{LR}$
$\tilde{x} = \tilde{x}^3$	$(52.34, 74.03, 15.56, 38.96)_{LR}$
$\tilde{x} = \tilde{x}^4$	$(52.34, 74.03, 13.04, 38.96)_{LR}$

Solve the following problem:

$$\begin{aligned}
\max \quad & \lambda \\
\text{s.t.} \quad & \mu_{z_3}(z_3(X)) = \frac{5x_{11} + 5x_{21} - 6\alpha_1 - 7\alpha_2 - 0.3681}{50 - 0.3681} \geq \lambda, \\
& \mu_{z_1}(z_1(X)) = \frac{6x_{11} + 7x_{21} - 8.5631}{70 - 8.5631} = 0.7126, \\
& \mu_{z_2}(z_2(X)) = \frac{7x_{12} + 9x_{22} - 29.5098}{103.3188 - 29.5098} = 0.6033, \\
& \mu_{z_4}(z_4(X)) \\
& \quad = \frac{9x_{12} + 10x_{22} + 7\beta_1 + 9\beta_2 - 107.2245}{142.5882 - 107.2245} = 0.6033, \\
& X \in D.
\end{aligned} \tag{46}$$

The optimal objective value is $\lambda^{4*} = 0.7844$, and one of the optimal solutions is $X^4 = (2.6754, 3.9106, 0, 0, 5.1843, 5.1848, 0, 4.6128)^T$.

Calculate the value of the membership function of $z_t(X)$ ($t = 1, 2, 3, 4$) at $X = X^4$, and we get $\mu_{z_1}(z_1(X^4)) = 0.7126$, $\mu_{z_2}(z_2(X^4)) = 0.6033$, $\mu_{z_3}(z_3(X^4)) = 0.7844$, and $\mu_{z_4}(z_4(X^4)) = 0.6033$.

Generate $\tilde{x}^j$ by X^j ($j = 1, 2, 3, 4$), following Step 10, and calculate the value of $z(\tilde{x}^j)$. As shown in Tables 2 and 3, the solution $\tilde{x}^{j+1}$ (or X^{j+1}) is better than $\tilde{x}^j$ (or X^j), $j = 1, 2, 3$.

Following Step 10, we find

$$X^* = X^4 = (2.6754, 3.9106, 0, 0, 5.1843, 5.1848, 0, 4.6128)^T. \tag{47}$$

Therefore,

$$\begin{aligned}
\tilde{x}^* &= \tilde{x}^4 \\
&= \big((2.6754, 3.9106, 0, 0)_{LR}, \\
&\qquad (5.1843, 5.1848, 0, 4.6128)_{LR}\big)^T,
\end{aligned} \tag{48}$$

serves as the compromised optimal solution with corresponding objective value

$$z^* = z(\tilde{x}^*) = (52.34, 74.03, 13.04, 38.96)_{LR}. \tag{49}$$

5. Concluding Remarks

To the end, we show the following concluding remarks.

(1) In this paper, we proposed a new method to find the compromised optimal solution to the fully fuzzy linear programming problems with all the parameters and variables represented by *LR* flat fuzzy numbers. The solution is also an *LR* flat fuzzy number. In this sense, we get an exact solution, which may give more help to the decision makers.

(2) The order relation in the objective function is the same as that in the constraint inequalities. Based on the definition of the order relation, the FFLP can be equivalently transformed into a MOLP, which is a crisp programming that is easy to be solved.

(3) Considering the MOLP problem, classical fuzzy programming method is modified for obtaining the compromised optimal solution.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the editor and the anonymous reviewers for their valuable comments, which have been very helpful in improving the paper. This work is supported by the PhD Start-up Fund of Natural Science Foundation of Guangdong Province, China (S2013040012506), and the China Postdoctoral Science Foundation Funded Project (2014M562152).

References

[1] R. E. Bellman and L. A. Zadeh, "Decision-making in a fuzzy environment," *Management Science*, vol. 17, pp. B141–B164, 1970.

[2] M. Dehghan, B. Hashemi, and M. Ghatee, "Computational methods for solving fully fuzzy linear systems," *Applied Mathematics and Computation*, vol. 179, no. 1, pp. 328–343, 2006.

[3] M. Dehghan and B. Hashemi, "Solution of the fully fuzzy linear systems using the decomposition procedure," *Applied*

Mathematics and Computation, vol. 182, no. 2, pp. 1568–1580, 2006.

[4] M. Dehghan, B. Hashemi, and M. Ghatee, "Solution of the fully fuzzy linear systems using iterative techniques," *Chaos, Solitons & Fractals*, vol. 34, no. 2, pp. 316–336, 2007.

[5] A. Kumar, J. Kaur, and P. Singh, "A new method for solving fully fuzzy linear programming problems," *Applied Mathematical Modelling*, vol. 35, no. 2, pp. 817–823, 2011.

[6] F. Hosseinzadeh Lotfi, T. Allahviranloo, M. Alimardani Jondabeh, and L. Alizadeh, "Solving a full fuzzy linear programming using lexicography method and fuzzy approximate solution," *Applied Mathematical Modelling*, vol. 33, no. 7, pp. 3151–3156, 2009.

[7] T. Allahviranloo, F. H. Lotfi, M. K. Kiasary, N. A. Kiani, and L. Alizadeh, "Solving fully fuzzy linear programming problem by the ranking function," *Applied Mathematical Sciences*, vol. 2, no. 1–4, pp. 19–32, 2008.

[8] J. Kaur and A. Kumar, "Exact fuzzy optimal solution of fully fuzzy linear programming problems with unrestricted fuzzy variables," *Applied Intelligence*, vol. 37, no. 1, pp. 145–154, 2012.

[9] J. Kaur and A. Kumar, "Mehar's method for solving fully fuzzy linear programming problems with *L-R* fuzzy parameters," *Applied Mathematical Modelling*, vol. 37, no. 12-13, pp. 7142–7153, 2013.

[10] J. J. Buckley and T. Feuring, "Evolutionary algorithm solution to fuzzy problems: fuzzy linear programming," *Fuzzy Sets and Systems*, vol. 109, no. 1, pp. 35–53, 2000.

[11] S. M. Hashemi, M. Modarres, E. Nasrabadi, and M. M. Nasrabadi, "Fully fuzzified linear programming, solution and duality," *Journal of Intelligent and Fuzzy Systems*, vol. 17, no. 3, pp. 253–261, 2006.

[12] Y. R. Fan, G. H. Huang, and A. L. Yang, "Generalized fuzzy linear programming for decision making under uncertainty: feasibility of fuzzy solutions and solving approach," *Information Sciences*, vol. 241, pp. 12–27, 2013.

[13] S. H. Nasseri and F. Zahmatkesh, "Huang method for solving fully fuzzy linear system of equations," *Journal of Mathematics and Computer Science*, vol. 1, pp. 1–5, 2010.

[14] H.-J. Zimmermann, "Fuzzy programming and linear programming with several objective functions," *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 45–55, 1978.

[15] B.-Y. Cao, *Fuzzy Geometric Programming*, Kluwer Academic Publishers, Boston, Mass, USA, 2002.

[16] D. Dubois and H. Prade, *Fuzzy Sets and Systems, Theory and Applications*, vol. 144 of *Mathematics in Science and Engineering*, Academic Press, New York, NY, USA, 1980.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 628798, 16 pages
http://dx.doi.org/10.1155/2014/628798

Research Article

Discrete Globalised Dual Heuristic Dynamic Programming in Control of the Two-Wheeled Mobile Robot

Marcin Szuster and Zenon Hendzel

Department of Applied Mechanics and Robotics, Rzeszow University of Technology, 8 Powstancow Warszawy Street, 35-959 Rzeszow, Poland

Correspondence should be addressed to Marcin Szuster; mszuster@prz.edu.pl

Received 8 April 2014; Revised 17 August 2014; Accepted 17 August 2014; Published 14 September 2014

Academic Editor: Yang Xu

Network-based control systems have been emerging technologies in the control of nonlinear systems over the past few years. This paper focuses on the implementation of the approximate dynamic programming algorithm in the network-based tracking control system of the two-wheeled mobile robot, Pioneer 2-DX. The proposed discrete tracking control system consists of the globalised dual heuristic dynamic programming algorithm, the PD controller, the supervisory term, and an additional control signal. The structure of the supervisory term derives from the stability analysis realised using the Lyapunov stability theorem. The globalised dual heuristic dynamic programming algorithm consists of two structures: the actor and the critic, realised in a form of neural networks. The actor generates the suboptimal control law, while the critic evaluates the realised control strategy by approximation of value function from the Bellman's equation. The presented discrete tracking control system works online, the neural networks' weights adaptation process is realised in every iteration step, and the neural networks preliminary learning procedure is not required. The performance of the proposed control system was verified by a series of computer simulations and experiments realised using the wheeled mobile robot Pioneer 2-DX.

1. Introduction

A rapid development of the mobile robotics applications in the last few years can be observed. Autonomous wheeled mobile robots (WMRs) have attracted much attention among researchers and engineers, while construction of robots, their sensory systems, and control algorithms were developed. One of the most challenging tasks, which occurs in the implementations of autonomous WMR, is the tracking control problem. It is widely discussed in literature, where different control strategies [1–4] are presented. This shows how significant the problem is. Difficulties met in the realisation of the desired trajectory by WMRs result from the fact that these control objects are described using nonlinear dynamic equations, where some parameters of the model can be unknown or change during the movement, for the sake of disturbances. This results in the necessity of application of computationally complex methods, which can adjust their parameters during the realisation of the trajectory and assure required quality of tracking. Artificial intelligence (AI) methods, like neural networks (NNs) [1, 2, 5, 6], are willingly applied in control systems of robots, for the sake of weights adaptation possibility. The development of AI methods makes the implementation of Bellman's dynamic programming (DP) [7] idea possible. This group of methods is called approximated dynamic programming algorithms (ADP) [8–12], adaptive critic designs (ACD), neurodynamic programming algorithms, or actor-critic structures. It is included in the larger family of methods adapted using the reinforcement learning (RL) idea. According to [9, 12], the ADP algorithms family is composed of six main schemes: heuristic dynamic programming (HDP), dual heuristic dynamic programming (DHP), globalised DHP (GDHP), and action dependant versions of mentioned earlier algorithms: action-dependent HDP (ADHDP), ADDHP, and ADGDHP. Very good surveys on ADP are given in [9, 13–16]. ADP algorithms have been firstly described for discrete-time systems [8, 9, 12] and few years later, for time-continuous systems [17–21].

Simultaneously with continuous high interest in RL algorithms, a growing number of its applications can be observed. The challenging applications of RL methods are the control problems of autonomous robots like the helicopter [22] or the underwater vehicle [23]. There are implementations of RL algorithms in mobile robot path planning [24], urban traffic signal control [25], or power system control [26], but these are mostly implementations of the Q-learning algorithm [10]. There are not many recent articles concerning ADP algorithms; the example is the application of ADHDP algorithm for a static compensator connected to a power system [27] or HDP and DHP algorithms in target recognition [28]. Application of the ADP algorithms in the control of the wheeled mobile robot is presented in [4] and in the trajectory generating process in [29]. In [30, 31] the HDP algorithm is applied to the control of the nonlinear system with some simulation results. Interesting results are shown in [32], where based on the HDP and the DHP algorithms, new kernel versions were proposed that can obtain better performance than original ones. The performance was tested using the inverted pendulum and the ball and plate benchmark systems. The implementation of the GDHP algorithm for the control of the linear object is described in [33] and for the control of the nonlinear system in [3, 34, 35], the control problem of the turbo-generator, solved using this algorithm, is presented in [36]. The article [37] summarizes the novel developments in policy-gradient and presents the novel RL architecture, the natural actor-critic (NAC), and the simulation test performed in the cart-pole balancing problem. Recent works on ADP algorithms have attempted to solve the problem of implementation of ADP based control systems without a system model knowledge [17–19]. Recent advances in this field also include implementation of ADP algorithms for partially unknown nonlinear systems [19] and robust optimal tracing control for the unknown nonlinear system [38].

The paper presents the application of the ADP algorithm in the GDHP configuration [3, 33–35] in the tracking control problem of the WMR. The discrete tracking control system guarantees a high tracking performance and a stable realisation of the desired trajectory in the face of disturbances. The GDHP algorithm consists of two structures, the actor and the critic, both realised in the form of random vector functional link (RVFL) NNs [2]. Solutions of the tracking control problems presented in literature are often theoretical considerations; there are not many real applications of ADP algorithms in control problems. The proposed discrete neural tracking control system is used for the tracking control of the WMR Pioneer 2-DX, where a series of computer simulations and experiments were realised to illustrate the performance of the control algorithm.

The results of the research presented in the paper continue the authors' earlier works related to the problem of control of the ball and beam systems [39] and the robotic manipulator [40] using DHP algorithm, tracking control of the WMR [41–44] using different ADP algorithms, and the problem of trajectory generating using ADHDP [45]. The remainder of this paper is organised as follows. The WMR dynamics is given in Section 2. The ADP algorithms family is described in Section 3. In Section 4 the GDHP algorithm implemented in the proposed discrete tracking control system is presented and in the following section, the stability is analysed using the Lyapunov function. In Section 6, the effectiveness of the proposed control algorithm is demonstrated through a numerical illustration and an experiment realised using the WMR Pioneer 2-DX. Finally, Section 7 gives the conclusion.

2. Dynamical Model of the Wheeled Mobile Robot Pioneer 2-DX

The WMR Pioneer 2-DX is the control object, shown in Figure 1(a). It is a nonholonomic object, which dynamics is described using nonlinear equations. The WMR is composed of two driving wheels 1 and 2, a third, free rolling castor wheel 3, and a frame 4 (Figure 1(b)). The movement of the WMR is analysed in the xy plane.

Point A is a central point of the WMR's frame, β is an angle of the frame's turn, r_1, r_2, l, and l_1 are dimensions that result from the WMR's geometry, $\alpha_{[1]}$, $\alpha_{[2]}$ are angles of the driving wheels 1 and 2 rotation, and $u_{[1]}$, $u_{[2]}$ are control signals. The dynamical model of the WMR was derived using Maggie's formalism [2, 46] and assumed in the form

$$\mathbf{M}\ddot{\boldsymbol{\alpha}} + \mathbf{C}(\dot{\boldsymbol{\alpha}})\dot{\boldsymbol{\alpha}} + \mathbf{F}(\dot{\boldsymbol{\alpha}}) + \boldsymbol{\tau}_d(t) = \mathbf{u}, \tag{1}$$

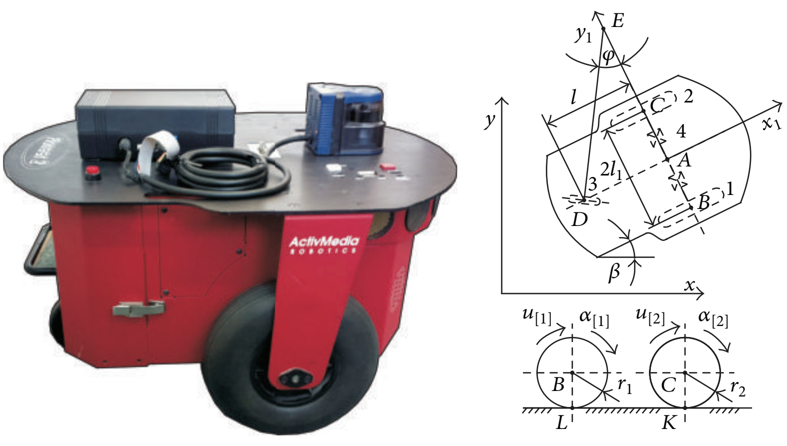
where $\dot{\boldsymbol{\alpha}} = [\dot{\alpha}_{[1]}, \dot{\alpha}_{[2]}]^T$ is the vector of angular velocities of driving wheels, $\mathbf{M}$ is the positive defined inertia matrix, $\mathbf{C}(\dot{\boldsymbol{\alpha}})\dot{\boldsymbol{\alpha}}$ is the vector of centrifugal and Coriolis forces/momentous, $\mathbf{F}(\dot{\boldsymbol{\alpha}})$ is the friction vector, $\boldsymbol{\tau}_d(t)$ is the vector of disturbances, and $\mathbf{u}$ is the control vector. Matrices $\mathbf{M}$, $\mathbf{C}(\dot{\boldsymbol{\alpha}})$ and the vector $\mathbf{F}(\dot{\boldsymbol{\alpha}})$ take the form

$$\mathbf{M} = \begin{bmatrix} a_{[1]} + a_{[2]} + a_{[3]} & a_{[1]} - a_{[2]} \\ a_{[1]} - a_{[2]} & a_{[1]} + a_{[2]} + a_{[3]} \end{bmatrix},$$
$$\mathbf{C}(\dot{\boldsymbol{\alpha}}) = \begin{bmatrix} 0 & 2a_{[2]}\left(\dot{\alpha}_{[2]} - \dot{\alpha}_{[1]}\right) \\ -2a_{[2]}\left(\dot{\alpha}_{[2]} - \dot{\alpha}_{[1]}\right) & 0 \end{bmatrix}, \tag{2}$$
$$\mathbf{F}(\dot{\boldsymbol{\alpha}}) = \begin{bmatrix} a_{[5]}\operatorname{sgn}\left(\dot{\alpha}_{[1]}\right) \\ a_{[6]}\operatorname{sgn}\left(\dot{\alpha}_{[2]}\right) \end{bmatrix},$$

where $\mathbf{a} = [a_{[1]}, \ldots, a_{[6]}]^T$ is the vector of WMR's parameters that result from the object's geometry, mass distribution, and resistances to motion [2, 46]. The nominal parameters of the WMR Pioneer 2-DX were assumed as $\mathbf{a} = [0.1207, 0.0768, 0.037, 0.0001, 2.025, 2.025]^T$.

The proposed tracking control system is discrete. A continuous model of the WMR's dynamics (1) was discretised using Euler's method and assumed in the form

$$\mathbf{z}_{1\{k+1\}} = \mathbf{z}_{1\{k\}} + h\mathbf{z}_{2\{k\}},$$
$$\mathbf{z}_{2\{k+1\}} = \mathbf{z}_{2\{k\}} - h\mathbf{M}^{-1}\left[\mathbf{C}\left(\mathbf{z}_{2\{k\}}\right)\mathbf{z}_{2\{k\}} + \mathbf{F}\left(\mathbf{z}_{2\{k\}}\right) + \boldsymbol{\tau}_{d\{k\}} - \mathbf{u}_{\{k\}}\right], \tag{3}$$



(a)

(b)

Figure 1: (a) The WMR Pioneer 2-DX, (b) scheme of the WMR.

where $\mathbf{z}_{2\{k\}} = [z_{2[1]\{k\}}, z_{2[2]\{k\}}]^T$ is a vector that corresponds to the continuous vector of angular velocities $\dot{\boldsymbol{\alpha}}$, k is an index of iteration steps, and h is a time discretisation parameter. The state vector was assumed in the form $\mathbf{z}_{\{k\}} = [\mathbf{z}_{1\{k\}}, \mathbf{z}_{2\{k\}}]^T$. The discrete tracking errors of angles of the driving wheels rotation $\mathbf{z}_{1\{k\}}$ and errors of angular velocities $\mathbf{z}_{2\{k\}}$ were defined as

$$\mathbf{e}_{1\{k\}} = \mathbf{z}_{1\{k\}} - \mathbf{z}_{d1\{k\}},$$
$$\mathbf{e}_{2\{k\}} = \mathbf{z}_{2\{k\}} - \mathbf{z}_{d2\{k\}}, \tag{4}$$

where the desired trajectory ($\mathbf{z}_{d\{k\}} = [\mathbf{z}_{d1\{k\}}, \mathbf{z}_{d2\{k\}}]^T$) was generated earlier. On the basis of (4) the filtered tracking error $\mathbf{s}_{\{k\}}$ was defined in the form

$$\mathbf{s}_{\{k\}} = \mathbf{e}_{2\{k\}} + \boldsymbol{\Lambda}\mathbf{e}_{1\{k\}}, \tag{5}$$

where $\boldsymbol{\Lambda}$ is a positive defined, fixed diagonal matrix.

Substituting the WMR dynamics model (3) and the tracking errors (4) into $\mathbf{s}_{\{k+1\}}$, calculated on the base of (5), the filtered tracking error was assumed in the form

$$\mathbf{s}_{\{k+1\}} = \mathbf{Y}_d\left(\mathbf{z}_{\{k\}}, \mathbf{z}_{d\{k\}}, \mathbf{z}_{d3\{k\}}\right) - \mathbf{Y}_f\left(\mathbf{z}_{2\{k\}}\right) - \mathbf{Y}_{\tau\{k\}} + h\mathbf{M}^{-1}\mathbf{u}_{\{k\}}, \tag{6}$$

where

$$\begin{aligned}
&\mathbf{Y}_d\left(\mathbf{z}_{\{k\}}, \mathbf{z}_{d\{k\}}, \mathbf{z}_{d3\{k\}}\right) \\
&\quad = \mathbf{z}_{2\{k\}} - \mathbf{z}_{d2\{k+1\}} + \boldsymbol{\Lambda}\left[\mathbf{z}_{1\{k\}} + h\mathbf{z}_{2\{k\}} - \mathbf{z}_{d1\{k+1\}}\right] \\
&\quad = \mathbf{s}_{\{k\}} + \mathbf{Y}_e\left(\mathbf{z}_{2\{k\}}, \mathbf{z}_{d2\{k\}}, \mathbf{z}_{d3\{k\}}\right), \\
&\mathbf{Y}_e\left(\mathbf{z}_{2\{k\}}, \mathbf{z}_{d2\{k\}}, \mathbf{z}_{d3\{k\}}\right) = h\left[\boldsymbol{\Lambda}\mathbf{e}_{2\{k\}} - \mathbf{z}_{d3\{k\}}\right], \\
&\mathbf{Y}_f\left(\mathbf{z}_{2\{k\}}\right) = h\mathbf{M}^{-1}\left[\mathbf{C}\left(\mathbf{z}_{2\{k\}}\right)\mathbf{z}_{2\{k\}} + \mathbf{F}\left(\mathbf{z}_{2\{k\}}\right)\right], \\
&\mathbf{Y}_{\tau\{k\}} = h\mathbf{M}^{-1}\boldsymbol{\tau}_{d\{k\}},
\end{aligned} \tag{7}$$

where $\mathbf{z}_{d3\{k\}}$ is the vector of desired angular accelerations that derives from the expansion of the vector $\mathbf{z}_{d2\{k+1\}}$ using Euler's method. The vector $\mathbf{Y}_f(\mathbf{z}_{2\{k\}})$ includes all nonlinearities of the controlled object.

3. Approximate Dynamic Programming

Bellman's dynamic programming (DP) is based on the calculation of the value function, the control law, and the state of the object for every step of the process, from the last to the first. That is why it is not applicable in online control. ADP algorithms are also called adaptive critic designs (ACD) [8–16] or neuro-dynamic programming (NDP) algorithms. They derive from the application of NNs into Bellman's approach to the optimal control theory, where the value function and the optimal control law are approximated by the critic and the actor. This approach makes real-time control of dynamical objects possible. The ADP algorithms family is schematically shown in Figure 2. It is composed of six algorithms, which differ from each other by the critic's structure and the weights adaptation rule of the actor's and the critic's NN.

The basic structure is the HDP algorithm, in which the critic approximates the value function and the actor generates the suboptimal control law. In the DHP algorithm the critic approximates the difference of the value function with respect to the state of the controlled system. The actor has the same structure as in HDP. Complexity of the critic grows proportionally to the size of the state vector, because the difference of the value function with respect to the n-dimensional state vector is approximated by n critic's NNs, and the critic's weights adaptation law is also more complex. The DHP algorithm assures higher quality of tracking control in comparison to HDP [43]. The GDHP algorithm is built in the same way as HDP; its characteristic feature is the critic's weights adaptation law. It is based on the minimisation of the value function and its difference with respect to the state and can be seen as a combination of the HDP and the DHP critic's NN adaptation law. The actor structure is the same as

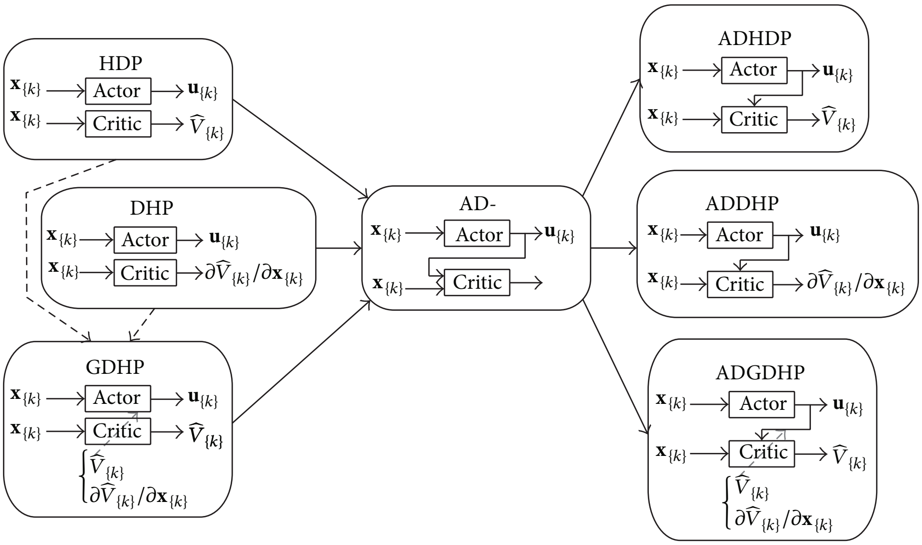
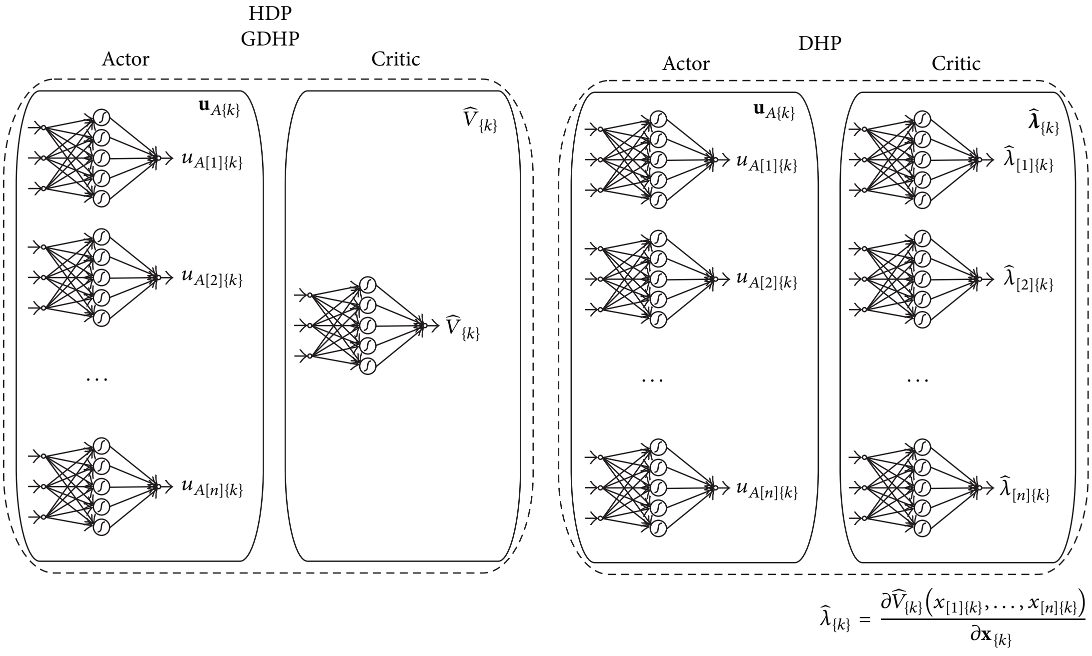


Figure 2: The scheme of the approximate dynamic programming algorithms family.


HDP
GDHP
Actor
Critic
$\mathbf{u}_{A\{k\}}$
$u_{A[1]\{k\}}$
$u_{A[2]\{k\}}$
...
$u_{A[n]\{k\}}$
$\widehat{V}_{\{k\}}$
DHP
$\widehat{\boldsymbol{\lambda}}_{\{k\}}$
$\widehat{\lambda}_{[1]\{k\}}$
$\widehat{\lambda}_{[2]\{k\}}$
$\widehat{\lambda}_{[n]\{k\}}$
$\widehat{\lambda}_{\{k\}} = \frac{\partial\widehat{V}_{\{k\}}(x_{[1]\{k\}},\ldots,x_{[n]\{k\}})}{\partial\mathbf{x}_{\{k\}}}$

(a)

(b)

Figure 3: (a) Scheme of the actor's and the critic's structure complexity in HDP and GDHP, (b) scheme of the actor's and the critic's structure complexity in DHP.

in HDP. The difference in complexity of the three basic ADP algorithms is schematically shown in Figure 3.

In the HDP and the GDHP algorithm the critic is composed of one NN that approximates the value function, while in the DHP algorithm critic consists of n NNs, where n is the size of the state vector. For example, in the case of the WMR, where the state vector for the system (6) is of $n = 2$ size, the DHP algorithm consists of the actor and the critic realised in a form of two NNs each. In the GDHP algorithm, the actor is composed of two NNs, but the critic is realised in the form of only one NN. The advantage of GDHP over DHP, in the case of complexity of the critic, is even more evident considering the instance of the 6 degrees of freedom robotic manipulator ($n = 6$). The DHP algorithm implemented in the control system for this controlled object should be composed of the actor and the critic realised in a form of six NNs each, while the GDHP would be composed of the actor realised in a form of six NNs, and only one NN in the critic structure. The difference of the complexity of the critic structure increases simultaneously as the state vector of the controlled object increases. The rest of the ADP algorithms are AD versions of the basic algorithms, where the control law generated by the actor's NN is also the input to the critic's NN.

4. Globalised Dual Heuristic Dynamic Programming in Tracking Control

The main part of the proposed tracking control system is the GDHP algorithm. There are not many applications of the GDHP algorithms in literature, and existing publications concern rather with theoretical studies [3, 33–36]. In this paper, both the numerical tests and the verification experiments of the neural tracking control system, realised using the WMR Pioneer 2-DX, are presented. The GDHP structure generates the control law that minimises the value function $V_{\{k\}}(\mathbf{s}_{\{k\}})$ [8–16], assumed in the form of equation

$$V_{\{k\}}\left(\mathbf{s}_{\{k\}}\right)=\sum_{k=0}^{N}\gamma^{k}L_{C\{k\}}\left(\mathbf{s}_{\{k\}}\right), \tag{8}$$

where N is a number of iteration steps, γ is a discount factor, $0<\gamma\leq 1$, and $L_{C\{k\}}(\mathbf{s}_{\{k\}})$ is the local cost function for the kth step, assumed in the form

$$L_{C\{k\}}\left(\mathbf{s}_{\{k\}}\right)=\frac{1}{2}\mathbf{s}_{\{k\}}^{T}\mathbf{R}\mathbf{s}_{\{k\}}, \tag{9}$$

where $\mathbf{R}$ is a positive defined, fixed diagonal matrix.

The GDHP algorithm, schematically shown in Figure 4(a), consists of the following:

(i) the predictive model that predicts the WMR's closed-loop state $\mathbf{s}_{\{k+1\}}$, according to the equation

$$\mathbf{s}_{\{k+1\}}=\mathbf{Y}_{d}\left(\mathbf{z}_{\{k\}},\mathbf{z}_{d\{k\}},\mathbf{z}_{d3\{k\}}\right)-\mathbf{Y}_{f}\left(\mathbf{z}_{2\{k\}}\right)+h\mathbf{M}^{-1}\mathbf{u}_{\{k\}}, \tag{10}$$

where $\mathbf{u}_{\{k\}}$ is the overall tracking control signal of the proposed control system. Its structure derives from the stability analysis presented in the next section. The controlled system's dynamical model is necessary in the synthesis of the actor's and the critic's weights adaptation law in the GDHP algorithm;

(ii) the actor, realised in the form of two RVFL NNs, that generate the suboptimal control law $\mathbf{u}_{A\{k,l\}}=[u_{A[1]\{k,l\}},u_{A[2]\{k,l\}}]^{T}$ and are expressed by the formula

$$u_{A[j]\{k,l\}}\left(\mathbf{x}_{Aj\{k\}},\mathbf{W}_{Aj\{k,l\}}\right)=\mathbf{W}_{Aj\{k,l\}}^{T}\mathbf{S}\left(\mathbf{D}_{A}^{T}\mathbf{x}_{Aj\{k\}}\right), \tag{11}$$

where $j=1,2,l$ is an index of the internal loop iteration, $\mathbf{x}_{Aj\{k\}}$ is the input vector of the jth actor's NN, it consists of normalised values of the filtered tracking error $\mathbf{s}_{\{k\}}$, errors $\mathbf{e}_{\{k\}}$, desired ($\mathbf{z}_{d2\{k\}}$) and realised ($\mathbf{z}_{2\{k\}}$) angular velocities of the driving wheels, $x_{Aj[i]\{k\}}\epsilon\langle -1;1\rangle$, $\mathbf{W}_{Aj\{k,l\}}$ is the vector of output layer weights of the jth actor's NN, $\mathbf{S}(\cdot)$ is the vector of sigmoidal bipolar neuron activation functions, and $\mathbf{D}_{A}$ is the matrix of fixed input weights selected randomly in the NNs initialisation process. Actor's NNs weights are adapted by the gradient method according to equation

$$\mathbf{W}_{Aj\{k,l+1\}}=\mathbf{W}_{Aj\{k,l\}}-e_{A[j]\{k,l\}}\mathbf{\Gamma}_{A}\mathbf{S}\left(\mathbf{D}_{A}^{T}\mathbf{x}_{Aj\{k\}}\right), \tag{12}$$

where $\mathbf{\Gamma}_{A}$ is the fixed diagonal matrix of positive learning rates. The quality rating $\mathbf{e}_{A\{k,l\}}$ was assumed in the form

$$\mathbf{e}_{A\{k,l\}}=\frac{\partial L_{C\{k\}}\left(\mathbf{s}_{\{k\}}\right)}{\partial\mathbf{u}_{\{k\}}}+\left[\frac{\partial\mathbf{s}_{\{k+1\}}}{\partial\mathbf{u}_{\{k\}}}\right]^{T}\frac{\partial\widehat{V}_{\{k+1,l\}}\left(\mathbf{x}_{C\{k+1\}},\mathbf{W}_{C\{k,l\}}\right)}{\partial\mathbf{s}_{\{k+1\}}}, \tag{13}$$

where $\widehat{V}_{\{k+1,l\}}(\mathbf{x}_{C\{k+1\}},\mathbf{W}_{C\{k,l\}})$ is the output of the critic's NN, generated on the basis of the predicted state for the step $k+1$;

(iii) the critic, realised in the form of one RVFL NN, estimates the value function (8). It is expressed by the formula

$$\widehat{V}_{\{k,l\}}\left(\mathbf{x}_{C\{k\}},\mathbf{W}_{C\{k,l\}}\right)=\mathbf{W}_{C\{k,l\}}^{T}\mathbf{S}\left(\mathbf{D}_{C}^{T}\mathbf{x}_{C\{k\}}\right), \tag{14}$$

where $\mathbf{x}_{C\{k\}}$ is the input vector of the critic's NN, $\mathbf{x}_{C\{k\}}=\boldsymbol{\kappa}_{C}[1,\mathbf{s}_{\{k\}}^{T}]^{T}$, $\boldsymbol{\kappa}_{C}$ is the constant diagonal matrix of positive scaling coefficients, $\mathbf{W}_{C\{k,l\}}$ is the vector of output layer weights of the critic's NN, and $\mathbf{D}_{C}$ is the matrix of fixed input weights selected randomly in the critic's NN initialisation process. The critic's RVFL NN is schematically shown in Figure 4(b).

The critic's weights adaptation procedure in the GDHP algorithm is the most complex among all the ADP structures family. It is based on the minimisation of errors characteristic for the critic's weights adaptation rule of the HDP algorithm ($e_{H\{k,l\}}$) and the DHP algorithm ($e_{D\{k,l\}}$), expressed by the formula

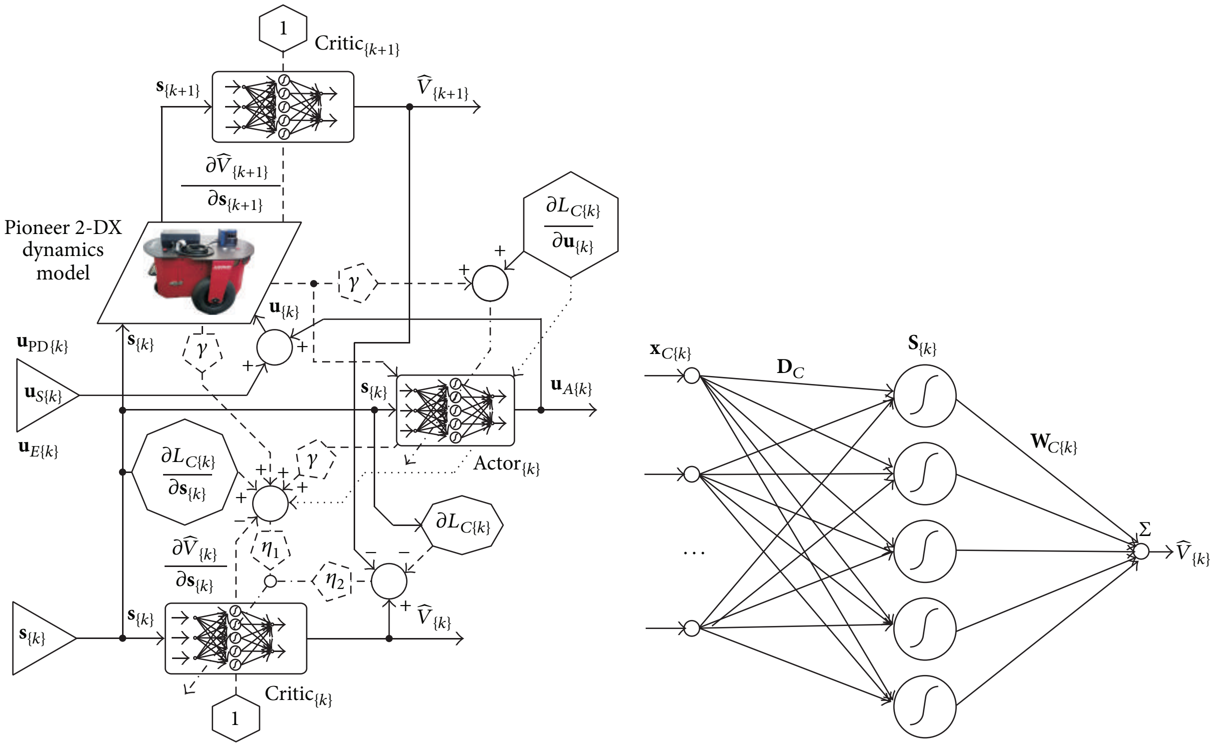
$$\begin{aligned}e_{H\{k,l\}}={}&\widehat{V}_{\{k,l\}}\left(\mathbf{x}_{C\{k\}},\mathbf{W}_{C\{k,l\}}\right)-L_{C\{k\}}\left(\mathbf{s}_{\{k\}}\right)\\&-\gamma\widehat{V}_{\{k+1,l\}}\left(\mathbf{x}_{C\{k+1\}},\mathbf{W}_{C\{k,l\}}\right),\end{aligned} \tag{15}$$

$$\begin{aligned}e_{D\{k,l\}}={}&\mathbf{I}_{D}^{T}\left\{\frac{\partial L_{C\{k\}}\left(\mathbf{s}_{\{k\}}\right)}{\partial\mathbf{s}_{\{k\}}}+\left[\frac{\partial\mathbf{u}_{\{k\}}}{\partial\mathbf{s}_{\{k\}}}\right]^{T}\frac{\partial L_{C\{k\}}\left(\mathbf{s}_{\{k\}}\right)}{\partial\mathbf{u}_{\{k\}}}\right.\\&+\gamma\left[\frac{\partial\mathbf{s}_{\{k+1\}}}{\partial\mathbf{s}_{\{k\}}}+\left[\frac{\partial\mathbf{u}_{\{k\}}}{\partial\mathbf{s}_{\{k\}}}\right]^{T}\frac{\partial\mathbf{s}_{\{k+1\}}}{\partial\mathbf{u}_{\{k\}}}\right]^{T}\\&\times\frac{\partial\widehat{V}_{\{k+1,l\}}\left(\mathbf{x}_{C\{k+1\}},\mathbf{W}_{C\{k,l\}}\right)}{\partial\mathbf{s}_{\{k+1\}}}\\&\left.-\frac{\partial\widehat{V}_{\{k,l\}}\left(\mathbf{x}_{C\{k\}},\mathbf{W}_{C\{k,l\}}\right)}{\partial\mathbf{s}_{\{k\}}}\right\},\end{aligned} \tag{16}$$

where $\mathbf{I}_{D}$ is a constant vector, $\mathbf{I}_{D}=[1,1]^{T}$. Weights of the critic's NN are adapted using the gradient method according to the equation

$$\begin{aligned}\mathbf{W}_{C\{k,l+1\}}={}&\mathbf{W}_{C\{k,l\}}-\eta_{1}e_{H\{k,l\}}\mathbf{\Gamma}_{C}\frac{\partial\widehat{V}_{\{k,l\}}\left(\mathbf{x}_{C\{k\}},\mathbf{W}_{C\{k,l\}}\right)}{\partial\mathbf{W}_{C\{k,l\}}}\\&-\eta_{2}e_{D\{k,l\}}\mathbf{\Gamma}_{C}\frac{\partial^{2}\widehat{V}_{\{k,l\}}\left(\mathbf{x}_{C\{k\}},\mathbf{W}_{C\{k,l\}}\right)}{\partial\mathbf{s}_{\{k\}}\mathbf{W}_{C\{k,l\}}},\end{aligned} \tag{17}$$

where $\mathbf{\Gamma}_{C}$ is the fixed diagonal matrix of positive learning rates and η_{1},η_{2} are positive constants.



(a) (b)

Figure 4: (a) Scheme of the GDHP algorithm, (b) scheme of the critic's RVFL NN.

Adaptation process of NNs' weights is an interesting feature of the ADP algorithms. It is realised in a form of an internal loop with the iteration index l. In every step k of the discrete control process calculations, which are connected to the actor's and the critic's weights adaptation procedure, are executed according to the scheme shown in Figure 5.

The actor-critic structure adaptation process is organised in the following way: at the beginning of every kth iteration step $l = 0$. Actor's NNs weights are adapted according to the assumed adaptation law (12) by minimisation of the error rate (13). This part of the algorithm, called the "control law improvement routine" [9], leads to the evaluation of the actor's NNs weights $\mathbf{W}_{Aj\{k,l+1\}}$. The next step consists of the adaptation of the critic's NN weights; it is called the "value function determination operation." The critic's NN weights are adapted according to the assumed adaptation law, by the minimisation of the error rate (15), called the temporal difference error (TDE) [12], and the error rate (16). This leads to the calculation of the critic's NN weights $\mathbf{W}_{C\{k,l+1\}}$. Next, the internal loop iteration index l is increased, and a new cycle of the ADP algorithm adaptation is started. In the presented algorithm, the internal loop breaks, when the number of internal iterations $l \geq l_{mx}$, where l_{mx} is the maximal number of iteration cycles, or when the error $e_{A[j]\{k,l\}}$ is smaller than the assumed positive limit $E_{A[j]}$, $e_{A[j]\{k,l\}} < E_{A[j]}$, $j = 1, 2$. When one of these conditions is satisfied, $\mathbf{W}_{Aj\{k,l+1\}}$ becomes $\mathbf{W}_{Aj\{k+1,l\}}$ and $\mathbf{W}_{C\{k,l+1\}}$ becomes $\mathbf{W}_{C\{k+1,l\}}$. Next index k is increased. The actor's NNs generate control signals and the GDHP structure receives information about a new state of the controlled object. In the next sections index l is omitted for the sake of simplicity.

5. Stability Analysis

This paper focuses on the implementation of the ADP algorithm in the network-based tracking control system of the two-wheeled mobile robot, Pioneer 2-DX. The proposed discrete tracking control system consists of the GDHP algorithm, the PD controller, the supervisory term, and the additional control signal.

The filtered tracking error $\mathbf{s}_{\{k\}}$ was defined in the form (5), where $\mathbf{\Lambda}$ is a positive defined, fixed diagonal matrix selected in the way that the eigenvalues are within a unit disc. Consequently, if the filtered tracking error (5) tends to zero then all the tracking errors go to zero. Filtered tracking error $\mathbf{s}_{\{k+1\}}$ can be expressed as (6), where the vector $\mathbf{Y}_f(\mathbf{z}_{2\{k\}})$ includes all nonlinearities of the controlled object.

Let us define the control input $\mathbf{u}_{\{k\}}$ as

$$\mathbf{u}_{\{k\}} = h^{-1}\mathbf{M}\left[\mathbf{Y}_d\left(\mathbf{z}_{\{k\}}, \mathbf{z}_{d\{k\}}, \mathbf{z}_{d3\{k\}}\right) - \widehat{\mathbf{Y}}_f\left(\mathbf{z}_{2\{k\}}\right) - \mathbf{K}_D\mathbf{s}_{\{k\}}\right], \tag{18}$$

where $\widehat{\mathbf{Y}}_f(\mathbf{z}_{2\{k\}})$ is an estimate of the unknown function.

Then, the closed-loop system becomes

$$\mathbf{s}_{\{k+1\}} = \mathbf{K}_D\mathbf{s}_{\{k\}} - \widetilde{\mathbf{Y}}_f\left(\mathbf{z}_{2\{k\}}\right) - \mathbf{Y}_{\tau\{k\}}, \tag{19}$$

where the functional estimation error is given by $\widetilde{\mathbf{Y}}_f(\mathbf{z}_{2\{k\}}) = \widehat{\mathbf{Y}}_f(\mathbf{z}_{2\{k\}}) - \mathbf{Y}_f(\mathbf{z}_{2\{k\}})$. Equation (19) relates the filtered tracking

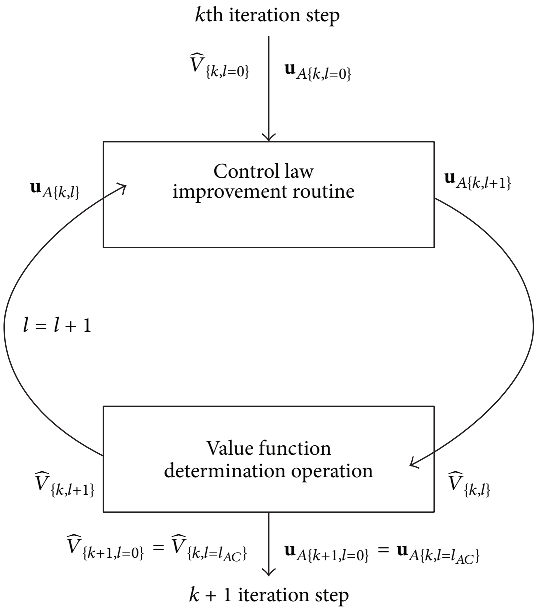

kth iteration step
$\widehat{V}_{\{k,l=0\}}$
$\mathbf{u}_{A\{k,l=0\}}$
Control law improvement routine
$\mathbf{u}_{A\{k,l\}}$
$\mathbf{u}_{A\{k,l+1\}}$
$l = l + 1$
Value function determination operation
$\widehat{V}_{\{k,l+1\}}$
$\widehat{V}_{\{k,l\}}$
$\widehat{V}_{\{k+1,l=0\}} = \widehat{V}_{\{k,l=l_{AC}\}}$
$\mathbf{u}_{A\{k+1,l=0\}} = \mathbf{u}_{A\{k,l=l_{AC}\}}$
$k + 1$ iteration step

Figure 5: Schematic conception of the ADP structure adaptation process.

error with the functional estimation error. In general, the filtered tracking error system (19) can also be expressed as

$$\mathbf{s}_{\{k+1\}} = \mathbf{K}_D \mathbf{s}_{\{k\}} + \mathbf{d}_{0\{k\}}, \tag{20}$$

where $\mathbf{d}_{0\{k\}} = -(\widetilde{\mathbf{Y}}_f(\mathbf{z}_{2\{k\}}) + \mathbf{Y}_{\tau\{k\}})$. If the functional estimation error $\widetilde{\mathbf{Y}}_f(\mathbf{z}_{2\{k\}})$ is bounded in such a way that $|Y_{f[j]}(\mathbf{z}_{2\{k\}})| \leq F_{[j]}$, $F_{[j]}$ is a positive constant and $Y_{\tau[j]\{k\}} < b_{d[j]}$, where $b_{d[j]}$ is a positive constant, then the next stability results hold.

Let us consider the system given by (3). Let the control action be provided by (18) and assume that the functional estimation error and the unknown disturbance are bounded. The filtered tracking error system (6) is stable provided that

$$0 < K_{D\max} < 1, \tag{21}$$

where $K_{D\max} \in R$ is the maximum eigenvalue of the matrix $\mathbf{K}_D$.

Let us consider the following Lyapunov function candidate:

$$L_{\{k\}} = \mathbf{s}_{\{k\}}^T \mathbf{s}_{\{k\}}. \tag{22}$$

The first difference is

$$\Delta L_{\{k\}} = \mathbf{s}_{\{k+1\}}^T \mathbf{s}_{\{k+1\}} - \mathbf{s}_{\{k\}}^T \mathbf{s}_{\{k\}}. \tag{23}$$

Substituting the filtered tracking error dynamics (19) into (23) results in

$$\begin{aligned} \Delta L_{\{k\}} = {} & \left[\mathbf{K}_D \mathbf{s}_{\{k\}} - \widetilde{\mathbf{Y}}_f\left(\mathbf{z}_{2\{k\}}\right) - \mathbf{Y}_{\tau\{k\}}\right]^{\mathbf{T}} \\ & \times \left[\mathbf{K}_D \mathbf{s}_{\{k\}} - \widetilde{\mathbf{Y}}_f\left(\mathbf{z}_{2\{k\}}\right) - \mathbf{Y}_{\tau\{k\}}\right] - \mathbf{s}_{\{k\}}^T \mathbf{s}_{\{k\}}, \end{aligned} \tag{24}$$

what implies that $\Delta L_{\{k\}} \leq 0$ provided that

$$\left\|\mathbf{K}_D \mathbf{s}_{\{k\}} - \widetilde{\mathbf{Y}}_f\left(\mathbf{z}_{2\{k\}}\right) - \mathbf{Y}_{\tau\{k\}}\right\| \leq K_{D\max} \left\|\mathbf{s}_{\{k\}}\right\| + F + b_d < \left\|\mathbf{s}_{\{k\}}\right\|, \tag{25}$$

where F and b_d are positive constants. This further implies that

$$\left\|\mathbf{s}_{\{k\}}\right\| \geq \frac{F + b_d}{1 - K_{D\max}}. \tag{26}$$

The closed-loop system is uniformly ultimately bounded (UUB) [47]. The PD controller parameter $K_{D\max} \in R$ has to be selected using (21) in order for the closed-loop system to be stable. This outer-loop signal is viewed as the supervisor's evaluation feedback to the actor and the critic. In the NN actor-critic control scheme derived in this paper there is no preliminary offline learning phase. The weights are simply initialized at zero, for then the control system is just the PD controller. Therefore, the closed-loop system remains stable until the NNs begin to learn.

The proposed discrete tracking control system is composed of the GDHP structure that generates the control signal $\mathbf{u}_{A\{k\}}$, the PD controller ($\mathbf{u}_{\text{PD}\{k\}}$), the supervisory term ($\mathbf{u}_{S\{k\}}$), and the additional control signal $\mathbf{u}_{E\{k\}}$. Structure of the supervisory term derives from the stability analysis performed using the Lyapunov stability theorem. The additional control signal $\mathbf{u}_{E\{k\}}$ derives from the process of the WMR dynamics model discretisation. The overall tracking control signal was assumed in the form

$$\mathbf{u}_{\{k\}} = -h^{-1} \mathbf{M} \left[\mathbf{u}_{A\{k\}} + \mathbf{u}_{\text{PD}\{k\}} + \mathbf{u}_{E\{k\}} - \mathbf{u}_{S\{k\}}\right], \tag{27}$$

where

$$\begin{aligned} \mathbf{u}_{\text{PD}\{k\}} &= \mathbf{K}_D \mathbf{s}_{\{k\}}, \\ \mathbf{u}_{E\{k\}} &= h\left[\mathbf{\Lambda} \mathbf{e}_{2\{k\}} - \mathbf{z}_{d3\{k\}}\right], \\ \mathbf{u}_{S\{k\}} &= \mathbf{I}_S \mathbf{u}_{S\{k\}}^{*}, \end{aligned} \tag{28}$$

where $\mathbf{K}_D$ is a fixed diagonal matrix of positive PD controller gains, $\mathbf{I}_S$ is a diagonal matrix, with elements $I_{S[j,j]} = 1$ if $|s_{[j]\{k\}}| \geq \rho_{[j]}$ or $I_{S[j,j]} = 0$ in the other case, $j = 1, 2$, $\rho_{[j]}$ is a positive constant.

The scheme of the discrete neural tracking control system with actor-critic structure in the GDHP configuration is shown in Figure 6.

The stability analysis was performed under the assumption that $I_{S[j,j]} = 1$. Substituting (27) into (6), the closed-loop system equation is expressed by the formula

$$\begin{aligned} \mathbf{s}_{\{k+1\}} = {} & \mathbf{Y}_d\left(\mathbf{z}_{\{k\}}, \mathbf{z}_{d\{k\}}, \mathbf{z}_{d3\{k\}}\right) - \mathbf{Y}_f\left(\mathbf{z}_{2\{k\}}\right) - \mathbf{Y}_{\tau\{k\}} \\ & - \left[\mathbf{u}_{A\{k\}} + \mathbf{u}_{\text{PD}\{k\}} + \mathbf{u}_{E\{k\}} - \mathbf{u}_{S\{k\}}\right]. \end{aligned} \tag{29}$$

The stability analysis was realised using the positive definite Lyapunov candidate function

$$L = \frac{1}{2} \mathbf{s}^T \mathbf{s}, \tag{30}$$

which discretised derivative was assumed in the form

$$\Delta L_{\{k\}} = \mathbf{s}_{\{k\}}^T \left[\mathbf{s}_{\{k+1\}} - \mathbf{s}_{\{k\}}\right]. \tag{31}$$

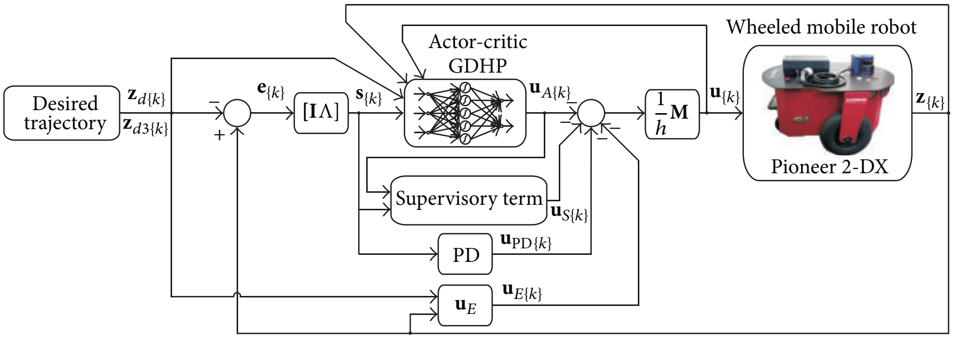


Figure 6: Scheme of the tracking control system.

Substituting (29) into (31), $\Delta L_{\{k\}}$ takes the form

$$\Delta L_{\{k\}} = \mathbf{s}_{\{k\}}^{T}\left[-\mathbf{Y}_{f}\left(\mathbf{z}_{2\{k\}}\right) - \mathbf{Y}_{\tau\{k\}} - \mathbf{u}_{A\{k\}} - \mathbf{K}_{D}\mathbf{s}_{\{k\}} + \mathbf{u}_{S\{k\}}^{*}\right]. \tag{32}$$

On the assumption that all elements of the vector of disturbances are bounded, $Y_{\tau[j]\{k\}} < b_{d[j]}$, where $b_{d[j]}$ is a positive constant, the difference of the Lyapunov candidate function takes the form

$$\begin{aligned}\Delta L_{\{k\}} \leq & -\mathbf{s}_{\{k\}}^{T}\mathbf{K}_{D}\mathbf{s}_{\{k\}} + \sum_{j=1}^{2}\left|s_{[j]\{k\}}\right| \\ & \times\left[\left|Y_{f[j]}\left(\mathbf{z}_{2\{k\}}\right)\right| + \left|u_{A[j]\{k\}}\right| + \left|Y_{\tau[j]\{k\}}\right|\right] \\ & + \sum_{j=1}^{2} s_{[j]\{k\}}u_{S[j]\{k\}}^{*}.\end{aligned} \tag{33}$$

The supervisory term's control signal was assumed in the form

$$u_{S[j]\{k\}}^{*} = -\operatorname{sgn}\left(s_{[j]\{k\}}\right)\left[F_{[j]} + \left|u_{A[j]\{k\}}\right| + b_{d[j]} + \sigma_{[j]}\right], \tag{34}$$

where $|Y_{f[j]}(\mathbf{z}_{2\{k\}})| \leq F_{[j]}$, $F_{[j]}$ is a positive constant, and $\sigma_{[j]}$ is a positive constant. On the above assumptions the difference of the Lyapunov function (30) is a negative definite.

6. Research Results

Performance of the proposed discrete tracking control system was tested during a series of computer simulations and then verified using the laboratory stand schematically shown in Figure 7.

The laboratory stand consists of the WMR Pioneer 2-DX, the power supply and a PC equipped with the dSpace DS1102 digital signal processing board and software: dSpace Control Desk and Matlab/Simulink. The WMR Pioneer 2-DX is equipped with the sensory system composed of eight ultrasonic sensors and a scanning laser range finder. The movement of the robot is realised using two independently supplied DC motors with gears (ratio 19.7 : 1) and encoders (500 ticks per shaft revolution). The WMR weights m_R = 9 kg, its frame is l_R = 0.44 m long, l_W = 0.33 m width, and its maximal velocity is equal to v_A = 1.6 m/s.

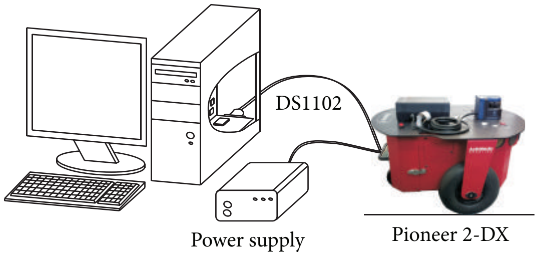


Figure 7: Scheme of the laboratory stand.

6.1. Simulation Results. Performance of the proposed control system was tested during a series of numerical simulations performed using the Matlab/Simulink software environment. In this section the notation of variables is simplified and the index k is omitted. The same set of parameters during simulations as in the experiment was used. The time discretisation parameter was equal to h = 0.01 s. In the GDHP structure NNs with eight neurons each were used. The output layer weights of NNs were set to zero in the initialisation process. Parameters of the PD controller $\mathbf{K}_D$ = diag{0.036, 0.036}, $\mathbf{\Lambda}$ = diag{0.5, 0.5} were assumed. One must select $\mathbf{K}_D$ using some trial and error experiments or computer simulations. In practice, this has not shown itself to be a problem. The PD controller gains were selected heuristically to satisfy (21). For the sake of the noise that occurs in the signals of the driving wheels angular velocities, incremental encoders were used in the experiment for measurement, the amplification of PD gains in a range of conditions (21) does not improve tracking control quality and can lead to instability. The matrix $\mathbf{R}$, in the cost function, was set to $\mathbf{R}$ = diag{1, 1}, the discount factor was equal to $\gamma = 0.5$, learning rates of the actor's NNs and the critic's NN were equal to $\Gamma_{A[i,i]} = 0.1$ and $\Gamma_{C[i,i]} = 0.9$ properly, $i = 1, \ldots, 8$, $\eta_1 = \eta_2 = 1$. Parameters of the supervisory term were set to $\rho_{[j]} = 3$ and $\sigma_{[j]} = 0.09$. The maximal velocity of point A of the WMR's frame was equal to v_A = 0.4 m/s. During the movement of the WMR two parametric disturbances were simulated (marked on diagrams by ellipses), first in t_1 = 12.5 s, when the nominal set of parameters was changed to $\mathbf{a}_d = [0.1343, 0.0945, 0.037, 0.0001, 2.296, 2.296]^T$ and the second one, when in t_1 = 32.5 s, nominal

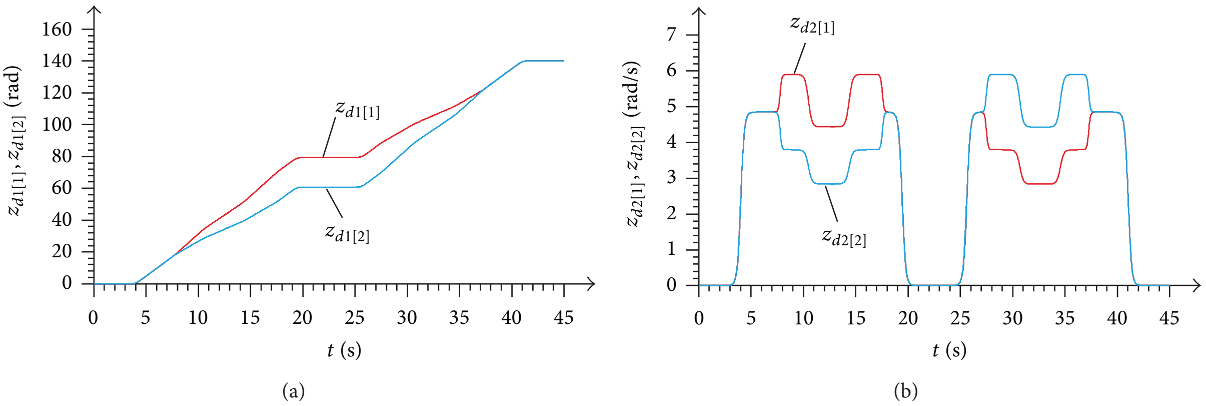


FIGURE 8: (a) The desired angles of wheels 1 and 2 rotation, $z_{d1[1]}$ and $z_{d1[2]}$, (b) the desired angular velocities of driving wheels 1 and 2, $z_{d2[1]}$ and $z_{d2[2]}$.

values of parameters were restored. The first change of parameters corresponds to the situation, when the WMR is loaded by an additional mass $m_L = 5$ kg, and a return to the nominal set of parameters corresponds to the situation, when the additional load is removed.

The desired trajectory of the WMR was computed earlier. In Figure 8(a) the desired angles of the driving wheels', 1 and 2, rotation are shown; in Figure 8(b) the desired angular velocities are presented. Realisation of the presented trajectory results in movement of point A of the WMR on the path in a shape of a digit "8," with a stop phase in the middle point.

The overall tracking control signal $\mathbf{u}$, shown in Figure 9(a), consists of the control signals generated by the actor's NNs $\mathbf{u}_A$, (Figure 9(b)), the PD control signals $\mathbf{u}_{\mathrm{PD}}$, (Figure 9(c)), the supervisory term's control signals $\mathbf{u}_S$, and the additional control signals $\mathbf{u}_E$, shown both in Figure 9(d). At the beginning of the numerical test, values of the PD control signals are big. Next, they are reduced during the NNs adaptation process. The control signals of the actor take the main part in the overall control signals. In time t_1, when the first parametric disturbance occurs, a change in values of the generated control signals can be observed. The additional load changes the dynamics of the WMR; realisation of the desired trajectory requires generating higher values of the control signals. The influence of the disturbance on the WMR's dynamics is compensated by the actor's NNs control signals. Analogically, the change of the WMR's parameters in time t_2, which simulates removal of the additional load, is compensated in the generated control law by reduction of the actor's NNs control signals values.

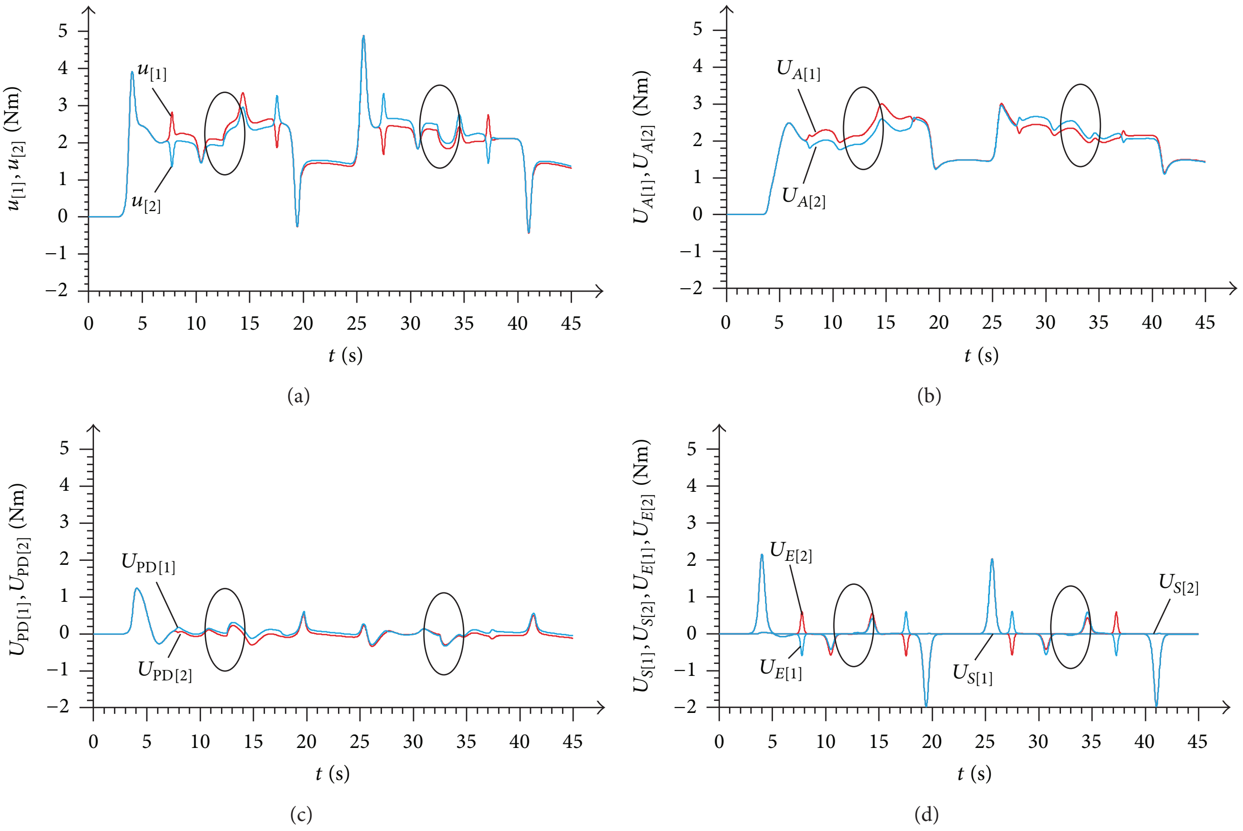
The desired and realised angular velocities of driving wheels 1 and 2 are shown in Figures 10(a) and 10(b), respectively. The biggest differences between the desired and realised angular velocities occur at the beginning of the numerical test. Small changes of realised angular velocities can be observed at the moment, when the parametric disturbances occur.

The desired trajectory was realised with tracking errors shown in Figures 11(a) and 11(b) for adequate driving wheels. In Figures 11(c) and 11(d), values of filtered tracking errors $s_{[1]}$ and $s_{[2]}$ are shown that are minimised by the ADP structure. The highest values of the tracking errors occur at the beginning of the numerical test, when values of the PD control signals are at their highest, and the process of NNs' zero initial weights adaptation starts. Next, the control signals of the actor's NNs take the main part of the overall control signals, and the values of tracking errors are reduced. A noticeable increase of the tracking error values occurs at the time of simulated disturbances, but it is reduced by the change of the actor's NNs control signals.

Values of the GDHP structure's NNs weights are shown in Figure 12(a) for the first actor's NN, in Figure 12(b) for the second one, and in Figure 12(c) for the critic's NN. In the numerical test, zero initial weights values were used. At the time of the disturbances, changes of weights' values occur as a result of the adaptation performed in order to reduce the tracking errors.

6.2. Verification Results. After numerical tests were performed, a series of experiments were realised using the WMR Pioneer 2-DX. The control algorithm operated in real time during the experiment, thanks to the application of the dSpace DS1102 digital signal processing board. In the experiment, the same parameters of the control system as in the simulation were used. The values of signals from the experiment were not filtered. The control signals are shown in Figure 13. The first disturbance occurs at time $t_1 = 13$ s and the second one at time $t_2 = 33$ s. The PD control signals (Figure 13(c)) based on the tracking errors calculated on the basis of the realised trajectory, determined by using signals form incremental encoders. These signals are noised, which has an effect on the overall control signals (Figure 13(a)). In contrast, the actor's NNs control signals (Figure 13(b)) and residual control signals (Figure 13(d)) are smooth. As it was observed in the simulation, at the time of the disturbances, the values of the actor's NNs control signals changed to compensate the effect of the WMR's dynamics change.

The biggest differences between the desired and realised angular velocities, shown in Figure 14, occur at the beginning of the experiment, when the process of the actor's NNs weights adaptation starts and at the time when the disturbances occur.



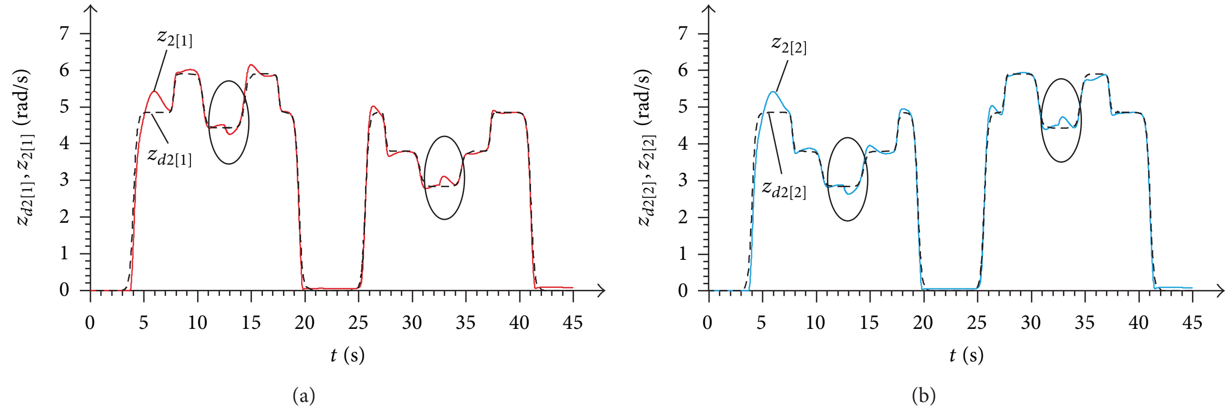
(a)

(b)

(c)

(d)

Figure 9: (a) The overall tracking control signals $u_{[1]}$ and $u_{[2]}$, (b) the actor's NNs control signals $U_{A[1]}$ and $U_{A[2]}$, $\mathbf{U}_A = -h^{-1}\mathbf{M}\mathbf{u}_A$, (c) the PD control signals $U_{\mathrm{PD}[1]}$ and $U_{\mathrm{PD}[2]}$, $\mathbf{U}_{\mathrm{PD}} = -h^{-1}\mathbf{M}\mathbf{u}_{\mathrm{PD}}$, (d) the supervisory term's control signals $(U_{S[1]}, U_{S[2]})$, $\mathbf{U}_S = -h^{-1}\mathbf{M}\mathbf{u}_S$, and the control signals $U_{E[1]}$ and $U_{E[2]}$, $\mathbf{U}_E = -h^{-1}\mathbf{M}\mathbf{u}_E$.



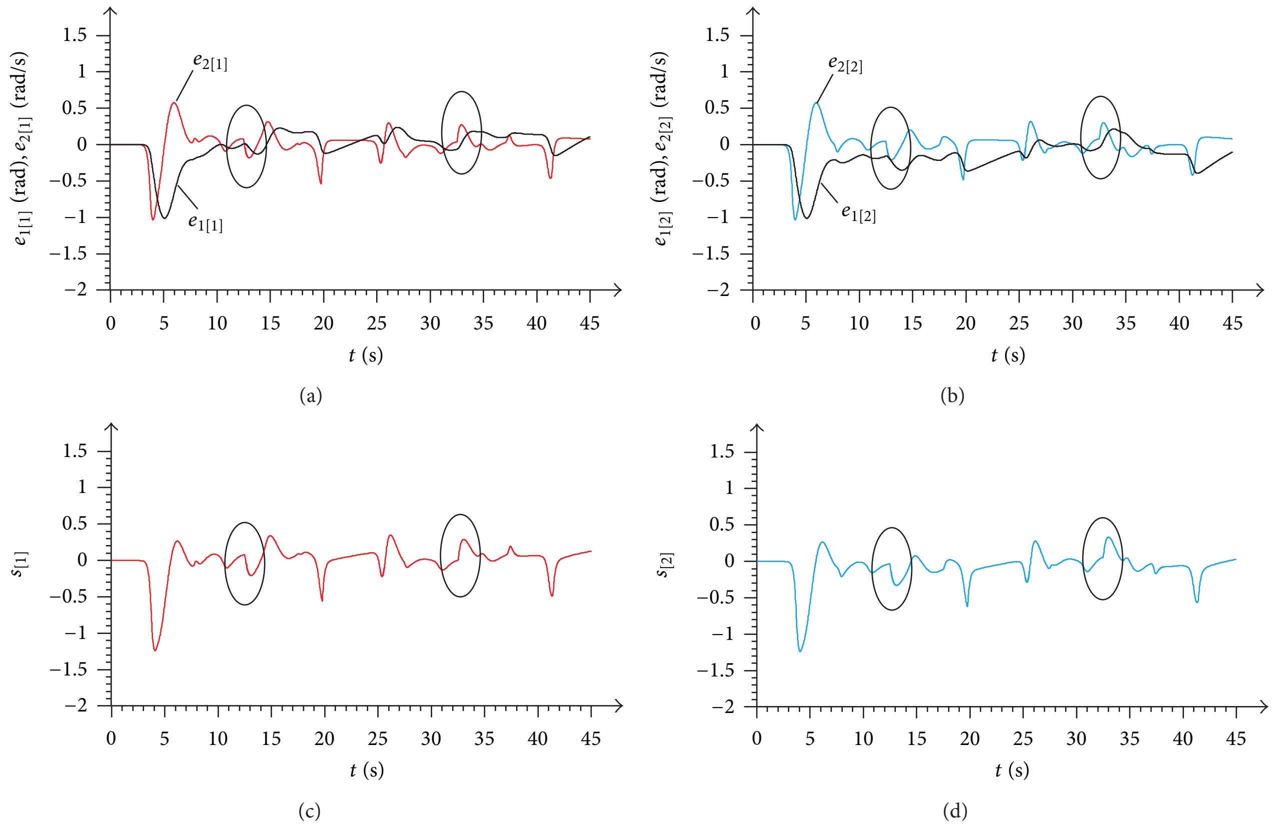
(a)

(b)

Figure 10: (a) The desired (dashed line) and realised (continuous line) angular velocity of wheel 1, $z_{d2[1]}$ and $z_{2[1]}$, (b) the desired (dashed line) and realised (continuous line) angular velocity of wheel 2, $z_{d2[2]}$ and $z_{2[2]}$.

The tracking errors of wheels 1 and 2 are shown in Figures 15(a) and 15(b); filtered tracking errors are shown in Figures 15(c) and 15(d). Values of errors are noisy, because of the realised method of measurement of the movement parameters. The errors at the beginning of the experiment are at their highest. The change of the load transported by the WMR has noticeable influence on the trajectory realisation process. The method of placing the load on the WMR and removing it has a big influence on temporary values of errors. The increase of errors values results in the adaptation of the actor's and the critic's NNs weights in order to minimise tracking errors.

Values of NNs' weights are shown in Figure 16. At a time, when the WMR transports an additional load, values of the



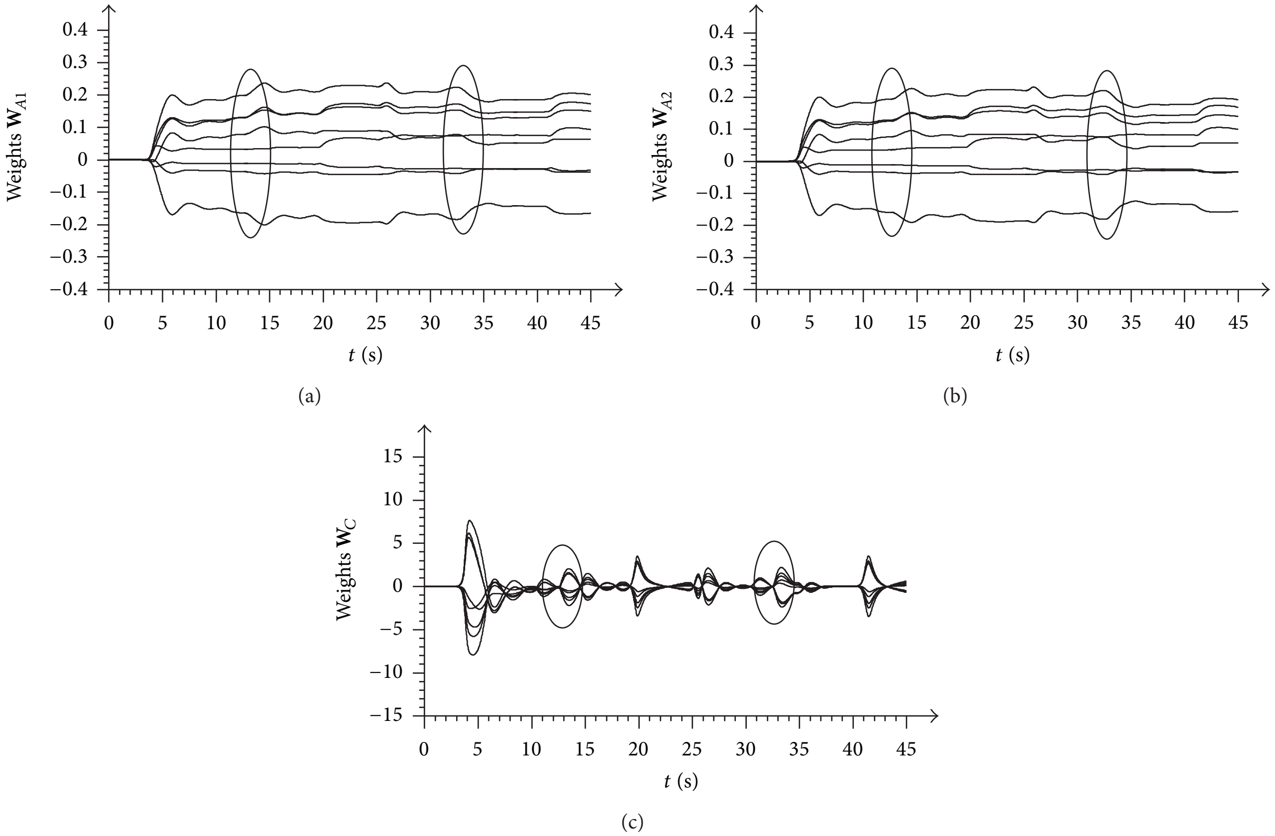
(a)

(b)

(c)

(d)

Figure 11: (a) Tracking errors of wheel 1, $e_{1[1]}$ and $e_{2[1]}$, (b) tracking errors of wheel 2, $e_{1[2]}$ and $e_{2[2]}$, (c) the filtered tracking error $s_{[1]}$, and (d) the filtered tracking error $s_{[2]}$.



(a)

(b)

(c)

Figure 12: (a) Weights of the actor's 1 RVFL NN $\mathbf{W}_{A1}$, (b) weights of the actor's 2 RVFL NN $\mathbf{W}_{A2}$, and (c) weights of the critic's RVFL NN $\mathbf{W}_{C}$.

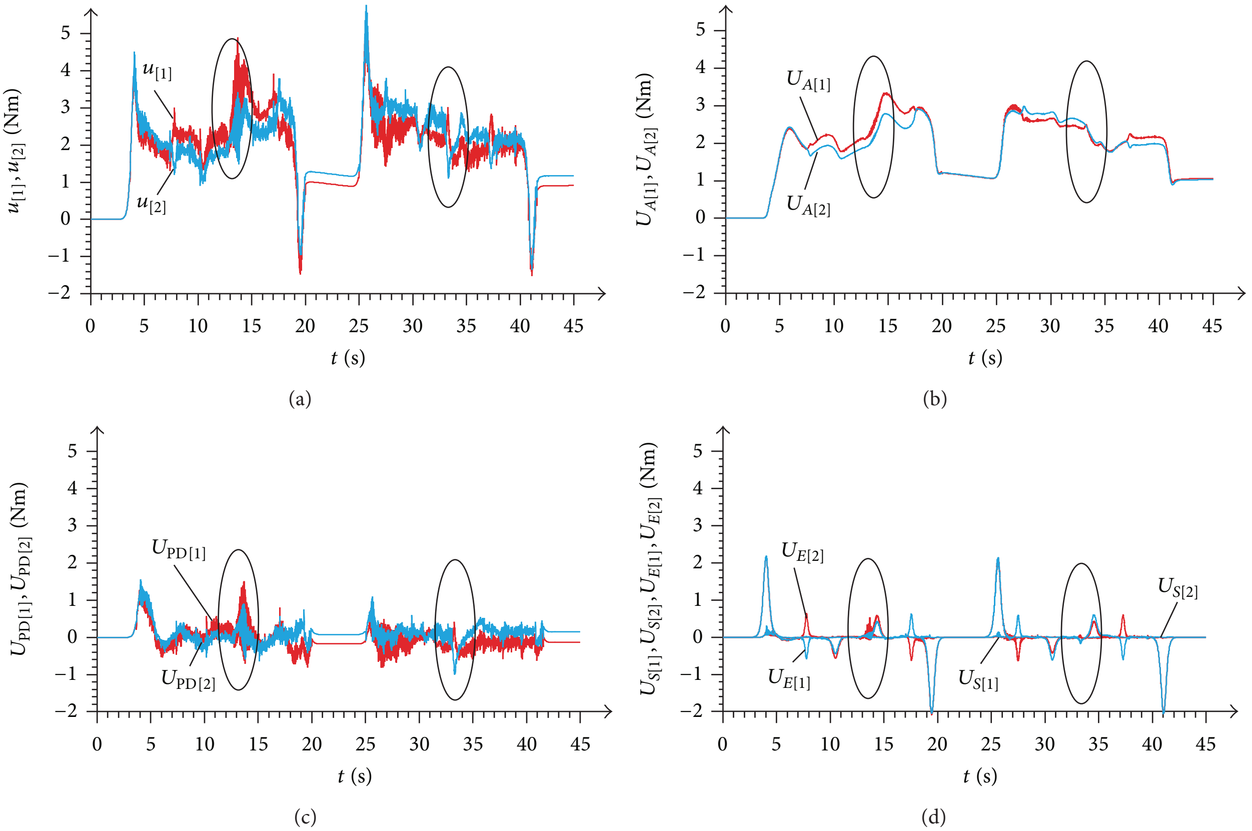


FIGURE 13: (a) The overall tracking control signals $u_{[1]}$ and $u_{[2]}$, (b) the actor's NNs control signals $U_{A[1]}$ and $U_{A[2]}$, $\mathbf{U}_A = -h^{-1}\mathbf{M}\mathbf{u}_A$, (c) the PD control signals $U_{\mathrm{PD}[1]}$ and $U_{\mathrm{PD}[2]}$, $\mathbf{U}_{\mathrm{PD}} = -h^{-1}\mathbf{M}\mathbf{u}_{\mathrm{PD}}$, (d) the supervisory term's control signals ($U_{S[1]}$, $U_{S[2]}$), $\mathbf{U}_S = -h^{-1}\mathbf{M}\mathbf{u}_S$, and the control signals $U_{E[1]}$ and $U_{E[2]}$, $\mathbf{U}_E = -h^{-1}\mathbf{M}\mathbf{u}_E$.

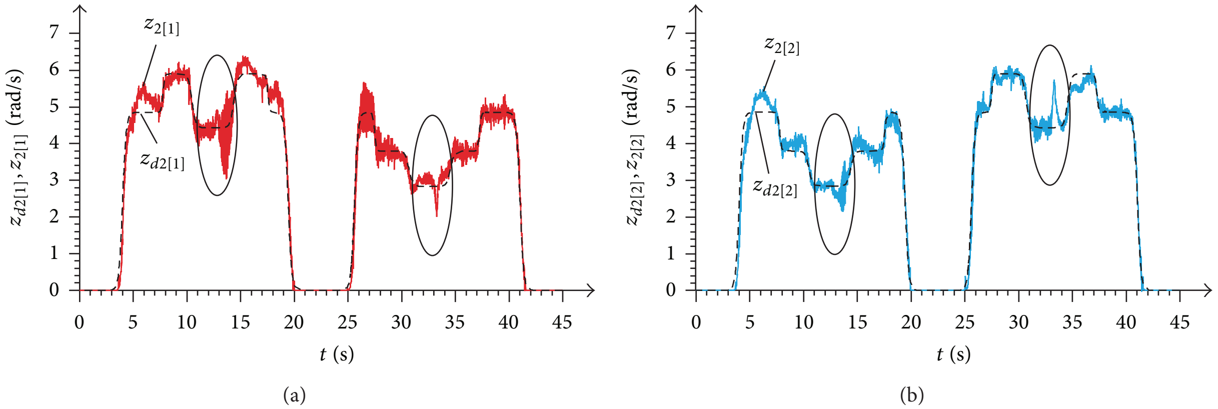
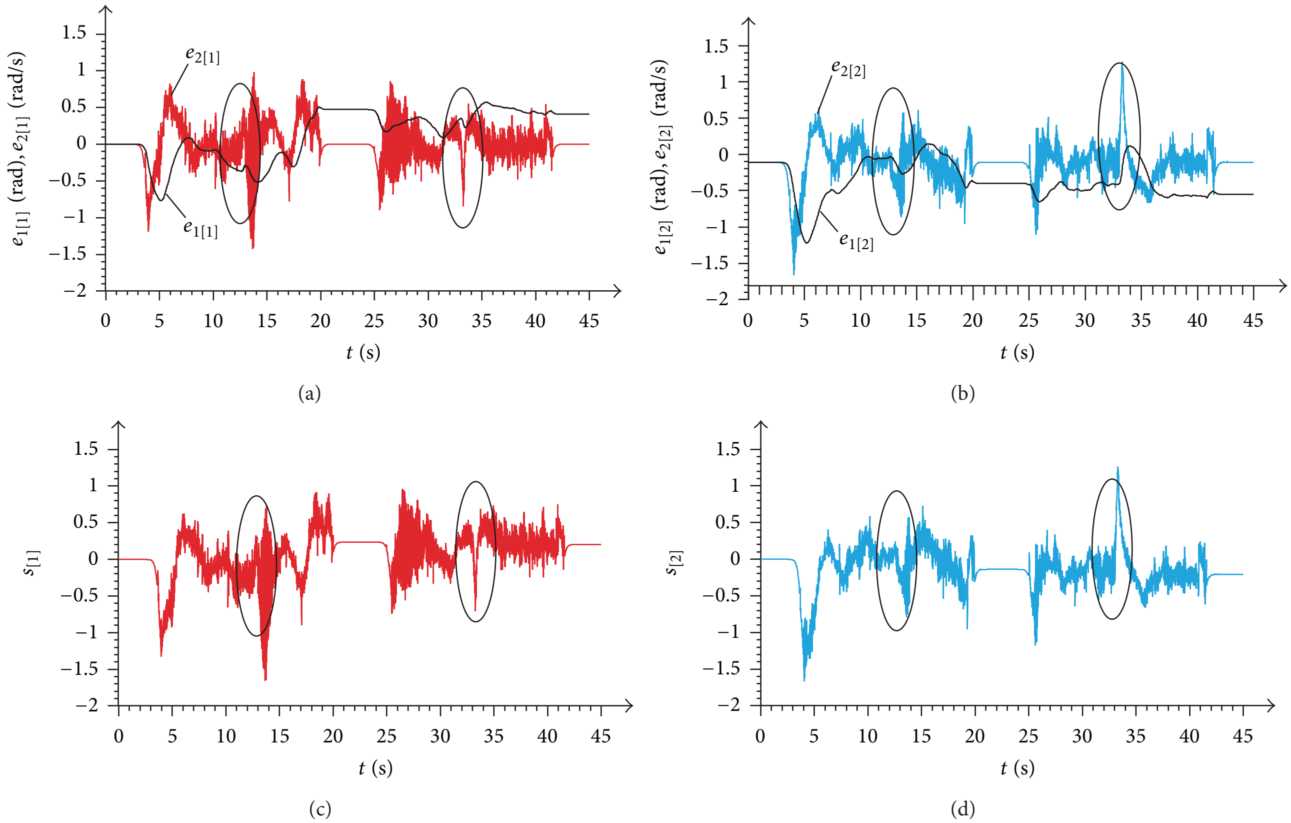


FIGURE 14: (a) The desired (dashed line) and realised (continuous line) angular velocity of wheel 1, $z_{d2[1]}$ and $z_{2[1]}$, (b) the desired (dashed line) and realised (continuous line) angular velocity of wheel 2, $z_{d2[2]}$ and $z_{2[2]}$.

actor's NNs weights increase. This is a result of generating higher values of the actor's control signals for the heavier WMR. The critic's NN approximates the value function based on the filtered tracking errors, values of its weights increase and when the values of filtered tracking errors increase.

The tracking quality of the proposed control system was compared to the results obtained by the tracking control systems presented earlier, where ADP algorithms in HDP and DHP [43] configuration, or the PD controller ($\mathbf{K}_D = \mathrm{diag}\{1, 1\}$, $\mathbf{\Lambda} = \mathrm{diag}\{0.5, 0.5\}$), were used. Every experiment was performed in the same conditions, using the same or



(a)

(b)

(c)

(d)

FIGURE 15: (a) Tracking errors of wheel 1, $e_{1[1]}$ and $e_{2[1]}$, (b) tracking errors of wheel 2, $e_{1[2]}$ and $e_{2[2]}$, (c) the filtered tracking error $s_{[1]}$, and (d) the filtered tracking error $s_{[2]}$.

analogical values of parameters, and the same type of the disturbance.

To evaluate the tracking control quality, the following quality ratings were used:

(i) average of maximal values of the filtered tracking error for wheels 1 ($s_{\max[1]}$) and 2 ($s_{\max[2]}$):

$$s_{\mathrm{mavr}} = \frac{1}{2}\left(s_{\max[1]} + s_{\max[2]}\right), \tag{35}$$

(ii) average of root mean square error (RMSE) of the filtered tracking errors $s_{[1]}$ and $s_{[2]}$:

$$\varepsilon_{\mathrm{avr}} = \frac{1}{2}\left(\sqrt{\frac{1}{N}\sum_{k=0}^{N} s_{[1]\{k\}}^{2}} + \sqrt{\frac{1}{N}\sum_{k=0}^{N} s_{[2]\{k\}}^{2}}\right), \tag{36}$$

where $N = 4500$.

Values of quality ratings are shown in Table 1.

Average of maximal values of the filtered tracking error for wheels 1 ($s_{\max[1]}$) and 2 ($s_{\max[2]}$) is shown in Figure 17(a), and values of RMSE of the filtered tracking errors $s_{[1]}$ and $s_{[2]}$ are shown in Figure 17(b).

On the basis of the obtained results, the higher quality of tracking for the control systems with ADP algorithms in comparison to the PD controller can be noticed. In the presented paper the goal was not to demonstrate the maximal quality of the tracking control attainable using highest feasible to apply the PD controller gains but to illustrate the increase of the quality of the tracking control after adding, to the control system, a part that compensates for nonlinearities of the control system. Values of the quality ratings for the control system with the GDHP structure are close to the ones obtained by the control system with the DHP structure. Simultaneously values of quality ratings are lower than obtained using the HDP algorithm, which means that the application of more complex critic's NN weights adaptation rule improves the quality of control.

TABLE 1: Values of quality ratings.

Control algorithm	PD	HDP	GDHP	DHP
s_{mavr}	4.06	2.24	1.66	1.52
$\varepsilon_{\mathrm{avr}}$	1.99	0.42	0.32	0.24

7. Conclusion

The paper presents the discrete tracking control system of the WMR Pioneer 2-DX. The main element of the control system is the ADP algorithm in the GDHP configuration. It consists of the actor and the critic, realised in a form of RVFL NNs. The additional elements of the control system, like the PD controller or the supervisory term, assure stability of the tracking control in case of disturbances, or at the

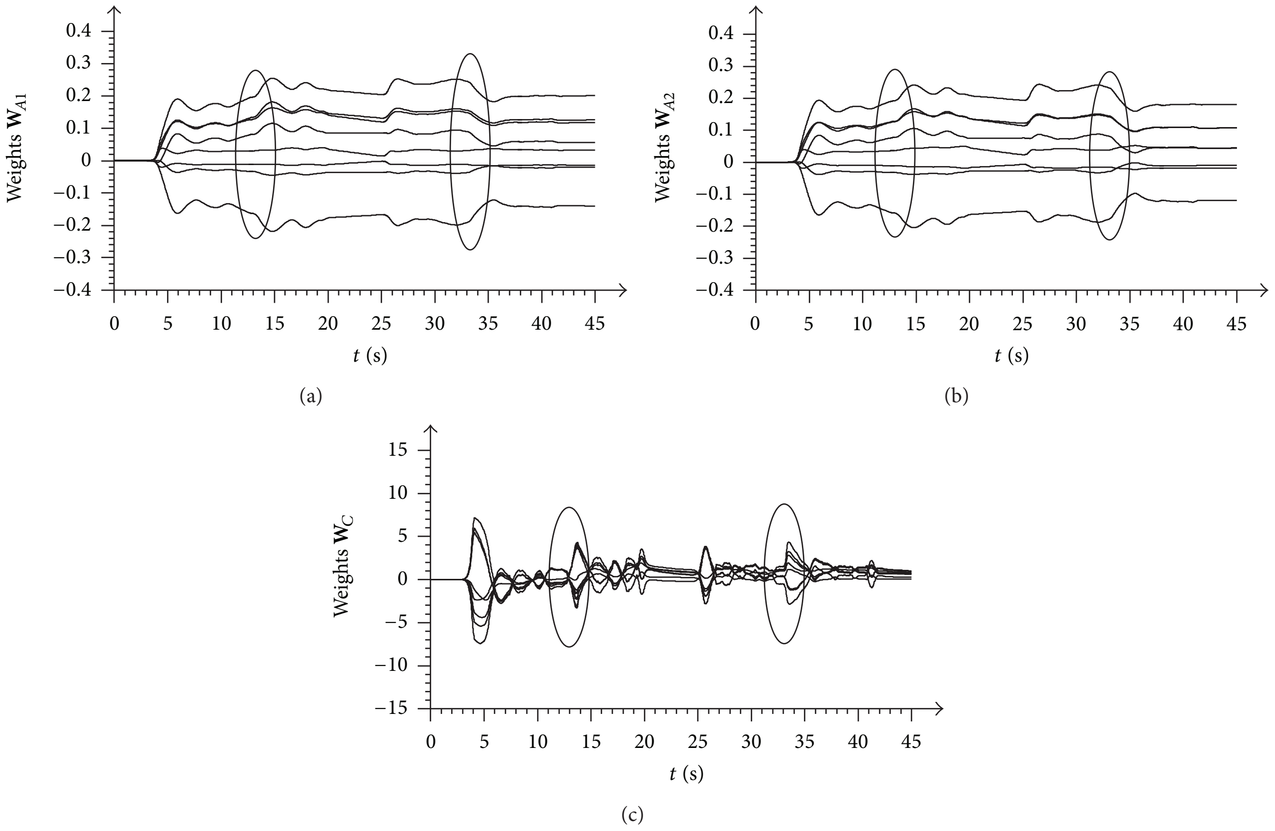


Figure 16: (a) Weights of the actor's 1 RVFL NN $\mathbf{W}_{A1}$, (b) weights of the actor's 2 RVFL NN $\mathbf{W}_{A2}$, and (c) weights of the critic's RVFL NN $\mathbf{W}_{C}$.

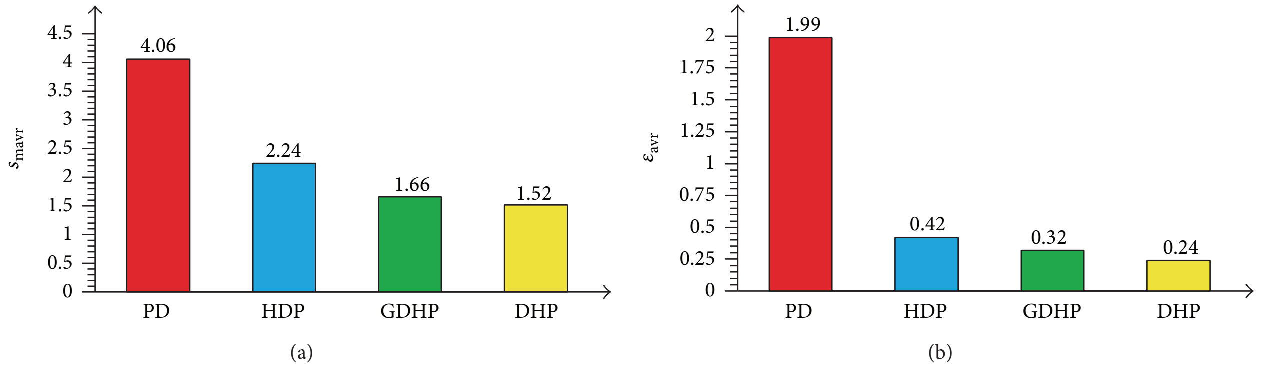


Figure 17: (a) Average of maximal values of the filtered tracking error for wheels 1 ($s_{\max[1]}$) and 2 ($s_{\max[2]}$), (b) RMSE of the filtered tracking errors $s_{[1]}$ and $s_{[2]}$.

beginning of movement, in the case when values of the actor's NNs weights are not adequately selected for the controlled system; for example, the process of preliminary learning was not realised, or zero initial weights were applied. PD controller gains were selected experimentally for the control system with the GDHP algorithm. Next the experiment for the control system with only the PD controller, with the same parameters, was performed to demonstrate the increase of the tracking control quality for the tracking control system compensating nonlinearities of the control object. It is important to indicate that in a case of realisation of the control system, with nonlinearities compensation, the primary part of the system is the nonlinear compensator. The nonlinear compensator, realised in the form of a GDHP algorithm, compensates for the nonlinearities of the controlled object, as well as the parametrical and the structural disturbances. The GDHP algorithm has the same structure as HDP and its critic's structure is simpler than in DHP. In the GDHP algorithm the critic's NN weights are adapted using a more complex adaptation law, which is composed of the critic's

NN weights adaptation rule of the HDP algorithm and the DHP algorithm. This feature assures a high quality of tracking, higher than the quality of tracking obtained when using the control system with the HDP algorithm, and close to the quality of tracking for the control system with the DHP algorithm, which is a significant advantage. The presented control system is stable; the values of errors and NNs' weights are bounded. Even in the case of zero initial weights of NNs application, or in the case of disturbances, the proposed control system guarantees a stable tracking process. The discrete tracking control system works online and does not require a process of preliminary learning of NNs. Performance of the control system was verified by a series of numerical tests and experiments realised using the WMR Pioneer 2-DX.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 589–600, 1998.

[2] M. Giergiel, Z. Hendzel, and W. Zylski, *Modelling and Control of Wheeled Mobile Robots*, PWN, Warsaw, 2002, (Polish).

[3] D. Liu, D. Wang, and X. Yang, "An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs," *Information Sciences*, vol. 220, pp. 331–342, 2013.

[4] R. Syam, K. Watanabe, and K. Izumi, "Adaptive actor-critic learning for the control of mobile robots by applying predictive models," *Soft Computing*, vol. 9, no. 11, pp. 835–845, 2005.

[5] W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds., *Neural Networks for Control, A Bradford Book*, The MIT Press, Cambridge, Mass, USA, 1990.

[6] Z. Wiesław and G. Piotr, "Verification of multilayer neural-net controller in manipulator tracking control," *Solid State Phenomena*, vol. 164, pp. 99–104, 2010.

[7] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1957.

[8] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 13, no. 5, pp. 834–846, 1983.

[9] A. G. Barto, W. B. Powell, J. Si, and D. Wunsch, *Handbook of Learning and Approximate Dynamic Programming*, Wiley-IEEE Press, New York, NY, USA, 2004.

[10] A. G. Barto and R. Sutton, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, Mass, USA, 1998.

[11] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Wiley, Hoboken, NJ, USA, 2007.

[12] D. V. Prokhorov and D. C. Wunsch II, "Adaptive critic designs," *IEEE Transactions on Neural Networks*, vol. 8, no. 5, pp. 997–1007, 1997.

[13] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: an introduction," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 39–47, 2009.

[14] F. L. Lewis, D. Liu, and G. G. Lendaris, "Guest editorial: special issue on adaptive dynamic programming and reinforcement learning in feedback control," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 38, no. 4, pp. 896–897, 2008.

[15] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, 2009.

[16] X. Xu, L. Zuo, and Z. Huang, "Reinforcement learning algorithms with function approximation: recent advances and applications," *Information Sciences*, vol. 261, pp. 1–31, 2014.

[17] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[18] K. G. Vamvoudakis and F. L. Lewis, "Multi-player non-zero-sum games: online adaptive learning solution of coupled Hamilton-Jacobi equations," *Automatica*, vol. 47, no. 8, pp. 1556–1569, 2011.

[19] D. Vrabie and F. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks*, vol. 22, no. 3, pp. 237–246, 2009.

[20] L. C. Baird III, "Reinforcement learning in continuous time: advantage updating," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '94)*, pp. 2448–2453, June 1994.

[21] T. Hanselmann, L. Noakes, and A. Zaknich, "Continuous-time adaptive critics," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 631–647, 2007.

[22] A. Y. Ng, H. J. Kim, M. I. Jordan, and S. Sastry, "Autonomous helicopter flight via reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 16, 2004.

[23] M. Carreras, J. Yuh, J. Batlle, and P. Ridao, "A behavior-based scheme using reinforcement learning for autonomous underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 30, no. 2, pp. 416–427, 2005.

[24] M. A. Kareem Jaradat, M. Al-Rousan, and L. Quadan, "Reinforcement based mobile robot navigation in dynamic environment," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 135–149, 2011.

[25] P. G. Balaji, X. German, and D. Srinivasan, "Urban traffic signal control using reinforcement learning agents," *IET Intelligent Transport Systems*, vol. 4, no. 3, pp. 177–188, 2010.

[26] D. Ernst, M. Glavic, and L. Wehenkel, "Power systems stability control: reinforcement learning framework," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 427–435, 2004.

[27] S. Mohagheghi, G. K. Venayagamoorthy, and R. G. Harley, "Adaptive critic design based neuro-fuzzy controller for a static compensator in a multimachine power system," *IEEE Transactions on Power Systems*, vol. 21, no. 4, pp. 1744–1754, 2006.

[28] K. M. Iftekharuddin, "Transformation invariant on-line target recognition," *IEEE Transactions on Neural Networks*, vol. 22, no. 6, pp. 906–918, 2011.

[29] J. del R. Millán, "Reinforcement learning of goal-directed obstacle-avoiding reaction strategies in an autonomous mobile robot," *Robotics and Autonomous Systems*, vol. 15, no. 4, pp. 275–299, 1995.

[30] X. Zhang, H. Zhang, Q. Sun, and Y. Luo, "Adaptive dynamic programming-based optimal control of unknown nonaffine nonlinear discrete-time systems with proof of convergence," *Neurocomputing*, vol. 91, pp. 48–55, 2012.

[31] D. Wang, D. Liu, and Q. Wei, "Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach," *Neurocomputing*, vol. 78, no. 1, pp. 14–22, 2012.

[32] X. Xu, Z. Hou, C. Lian, and H. He, "Online learning control using adaptive critic designs with sparse kernel machines," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 762–775, 2013.

[33] D. Wang, D. Liu, D. Zhao, Y. Huang, and D. Zhang, "A neural-network-based iterative GDHP approach for solving a class of nonlinear optimal control problems with control constraints," *Neural Computing and Applications*, vol. 22, no. 2, pp. 219–227, 2013.

[34] M. Fairbank, E. Alonso, and D. Prokhorov, "Simple and fast calculation of the second-order gradients for globalized dual heuristic dynamic programming in neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 10, pp. 1671–1676, 2012.

[35] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, "Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming," *Automatica*, vol. 48, no. 8, pp. 1825–1832, 2012.

[36] G. K. Venayagamoorthy, D. C. Wunsch, and R. G. Harley, "Adaptive critic based neurocontroller for turbogenerators with global dual heuristic programming," in *Proceeding of the IEEE Power Engineering Society Winter Meeting*, vol. 1, pp. 291–294, Singapore, January 2000.

[37] J. Peters and S. Schaal, "Natural actor-critic," *Neurocomputing*, vol. 71, no. 7–9, pp. 1180–1190, 2008.

[38] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2226–2236, 2011.

[39] Z. Hendzel, A. Burghardt, and M. Szuster, "Reinforcement learning in discrete neural control of the underactuated system," *Lecture Notes in Artificial Intelligence*, vol. 7894, pp. 64–75, 2013.

[40] P. Gierlak, M. Szuster, and W. Zylski, "Discrete dual-heuristic programming in 3DOF manipulator control," in *Artifical Intelligence and Soft Computing*, vol. 6114 of *Lecture Notes in Artificial Intelligence*, pp. 256–263, 2010.

[41] Z. Hendzel, "An adaptive critic neural network for motion control of a wheeled mobile robot," *Nonlinear Dynamics*, vol. 50, no. 4, pp. 849–855, 2007.

[42] Z. Hendzel and M. Szuster, "Discrete action dependent heuristic dynamic programming in wheeled mobile robot control," *Solid State Phenomena*, vol. 164, pp. 419–424, 2010.

[43] Z. Hendzel and M. Szuster, "Discrete model-based adaptive critic designs in wheeled mobile robot control," *Lecture Notes in Computer Science*, vol. 6114, no. 2, pp. 264–271, 2010.

[44] Z. Hendzel and M. Szuster, "Discrete neural dynamic programming in wheeled mobile robot control," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 5, pp. 2355–2362, 2011.

[45] Z. Hendzel and M. Szuster, "Neural dynamic programming in reactive navigation of wheeled mobile robot," in *Artificial Intelligence and Soft Computing*, vol. 7268 of *Lecture Notes in Computer Science*, pp. 450–457, 2012.

[46] J. Giergiel and W. Zylski, "Description of motion of a mobile robot by Maggies Equations," *Journal of Theoretical and Applied Mechanics*, vol. 43, no. 3, pp. 511–521, 2005.

[47] F. L. Lewis, J. Campos, and R. Selmic, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2002.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 508314, 15 pages
http://dx.doi.org/10.1155/2014/508314

Research Article

Methodology for Supply Chain Integration: A Case Study in the Artisan Industry of Footwear

Jorge Raúl Pérez-Gallardo,[1,2] Beatriz Hernández-Vera,[2] Constantino Gerardo Moras Sánchez,[2] Alberto Alfonso Aguilar Lasserre,[2] Rubén Posada-Gómez,[2] Ulises Juárez-Martínez,[2] and Giner Alor-Hernández[2]

[1] *Université de Toulouse, INPT, UPS, Laboratoire de Génie Chimique, 4 allée Emile Monso, 31432 Toulouse, France*
[2] *Division of Research and Postgraduate Studies, Instituto Tecnológico de Orizaba, 852 Oriente 9, 94320 Orizaba, VER, Mexico*

Correspondence should be addressed to Beatriz Hernández-Vera; bh.vera@hotmail.com

Received 11 April 2014; Accepted 20 June 2014; Published 7 September 2014

Academic Editor: Ker-Wei Yu

The supply chain is a major issue in a global market. The efficient relationship between supplier-producer-retailer conforms the perfect combination to achieve a high level of competitiveness according to the needs of increasingly demanding and changing markets. The difficulty of synchronizing the members within the SC is related to the inherent uncertain factors. This paper proposes a methodology for integrating the supply chain that can be applied in any type of business. To demonstrate its relevance, a case study is performed at a handmade shoe company. An improved demand forecasting, an optimal balanced production line, a proper inventory management of raw materials, and an efficient calculation of the delivery time to the customer represent some of the main results.

1. Introduction

Proper integration of supply chain (SC) is a major issue because in a global world the company competitiveness is the key to stay in the market. Nevertheless, in the competition for its permanence, companies must sail endless problems.

These problems force the enterprises not only to efficiently coordinate their activities internally, but also to establish and preserve harmonious relations with its suppliers and retailers within the SC. The synchronization of the supply chain is a topic that has been widely discussed in different approaches. A healthy relationship between the three main members (supplier-producer-retailer) leads companies to get a high level of competitiveness. The complexity of harmonizing these three elements lies in the various viewpoints and approaches that each member may have in order to generate an adequate coordination. For the manufacturer it is important to have a system that supports an efficient planning, scheduling, and production control, a proper selection of their suppliers and to define efficient distribution protocols in order to be able to integrate, synchronize, and optimize the performance of the entire chain.

The competitiveness of markets has forced manufacturers to select the most appropriate supply chain network to reduce total costs and loss of time. In the literature there are examples of various methodologies to achieve the supply chain integration, for example, Ghasimi et al. [1] used genetic algorithms to optimize a novel mathematical model for the supply chain network for defective products, in order to minimize the production, distribution, and maintenance costs. The model permits to determine the amount of economic production, the length of each cycle, and the number of defective products, waste products, and retail scarcity using just-in-time technique.

Efendigil and Önüt [2] present a methodology for SC integration from customers to suppliers through warehouses, retailers, and plants via both adaptive network based fuzzy inference system and artificial neural networks approaches.

The methodology presented provides this integration by finding the requested supplier capacities using the demand and order lead time information across the whole SC in an uncertain environment. The sensitivity analysis is made by comparing the obtained results with the traditional statistical techniques.

Finally Hon and Ng [3] propose a Process Integration (PI) approach to supply chains, especially developing further the analogy between Pinch Analysis and the field of operational research. Their work provides clear insight into how the PI concept brings a significant contribution to the development of supply chain design.

In Mexico, according to the National Institute of Statistics, Geography and Informatics (INEGI) [4], over 90% of business units across the country are SMEs, which represent 52% of GDP. Most companies have seen a decline of their profits due to poor internal organization and administration. In addition, many of the products or goods are unattractive to customers. There is a fierce competition both national and international to find a distinctive feature that makes a product or service unique. One of the industries where the difficulties are more visible is the shoe industry. In this article the case of a shoe maker company is presented to demonstrate the performance of the methodology proposed.

We present a state of art in the following section, the problem to address will be exposed in the third section, in the fourth section the proposed methodology for the integration of supply chains is presented. In the fifth section the methodology is applied in a case study and finally the conclusions about the work are presented.

2. State of the Art

At the present, the highly competitive and changing markets have caused that companies demand quick responses that represent an advantage to face its competitors [5]. Besides the rapid evolution of technological innovation, the stiff competition and the fluctuating consumer demands have forced to establish a closer cooperation between suppliers, producers, and distributors, as well as the study of the relationships between the members within the chain supply in order to improve their performance [6].

Global competition is imposing to current industry to make high quality products in a faster and cheaper way. The demand for a quick exchange of information has increased, especially among remote sites, for creating a better cooperation and coordination. Several papers seek the effective coordination between the entities that make up the supply chain network. Traditionally, the strategic management of human resource had been focused as an isolated entity. Lengnick-Hall et al. [7] present a support framework for identifying factors that influence the efficient orientation of the supply chain and define strategic management practices to capitalize this orientation. The framework also generates a better human resource strategic management, a better relationships both internal and external at the organization, and understanding of the links among the human resource systems, the orientation of the supply chain, and the strategic results.

Palma-Mendoza et al. [8] propose a detailed plan to support the supply chain integration based on the unified framework of three different domains: redesign of business process, redesign of the supply chain process, and design e-business, which can be applied to any company or sector.

Several tools and techniques had been used to achieve the supply chain integration depending on the current needs. Eliman and Dodin [9] propose a supply chain integration as a network of projects that covers certain activities: sending, processing, and receiving orders. This network is modeled by mixed integer programming that captures various advantages and disadvantages as the stationary and transient inventory costs, processing activity costs, the shipping cost, and the penalty for delay in delivery of customer orders. The optimal solution found provides cycle times, modes of transport, processing times, and due dates at minimal cost. From another perspective Jafar [10] studied the supply chain in two stages in order to reduce the delivery time. He involved the distributors to participate in the coordination plan making emphasize in the importance of more reliable shipping equipments.

Mehdi [11] presents a programming model for multiple objectives with the purpose to adopt an integrated approach that support the decision making on the optimal allocation of the limited resources in the supply chain, the suppliers selection, the production, distribution, and schedule of supplies, and the lower cost and the higher income, thus, maximizing the benefit of the chain. The model considers the integration level of suppliers, producers, and distributors, as well as the free relationships among the producers to manage the product supply through the process or even the products of each other, and finally consider the effect of the economic factors in decisions such as inflation.

Zhang et al. [12] conducted a biobjective model for the design of the supply chain for dispersed manufacturing considering essential trade-offs between the supply chain costs and the lead time to determine the optimal location of the manufacturing activities.

It is imperative for contemporary companies proactively seek to improve continuously the performance of their supply chain. The coordination and the integration of the decision making across the supply chain between the various partners that conform it are often used for this purpose. Such coordination strategies of the supply chain include the use of a common cycle time, fix quantity discounts, the use of an optimal batch sizing, the quality improvements and inspections, to mention a few. An important issue in the supply chain refers to the incorporation of human factors presented thought the supply chain task such as the failed quality inspections and production improvements due to learning. Khan et al. [13] present a simple mathematical model for determining an optimal inventory policy seller-buyer that considers the quality inspection errors. The objective is to minimize the annual cost incurred in the supply chain.

To ensure the effective supply of biomass to produce biofuels on a large scale, Tao et al. [14] presented an optimization model of the supply chain to minimize the annual costs of biomass-ethanol production by optimizing both strategic and tactical planning decisions simultaneously. The mixed integer linear programming model optimizes the activities like biomass collection, packaging, transportation, internal transport, preprocessing, storage, production, and distribution of ethanol. The locations, capacities and quantity of facilities and distribution patterns of biomass and ethanol are considered as key strategic decisions, while biomass production, delivery and operation schedules and inventory monitoring are key tactical decisions.

A common problem at network design for integrated supply chain consists in determinate the locations of distribution centers and the allocation of customers and suppliers to the centers of distribution. The problem tries to seek at the same time the distribution of products from the manufacturer to the customer and the collection of components from suppliers to manufacturers through cross-docking at the distribution centers. Zhang et al. [15] propose an algorithm based on Lagrangian relaxation which includes the co-location of different types of distribution centers and the coordination of the transport to achieve savings in transportation costs.

Eren et al. [16] propose an integrated model that jointly optimizes the strategic and tactical SC closed loop decisions. Strategic level decisions concern the amounts of goods flowing in chains downstream and upstream, while tactical level decisions concern the balancing of removing lines of inverse chain. The objective is to minimize transportation, purchases, renovation, and operation costs of disassembly workstations. Finally, the authors formulated and described a nonlinear mixed integer programming model for this situation.

Sarkar and Majumder [17] propose an integrated SC supplier-seller model. Two models constructed on the basis of the probability distribution of the time were presented. The waiting time is normally distributed in the first model, while in the second model a free distribution for the delivery time is considered. For this second model, only the mean and standard deviation are known. The objective of the model is to reduce the total cost of the system taking into account the reduction of installation costs for the supplier.

A manufacturing SC with multiple suppliers in the presence of multiple uncertainties as uncertain supplies of materials, stochastic times of production and randomly costumer demand is a scenario that Song et al. [18] addressed in their study. They formulated an inventory management policy, a raw materials acquisition policy, and a production control policy using the stochastic dynamic programming approach. Then they investigated other strategies as the reduction of suppliers and the differentiation of suppliers under the integrated inventory management policy, establishing qualitative relationships between the amount of suppliers, the supplier capabilities, and the total expected cost. The model allows achieving quantitatively the best trade-off between reducing suppliers and improving the supplier capability, in addition to quantifying the supplier differentiation in purchase decisions.

The multiskill environment is a problem for staff allocation, so Telhada [19] formulates a mixed integer programming model whose application supports the assignment of shifts and tasks on the same problem.

Finally Zhang [20] formulated a model and a method for the coordination of production and inventory cycles throughout the manufacturing SC involving reverse logistics for items with multiperiod finite horizon inventory. An entire manufacturing SC involving reverse logistics includes category 2 Suppliers, category 1 suppliers, manufacturers, distributors, consumers, and Recycling. The authors propose a mathematical model to represent the behavior of this system; the solution methods proposed are based on a decentralized decision-making process and a combination of decentralized and centralized decision, known as the semicentralized decision making process. The centralized decision-making process is achieved by a mixed integer linear programming.

This brief literature review allows seeing the different angles from which the SC has recently been addressed. In general it can be concluded that the above efforts are looking to integrate the various links in the chain, in order to reduce costs and increase yields. But in particular, this study has dealt with the management and inventory control, the distribution of finished products, the identification of multiple suppliers in the presence of uncertainties, and the inclusion of human errors and their impacts to the chain, with this last point being the most important. The ability, performance, and proceess that the human resource had, can move the chain differently.

The above represents individual efforts to demonstrate how particular issues affect the operation of the SC. Also various techniques, algorithms or tools were used to potentially improve the SC and support the decision making process. All these contributions were tested into SC that does not include artisanal production processes, which opens the possibility to question the validity of these applications in such processes.

This article proposes a methodology for ensuring the SC integration regardless of the type of business, using traditional engineering techniques and artificial intelligence techniques. This integration does not detract from the traditional methods of problem solving but they complement each other.

3. Problem Statement

The SC synchronization is a topic that has been widely discussed in recent years as it could be shown in the previous section. Due to various factors, companies not only search the efficient coordination of their activities internally, but also maintain a harmonious relationship with its suppliers and dealers within this chain.

Globalization and new technologies are commonly terms used to reflect the current reality and are associated with ideas of change, speed, real-time communications, and electronic commerce, among others. However in this changing world still prevail situations such as the development of customized products for specific needs, the correct allocation of task in a process, and the reduction of delivery time to name a few.

At the present, company competitiveness is the key to staying in the market, but the fact is that in the competition for their permanence businesses must deal with endless problems. In addition some businessmen persist to see their companies as a standalone system and not as a member of a bigger system that influences the future behavior of the market. They do not consider that the competition

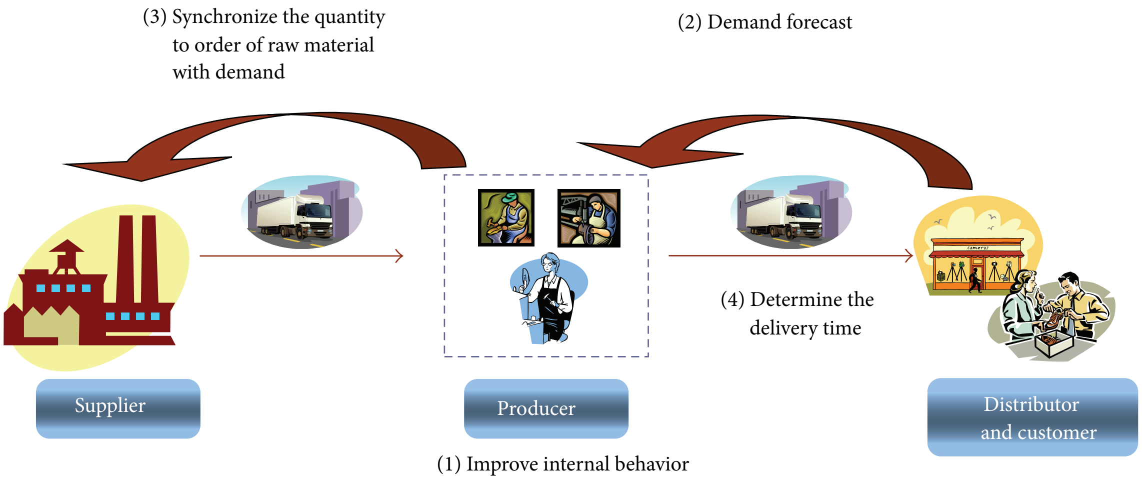

(3) Synchronize the quantity to order of raw material with demand
(2) Demand forecast
(4) Determine the delivery time
Supplier
Producer
Distributor and customer
(1) Improve internal behavior

Figure 1: Methodology for the integration of the supply chain.

between networks and the integration of key processes with trading partners allow them to form a much larger system, where suppliers and customers are no longer part of its environment, but part of his whole system known as SC.

It is necessary for companies to have support tools that allow them to guide their efforts to identify their place within an efficient SC, establishing harmonious relations with suppliers and distributors, in order to establish itself in a competitive market.

That why this paper presents a methodology to integrate the SC that aims to be a tool to optimize the resources and the flow of certain products thoughout the SC and even more to demonstrate the use of the proposed methodology a case study in the artesian company of footwear is presented. In general, the integration is made in two levels: first, the internal processes of the footwear company are analyzed and the solution of possible problem that could block the SC integration are looked for; second, strategies that permit an integration with the suppliers and retailers are proposed and evaluated.

This methodology is a flexible tool that enables each of its steps apply both traditional engineering tools and sophisticated or specific tools. Finally, it is appropriate for any type of business, including those whose production process is artisanal.

4. Methodology for Supply Chain Integration

As we already mention all companies that conform the SC must interact in order to achieve competitive advantage, which represent the reason why a methodology to guide the integration of each of the links in the SC was formulated. Each step of the proposed methodology is described below and schematized in Figure 1.

(1) Improve internal behavior: the first step for the successful integration of the company with the others that take part of the SC is to improve internal performance by applying traditional techniques to diagnose system performance, setting goals and strategies to improve the indicators and assess the progress.

(2) Demand forecast: this is a critical step to integrate the SC. A good demand forecast achieves synchronization of all the companies that conform the chain. This information is provided by historical records of the main selling points.

(3) Synchronize the quantity to order of raw material with demand: the aim is to reduce the costs generated by the stock of raw materials but they must be available at the time which has to make use of it from the forecasted demand.

(4) Determine the delivery time: the aim is to establish reliable times or delivery of the final product represents the last point.

The four steps of the methodology together enable the company improve its internal performance and to establish a harmonious and synchronized relationship these changes will place the company in its respective SC and their respective partners within the same chain. The integration result of the application of the proposed methodology will allow companies able to face the rest of the factors promoting the competitiveness of enterprises.

5. The Case of Study: Application of the Methodology for the Integration of the Supply Chain

The footwear industry in Mexico is considered one of the most important in the artisanal field, with products whose quality is widely recognized. But unfortunately this industry like many others is submerged in a chronic crisis since several years. According to the National Chamber of the Footwear Industry in Mexico [21] the cost of production ranges between 18 and 20 dollars, while in Asian countries varies between 6 or 8 dollars. Although, some Mexican companies have made the effort to get new technology and maintain high levels of international competitiveness, it is virtually impossible to compete against countries like China due to their business strategies and their labor cost.

The case of a handmade shoe company located in Southeastern Mexico is exposed and for which the methodology proposed is applied in this paper in order to prove its effectiveness and allow the company to become more competitive.

The shoe company studied follows artisanal production processes. In this kind of companies, the rate of work depends on the capabilities and skills of workers. In addition, the lack of control over their inventory and delivery times which are key criteria for determining the competitiveness of any company makes almost impossible to get a successful integration of SC. That is why a correct distribution of the task in order to achieve the goal and objectives is fundamental.

The shoe company under study represents only one link in the network of SC in which it is immersed, and to become more competitive it must be integrated to the rest of the links.

5.1. Improve the Internal Behavior. The first step of the methodology is to improve the internal behavior and focusing mainly on the production of shoes, so we will seek to make more efficient this area. This first step is divided into three activities which are detailed below:

Step 1 (simulation of the production process). To better understand the operation of the system under study simulation was used to obtain information on the performance of certain actions and to define strategies to improve these indicators.

Step 2 (solution to the identified problem). After detecting the problem, the best alternative to solve it was looked for.

Step 3 (evaluation of proposed alternative). To prove if what is being proposed really solves the problem and improves the performance of the measure evaluates, the simulation model of the production area is modified. The statistical results obtained are compared. If the goal is not achieved, another solution is proposed and evaluated.

5.1.1. Simulation of the Production Process: Modeling and Analysis System. Discrete event simulation is a technique that allows developing computer models of some desired system, in order to identify unknown characteristics and/or conduct experiments in a defined time frame, respecting some imposed restrictions.

One of the initial conditions to successfully implement a correct SC management is to improve the efficiency of internal operations. The simulation model was focused on the analysis of the department in which according to the owner there are more problems.

The objectives pursued with the development of the simulation model were as follows:

(i) analyze the current production process for the 3 products (shoe models) with greater demand;

(ii) Analyze the productivity indicators for workers in the production process at the current system.

The department studied has 6 operators. The direction sets the goal of producing 250 pairs of shoes by day for any type of product with a transfer batch of 25 pairs of shoes, based on time study conducted by the Chamber of the Footwear Industry of the State of Guanajuato [21]. Operations were assigned according to the skills of each employee and no previous study of line balancing was performed. The main problem of the production system detected was the failure of workers to get the goal imposed during the working time journey, causing unpaid overtime. Another aspect is the existences of bottlenecks that cause the workers to have no work to do. When this happens, they have to support a worker who has a lot of inventory in batch process causing the transfer not to be respected and the bottleneck keeps changing.

A simulation model that represents the system as it currently operates was made. The model takes into account all the operations assigned to 6 workers for the manufacture of 3 products. The resulting simulation model allowed knowing some indicators such as the cycle time, the productivity of each worker, and machine as well as the main bottleneck.

(1) Data Collection and Definition of Current System Performance Parameters. The process flowchart, time study, the name of the workers, their skills, and their assigned tasks within the production process was analyzed to build the simulation model. Also it was necessary to know about the available machinery, the capacity and size of the machinery, and the total size of the production area to develop the layout.

To build the simulation model, the cutting tasks, the processing time, and the workers who are part of the die cutting department were took into account.

The following measures of performance were evaluated: the cycle time, the productivity of workers and machines, the bottlenecks, and the efficiency of line balancing.

(2) Construction and Verification of the Simulation Model. To build the simulation model ProModel, which is a powerful simulator with integrated animation, was used.

The model contains each of the elements of interest in the system, such as machines, pieces, and personnel. Figure 2 shows the layout of the model made.

Once the construction of the model was finished, we proceeded to verify it using the tracking system which ProModel has integrated. It shows each of the events every minute of time simulated, which allows a broad view of what is happening internally.

(3) Verification, Validation, and Experiment Design. The simulation model was validated in order to confirm that truly reflect reality, these phases include the pilot testing, the validation, and the experiment design.

Pilot tests were conducted to obtain preliminary information provided by the simulator. This information will serve for the later stages; in this case 10 pilot runs were performed.

Validation is the comparison of the observed data and the simulated data, in order to prove that the differences are statistically insignificant.

Statistical paired t-test was used to compare the results obtained from the simulation with real data. The time that

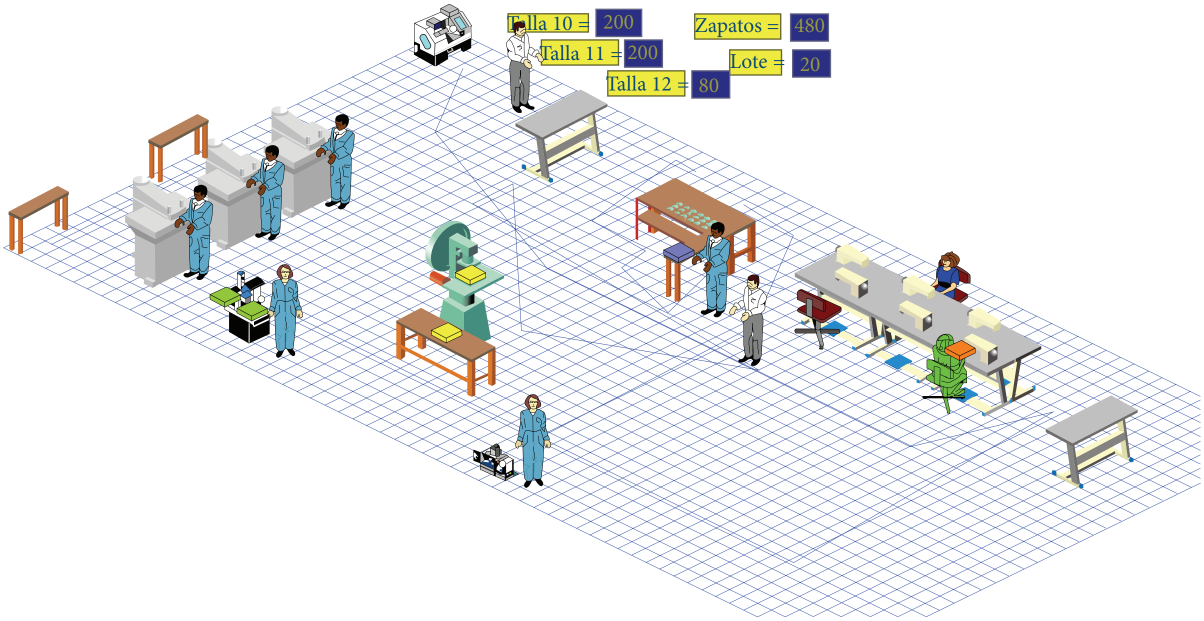


Figure 2: Model layout.

250 pairs of shoes for each of the three products takes to be produced was the performance measure. The results conclude that the difference observed between μ_x (mean of observed data) and μ_y (mean of simulated data) is not statistically significant at a confidence level of 95%, and may be explained by random fluctuations, thereby the model was accepted.

Finally, to determine the optimal number of runs for the simulated model, the method of estimating the mean $\mu = E(x)$ with a specific error is used. In other words, if 100 independent confidence intervals are constructed of 90%, is expected $\overline{X}$ to have an absolute error β of at most 90 cases; 10 intervals around the absolute error would be greater to β.

The time that takes to produce 250 pairs of shoes in minutes, obtained from the 10 independent pilot runs were used and the average manufacturing time in minutes with an absolute error $\beta = 5$ minutes and a confidence level of 99% was estimated. With this procedure, we found that the optimal number of replications was 14 runs.

(4) Analysis of Results. Results after running the model 14 times are shown in Table 1. Table 1 shows the value of the three parameters selected: the total manufacturing time, manufacturing cycle (sum of durations of all the tasks assigned to each worker) and cycle time for each of the three shoes products. Finally the individual productivity and efficiency of the whole production line based on a production rate of 250 pairs of each of the three products is included.

The results show that the actual efficiency of the production line on average is 56%; this represents the percentage of time that the resources of the factory are working, while the remaining 44% is considered idle time. It also states that staff's productivity is uneven. While the worker 6 is occupied 23% of the total simulation time, the worker 2 has nearly an 89% of total time occupied. This means that the workload is poorly distributed and does not justify the hiring of new staff. A new task assignation is required.

Table 1: Results of the simulation model of the current manufacturing process.

Parameter	Mean (minutes)	Standard deviation (minutes)
Total time for manufacturing three products	1252.4	5.9
Product 1		
Manufacturing cycle	591.4	7.2
Cycle time (batch)	22.0	0.2
Product 2		
Manufacturing cycle	469.8	3.0
Cycle time (batch)	21.0	0.2
Product 3		3
Manufacturing cycle	457.0	3.0
Cycle time (batch)	21.0	0.1
Parameter	**Mean (%)**	**Standard deviation (%)**
Workers		
Worker 1	63.3	0.4
Worker 2	88.9	0.1
Worker 3	74.2	0.5
Worker 4	61.2	0.4
Worker 5	46.8	0.8
Worker 6	23.6	0.1
Line efficiency	56.0	

5.1.2. Optimization of Supply Chain. The results of the simulation showed that there is low productivity of both workers and machines. Likewise, we emphasize that the lead time is too high. Because of this situation the perfect balance at the production line is necessary, that is, the sum of the operation time allocated to each operator must be equal to the cycle time and, therefore, the product flows without delays.

(1) Balancing Production Lines. Following the proposed methodology (Section 5.1), we propose the line balancing that consists of assigning tasks to individual operators or workstations so that any performance measure is optimized.

The problem of balancing the manufacturing lines of shoes should considers the following.

(a) Operation times are probabilistic.

(b) The tasks cannot be executed in an arbitrary order given the precedence requirements.

(c) It is desired to minimize the number of workers.

(d) The line is balanced with the available machinery.

(e) The tasks cannot be divided between two or more stations.

(f) The setup times are negligible.

(g) Each station processes only one task at a time.

(h) All tasks must be processed once.

(i) Transportation time between a work stations is not considered

(j) The tasks run only once for a season (considered a serial line).

(k) Some workers cannot perform certain tasks.

The variables used to construct the model of line balancing are as follows:

k = Total number of workstations

n = Total number of manufacturing processes

c = Cycle time

t_i = Manufacturing time operation j, $j = 1, \ldots, n$

I = Idle time of the production system

$$X_{ij} = \begin{cases} 1, & \text{If the process } j \text{ is assigned to workstation } i,\ i = 1, \ldots, k \\ 0, & \text{otherwise.} \end{cases}$$

The objective function is to minimize the idle time given by (1)

$$I = kc - \sum_{j=1}^{n} t_j. \tag{1}$$

This function is subject to the following restrictions:

$$\sum_{j=1}^{n} X_{ij} t_j \leq c, \quad i = 1, 2, \ldots, k, \tag{2}$$

$$\sum_{i=1}^{k} X_{ij} = 1, \quad j = 1, 2, \ldots, n, \tag{3}$$

$$X_{hv} \leq \sum_{i=1}^{h} X_{iu}, \quad h = 1, 2, \ldots, k;\ (u, v) \in R, \tag{4}$$

where the total time of all processes assigned to each workstation must not exceed the cycle time (2), a task can only be assigned to a workstation, (3) and precedence relationships exist (4). The precedence relationships were modeled as published by Ragsdale and Brown [22].

Most of the operations for the manufacture of shoes in this company are manual; because of this, the operating time of some activity varies from one shoe to another so each time operation follows a probabilistic distribution at the mathematical model. Each operation must be assigned to only one worker; this situation is represented by a binary variable with value 1 when the process j is assigned to workstation i and 0 otherwise. We proposed to optimize the lines balancing problem by a simple genetic algorithms.

(1) Solution of the Mathematical Model

(i) Optimization of the Mathematical Model through a Genetic Algorithm (GA). The objective function was minimizing the idle time. As long as the idle time decreases the efficiency of the line increases. The efficiency of the line is considered as percentage, thus the optimal value is 100%.

The basic structure of classical Assignment Problem (AP) is the minimization of an objective function involving cost or time. It is a well-studied problem in the optimization literature due to its many applications such as facility location, personnel scheduling, task assignment, and so forth. It is formulated as an 0-1 optimization problem. However, many real-life problems consider constraint such as budgetary limitations, time restrictions, or personal abilities, and so forth, that affect the assignment decisions. Due to this fact, finding a solution becomes more complex. Many authors solve this type of problems by genetic algorithm with excellent results [23, 24].

The genetic algorithm is a useful adaptive method to solve search and optimization problems. Based on the theory of Darwinian evolution, it is a robust technique that can solve problems in various areas including those in which the application of other methods is difficult. While these algorithms do not guarantee an optimal solution to the problem is obtained, empirical evidence shows that it can find solutions to an acceptable level in a competitive time compared to other combinatorial optimization algorithms. The assignment problem for the case of the proposed study is a clear consequence of the same situations. The Simple Genetic Algorithm used, also called Canonical, was encoded based on the binary variable that assign the task to its respective workstation. Genetic Algorithm (GA) includes three fundamental genetic operations: selection, crossover, and mutation. The first step in any GA is to generate an initial population with a group of individuals randomly created. The individuals in the population are then evaluated and assigned a fitness value. The evaluation function is provided by the operator and gives the individuals a score based on how well they perform at the given task (fitness value). Two individuals are then selected based on their fitness; the higher the fitness, the higher the chance of being selected.

After selection has been carried out, crossover is applied to randomly paired individuals. The recombined individuals create one or more off-spring. This can be viewed as creating the new population. The randomly mutation of the offspring is then applied. In terms of GAs, mutation means a random change of the value or a gene in the population. After

Table 2: Comparison of the results obtained after line balancing according to the modification of the simulation model.

Parameter	Unbalanced system		Balanced System	
	Mean (minutes)	Standard deviation (minutes)	Mean (minutes)	Standard deviation (minutes)
Total manufacturing time (three products)	1252.4	5.9	1121.1	7.2
Product 1				
Manufacturing cycle	591.4	7.2	550.3	4.2
Cycle time (batch)	22.0	0.2	20.1	0.1
Product 2				
Manufacturing cycle	469.8	3.0	417.5	2.4
Cycle time (batch)	21.0	0.2	19.8	0.1
Product 3				
Manufacturing cycle	457.0	3.0	392.0	1.8
Cycle time (batch)	21.0	0.1	19.0	0.2
Parameter	Mean (%)	Standard deviation (%)	Mean (%)	Standard deviation (%)
Workers				
Worker 1	63.3	0.4	81.5	0.5
Worker 2	88.9	0.1	66.6	0.5
Worker 3	74.2	0.5	67.9	0.6
Worker 4	61.2	0.4	57.1	0.4
Worker 5	46.8	0.8	66.6	0.5
Worker 6	23.6	0.1	64.5	0.5
Line efficiency	56.0		73.6	

the process of selection, recombination and mutation, the next population can be evaluated. The process continues until a suitable solution has been found or when a given number of generations have been reached.

GA parameters used are population size 200 individuals, 400 generations (stopping criterion), 40% crossover rate, and mutation rate of 30%. A new allocation of tasks for each of the three products (different models of shoes) separately was generated.

(2) Evaluation of Proposed Alternative. With the new tasks configuration for each products is necessary to make changes to the simulation model to verify that this configuration reduces manufacturing time and achieves a proper production line balancing between existing workstations. This corresponds to Step 3 (improvement rate) of the proposed methodology.

The results show an increase in the efficiency of the line of 17.5%, Table 2 shows a comparison of the current system and the system after balancing it.

To determine whether this reduction in manufacturing time is significant, a statistical test between means of two small samples was completed. It was concluded with 95% confidence that the proposed improvements will cause a reduction in the cycle time was performed.

5.2. The Demand Forecast. Step 2 of the methodology is the demand forecasting. To forecast the demand and calculate the monthly cycle time, historical record of the sale was used.

To determine the type of forecast to use it was necessary to plot the demand data. As shown in Figure 3, the behavior of the demand for each pattern is irregular, therefore has no tendency, or seasonality. To test this assumption the nonparametric statistical test of Daniels was performed to detect the seasonality of a series. This statistical test does not require the assumption of normality in the population. It was concluded to a 95% confidence that the monthly demand for shoes has not trend.

Once it was found that a model without trend is the most appropriate, the next step was to select the appropriate procedure to fit the data to the model. Methods of exponential smoothing, moving averages and updated mean were applied to forecast the demand for each of the selected shoe's models. For each product the best forecasting model was selected based on the performance of the three measures of statistical adequacy (median absolute deviation, MAD; root mean square error, RMSE; and mean absolute percentage error, MAPE).

5.2.1. Adjusting the Forecast by Fuzzy Logic. Deterministic forecasting methods have as a principle that every event is determined by the chain of cause and consequence, regardless of random events.

In the case study, to determine the demand it was convenient to use a fuzzy logic model, since it allows making decisions based on uncertain data or subjective knowledge, which are usually presented through linguistic variables. In this research, the fuzzy logic model proposed by Escobar

Month	Product 1	Product 2	Product 3
February	5696	4386	2807
March	8462	5416	4387
April	5971	4837	3482
May	9400	5640	4343
June	8206	6565	3939
July	5771	4155	3324

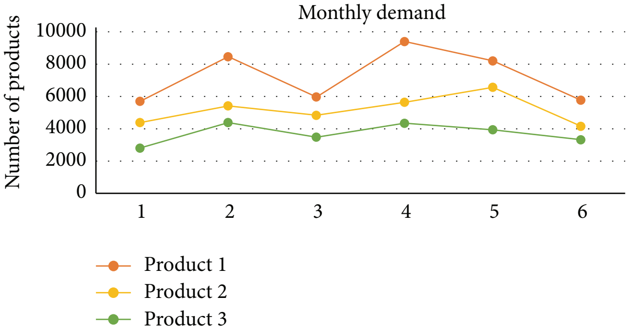


Figure 3: Graph of the monthly demand.

Gómez [25] was applied. This is still part of Step 2 of the methodology.

(1) Linguistic Variables and Fuzzy Sets. Fuzzy logic is an artificial intelligence technique that allows modeling situations whose variables are measured according to a subjective scale, or that depend on knowledge or opinion of experts in the subject concerned and also arise under uncertain conditions. The process involves identifying some variables, defined as input, which directly influences another variable, identified as output and their respective levels of response. The result will be a model that is able to translate subjective information on reliable and easily quantifiable results.

This model considered three factors which have influence demand according to expert knowledge: season, perception, and offer. These factors are described below.

Season as linguistic variable is the expression that refers to the behavior of demand in a given period compared to the same period of previous years. Three output cases were considered representing three fuzzy sets for this variable: low, medium and high season.

The membership functions for each of the fuzzy sets are shown in (5) as follows:

$$\mu_{\text{Low}}(\text{Season}) = \begin{cases} 1; & W \leq A, \\ 1 - \dfrac{B-W}{B-A}; & A < W \leq B, \\ 0; & B \leq W, \end{cases}$$

$$\mu_{\text{Medium}}(\text{Season}) = \begin{cases} 0; & W \leq A, \\ 1 - \dfrac{B-W}{B-A}; & A < W \leq B, \\ 1 - \dfrac{W-B}{C-B}; & B < W < C, \\ 0; & C \leq W, \end{cases} \tag{5}$$

$$\mu_{\text{High}}(\text{Season}) = \begin{cases} 1; & W \leq B, \\ 1 - \dfrac{C-W}{C-B}; & B < W \leq C, \\ 1; & C \leq W. \end{cases}$$

Perception as a linguistic variable is defined as the feeling of satisfaction that the customer has because the product met their expectations and needs. Fuzzy sets that are part of this variable are three: poor, fair, and good perception. Membership functions are shown in (6) as follows:

$$\mu_{\text{Poor}}(\text{Perception}) = \begin{cases} 1; & X \leq A', \\ 1 - \dfrac{X-A'}{B'-A'}; & A' < X < B', \\ 0; & B' \leq X, \end{cases}$$

$$\mu_{\text{Fair}}(\text{Perception}) = \begin{cases} 1; & X \leq A', \\ 1 - \dfrac{B'-X}{B'-A'}; & A' < X < B', \\ 1; & B' \leq X \leq C', \\ 1 - \dfrac{X-C'}{D'-C'}; & C' < X < D', \\ 0; & D' \leq X, \end{cases} \tag{6}$$

$$\mu_{\text{Good}}(\text{Perception}) = \begin{cases} 0; & X \leq C', \\ 1 - \dfrac{D'-X}{D'-C'}; & C' < X < D', \\ 1; & D' \leq X. \end{cases}$$

Offer as linguistic variable is defined as the level of competition that exists for the product analyzed at the market. For this variable three output alternatives were defined as fuzzy sets and are as follows: the level of competition is little, medium or large. The membership functions are presented in (7) as follows:

$$\mu_{\text{Little}}(\text{Offer}) = \begin{cases} 1; & Y \leq A'', \\ 1 - \dfrac{Y-A''}{B''-A''}; & A'' < Y \leq B'', \\ 0; & C'' \leq Y, \end{cases}$$

$$\mu_{\text{Medium}}(\text{Offer}) = \begin{cases} 0; & Y \leq A'', \\ 1 - \dfrac{B''-Y}{B''-A''}; & A'' < Y \leq B'', \\ 1 - \dfrac{Y-B''}{C''-B''}; & B'' < Y < C'', \\ 0; & C'' \leq Y, \end{cases} \tag{7}$$

$$\mu_{\text{Large}}(\text{Offer}) = \begin{cases} 0; & Y \leq B'', \\ 1 - \dfrac{C''-Y}{C''-B''}; & B'' < Y \leq C'', \\ 1; & C'' \leq Y. \end{cases}$$

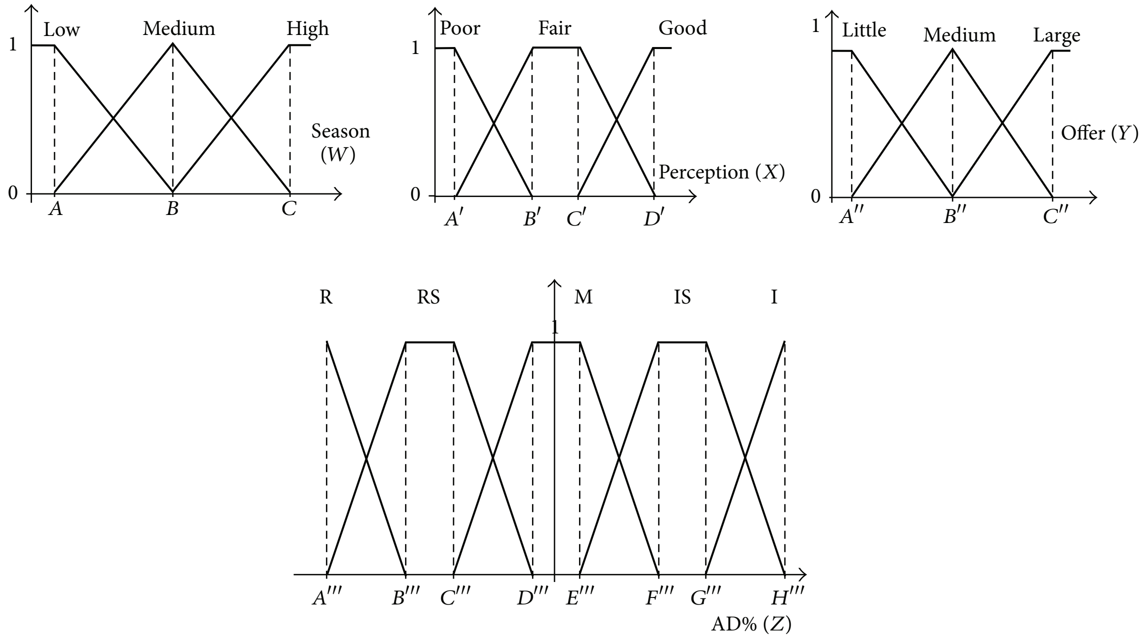


Figure 4: Graphical representation of the fuzzy sets of the linguistic input variables and output AD.

Table 3: Fuzzy rules base.

					Perception				
		Poor			Fair			Good	
		Offer			Offer			Offer	
	Little	Medium	Large	Little	Medium	Large	Little	Medium	Large
Season									
Low	R	R	R	RS	R	R	M	M	RS
Medium	M	RS	R	IS	M	RS	I	IS	M
High	M	RS	R	IS	IS	M	I	I	IS

Therefore, the output variable of the model is *Adjusted Demand* and is defined as the change in the level of the projection of the aggregate demand percentage. Under this consideration five cases are presented, which allow to define the fuzzy sets belonging to this: reduce, reduce slightly, maintain, increase, and increase slightly (8). Consider

$$\mu_{\text{Reduce}}\,(\text{Adjusted demand}) = \begin{cases} 0; & Z < A''', \\ 1 - \dfrac{Z - A'''}{B''' - A'''}; & A''' \leq Z < B''', \\ 0; & B''' \leq Z. \end{cases} \quad (8)$$

Both the input variables and the output variable, in this image, are presented in Figure 4 which shows the relationships described in the membership functions ((5)–(8)) where the different scenarios that can occur for each variable are clear, and allows modeling these variables by geometric figures.

(2) Fuzzy Rules. Given the input linguistic variables and their fuzzy sets (3 variables with three sets each) a base of 27 fuzzy rules was obtained. In this set of rules the knowledge of the experts consulted on decision-making related to the adjustment of the forecast demand is summarized. The summary of the rules used for the inference process is presented in Table 3.

Where D means high projected demand reduction, DP means low projected demand reduction, M maintains the projected demand, IP means low projected demand increase, and finally I means high projected demand.

The defuzzification process was the next step. This process consists in obtaining the values through fuzzy membership functions of the values assigned to the input variable. In this model the degree of membership in fuzzy sets was obtained by evaluating the values assigned to linguistic variables using the membership functions. If, after applying the fuzzy equations the value obtained for a given set is greater than zero, it is considered to belong to the set analyzed with a degree equal to the value found. Otherwise, it is considered that the degree of belonging to the set is zero.

The fuzzy logic model applied to the demand of each of the products is shown in Tables 4, 5, and 6. This modification caused the forecasting model fit better to actual demand.

Table 4: Adjusted demand product 1.

Month	Input variables			Output variable		
	Season	Perception	Offer	% Fit	Adjusted demand $\hat{y}_t$	e_t
May	2.8	8.2	2.5	6.62	7153.85	2246.15
June	2	7	2	0	7944.33	261.67
July	1.5	7	2	−12.95	6841.26	−1070.26
August	1.5	4.5	2	−14.14	6690.50	?
		Product 1			MAD	1192.69
					RMSE	1444.43
					MAPE	15.21%

Table 5: Adjusted demand product 2.

Month	Input variables			Output variable		
	Season	Perception	Offer	% Fit	Adjusted Demand $\hat{y}_t$	e_t
May	2.8	8.2	2.5	6.62	5202.38	−859.58
June	2	7	2	0	5069.53	−1130.65
July	1.5	7	2	−12.95	4673.35	−1349.25
August	1.5	4.5	2	−14.14	4435.82	?
		Product 2			MAD	218.61
					RMSE	1131.04
					MAPE	13.31%

By having a reliable forecast of the demand for the three products, it was possible to best determine the daily quantity of products to manufacture and the cycle time. Finally this information represents the input data to the simulator. The results are shown in Table 7.

5.3. Synchronizing the Quantity to Order of Raw Material with Demand. The third step of the methodology was to synchronize the order quantity of raw material with the estimated demand. Figure 5 illustrates the relationship between the variables involved in the establishment of the order points and the safety stock. The main relationship is shown in (9) as follows:

$$\mathrm{OP} = \mathrm{EDDLT} + \mathrm{SS} \tag{9}$$

5.3.1. Fuzzy Model to Determine the Safety Stock. There are several methodologies to determine the safety stock. These techniques assume that the demand follows either a normal probability distribution or a discrete probability distribution. A fuzzy logic model allowed finding the safety stock taking into account the uncertainty.

The model included the following assumptions.

(i) The assumption of normality of daily demand was dismissed.

(ii) There are no contracts between the supplier and the company therefore there is no planning orders.

(iii) Sometimes it is difficult to find another supplier who can supply the product because either there are no competitors or the product's specifications do not permit it.

(iv) There is no feedback of information between customer, company, and suppliers.

(1) Linguistic Variables and Fuzzy Sets. As in the case of the product demand, the amount of raw material that should be considered as safety stock was estimated by a fuzzy logic model considering the following factors: Delivery time, Quality and Shortage. *Delivery time* as the linguistic variable is defined as the fulfilling of the delivery date of the raw material by the supplier. Fuzzy sets that are part of this variable are three, which are: high, medium and low punctuality.

Quality is the next factor and as a linguistic variable is the term that refers to compliance with the specifications of raw materials stocked by the supplier. This variable considers three cases, and each of these cases defines a fuzzy set, which is good, fair, and poor quality.

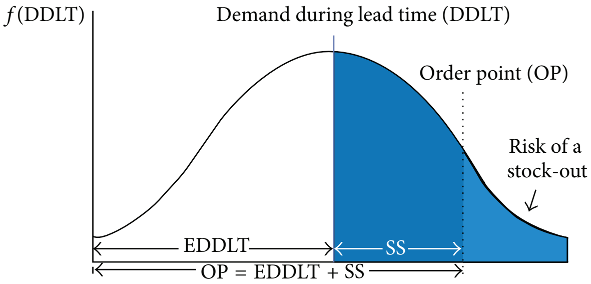
The last factor to consider is the *Shortage* of raw materials. The shortage does not refer to the lack of material at the warehouse; it refers to the shortage of materials on the market in general. For these variable, three fuzzy sets are assumed: low, medium and serve shortage.

The output variable of the model represents an adjustment factor that will be multiplied by the demand during the lead time (EDDLT) to determine safety stock. This variable is named *raw material*. Under this consideration five cases are presented, which allow to define the fuzzy sets belonging to this variable: not critical, uncertain, critical, uncertain-critical, and supercritical. Figure 6 shows the sets of input variables and the output model for safety stock.

The summary of the rules used for the inference process is presented in Table 8.

Table 6: Adjusted demand product 3.

Month		Input variable			Output variable	
	Season	Perception	Offer	% Fit	Adjusted demand $\hat{y}_t$	e_t
May	2.8	8.2	2.5	6.62	4194.96	147.84
June	2	7	2	0	3912.54	26.34
July	1.5	7	2	−12.95	3604.60	−280.51
August	1.5	4.5	2	−14.14	3118.00	?
					MAD	121.51
		Product 3			RMSE	183.70
					MAPE	8.10%



EDDLT: expected demand during lead time
SS: safety stock

Figure 5: Relationship between DDLT, EDDLT, SS, OP and the probability of shortage of each cycle reorder.

Table 7: Simulation results according to the monthly demand for all three products.

Month	Adjusted demand for the three products	Daily amount of pair of shoes produced by each type of product			Line efficiency
		1	2	3	
May	7154	265	208	168	85%
June	7944	294	203	157	95%
July	6841	253	187	144	81%
August	6691	248	178	125	79%

Where NC means not critical, INC means Unclear-not critical, C means Critical, IC means critical uncertain, and SC means super critical.

The fuzzy model will provide a percentage factor (variable output) that will be multiplied by the EDDLT to calculate the safety stock.

5.4. Determine the Delivery Time: Fuzzy Model for the Product Delivery Time. The products arrive to end consumers in two ways: to sell directly the products to shoe stores established in the region and to get their products through intermediaries, that is, their products are collected by a third company that store the products at central storehouses and distributes them to the point of final sale.

The model includes the following assumptions.

(i) There are no contracts between the supplier and the company; therefore there is no planning orders.

(ii) There is no feedback of information between customer, company, and supplier.

(iii) The company is responsible for bringing products to point of sale.

5.4.1. Linguistic Variables and Fuzzy Sets. The two factors create uncertainty in the estimated delivery time: *Ease of manufacturing* product and the *location* of the point of sale. *Ease of manufacturing* as linguistic variable is defined as a facility, according to experience, that company has to satisfy the quantity of product requested by the customer. Fuzzy sets that are part of this variable are three: very possible, possible, and hardly possible. The *location* is the second factor and as a linguistic variable represent the distance between the factory and the place from which the order was placed. The variable considers two cases are near and far. The output variable of the model is an adjustment factor that multiplies the delivery time thrown by internal planning. This variable is named *Fit*. Under this consideration the 3 cases are presented, which allow to define the fuzzy sets belonging to this variable: the adjustment is null, slight, or large. Figure 7 shows the two input variables and the output of this model.

Finally Table 9 summarizing the rules used for the inference process.

The input data (fuzzy variables) were introduced to the fuzzy model and as results the model provided a percentage factor (variable output). This factor will be multiplied by the initial delivery time in order to give an accurate delivery date to the client. With these results the company is able to offer a delivery date where the breach of agreed orders is reduced.

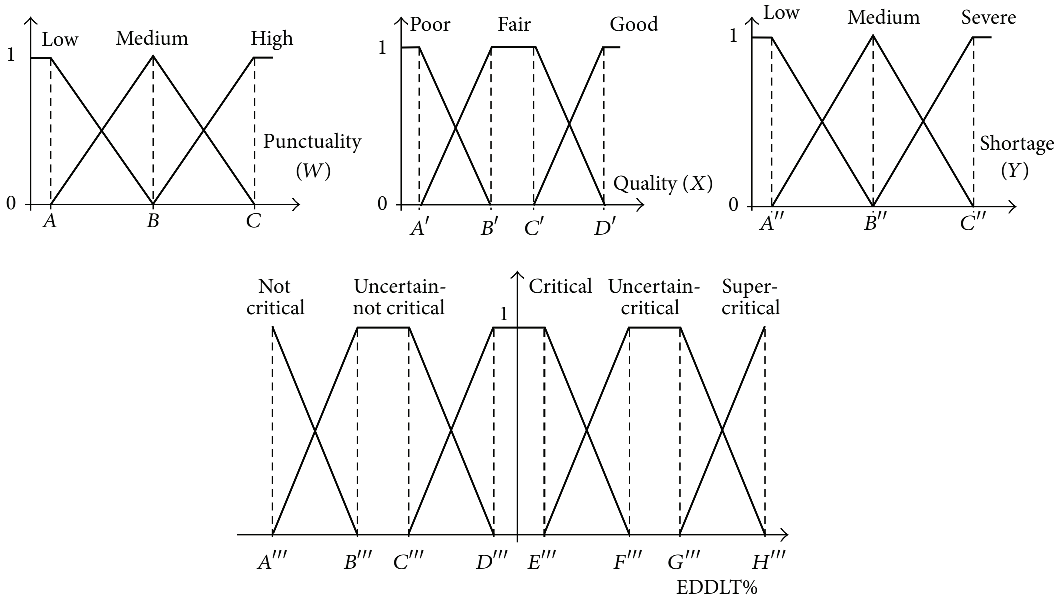


Figure 6: Fuzzy sets variable raw material.

Table 8: Fuzzy rules base.

	Quality								
	Poor			Fair			Good		
	Shortage			Shortage			Shortage		
	Low	Medium	Severe	Low	Medium	Severe	Low	Medium	Severe
Punctuality									
Low	C	SC	SC	C	UC	UC	UNC	UNC	C
Medium	C	C	UC	C	C	UC	UNC	UNC	C
High	UNC	C	UC	UNC	C	UC	NC	NC	UNC

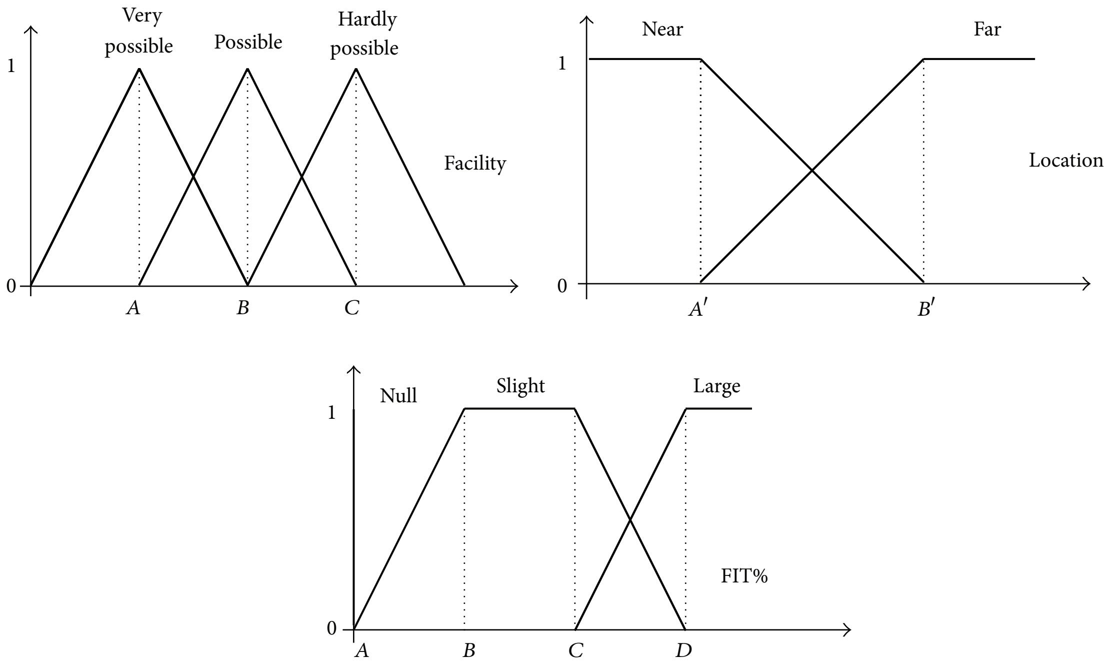


Figure 7: Fuzzy set of the output variable Fit.

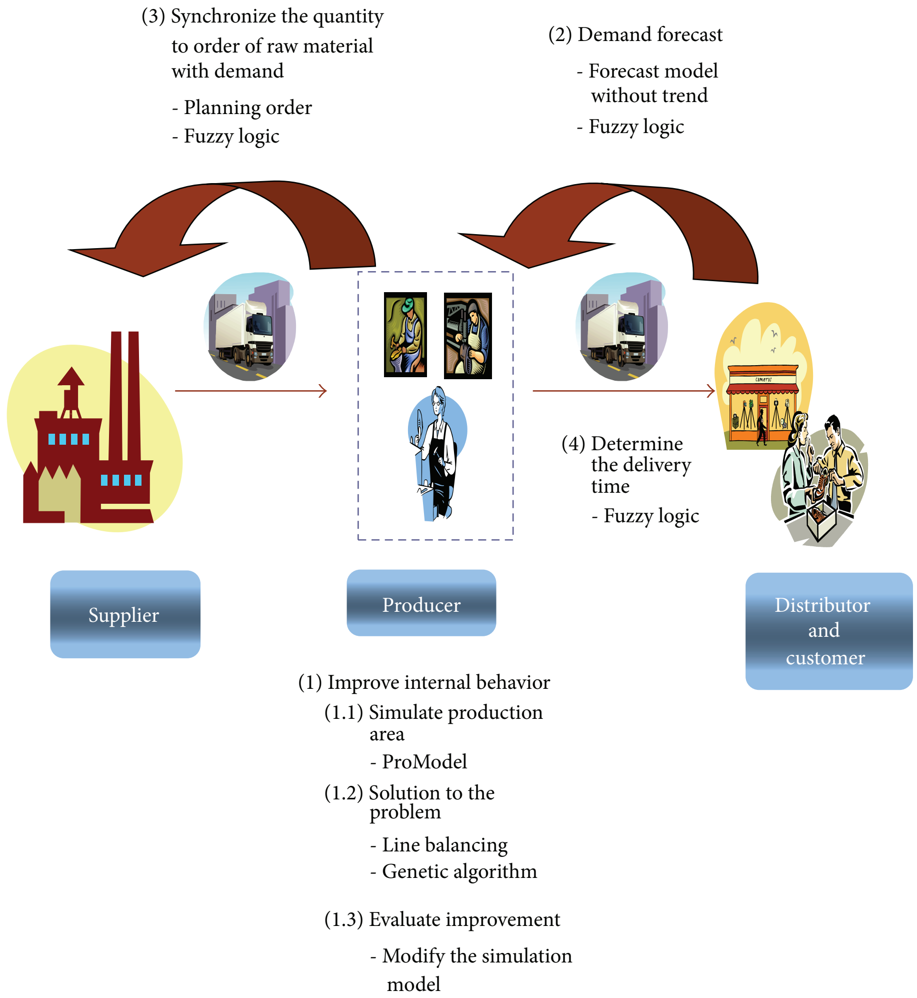

(3) Synchronize the quantity to order of raw material with demand
- Planning order
- Fuzzy logic
(2) Demand forecast
- Forecast model without trend
- Fuzzy logic
(4) Determine the delivery time
- Fuzzy logic
Supplier
Producer
Distributor and customer
(1) Improve internal behavior
(1.1) Simulate production area
- ProModel
(1.2) Solution to the problem
- Line balancing
- Genetic algorithm
(1.3) Evaluate improvement
- Modify the simulation model

Figure 8: Methodology for the integration of the supply chain and techniques used in each step.

Table 9: Fuzzy rules base.

	Location	
	Near	Far
Easy of manufacturig		
Very possible	Null	Slight
Possible	Slight	Large
Hardly possible	Slight	Large

A greater confidence between the producer and client will be established.

6. Conclusions

The SC integration is a major issue today, as it allows companies to position themselves in a highly competitive market. The main contribution of this research is to propose a sequence of logical steps to achieve this integration regardless of the nature of the company, especially when talking about companies that have manual processes.

As an advantage the proposed methodology permit to integrate the traditional engineering techniques with new developed tools as soft computing to overcome the problems that prevent the successful integration of the members of the SC adapting the use of them depending on the features and restrictions of each company without condition the use of any particular tool. This integration does not detract from the traditional methods of problem solving but complement each other. As already mentioned, the resulting models can be taken up to other companies. The application of this methodology in a company in the footwear industry shows its efficiency when it is applied to a business process which is considered artisanal and affected by an uncertain market. This approach helped the company to synchronize the SC to which it belongs. Figure 8 summarizes the methodology for the SC integration more broadly; in this scheme the techniques used for each of the steps in the methodology samples are mentioned.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] S. A. Ghasimi, R. Ramli, and N. Saibani, "A genetic algorithm for optimizing defective goods supply chain costs using JIT logistics and each-cycle lengths," *Applied Mathematical Modelling*, vol. 38, no. 4, pp. 1534–1547, 2014.

[2] T. Efendigil and S. Önüt, "An integration methodology based on fuzzy inference systems and neural approaches for multi-stage supply-chains," *Computers and Industrial Engineering*, vol. 62, no. 2, pp. 554–569, 2012.

[3] L. L. Hon and W. P. Q. Ng, "A process integration approach for supply chain development," in *Handbook of Process Integration (PI) Minimisation of Energy and Water Use, Waste and Emissions*, Woodhead Publishing Series in Energy, pp. 571–593, 2013.

[4] INEGI, National Institute of Statistics, Geography and Informatics, 2013, http://www.inegi.org.mx.

[5] M. Fernando and C. F. Elisa, "Cómo administrar la cadena de suministro," 2008, http://www.cnnexpansion.com/manufactura/actualidad/2008/03/26/como-administrar-la-cadena-de-suministro.

[6] M. Paul, Supply Chain Analytics: What is it and Why is it so Important?, 2013, http://www.industryweek.com/blog/supply-chain-analytics-what-it-and-why-it-so-important.

[7] M. L. Lengnick-Hall, C. A. Lengnick-Hall, and C. M. Rigsbee, "Strategic human resource management and supply chain orientation," *Human Resource Management Review*, vol. 23, no. 4, pp. 366–377, 2013.

[8] J. A. Palma-Mendoza, N. Kevin, and R. Rajat, "Business process re-design methodology to support supply chain integration," *International Journal of Information Management*, vol. 34, no. 2, pp. 167–176, 2014.

[9] A. A. Elimam and B. Dodin, "Project scheduling in optimizing integrated supply chain operations," *European Journal of Operational Research*, vol. 224, no. 3, pp. 530–541, 2013.

[10] H. Jafar, "Lead time variation control using reliable shipment equipment: an incentive scheme for supply chain coordination," *Transportation Research Part E: Logistics and Transportation Review*, vol. 63, pp. 44–58, 2014.

[11] S. Mehdi, "An integrated multi-objective model for allocating the limited sources in a multiple multi-stage lean supply chain," *Economic Modelling*, vol. 37, pp. 224–237, 2014.

[12] A. Zhang, H. Luo, and G. Q. Huang, "A bi-objective model for supply chain design of dispersed manufacturing in China," *International Journal of Production Economics*, vol. 146, no. 1, pp. 48–58, 2013.

[13] M. Khan, M. Y. Jaber, and A. Ahmad, "An integrated supply chain model with errors in quality inspection and learning in production," *Omega*, vol. 42, no. 1, pp. 16–24, 2014.

[14] T. Lin, F. Rodríguez Luis, N. Shastri Yogendra, C. Hansen Alan, and K. C. Ting, "Integrated strategic and tactical biomassbiofuel supply chain optimization," *Bioresource Technology*, vol. 156, pp. 256–266, 2014.

[15] Z. H. Zhang, B. F. Li, X. Qian, and L. N. Cai, "An integrated supply chain network design problem for bidirectional flows," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4298–4308, 2014.

[16] Ö. Eren, P. Turan, and B. Tolga, "Modeling and optimizing the integrated problem of closed-loop supply chain network design and disassembly line balancing," *Transportation Research Part E: Logistics and Transportation Review*, vol. 61, pp. 142–164, January 2014.

[17] B. Sarkar and A. Majumder, "Integrated vendor-buyer supply chain model with vendor's setup cost reduction," *Applied Mathematics and Computation*, vol. 224, pp. 362–371, 2013.

[18] D. Song, J. Dong, and J. Xu, "Integrated inventory management and supplier base reduction in a supply chain with multiple uncertainties," *European Journal of Operational Research*, vol. 232, no. 3, pp. 522–536, 2014.

[19] J. Telhada, "Alternative MIP formulations for an integrated shift scheduling and task assignment problem," *Discrete Applied Mathematics*, vol. 164, part 1, pp. 328–343, 2014.

[20] D. Z. Zhang, "An integrated production and inventory model for a whole manufacturing supply chain involving reverse logistics with finite horizon period," *Omega*, vol. 41, no. 3, pp. 598–620, 2012.

[21] CANAICAL, "Chamber of the Footwear Industry of the State of Guanajuato," 2014, http://canaical.org/.

[22] C. T. Ragsdale and E. C. Brown, "On modeling line balancing problems in spreadsheets," *INFORMS Transactions on Education*, vol. 4, no. 2, 2004.

[23] J. Majumdar and A. K. Bhunia, "An alternative approach for unbalanced assignment problem via genetic algorithm," *Applied Mathematics and Computation*, vol. 218, no. 12, pp. 6934–6941, 2012.

[24] Ö. Mutlu, O. Polat, and A. A. Supciller, "An iterative genetic algorithm for the assembly line worker assignment and balancing problem of type-II," *Computers and Operations Research*, vol. 40, no. 1, pp. 418–426, 2013.

[25] N. E. Escobar Gómez, J. J. Díaz Núñez, and L. F. Taracena Sanz, "Model for adjustment of aggregate forecasts using fuzzy logic," *Ingeniería Investigación y Tecnología*, vol. 11, no. 3, pp. 289–302, 2010.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 746319, 18 pages
http://dx.doi.org/10.1155/2014/746319

Research Article

A Hybrid Metaheuristic-Based Approach for the Aerodynamic Optimization of Small Hybrid Wind Turbine Rotors

José F. Herbert-Acero,[1] Jaime Martínez-Lauranchet,[1] Oliver Probst,[1] Santos Méndez-Díaz,[2] Krystel K. Castillo-Villar,[3] Manuel Valenzuela-Rendón,[4] and Pierre-Elouan Réthoré[5]

[1] *Chair for Wind Energy, Department of Physics, Instituto Tecnológico y de Estudios Superiores de Monterrey, Eugenio Garza Sada 2501 Sur, 64849 Monterrey, NL, Mexico*
[2] *School of Mechanical and Electrical Engineering, Universidad Autónoma de Nuevo León, Avenida Universidad s/n. Ciudad Universitaria, 66451 San Nicolás de los Garza, NL, Mexico*
[3] *Department of Mechanical Engineering, The University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249, USA*
[4] *Chair for Evolutionary Computation, Department of Computer Science, Instituto Tecnológico y de Estudios Superiores de Monterrey, Eugenio Garza Sada 2501 Sur, 64849 Monterrey, NL, Mexico*
[5] *Wind Energy Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, Frederiksborgvej 399, 4000 Roskilde, Denmark*

Correspondence should be addressed to José F. Herbert-Acero; jf.herbert.phd.mty@itesm.mx

Received 16 April 2014; Accepted 10 July 2014; Published 28 August 2014

Academic Editor: Yang Xu

This work presents a novel framework for the aerodynamic design and optimization of blades for small horizontal axis wind turbines (WT). The framework is based on a state-of-the-art blade element momentum model, which is complemented with the XFOIL 6.96 software in order to provide an estimate of the sectional blade aerodynamics. The framework considers an innovative nested-hybrid solution procedure based on two metaheuristics, the virtual gene genetic algorithm and the simulated annealing algorithm, to provide a near-optimal solution to the problem. The objective of the study is to maximize the aerodynamic efficiency of small WT (SWT) rotors for a wide range of operational conditions. The design variables are (1) the airfoil shape at the different blade span positions and the radial variation of the geometrical variables of (2) chord length, (3) twist angle, and (4) thickness along the blade span. A wind tunnel validation study of optimized rotors based on the NACA 4-digit airfoil series is presented. Based on the experimental data, improvements in terms of the aerodynamic efficiency, the cut-in wind speed, and the amount of material used during the manufacturing process were achieved. Recommendations for the aerodynamic design of SWT rotors are provided based on field experience.

1. Introduction

There has been an increased necessity for the development of small grid-independent energy applications [1] (e.g., domestic, street lighting, and rural electrification) in which wind energy plays an important role. However, the integration of small wind energy technologies is challenging, mostly because of the low average wind speed experienced at the locations where small wind turbines (SWT) are commonly installed [2]. In addition, SWT show reduced aerodynamic efficiencies as compared to large-scale wind turbines (WT) [3] given that they generally operate at low Reynolds numbers and under higher turbulent conditions. SWT designers and manufacturers have shown considerable interest in improving the overall efficiency of their prototypes in order to increase their competitiveness against other small-scale renewable energy technologies. Hence, in order to address this need, improved SWT aerodynamic, acoustic, structural, mechanical, and electrical technologies are required.

Several contributions of variable complexity have been reported in the literature with regard to the design and optimization of WT blades [3–13]. Even though the problem has been extensively studied, no general theories [10] or general conclusions for the optimal design of WT blades have been proposed. In addition, only few works have made major contributions to the technoeconomical modeling and optimization of WT blades while considering all-encompassing schemes [5–8, 10]. The classical blade design methodology adopted in such works consists of three steps: (1) the definition of the blade performance metric (e.g., the aerodynamic efficiency, the cost of energy, etc.) and the objective function to be extremized, (2) the definition of the decision variables and their respective bounds, and (3) the choice of the optimization strategy. Most works determined the blade's aerodynamic efficiency through the use of a blade element momentum (BEM) model, considered complex and highly constrained decision variables, and considered a population-based metaheuristic (e.g., the simple genetic algorithm (GA)) or a calculus-based procedure (e.g., sequential quadratic programming) to provide a solution to their optimization problem. The common pitfall of such works is that the adopted solution procedures (e.g., calculus-based optimization) are likely to reach local maxima due to the complexity of their objective functions. In this regard, the use of state-of-the-art mathematical tools of soft computing may provide a better solution to this rather complex problem. Moreover, the majority of the works have evaluated only a few scenarios and have failed in providing a comprehensive description of their optimized results and, therefore, do not provide appropriate guidelines and/or recommendations to WT blade designers.

This research contributes to the objective of improving SWT aerodynamic technologies by developing a pragmatic framework for the aerodynamic design and optimization of SWT blades and by conducting a validation study, which in turn allowed identifying key features during the SWT blade design. The proposed framework can be considered as a constituent part of a more general, holistic, and multidisciplinary framework known as the wind turbine blade optimization (WTBO) problem. The WTBO problem consists of the design and manufacture of WT blades that exhibit high aerodynamic efficiency, low noise generation, and proper structural design and manufacture to withstand the dynamic loads generated by the expected wind field at the minimum cost and while considering a control/regulation system [11] (e.g., active stall or furling systems [14], among others). The WTBO problem has been applied in the design of blades for medium and large-scale pitch-regulated WT; however, it has been scarcely applied in the design of blades for small-scale stall-regulated or furling-regulated WT [3, 4, 10], hence denoting an important subfield of study that requires more attention.

The starting point in the design and optimization of SWT rotors are the existing guidelines and international standards which give a clear overview of the important factors and safety regulations that affect the decision making of the WT designer. For SWT, requirements may be altered if the SWT safety is not compromised as stated by the International Electrotechnical Commission (IEC) standards [15], which is often the case given the highly constrained scenarios where SWT operate. In this regard, SWT must be as flexible as possible in order meet the energy demand; therefore, any other relevant secondary objectives, such as noise generation, may be relaxed in order to maximize the cost-effectiveness of SWT. This work is focused on the study of the aerodynamic behavior of optimized SWT rotors. Secondary objectives related to the minimization of acoustic, structural, mechanical, and electrical issues were not explicitly considered in the proposed framework with the aim of studying the key geometrical factors influencing the aerodynamic efficiency of SWT rotors.

The remaining of this work is structured as follows. Section 2 presents the theoretical foundations and the development of the SWT blade optimization framework. Section 3 describes the development of an optimized SWT prototype which is used to perform a validation study of a set of optimized rotors. Section 4 discusses the results. Finally, Section 5 presents the overall conclusions and outlines future research.

2. Theoretical Development

2.1. Two-Dimensional Steady-State Airfoil Aerodynamics. A large range of studies have focused on determining the experimental aerodynamics of two-dimensional airfoils [16–20], which are the base for the design of WT rotors. However, the characterizations are commonly performed for a limited set of angles-of-attack (α) and medium-to-large Reynolds numbers (Re). In addition, it is unusual to find aerodynamic characterizations of complete series of airfoils. To overcome these issues, recent approaches in the literature have focused on the development of automatized procedures for the creation of new airfoils and its aerodynamics are studied theoretically through the implementation of computational fluid dynamics (CFD) approaches [19, 21–23]. Despite the advances in airfoil shape development and airfoil aerodynamics prediction, only a few studies have addressed which types of airfoils are appropriate for the construction of efficient SWT rotors [4, 18, 24, 25]. The result thereof is the fact that SWT aerodynamic technologies have been scarcely explored; thus, a formal study to identify the geometric blade features that improve the SWT aerodynamic efficiency is of practical and theoretical relevance. In order to meet this objective, the present work considers the complete National Advisory Committee for Aeronautics (NACA) 4-digit airfoil series as a test case.

There is an infinite set of two-dimensional airfoils that might be used in the development of WT blades. The NACA 4-digit series of airfoils [16] are a small subset of analytically developed airfoils that have been used in different applications, including wind energy [26]. Each of the four digits describes different geometrical features in the set. The first digit describes the maximum camber as a percentage of the airfoil chord length. The airfoil camber is defined as a measure of the asymmetry between the top and the bottom surfaces of the airfoil. Therefore, airfoils without camber are defined as symmetric airfoils. The second digit describes the distance of the maximum camber from the airfoil leading edge in percentage tens of the chord length. The last two digits describe the maximum thickness of the airfoil as a percentage of the airfoil chord length. For example, the NACA 7512 is an airfoil having a maximum camber of 7% located at 50% from the leading edge and having a maximum thickness of 12% with respect to the airfoil chord length.

Only a small set of the NACA 4-digit airfoils have been experimentally characterized [16, 17, 20]. Therefore, their aerodynamics is approximated through the implementation of CFD approaches. In the present work, the two-dimensional, incompressible, steady-state aerodynamic forces, in terms of the aerodynamic coefficients of lift (C_L) and drag (C_D), are obtained by means of the XFOIL 6.96 software. XFOIL [27] is a Fortran-based software, created by Drela in 1989, for the analysis of the subsonic aerodynamics of isolated airfoils. XFOIL computations are based on a combined panel method and an integral boundary layer formulation, which in turn is complemented with an e^n laminar-to-turbulent transition method. Given an initial inflow condition, the flow velocity distribution around the airfoil is computed from the panel method while accounting for viscous forces and the induced vorticity from the airfoil surface. The resultant boundary layer and wake are interacted with a surface transpiration model. The resultant flow field is incorporated into the fluid mechanics viscous equations, yielding a nonlinear elliptic system of equations which is solved by a Newton-Raphson algorithm, resulting in both a complete pressure and velocity distributions in the airfoil vicinity. The lift coefficient (C_L) is calculated by direct surface pressure integration, as viscous contributions to the lift force can be neglected, and the pressure coefficient (C_{PR}) is calculated using the Karman-Tsien compressibility correction. The drag coefficient (C_D) is determined from the wake momentum thickness far downstream and calculated with use of the Squire-Young formulation. The methods, corrections, and the boundary layer formulation used in XFOIL are extensively described in [27, 28]. The XFOIL 6.96 software was incorporated into the BEM model (described in next section) and the hybrid solution procedure, both of which were developed in a MATLAB 2012a environment.

The airfoil aerodynamics can be computed only for pre-stall conditions (e.g., at low angles-of-attack) since the complex flow behavior of the stall state cannot be accurately modeled by XFOIL. Stall is a local boundary layer flow detachment from the blade surface experienced when the local effective angle-of-attack is large and the flow stream lines cannot follow the blade surface due to the large induced adverse pressure gradients. The stall condition produces dramatic reductions in the local lift force while increasing the local drag force, thus, reducing the aerodynamic efficiency of the WT rotor. Nevertheless, induced stall conditions are often useful for control and regulation purposes which, however, are not considered in this work. A complete database of aerodynamic coefficients was constructed from XFOIL 6.96 for the range of airfoil thicknesses $10 \leq \gamma \leq 24$ [%] (i.e., the last two digits of the NACA 4-digit series), the range of angles-of-attack $-20^\circ \leq \alpha \leq 20^\circ$, and the range of Reynolds numbers $1 \times 10^3 \leq \text{Re} \leq 1 \times 10^6$ for each combination of the maximum camber value and the maximum camber position (i.e., the first two digits of the NACA 4-digit series). The database quality was inspected and a few corrections (e.g., related to numerical divergences) were performed at intermediary angles-of-attack and Reynolds numbers by implementing bicubic spline interpolations. The resultant aerodynamic coefficients were extrapolated up to $\alpha = 90^\circ$ by means of the Viterna-Corrigan semiempirical method [29, 30] in order to model the stall state. The database was stored and specific aerodynamic values were made available to the BEM model via three-dimensional bicubic spline interpolations during the rotor aerodynamic performance evaluation.

2.2. Blade Element Momentum (BEM) Model. The WT aerodynamic efficiency is quantified through the use of a state-of-the-art blade element momentum (BEM) model [29–31], which has shown good accuracy when predicting the aerodynamic efficiency of large-scale WT blades. The main differences between the BEM models proposed in the literature are related to its numerical solution procedures and the semiempirical correction factors used to account for the effects that were not explicitly considered during the derivation of the model. The steady-state BEM model relates the blade geometry with the rotor aerodynamic performance depending on the magnitude of the free-stream wind speed (U_∞ [m/s]) approaching the WT and the WT rotational speed (η [RPM]).

The BEM model is composed of two different parts. The first part is focused on the description of the local aerodynamics of the different blade elements (e.g., blade sectional airfoils), which are subject to a force balance analysis that is dependent on the local inflow condition. The second part is focused on the description of the global processes of mass and momentum conservation in which the integral versions of the governing equations of the fluid mechanics are used to quantify and balance the changes in linear and angular momentum of the air flow with the forces and torques on the WT rotor. The resultant set of coupled equations are reinterpreted using the concepts of axial (a_b) and tangential (a'_b) induction factors, which describe the effective local axial and tangential wind speed magnitudes in a normalized fashion. The Prandtl factor (f), as shown in (2), is included in the BEM nonlinear model to describe additional losses caused by the generation of vortices at the tip of the blades.

The BEM model variables are the axial and tangential induction factors, which are readily solved using a Newton-Raphson algorithm for a given inflow condition and certain blade geometric characteristics while considering a predefined maximum number of iterations and a finite numerical tolerance. The blade geometric characteristics are included in the BEM model both explicitly, through the calculation of the chord radial solidity (σ_r), and implicitly, during the calculation of the effective lift (C_L) and drag (C_D) aerodynamic coefficients at each of the blade span positions. The Reynolds number effects are incorporated into the BEM model during

Table 1: Nomenclature of the BEM model.

Local axial induction factor $a_b = \dfrac{U_\infty - U_{d_b}}{U_\infty}$	U_∞ U_{d_b}	Free-stream wind speed [m/s]. Wind speed before crossing the blade segment.
Local tangential induction factor $a'_b = \dfrac{U_{\theta 1_b}}{\Omega r}$	$U_{\theta 2_b}$	Tangential flow speed after crossing the blade segment [30].
	ρ ν	Air density [kg/m^3]. Kinematic viscosity [m^2/s].
Azimuthally averaged axial induction factor $a_r = \dfrac{U_\infty - U_{d_r}}{U_\infty} = a_b f$	$C_L(\alpha, \mathrm{Re})$ $C_D(\alpha, \mathrm{Re})$ $c(r)$	Lift coefficient. Drag coefficient. Airfoil chord length [m].
Azimuthally averaged tangential induction factor $a'_r = \dfrac{U_{\theta 1}}{\Omega r} = a'_b f$	$\alpha(r)$ $\beta(r)$ $\phi = \alpha + \beta$ Ω	Angle-of-attack [°]. Twist angle [°]. Inflow angle [°]. Rotor angular speed [rad/s].
Radial and azimuthally averaged axial induction factor $a = \dfrac{U_\infty - U_d}{U_\infty}$	$\sigma_r = \dfrac{Nc}{2\pi r}$ $N = 3$ $\mu = \dfrac{r}{R}$	Chord radial solidity. Number of blades. Normalized radial position.
Reference tangential flow speed at a local segment [30] $U_{\theta 1_b} = \dfrac{1}{2} U_{\theta 2_b}$	$\mu_R = \dfrac{r_0}{R}$ r	Normalized radius of the blade root. Radial distance [m].
Average axial flow speed crossing the rotor $U_d = \dfrac{U_\infty + U_w}{2}$	R A $\lambda = \dfrac{\Omega R}{U_\infty}$	Blade radius [m]. Rotor swept area [m^2]. Tip speed ratio.
Average axial flow speed in the far wake, where the pressure has reestablished to the atmospheric value $U_w = 2U_d - U_\infty$	$C_t = C_L \cos\phi + C_D \sin\phi$ $C_n = C_L \sin\phi + C_D \cos\phi$	Tangential force coefficient. Normal force coefficient.

the calculation of the aerodynamic coefficients. The effective Reynolds number at each blade span position is quantified using (3).

BEM model:

$$\frac{a_r(1-a_r)}{(f-a_r)^2} + \left[\frac{\sigma_r C_t}{4f^2\sin^2(\phi)}\left(\frac{f-a_r}{1-a_r}\right)\right]^2 = \frac{\sigma_r C_n}{4f^2\sin^2(\phi)}, \tag{1a}$$

$$\frac{a'_r(1-a_r)}{(f-a_r)^2} = \left(\frac{1}{\lambda\mu}\right)\frac{\sigma_r C_t}{4f^2\sin^2(\phi)}. \tag{1b}$$

Prandtl factor:

$$f(\mu) = \frac{2}{\pi}\cos^{-1}\left(e^{-(N/2)((1-\mu)/\mu)\sqrt{1+(\lambda\mu/(1-a_b))^2}}\right). \tag{2}$$

Effective Reynolds number:

$$\mathrm{Re} = \frac{c}{\nu}(\Omega R\mu)\sqrt{(1+a'_b)^2 + \left(\frac{1-a_b}{\lambda\mu}\right)^2}. \tag{3}$$

Power coefficient:

$$C_P = 8\lambda^2\int_{\mu_R}^{1}\mu^3 a'_r(1-a_r)\,d\mu = \frac{P}{(1/2)\rho U_\infty^3 A}. \tag{4}$$

Thrust coefficient:

$$C_T = \int_{\mu_R}^{1} 2\sigma_r\left(\frac{1-a_b}{\sin\phi}\right)^2 C_n\mu\,d\mu = \frac{F_{\mathrm{Ax}}}{(1/2)\rho U_\infty^2 A}. \tag{5}$$

Total torque at rotor and torque coefficient:

$$F_{\mathrm{Q}} = 2\pi\rho R\int_{\mu_R}^{1} U_{\theta 2_b} U_{d_b} r^2\,d\mu,$$
$$C_{\mathrm{Q}} = \frac{F_{\mathrm{Q}}}{(1/2)\rho U_\infty^2 AR} = \frac{C_P}{\lambda}. \tag{6}$$

From the solved induction factors, normalized variables describing the WT rotor aerodynamic performance can be derived, such as the power coefficient (C_P), the thrust coefficient (C_T), and the torque coefficient (C_{Q}), which in turn can be transformed into the dimensioned variables of mechanical power (P), axial force (F_{Ax}), and torque force (F_{Q}), respectively, as shown in (4)–(6). The BEM model assumptions, limitations, and complete derivation can be consulted in [29, 30] and its nomenclature is presented in Table 1.

2.3. Problem Definition and Optimization Strategy. The blade variables to be optimized are the blade sectional airfoil shapes and the geometrical distributions of chord length (c [m]), twist angle (β [°]), and thickness (γ [%]) along the WT blade span. However, there is no general theory (i.e., there are no analytical closed-form expressions or concise guidelines that are not based on crude assumptions) describing the optimal radial variation of these variables. Therefore, robust and flexible analytical functions that depend on a reduced set of continuous parameters were developed to describe such geometric distributions, as shown in (8)–(10). Each of the different analytical functions describing the geometrical distributions contains four variables. The variables $\{c, \beta, \gamma\}_{1,2}$ define the initial and final value (i.e., the value at $r = r_0 = \mu_R R$ and $r = R$) of the geometrical distributions, respectively. The variables $\{c, \beta, \gamma\}_{3,4}$ define the concavity and vertical displacement of the geometric distributions, respectively. The blade radial discretization was performed in a nonuniform way, as described in (7a) and (7b), in order to reduce the amount of discrete blade segments, which proved to reduce the demand of computational resources during the optimization process without incurring numerical inaccuracies during BEM calculations.

After performing a sensitivity analysis, it was concluded that 17 discrete radial positions (N_x) were enough to obtain accurate numerical values of the BEM model predictions. This value differs from the values considered in previous studies [7, 13] and special attention is required since considering less than 17 discrete positions might lead to numerical inaccuracies that in turn might influence the optimization results. The parametric expressions used for the representation of the normalized radial coordinate, the local chord, twist angle, and thickness are the following:

$$\mu(x) = \frac{r(x)}{R} = \frac{1}{2} + \left(\frac{1}{2} - \zeta(x)\,\mu_R\right)\sin\left(\frac{\pi x}{2}\right), \qquad \text{(7a)}$$

$$\zeta(x) = \begin{cases} 1, & -1 \le x \le 0, \\ 0, & 0 < x \le 1, \end{cases} \qquad \text{(7b)}$$

$$c(r) = \frac{c_1}{\left(\mu_R R + c_3\right)^{\ln(c_2/c_1)/\ln((R+c_3)/(\mu_R R+c_3))}} \times \left(r + c_3\right)^{\ln(c_2/c_1)/\ln((R+c_3)/(\mu_R R+c_3))} + c_4, \qquad \text{(8)}$$

$$\beta(r) = \frac{\beta_1}{\left(\mu_R R + \beta_3\right)^{\ln(\beta_2/\beta_1)/\ln((R+\beta_3)/(\mu_R R+\beta_3))}} \times \left(r + \beta_3\right)^{\ln(\beta_2/\beta_1)/\ln((R+\beta_3)/(\mu_R R+\beta_3))} + \beta_4, \qquad \text{(9)}$$

$$\gamma(r) = \frac{\gamma_1}{\left(\mu_R R + \gamma_3\right)^{\ln(\gamma_2/\gamma_1)/\ln((R+\gamma_3)/(\mu_R R+\gamma_3))}} \times \left(r + \gamma_3\right)^{\ln(\gamma_2/\gamma_1)/\ln((R+\gamma_3)/(\mu_R R+\gamma_3))} + \gamma_4. \qquad \text{(10)}$$

The explicit variables to be optimized are the continuous values of $\{c, \beta, \gamma\}_{1,2,3,4}$. Each of the 12 variables to be optimized must be linearly bounded to avoid unphysical solutions. The WT rotor aerodynamic efficiency, in terms of the BEM-derived power coefficient, is the performance metric to be maximized in the present work. However, in order to account for different operational states, the mono-objective performance metric to be maximized is the integral of the power coefficient over a range of tip speed ratios, as defined in (11). The selection of this performance metric has the ultimate effect of improving the annual energy production of the SWT even without explicitly considering a site-specific wind resource.

Additional metrics are monitored during the optimization process, including the integral of the thrust coefficient over the same range of tip speed ratios, as defined in (12), the cut-in wind speed, and the total volume of the blade. The cut-in wind speed is approximated from the simple balance between (1) the product of the free-stream dynamic pressure and the blade solidity, which results in an aerodynamic force over the rotor that is dependent on the blade geometry, and (2) the torque needed to overcome the static friction and/or the cogging torque of the electric generator (T_{Ar}), as shown in (13). Consider

$$\text{Max}\left[\Gamma_{C_P} = \int_{\lambda_I}^{\lambda_F} C_P(\lambda)\, d\lambda\right], \qquad \text{(11)}$$

$$\Gamma_{C_T} = \int_{\lambda_I}^{\lambda_F} C_T(\lambda)\, d\lambda, \qquad \text{(12)}$$

$$V_{\text{cut-in}} = \frac{\sqrt{2T_{Ar}/\rho N}}{\sqrt{(1/2)\int_{r_0}^{R} rc\sin(2\beta)\, dr}}. \qquad \text{(13)}$$

The complete WTBO model, as presented, is a mixed-integer, linearly constrained, nonconvex, and nonlinear optimization model. The number of elementary operations needed to compute the aerodynamic efficiency of a WT rotor increases in an approximately linear way as a function of the number of blade discrete segments (N_x). However, the number of possible combinations of blade sectional airfoils increases in an exponential way ($N_{Airf}^{N_x}$) as a function of the number of discrete blade segments and the number of available airfoils (N_{Airf}). Therefore, the WTBO model can be categorized as an NPO-complete problem [32], one of the hardest problems to solve in computational optimization.

Calculus-based optimization approaches such as linear/quadratic programming, gradient methods, and Newton-Raphson, among others cannot be used to solve the problem to optimality. In addition, it is considered inadequate to implement relaxation procedures (i.e., procedures that transform the original problem into another equivalent problem of reduced complexity) over the BEM model in order to use tree search-based procedures (e.g., branch and bound, branch and cut, etc.) since the already simplified physics will be greatly influenced, hence, significantly affecting the original problem and its optimal solution. To attempt solving optimization problems of such complexity, heuristic and metaheuristic approaches [33] are commonly employed. However, heuristic and metaheuristic algorithms cannot guarantee optimality. Therefore, multiple executions of the optimization procedure must be performed to obtain a near-optimal solution to the problem instance being solved, as shown in Section 4. A wide range of metaheuristics can be used to perform the optimization task. However, evolutionary-based algorithms have proven to be efficient while optimizing WT blades [5, 10, 13].

This work extends the optimization effectiveness of previous approaches by developing a novel hybrid optimization procedure in which the geometrical shape of the blade is optimized in a nested procedure, while including more decision variables (e.g., number of airfoils and number of discrete segments). The proposed hybrid procedure consists of two parts: the first part focuses on optimizing the geometrical distributions of chord length, twist angle, and thickness, given a combination of sectional airfoils, by employing a local search metaheuristic based on the simulated annealing (SA) algorithm. The combination of sectional airfoils is proposed by a leading population-based metaheuristic based on the virtual gene genetic algorithm (vgGA). Hence, the SA algorithm is nested into the vgGA during the iterative optimization process. In the next subsections, a detailed description of the metaheuristic algorithms and the hybrid procedure is presented.

2.4. Local Search Metaheuristic Based on the Simulated Annealing Algorithm. The present work implements the SA algorithm to optimize the blade geometric distributions, shown in (8)–(10), in order to maximize the objective function, shown in (11). The SA algorithm [33–35] is based on the natural process of annealing of solids, which is a thermal process for obtaining low energy states in solids by exposing them to a heat bath. The annealing process consists of the following two steps: (1) increase the temperature of the heat bath until the solid melts and (2) slowly decrease the temperature of the heat bath until the particles arrange themselves in the low energy ground state of the solid. In the melt phase, the constituent particles arrange themselves randomly, and in the cooling stage the particles arrange themselves in an organized lattice. The analogy between the annealing process and the optimization process is that potential solutions of the studied problem are comparable to states of a physical system undergoing a process of annealing and the quality of a potential solution is equivalent to the energy of a state.

While traditional local search (and variable local search) procedures will generally accept transitions between feasible neighbor solutions only if its quality is better than the current solution, the SA algorithm has the ability of allowing transitions between feasible solutions whose qualities are worse than the current solution in the hope of avoiding getting stuck at a local extreme point. The transition to a worse solution is modeled as a probabilistic process, which relies on the Metropolis criterion [34] and depends on the actual state and the new state quality. The probability of rejection of states with lower quality increases as the search continues (i.e., the SA algorithm behaves as a greedy algorithm), therefore, simulating the arrangement of particles in an organized lattice during the cooling process.

The SA's ease of implementation, convergence properties, and ability to escape local extreme points have made it a popular technique during the past two decades. Therefore, the SA algorithm is considered as a suitable solution procedure for solving the WTBO model presented in this work. The interested reader is referred to [33–35] for more detailed descriptions of the SA algorithm. The SA's state is represented by a 4 × 3 matrix (G_s) containing the continuous values of the geometrical variables $\{c, \beta, \gamma\}_{1,2,3,4}$. The initial state is computed randomly within the geometrical variables bounds. During the optimization process, neighbor solutions are generated according to

$$G_{s+1} = G_s + U\left(-G_V, G_V\right), \qquad (14)$$

where s represents the actual state and U is the continuous uniform probability density function that computes random neighbor matrices within the interval $[-G_V, G_V]$. After computing a neighboring solution, a feasibility check is performed to verify that the new state does not contain values outside the defined bounds. In the case of unfeasibility, the elements outside the bounds are set to the nearest boundary value. The SA implementation allows for the variation of the temperature within a Markov chain every time a transition is accepted, thereby allowing the algorithm to escape local maxima by relaxing the Metropolis criterion [33]. A schematic diagram of the SA-based procedure is shown in Figure 1. The SA's parameters were tuned by conducting a statistical design of experiments, as described in Section 2.6.

2.5. Population Metaheuristic Based on the Virtual Gene Genetic Algorithm (vgGA) and the Hybrid vgGA-SA Solution Procedure. The present work implements the virtual gene genetic algorithm (vgGA) [36] to optimize the selection and combination of different airfoil shapes along the WT blade span. The GA refers to a class of adaptive search procedures based on the principles derived from natural evolution and genetics. In the GA, genes represent the design variables and potential solutions (i.e., individuals) are represented by chromosomes containing genes. Like other population-based methods, as the GA proceeds through generations (i.e., iterations), it holds a set of possible solutions referred to as the population. Based on the fitness value (i.e., the quality of a solution), which is determined through the evaluation of the population's individuals, two individuals (i.e., the parents) are selected and are combined through genetic operators (crossover and mutation) resulting in two new individuals (e.g., the offspring). It is expected that the population fitness will improve through the generations (iterations) until a near-optimal solution is obtained.

Many different variants of GAs have been considered in the literature [33] differing in the way genetic operators act over the population and the way individuals are represented. Research has demonstrated that the performance of a GA relies upon the proper choice of the selection and crossover mechanisms and the proper design of the individual representation. In the present work, a binary tournament selection, a uniformly distributed one-point crossover, and a uniformly distributed mutation methods were considered. The individual's representation is based on a chromosome containing binary digits (genotype). The combination of binary digits in turn defines the airfoil shape to be used at each of the blade span positions (phenotype). One of the

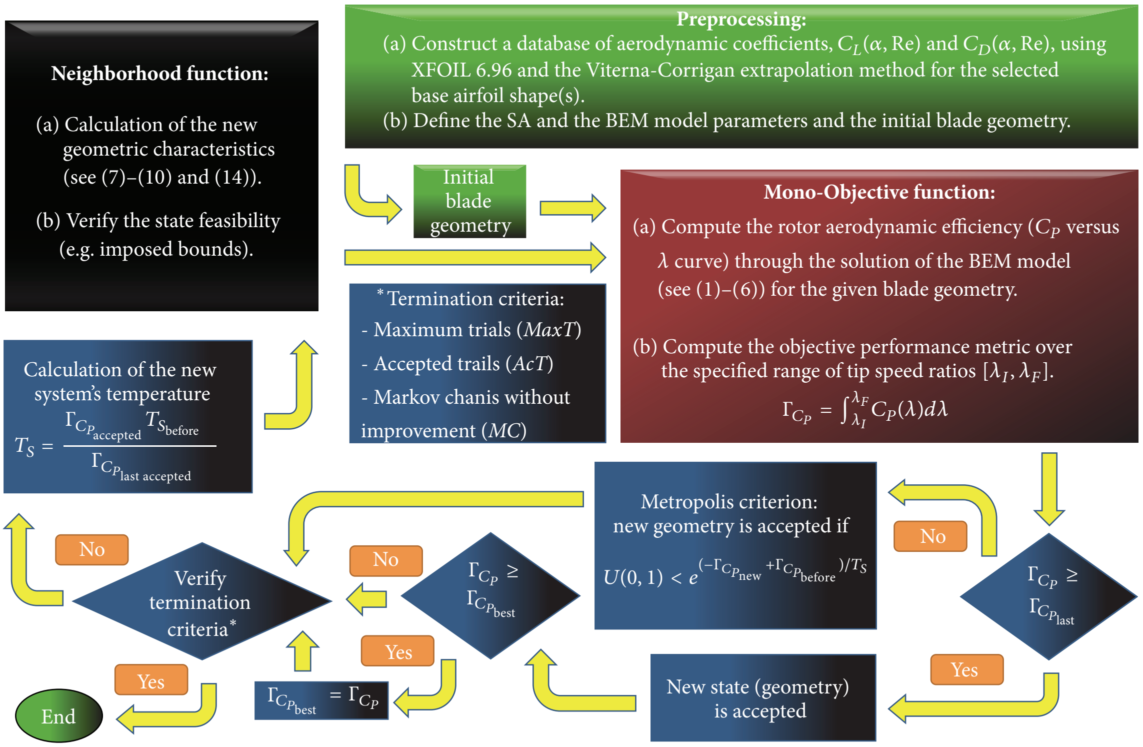

Neighborhood function:
(a) Calculation of the new geometric characteristics (see (7)–(10) and (14)).
(b) Verify the state feasibility (e.g. imposed bounds).
Preprocessing:
(a) Construct a database of aerodynamic coefficients, $C_L(\alpha, \mathrm{Re})$ and $C_D(\alpha, \mathrm{Re})$, using XFOIL 6.96 and the Viterna-Corrigan extrapolation method for the selected base airfoil shape(s).
(b) Define the SA and the BEM model parameters and the initial blade geometry.
Initial blade geometry
Mono-Objective function:
(a) Compute the rotor aerodynamic efficiency (C_P versus λ curve) through the solution of the BEM model (see (1)–(6)) for the given blade geometry.
(b) Compute the objective performance metric over the specified range of tip speed ratios $[\lambda_I, \lambda_F]$.
$\Gamma_{C_P} = \int_{\lambda_I}^{\lambda_F} C_P(\lambda)d\lambda$
*Termination criteria:
- Maximum trials (MaxT)
- Accepted trails (AcT)
- Markov chanis without improvement (MC)
Calculation of the new system's temperature
$T_S = \frac{\Gamma_{C_{P_{\text{accepted}}}} T_{S_{\text{before}}}}{\Gamma_{C_{P_{\text{last accepted}}}}}$
Metropolis criterion: new geometry is accepted if
$U(0,1) < e^{(-\Gamma_{C_{P_\text{new}}} + \Gamma_{C_{P_\text{before}}})/T_S}$
No
$\Gamma_{C_P} \geq \Gamma_{C_{P_\text{last}}}$
Yes
New state (geometry) is accepted
No
$\Gamma_{C_P} \geq \Gamma_{C_{P_\text{best}}}$
Yes
$\Gamma_{C_{P_\text{best}}} = \Gamma_{C_P}$
No
Verify termination criteria*
Yes
End

FIGURE 1: Schematic diagram of the SA-based solution procedure.

issues with considering this kind of representation is related to the estimation of the minimum number of bits needed to completely describe the range of values each variable can assume. In order to quantify this value, the chromosome is divided into N_x segments, corresponding to the number of blade discrete points. Each segment will contain MAB bits, quantified using (15), which are sufficient to represent the number of possible airfoils that each blade section can assume. Consider

$$\mathrm{MAB} = \mathrm{ceil}\left(\frac{\log\left(N_{Airf}\right)}{\log(2)}\right), \tag{15}$$

where the ceil function rounds the argument towards infinity and N_{Airf} is the number of available airfoils. Each of the N_x segments in the chromosome (genotype) is reinterpreted as a real number (phenotype), which is rounded to obtain a feasible integer value representing an airfoil identifier. For example, a hypothetical first chromosome segment $[000\cdots000]$ represents that the first available airfoil (i.e., airfoil #1) will be located at the first discrete segment of the blade, which is the closest to the blade root. Another hypothetical first chromosome segment $[111\cdots111]$ represents that the last available airfoil will be located at the first discrete segment of the blade, which is the closest to the blade root. A complete example showing the genotype and the phenotype representations of a hypothetical individual, considering 8 available airfoils and 4 discrete segments, is shown next:

$$\begin{aligned} \text{Individual} &\longrightarrow [\mathrm{MAB} \mid \mathrm{MAB} \mid \mathrm{MAB} \mid \mathrm{MAB}] \\ &= [111 \mid 101 \mid 000 \mid 001] = [8 \mid 6 \mid 1 \mid 2]\,. \end{aligned} \tag{16}$$

Note that in the previous example the number of discrete segments was conveniently limited to $N_x = 4$. As discussed before, the minimum number of segments needed to avoid numerical inaccuracies during BEM calculations is approximately $N_x = 17$. In the remaining of the work, $N_x = 17$ discrete segments were considered.

Once an entire population is constructed, its evaluation proceeds by executing five times the SA-based procedure proposed in the previous section, in which the near-optimal radial distributions of the geometrical variables of chord length, twist angle, and thickness are obtained for each individual. The quality of the best found solution, obtained through the five SA-based executions, is considered as the fitness of the individual. A schematic diagram of the vgGA-SA hybrid solution procedure is shown in Figure 2. The SA's and the vgGA's parameters were tuned by conducting two independent statistical designs of experiments, as presented in the next subsection.

2.6. The vgGA-SA's Parameters Tuning through Statistical Designs of Experiments. Optimization procedures based on heuristic and metaheuristic algorithms rely on their parameters to obtain high-quality/near-optimal solutions to the problem being investigated. It has been observed that the

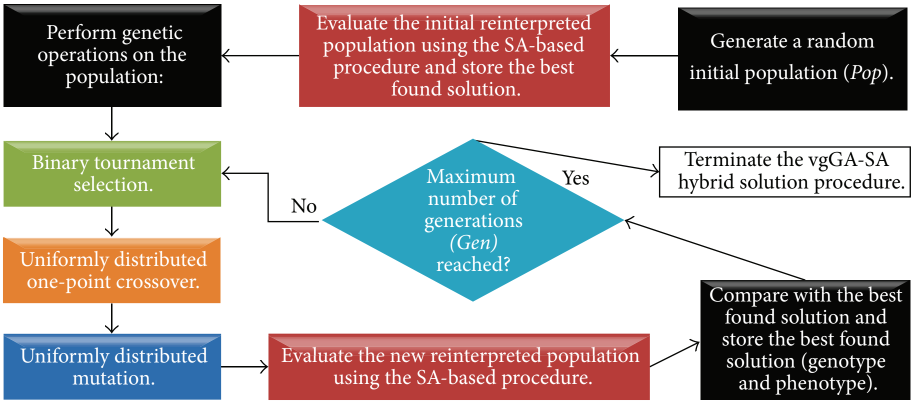

Perform genetic operations on the population:
Evaluate the initial reinterpreted population using the SA-based procedure and store the best found solution.
Generate a random initial population (*Pop*).
Binary tournament selection.
Maximum number of generations (*Gen*) reached?
Yes
No
Terminate the vgGA-SA hybrid solution procedure.
Uniformly distributed one-point crossover.
Uniformly distributed mutation.
Evaluate the new reinterpreted population using the SA-based procedure.
Compare with the best found solution and store the best found solution (genotype and phenotype).

Figure 2: Schematic diagram of the vgGA-SA hybrid solution procedure.

Table 2: SA's Parameters to be tuned and the considered levels.

Factor (SA's parameters)	Levels				
Temperature variation within Markov chains (T_{Var})		False		True	
System's initial temperature (T_{S0})	0.001	0.01	0.1	1	10
Maximum number of accepted trials (AcT)	5	10	50		150
Maximum number of trials ($MaxT$)	5	10	50		150
Number of Markov chains (MC)	1			2	
Neighborhood size (G_V) as a percentage of the size of the search space ($G_{UB} - G_{LB}$)	1%	5%	10%	30%	50%
Cooling rate (α_{SA})	0.7	0.8	0.9	0.97	0.99

use of poorly tuned parameters may limit the ability of optimization procedures to find high-quality solutions. The parameter tuning of complex algorithms is a well-known problem; however, there are no standard procedures to perform the tuning task.

In order to properly tune the vgGA-SA's parameters, two independent full factorial designs of experiments (DOEs) were conducted. The outcomes of the DOEs allowed (1) understanding which factors (parameters) are significant during the optimization process, (2) finding the best combination of values for the parameters of both the SA and the vgGA algorithms, and (3) quantifying the effects of using poor-tuned parameters. Both DOEs were processed with the aid of the statistical software Minitab v16.

SA's Parameters Tuning. For the SA's parameters tuning process, the rotor parameters described in Section 3 and the NACA 75XX airfoil were considered. Table 2 summarizes the SA's factors to be tuned and the considered levels. The DOE considered 10 replicates for each combination of parameters' levels, resulting in 80,000 SA-based simulations.

The DOE evaluation involves performing an analysis of variance (ANOVA), the aim of which is to determine what factors are significant during the optimization process. The ANOVA model adequacy verification did not exhibit issues with regard to (1) the residual's normality behavior, (2) the experiments independency, and (3) the constant variance assumptions. The validations of the first and third assumptions were performed by conducting an Anderson-Darling normal probability test and a Bartlett's and Levene's constant variance tests, respectively.

With a significance level of 5%, the ANOVA results suggested that there is relative significance between all the factors, which means that specific combinations of them will allow the SA-based procedure to reach high quality solutions. To enhance the understanding of how these factors influence the optimization results, Figure 3 presents the main effects of the factors on the response (i.e., the objective metric, quantified using (11)). From the main effects charts shown in Figure 3, it can be observed that some factors are more important than others on average (although all factors are statistically significant, as suggested by the ANOVA).

From Figure 3(a) it can be observed that by varying the system's temperature within Markov chains, a slightly better solution quality can be obtained on average. The temperature variation along Markov chains allows the system's temperature to increase when high quality states are explored. The result of increasing the system's temperature is that the SA greedy behavior is relaxed in the cooling stage, and thus the algorithm is granted with the ability to continue searching for better states and avoids the possibility of being trapped at local maxima.

Figure 3(b) indicates that the initial temperature has a significant impact on the performance of the SA algorithm. The initial system's temperature must be low in order to obtain a better solution quality on average. The SA algorithm tends to

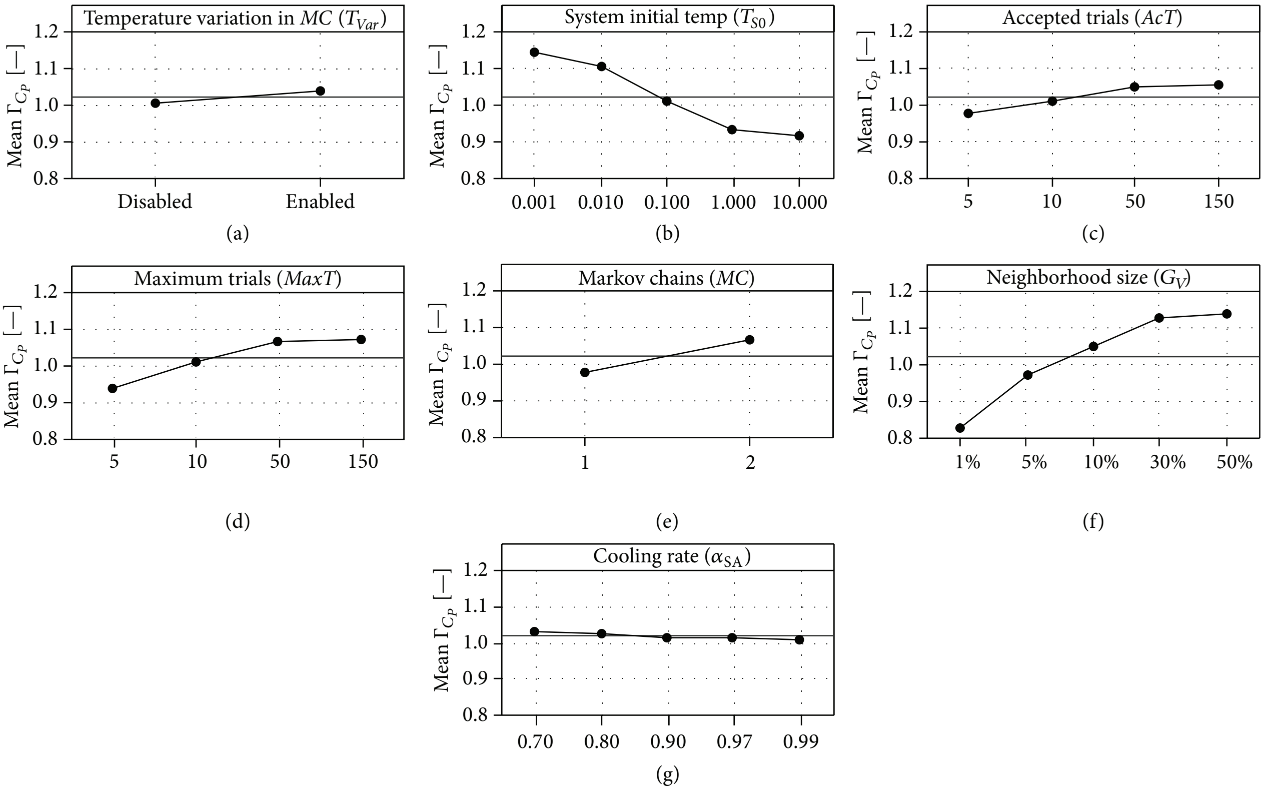


Figure 3: Average performance of the SA-based procedure while optimizing a NACA 75XX-based rotor for different combinations of parameters' levels.

be more selective (i.e., behaves as a greedy algorithm) if the system's temperature is low enough because the Metropolis criterion will reject transitions to low quality states more often.

From Figures 3(c), 3(d), and 3(e) it can be noted that by increasing the maximum number of accepted trials, the maximum number of trials, and the number of Markov chains, a better solution quality can be reached on average. This behavior is to be expected since the algorithm is allowed to continue its search for the best solution. Nevertheless, no significant improvements in solution quality are observed when allowing more than 50 trials.

Figure 3(f) shows that the neighborhood size, as a percentage of the search space size, has great influence on the performance of the SA algorithm. By allowing large state transitions, an improved solution quality can be obtained on average. This implicitly means that the solution space is not smooth, which is a common characteristic of nonconvex search spaces. To overcome the issue of optimizing variables on nonconvex spaces, the SA algorithm performs abrupt changes in the state in order to properly explore the solutions space at the cost of increasing the variance of the state quality during the optimization process.

From Figure 3(g), it can be observed that the cooling rate has little effect on the mean response. By lowering the magnitude of the cooling rate a slightly better solution quality can be obtained on average. As the system's temperature decreases, the SA algorithm becomes more selective regarding the visited states; thus, low quality states are discarded by the Metropolis criterion.

Table 3: vgGA's parameters to be tuned and the considered levels.

Factor (vgGA's parameters)	Levels		
Number of generations (Gen)	40	90	140
Population size (Pop)	20	60	80
Crossover probability (p_{cr})	0.85	0.95	0.98
Mutation probability (p_m)	0.002	0.02	0.2

After a careful inspection of the DOE results and after performing a detailed study of the best solution among the 80,000 simulations, the combination of parameters that maximizes the performance of the SA-based procedure was identified and is summarized next: neighborhood size $G_V = (G_{UB} - G_{LB})/3.33$, system's initial temperature $T_{S0} = 0.001$, system's cooling rate $\alpha_{SA} = 0.97$, maximum number of accepted trials $AcT = 150$, maximum number of trials $MaxT = 150$, and number of Markov chains $MC = 2$. The matrices G_{UB} and G_{LB} refer to the upper and lower bounds of the 12 variables to be optimized and its difference defines the size of the search space.

vgGA's Parameters Tuning. The vgGA's parameters were tuned following the previously described procedure. Once again,

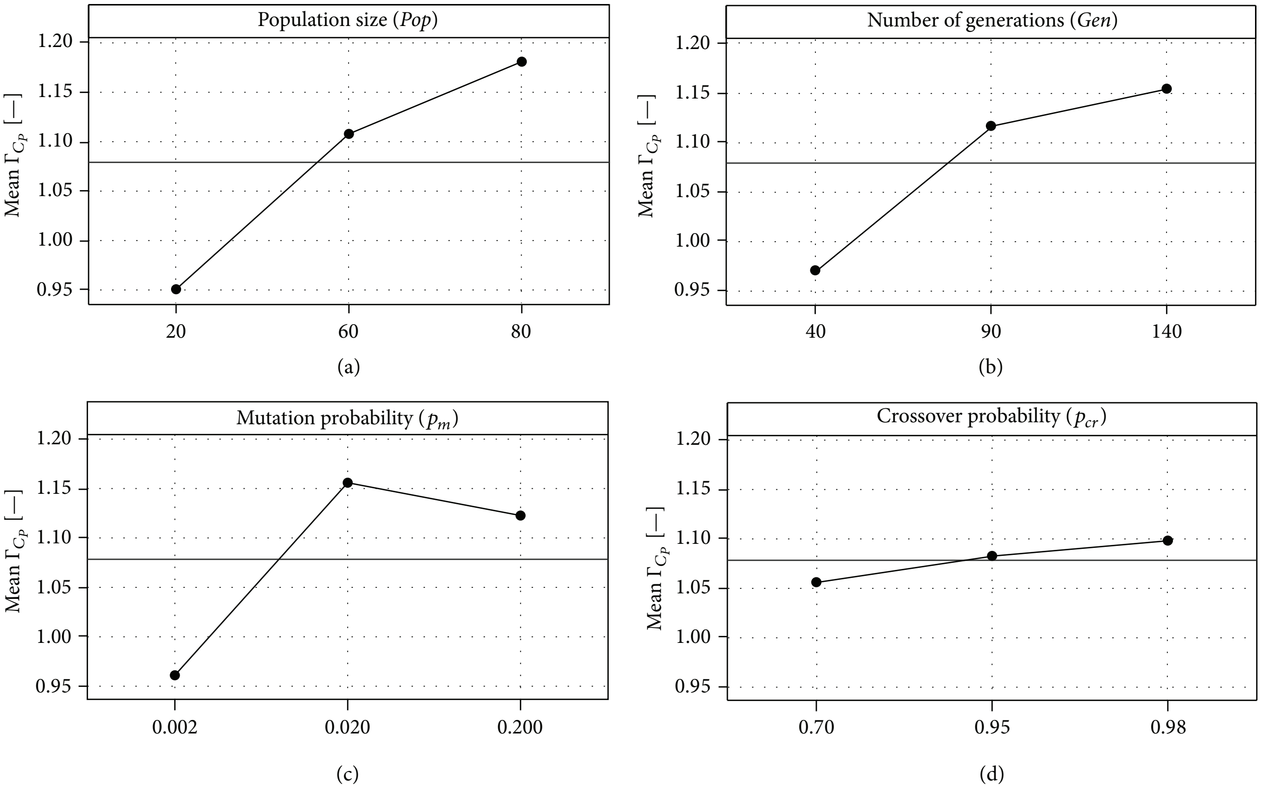


Figure 4: Average performance of the vgGA-SA procedure while optimizing a NACA hybrid rotor for different combinations of parameters' levels.

the rotor parameters described in Section 3 were considered. In addition, the nested SA-based procedure considered the previously tuned parameters. Table 3 resumes the vgGA's factors to be tuned and the considered levels. The DOE considered 10 replicates for each combination of parameters' levels, resulting in 810 vgGA-SA-based simulations.

The ANOVA model adequacy verification did not exhibit issues with regard to (1) the residual's normality behavior, (2) the experiments independency, and (3) the constant variance assumptions. With a significance level of 5%, the ANOVA results suggested that the significant factors are the population size, the number of generations, and the mutation probability. It is worth nothing that the crossover probability was not significant.

Figure 4 presents the main effects of the factors on the response. From Figures 4(a) and 4(b), it can be noted that by increasing the number of generations and the population size a much better solution quality can be reached on average. This behavior is to be expected since the algorithm is allowed to continue its search for the best solution. Although both figures suggest that additional improvements in the average solution quality may be obtained when increasing the value of both parameters' levels (e.g., beyond the proposed levels), it is considered appropriate to maintain such levels since the trade-off between the computational resources spent and the obtained solution quality worsens as the levels' values are increased.

Figure 4(c) indicates that the mutation probability has a significant impact on the performance of the vgGA algorithm. On the one hand, small values of the mutation probability inhibit the ability of the vgGA to explore the search space. On the other hand, large values of the mutation probability lead to large variance values on the population fitness, thus the vgGA algorithm cannot identify and combine the characteristics of the best individuals.

Although the crossover probability is not a significant factor (as suggested by the ANOVA), it is observed from Figure 4(d) that by increasing its value a slightly better solution quality can be obtained on average.

After a careful inspection of the DOE results and after performing a detailed study of the best solution among the 810 simulations, the combination of parameters that maximizes the performance of the vgGA-based procedure was identified and is summarized next: population size $Pop = 80$, number of generations $Gen = 140$, crossover probability $p_{cr} = 0.98$, and mutation probability $p_m = 0.02$.

Both the SA's and the vgGA's best combinations of parameters values are highly dependent on the problem size. Therefore, a specific parameter tuning process must be performed for when considering different scenarios.

3. Experimental Development

A validation study of a set of optimized rotors based on the NACA 4-digit airfoils, as described in next subsections, was performed at the subsonic closed-return wind tunnel of the Center for Research and Innovation in Aerospace Engineering (CIIIA) [37]. The dimensions of the wind tunnel's rectangular test section are 1 × 1 × 1.5 m^3. A maximum wind speed of 60 m/s can be reached at the test section with the aid of a 300 kW frequency-controlled fan. All wind speed measurements were made using a hot-wire anemometer located at the center position in the upwind part of the test section.

In order to avoid flow-wall blockage effects, the rotor radius was set to 0.175 m during the rotor optimization process, thus limiting the swept rotor area to 10% of the wind tunnel cross sectional area. The radius of the blade root (i.e., the first blade position that contributes to the energy conversion process, $r_0 = \mu_R R$) was set to 0.02 m. The integration limits of the objective function, (11), were set to $\lambda_I = 2$ and $\lambda_F = 5$ with the purpose of (1) forcing the vgGA-SA hybrid procedure to find efficient rotors that have lower cut-in wind speeds and (2) reducing the magnitude of the rotational speed needed to operate at the maximum efficient state, therefore, improving the WT operational safety and reducing the magnitude of the rotational speeds needed during the aerodynamic characterization procedure. The following lower (G_{LB}) and upper (G_{UB}) bound geometric matrices were used during the optimization process:

$$
G_{LB} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}
\begin{array}{c} \begin{array}{ccc} c\,[\text{m}] & \beta\,[^\circ] & \gamma\,[\%] \end{array} \\ \begin{bmatrix} 0.005 & 0 & 10 \\ 0.005 & 0 & 10 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array},
$$
$$
G_{UB} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}
\begin{array}{c} \begin{array}{ccc} c\,[\text{m}] & \beta\,[^\circ] & \gamma\,[\%] \end{array} \\ \begin{bmatrix} 0.03 & 20 & 22 \\ 0.025 & 20 & 22 \\ 1 & 1 & 1 \\ 0.01 & 2 & 2 \end{bmatrix} \end{array}. \tag{17}
$$

The air properties (e.g., density and kinematic viscosity) were calculated as a function of the local pressure, temperature, and relative humidity as suggested by Picard et al. [38].

3.1. Small Wind Turbine Prototype. To perform the validation study of the optimized rotors, a SWT prototype was constructed. The constituent parts of the SWT prototype are (1) the optimized NACA 4-digit-based rotors, (2) a 14-pole neodymium permanent magnet, brushless, radial flux, synchronous, three phase EMP6354-KV200 electric generator, (3) an electrical system composed of a six-phase Guerte KBPC3510 rectifier, two voltage, and two current sensors (Allegro systems ACS758 and ACS715, resp.) located in both the AC and DC parts of the system, and (4) a set of electric resistances based on Kanthal A1 wire. The voltage and current measurements were monitored using a National Instruments data acquisition system (NI DAQ-USB-6211) and a LabVIEW interface, which reported and logged the real-time operation of the SWT prototype.

The optimized rotors were manufactured using a rapid prototyping Dimension Elite 3D printer. A high density ABSplus P430 thermoplastic material, printed in layers of 0.178 mm in thickness, was used during the rotor manufacturing process. After the manufacture, a rotor balance post-process was performed, which consisted in (1) performing a slight sanding of the blade surface to reduce roughness and (2) adding thin layers of an epoxy material in an homogeneous way along the blade surface in order to balance the weight of the rotor and to improve its structural strength.

The EMP6354-KV200 electric generator was characterized following the theoretical and experimental procedures proposed in [29, 39, 40], where a set of characteristic curves of mechanical torque, rectified voltage, and rectified current as a function of both the rotational speed (η [rpm]) and the effective electric load (R_{Load} [Ω]) were obtained by means of a customized electromechanical test bench. The electromechanical characterization allowed determining the aerodynamic efficiency of the SWT at any operational condition by comparing the available free-stream power and the mechanical power delivered to the electric generator. The SWT rotational speed (η [RPM]) was controlled by changing the effective electric load and calculated by measuring the AC voltage frequency ($f_{V_{\text{AC}}}$), as shown in (18). In addition, the generator's no-load opposing torque (T_{Ar}) value was experimentally determined to be 0.0437766 Nm. Consider

$$\eta = \frac{120}{14} f_{V_{\text{AC}}}. \tag{18}$$

The aerodynamic characterization was performed only in steady states, which were reached after a few seconds of operation. All the relevant variables were logged at a frequency of 14 Hz over a two-minute period for each operational state.

4. Results and Discussion

Two different types of evaluation procedures were conducted in order to demonstrate the capabilities of the proposed framework. The first procedure focused on evaluating the ability of the SA-based solution procedure for finding near-optimal radial geometrical distributions for blades composed of one type of airfoil. The second procedure focused on studying and comparing the best result obtained by the vgGA-SA hybrid solution procedure with the best results obtained by the first evaluation procedure. Both procedures considered the evaluation of the complete NACA 4-digit airfoil series (i.e., varying the maximum camber from 0 to 9% and the position of the maximum camber from 0 to 90% of the chord length, resulting in 100 different base airfoils).

First Evaluation. For the first evaluation procedure, the SA-based algorithm was executed thirty times, considering that the resultant blade was composed of only one type of NACA 4-digit airfoil. The average performance of the SA-based algorithm, considering the specific case of optimizing the geometrical distributions of a rotor based on the NACA 75XX airfoil, is shown in Figure 5. The SA-based algorithm was able to reach a high-quality solution in the first 80 iterations, while additional iterations were required to obtain a high quality near-optimal solution. This is consistent with the findings reported in Section 2.6. Similar results were obtained for the

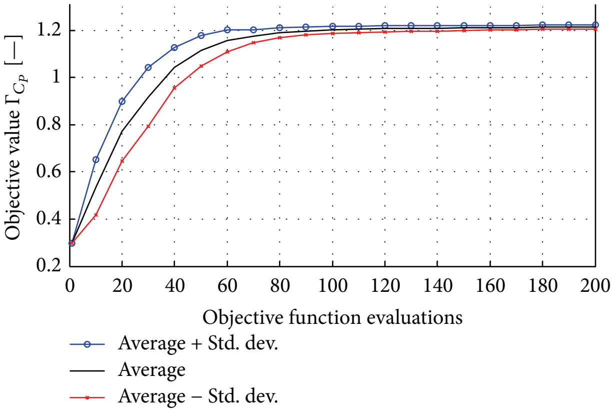


Figure 5: Average performance of the SA-based procedure while optimizing a NACA 75XX-based rotor.

remaining 99 NACA airfoils. The average time required to perform an evaluation of the objective function was 0.1229 seconds, considering that the computational experiments were performed on a Sager NP8230-P151SM1 workstation with an Intel Core i7-4800MQ processor operating at 4 GHz, 16 GB of memory DDR3 @1600 MHz on an Intel HM87 chipset motherboard.

The best result among the thirty SA-based executions, for each of the 100 NACA 4-digit airfoils, was stored and is reported in Figure 6. From Figure 6 it can be noted that the aerodynamic efficiency of the optimized rotors is significantly affected by the selected base airfoil shape. Rotors based on commonly studied NACA 4-digit airfoils (e.g., NACA 44XX) are not the best performing rotors. It can be observed that by increasing the maximum airfoil camber value better aerodynamic efficiencies can be achieved. However, the position of the airfoil maximum camber also has great influence in the rotor efficiency. On the one hand, poor rotor efficiencies were obtained when the position of the maximum camber was located close to the leading edge or close to the trailing edge of the base airfoil. The worst efficiencies were obtained from the NACA 0XXX and the NACA X0XX airfoils, which are symmetrical airfoils. In addition, low efficiencies were obtained from the NACA [7-9][1-2]XX airfoils with the notable example of the NACA 92XX airfoils (note that for visualization purposes the NACA X1XX-based rotors performance metrics were not plotted). This is due to the high asymmetric features they exhibit close to the leading edge which promotes an early flow detachment from the blade surface (stall condition) that prevails over a large range of operational conditions. On the other hand, the best efficiencies were obtained when the position of the maximum camber was in the range between 40% and 60% of the chord length (e.g., NACA [7-9][4-6]XX airfoils).

The thrust force increases with increasing maximum camber value and when the maximum camber is located around 40% to 60% of the chord length. This is consistent with the fact that the rotor efficiency is high when such geometrical features are considered. In addition, it was observed that for most symmetric airfoils the thrust coefficient was slightly lower in contrast with asymmetric airfoils. This is consistent with the fact that lift and drag forces are generally larger for asymmetric airfoils, which results in larger axial forces on the rotor.

The cut-in wind speed decreases with increasing maximum camber. However, since the simple cut-in wind speed model is fully dependent on the geometric distributions (e.g., (8), (9), and (13)), it is not possible to identify explicitly if the position of the maximum camber influences the cut-in wind speed value. However, since the torque coefficient is directly related to the power coefficient, as shown in (6), the position of the maximum camber positively affects the cut-in wind speed when its values are in the range between 40% and 60% of the chord length, as described later in the experimental validation. The obtained results suggest that improvements in both the power coefficient and the SWT rotor cut-in wind speed can be achieved by properly selecting the base airfoil shapes of the SWT blade. This finding contradicts previous work [10], in which it is stated that it is necessary to trade some power-producing capability in order to obtain improvements in starting performance. It is conspicuous that for asymmetric airfoils (e.g., NACA [7-9]5XX) the SA algorithm identifies as best solution WT blades with high solidity, as noted during the calculation of the cut-in wind speed. This behavior is explained since high solidity rotors typically operate at larger effective Reynolds numbers, thus improving the aerodynamic efficiency of the rotor.

Finally, the blade volume decreases with increasing maximum camber. This result can be understood since blades composed of asymmetric airfoils (e.g., NACA 75XX) have lower cross sectional areas. However, the total volume was highly influenced by the resultant airfoil thickness distribution, which resulted in low values for blades having asymmetric airfoils (e.g., the NACA 75XX optimized rotor has a thickness distribution in the range $10 \leq \gamma \leq 15$ [%]). This observed geometrical trend can limit the structural integrity of the WT rotor. Therefore, detailed structural studies must be performed for larger rotors experiencing larger axial forces.

Second Evaluation. For the second evaluation procedure, the hybrid vgGA-SA solution procedure was executed thirty times. The best result among the thirty executions is presented in Figure 7, in which a hypothetical fixed rotational speed of 2,500 RPM was assumed in order to construct the C_P versus λ, the C_T versus λ, and the $K_P = C_P/\lambda^3$ versus $1/\lambda$ performance curves. The hybrid procedure selected and combined different NACA 4-digit airfoils along the WT blade span. For this test case, a total of 100^{17} possible combinations of airfoils along the blade span can be constructed.

Interestingly, the vgGA-SA procedure considered the NACA 6[4-6]XX, the NACA 7[4-6]XX, the NACA 85XX, and the NACA 95XX airfoils in the best found solution, which were also the best performers in the first evaluation procedure. In addition, the vgGA-SA procedure determined nearly constant distributions of the chord length and thickness distributions along the blade span. The obtained high chord length value is attributed to the aerodynamic benefits of operating at larger Reynolds numbers while the low thickness

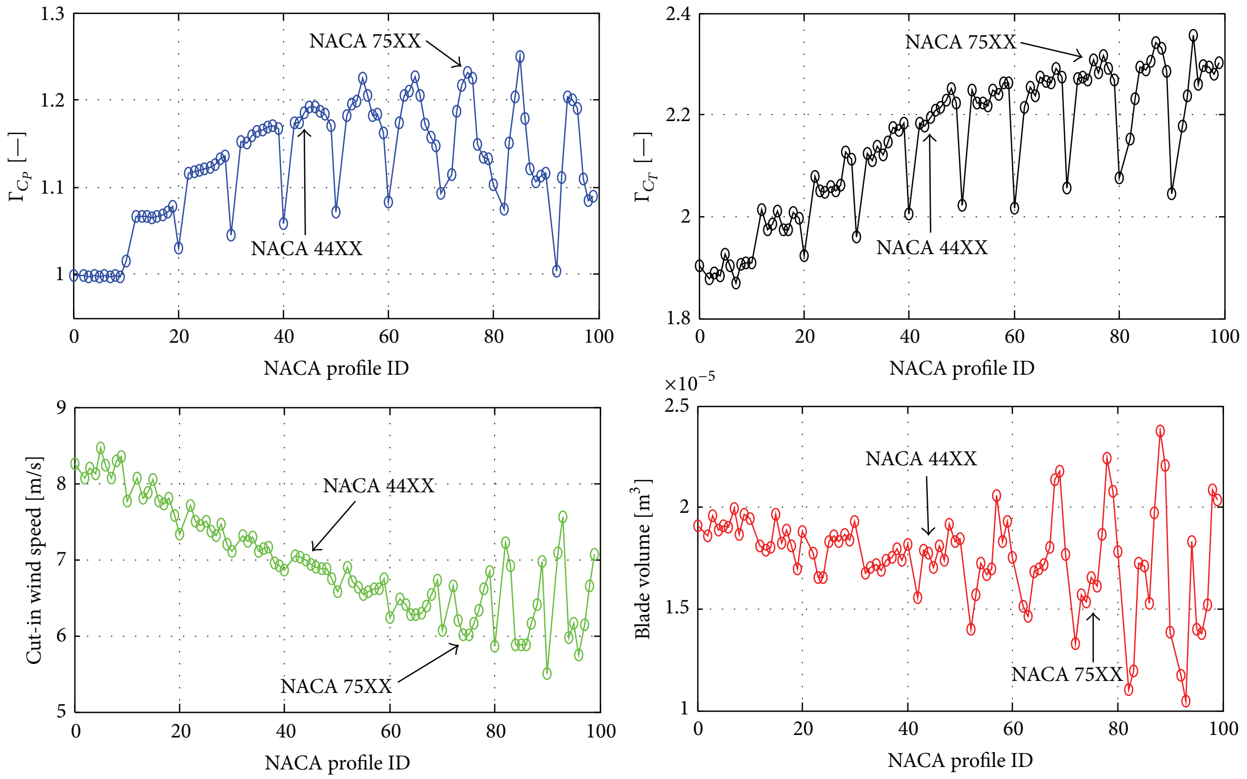


Figure 6: Performance metrics of the optimized NACA 4-digit-based rotors.

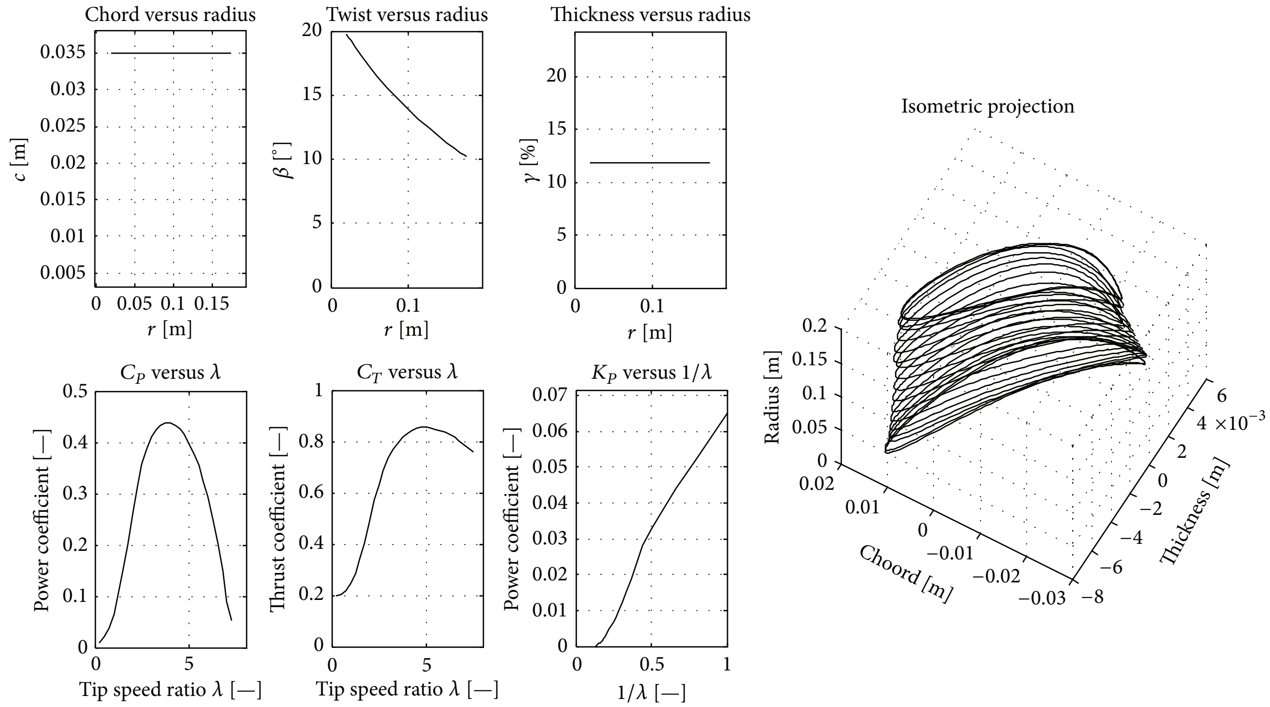


Figure 7: Optimized NACA 4-digit hybrid rotor geometry and aerodynamic performance assuming a fixed rotational speed of 2,500 RPM.

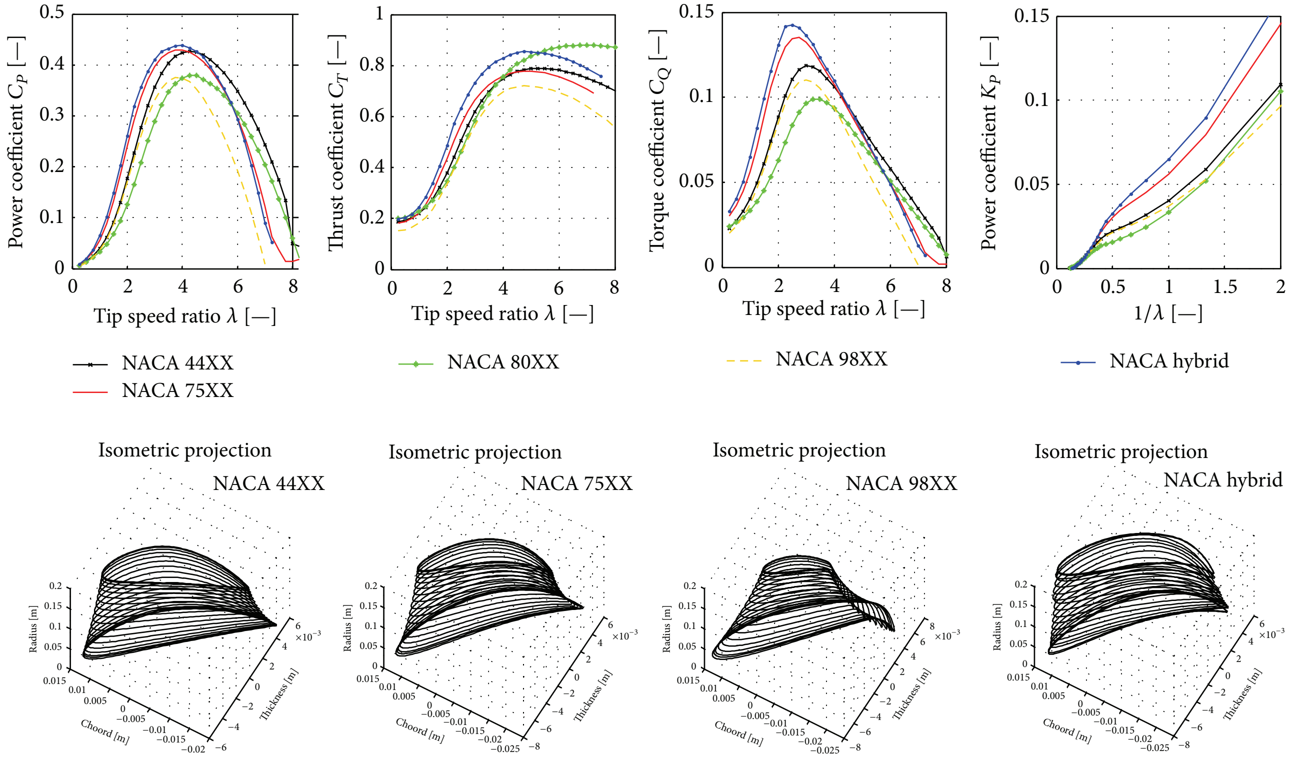


FIGURE 8: Geometry and aerodynamic performance of optimized rotors based on different NACA 4-digit airfoils operating at a fixed rotational speed of 2,500 RPM.

values help reducing the side effects of operating in near stall conditions.

Figure 8 compares the geometry and the aerodynamic performance curves for different optimized NACA 4-Digit rotors. Based on the results shown in Figures 6 and 8, three optimized rotors based on the NACA hybrid, the NACA 44XX, and the NACA 75XX airfoils were manufactured, as described in Section 3, and tested at the CIIIA wind tunnel. Figures 9, 10, and 11 summarize the theoretical and experimental evaluation of the optimized rotors for different inflow conditions and electrical loads (R_{Load}).

From Figures 9–11 several conclusions can be highlighted. The BEM model predicts the rotor aerodynamics reasonably well. However, a slight systematic overprediction of the power coefficient was found at most of the operational conditions. The overprediction was high at low Reynolds numbers and decreased monotonically with increasing Reynolds number. The observed overprediction can be partially attributed to an inadequate estimation of the aerodynamic coefficients by XFOIL 6.96 at low Reynolds numbers [22]. Depending on the operational condition (i.e., the tip speed ratio and electrical load) the efficiency of the WT can fall up to 10% due to Reynolds number effects. The maximum observed power coefficient was 45%, 46.7%, and 47% for the NACA 44XX, the NACA 75XX, and the NACA hybrid rotors, respectively. The theoretical cut-in wind speeds were 6.988 m/s, 6.018 m/s, and 5.848 m/s while the experimental cut-in wind speeds were 8.60 m/s, 6.26 m/s, and 6.03 m/s for the NACA 44XX, the NACA 75XX, and the NACA hybrid rotors, respectively. Additional improvements in the SWT starting performance can be obtained in practice by reducing the generator's shaft static friction, the generator's cogging torque, or by increasing the WT rotor diameter.

The rotor aerodynamics was predicted reasonably well in stall conditions. A metastable zone, related to aerodynamic instabilities, was observed during the transition from stall to nonstall operational states. Depending on the previous operational state of the WT, different operational paths can be observed upon changing the inflow speed condition. The size of the metastable zones increases when increasing the electric load as observed in Figures 7–9. Control systems relying on load-induced stall must account for the adverse effects of aerodynamic instabilities in order to (1) improve the SWT aerodynamic efficiency in low wind regimes and (2) properly regulate the WT under unwanted conditions.

Finally, the BEM model predictions are poor at large tip speed ratios. This is because the validity of the BEM model is limited to operational steady states where global flow detachment is not experienced. Global flow detachment occurs at high axial induction factors (e.g., when the WT rotational speed is large, so the rotor swept area is perceived as a solid disk by the free-stream wind) giving rise to vortex ring states. Empirical relationships are commonly used to model the conditions where the BEM theory breaks down [29, 30]. However, these generic relationships are not always valid and improvements to these expressions should be sought. Global

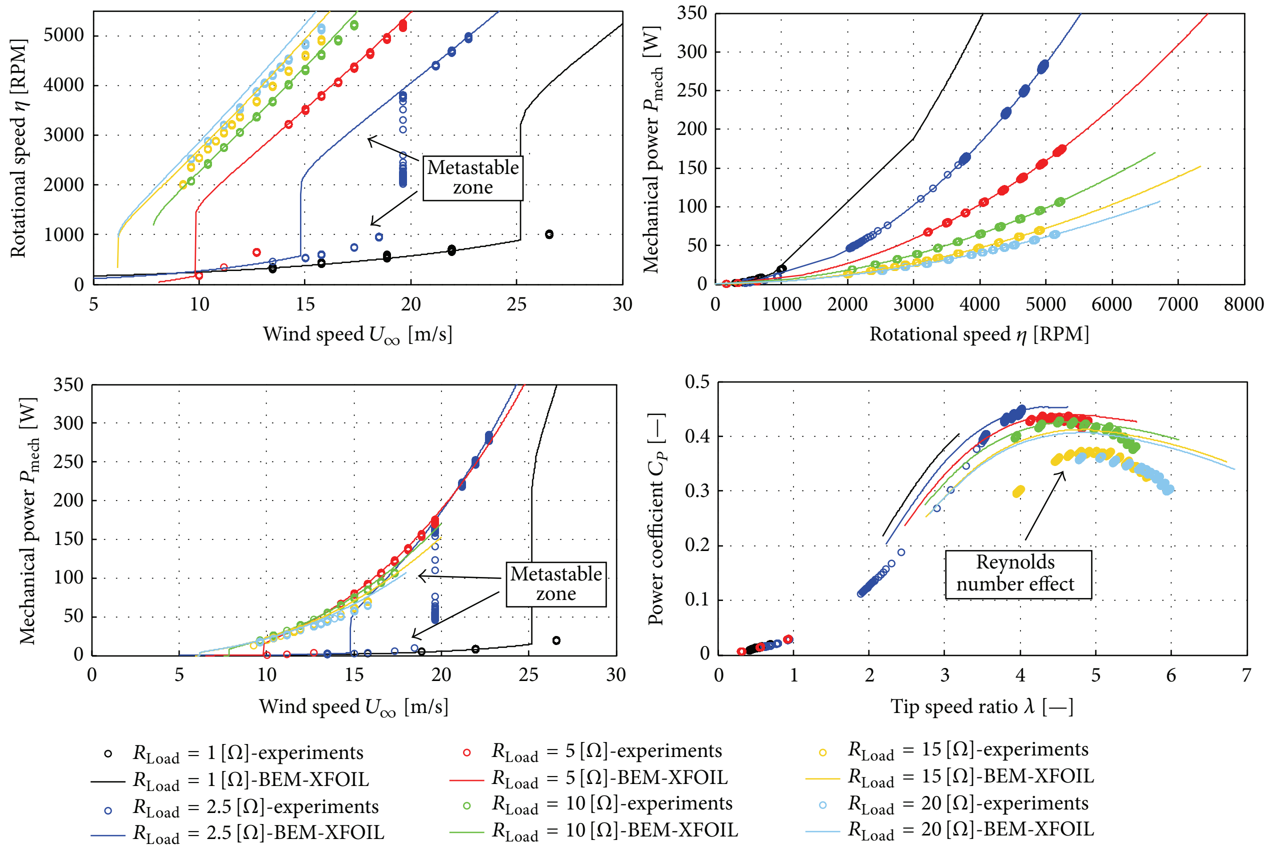


FIGURE 9: Optimized NACA 44XX-based rotor performance.

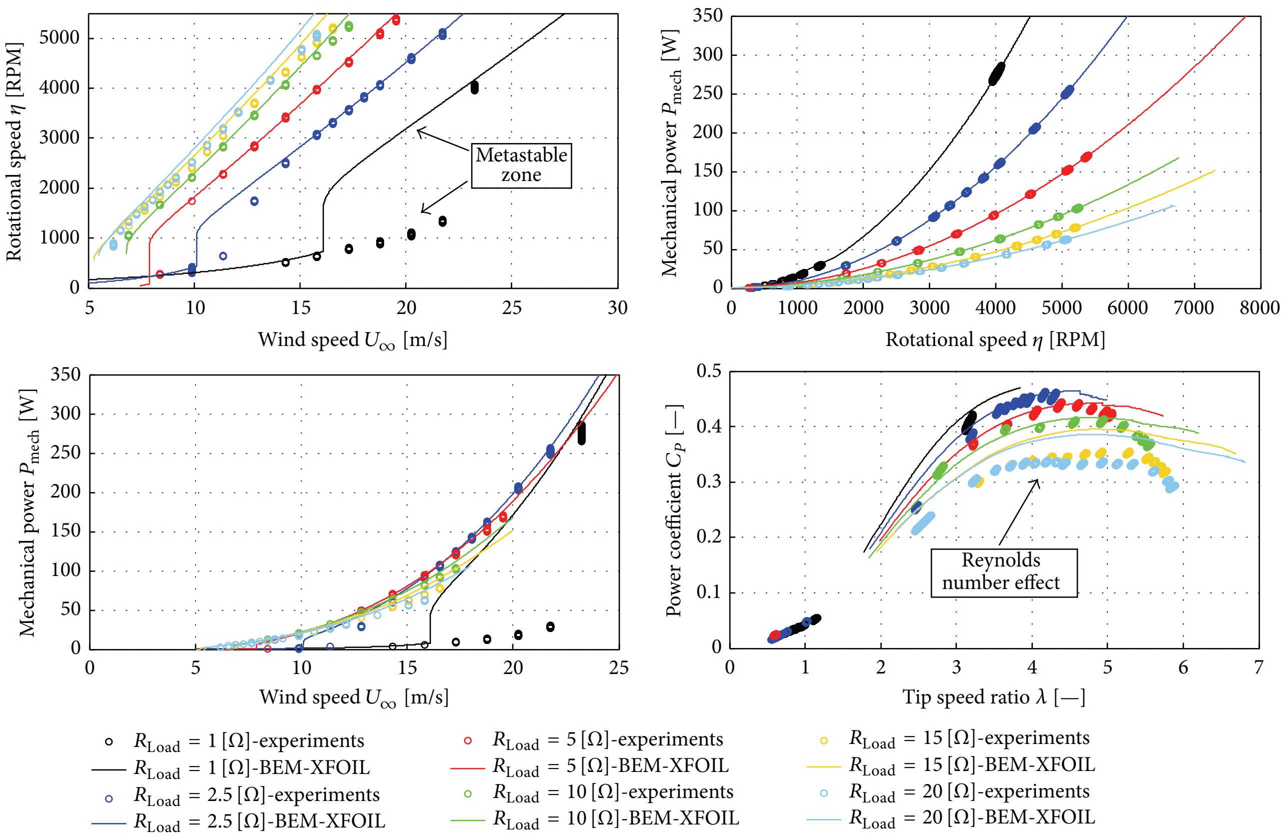


FIGURE 10: Optimized NACA 75XX-based rotor performance.

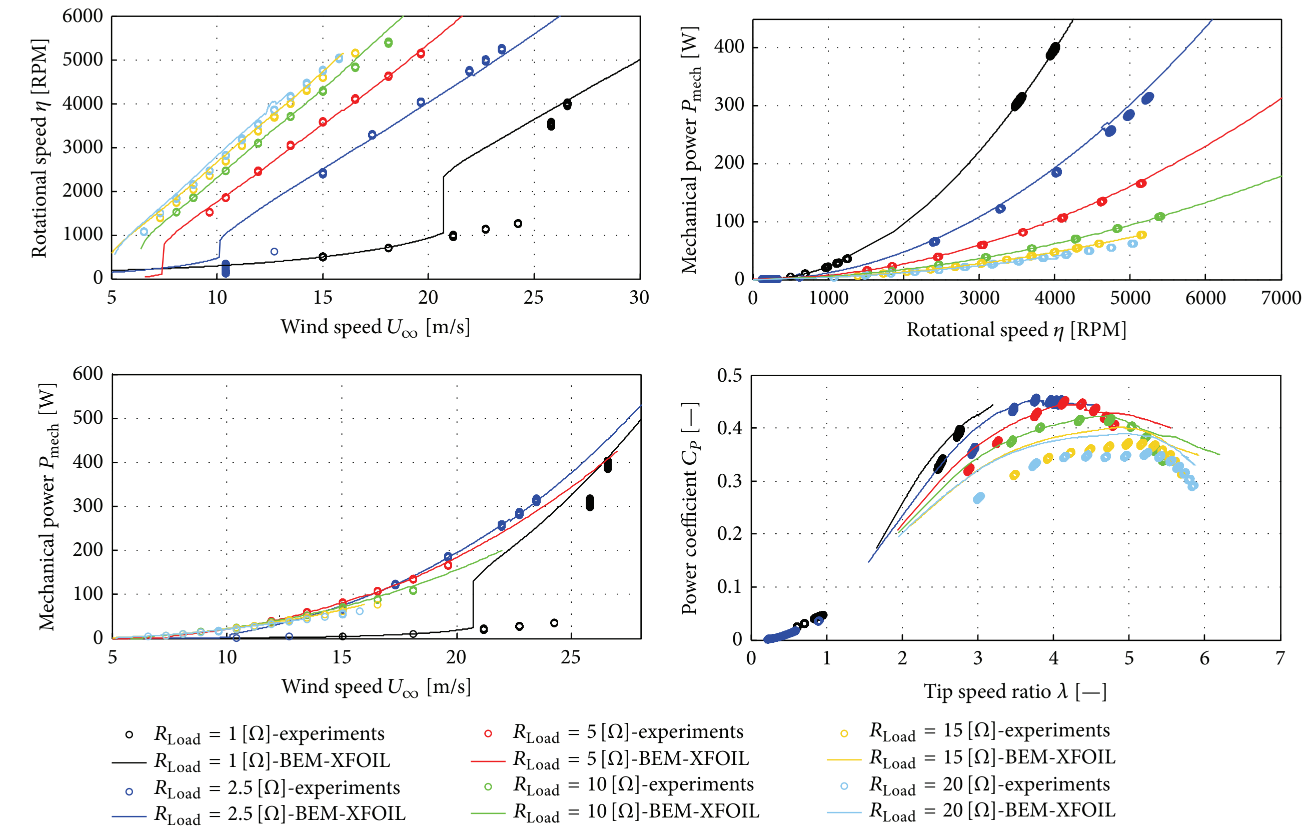


FIGURE 11: Optimized NACA hybrid rotor performance.

flow detachment is also caused by high rotor solidity values. Therefore, SWT rotors must be designed to have moderately high rotor solidity values, since solidity increases the SWT efficiency by increasing the effective Reynolds number as previously discussed and must be designed to operate at lower rotational speeds. Lower rotational speeds can be achieved by considering hybrid rotors based on asymmetric airfoils (e.g., NACA [7-9]5XX) that perform well at low and medium tip speed ratios.

5. Conclusions and Future Research

This paper presented a novel and comprehensive framework for the aerodynamic design of small hybrid wind turbine rotors and proposed a hybrid solution procedure based on two metaheuristics (the genetic algorithm and the simulated annealing algorithm) to solve this complex problem.

The rotor performance was quantified through the implementation of a state-of-the-art blade element momentum model, which was coupled with the XFOIL 6.96 software and the Viterna-Corrigan extrapolation procedure for the determination of the local blade aerodynamics. It was observed that the steady-state blade element momentum model provides accurate predictions of the SWT rotor aerodynamic efficiency as long as stall conditions (observed at low tip speed ratios) or global flow detachment conditions (observed at large tip speed ratios) are not experienced and the supplied airfoil aerodynamics information (lift and drag coefficient as a function of both the Reynolds number and the angle-of-attack) is accurate.

The hybrid solution procedure optimized both the radial geometric distributions of the blade, by using a simulated annealing algorithm, and the selection of the airfoil shape at each of the blade discrete stations, by using the virtual gene genetic algorithm. The objective was to maximize the aerodynamic performance of the rotor for a wide range of tip speed ratios. Two different evaluation procedures were conducted to determine the ability of the proposed hybrid optimization strategy to find high-quality solutions, while considering the entire NACA 4-digit airfoil series as a test case. Several wind tunnel tests were performed to validate the achieved theoretical results. The validation procedure showed good results and demonstrated the suitability of the proposed framework for the development of SWT rotors. From the obtained results, several key geometric characteristics that influence the aerodynamic efficiency of SWT rotors were identified and studied. The findings are believed to be useful for SWT rotor development and increase the feasibility of using SWT in urban areas or at sites with a low wind resource. In addition, they allow WT designers to improve their prototypes with the aim of increasing their profitability prospections.

In addition to the specific objective of proposing a novel framework for the design of SWT rotors, this work studied the aerodynamic performance of the NACA 4-digit airfoil series both theoretically and experimentally with the aim of identifying the best performers for the construction of optimized SWT rotors. The theoretical and experimental results demonstrated that the NACA [7-9][4-6]XX airfoils perform very well for small wind energy applications. During the validation study, significant Reynolds number effects were observed. At low Reynolds numbers the power coefficient can be 5% up to 10% lower depending on the operation regime of the WT. Additionally, it was observed that certain asymmetric features in the airfoil shape geometry (e.g., large values of the maximum camber and the location of the maximum camber at a distance between 40% and 60% from the leading edge of the airfoil) increases the aerodynamic efficiency of the WT rotor. Three major improvements were achieved while using specific NACA 4-digit airfoils: (1) a reduction of the cut-in wind speed (up to 2.57 m/s in contrast with other NACA 4-digit airfoil-based rotors), (2) increased aerodynamic efficiency (the maximum achieved efficiency was 47% at tip speed ratios close to four), and (3) a reduction on the amount of material used in the manufacturing process, in contrast with other NACA 4-digit airfoil-based rotors.

Regarding future lines of research, the BEM model's ability to predict the rotor aerodynamics at low and large tip speed ratios provides room for improvement and a fertile area of inquiry. In the stall regime, advanced computational fluid dynamics (CFD) tools that solve the Reynolds Averaged Navier-Stokes (RANS) equations complemented with turbulence models may provide better predictions of the airfoil aerodynamics, albeit at the cost of increased computational requirements. Improvements to the empirical relationships used to model the global flow detachment states are required in order to improve the predictions of the rotor aerodynamic efficiency at large tip speed ratios.

The optimized rotors did not incorporate technologies such as spoilers, winglets, stall barriers, vortex generators, or flaps. In case of incorporating these technologies, it might be possible to improve even more the SWT aerodynamic efficiency. Research on the performance improvements by the inclusion of these technologies is currently under way. Moreover, other families of airfoils are being tested as well.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Support from the Research Chair for Wind Energy at Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), under contract CAT158 is gratefully acknowledged. Special thanks are due to Dr. Patricia Zambrano and to all staff of the Center for Research and Innovation in Aerospace Engineering (CIIIA) of the Universidad Autónoma de Nuevo León for providing access to their wind tunnel facility and supporting the experimental section of this work.

References

[1] R. Chávez, H. Gómez, J. Herbert, A. Romo, and O. Probst, "Mesoscale modeling and remote sensing for wind energy applications," *Revista Mexicana de Física*, vol. 59, no. 2, p. 114, 2013.

[2] K. Ferrigno, *Challenges and Strategies for Increasing Adoption of Small Wind Turbines in Urban Areas [M.Sc. Thesis]*, Engineering Systems Division, Massachusetts Institute of Technology (MIT), May 2010, http://dspace.mit.edu/handle/1721.1/59240.

[3] D. Wood, *Small Wind Turbines: Analysis, Design, and Application*, Green energy and Technology, Springer, New York, NY, USA, 2011.

[4] N. A. Cencelli, *Aerodynamic optimization of small scale wind turbine blade for low wind speed conditions [M.S. thesis]*, Stellenbosch University, Stellenbosch, South Africa, 2006.

[5] R. W. Vesel, *Optimization of a wind turbine rotor with variable airfoil shape via a genetic algorithm [B.Sc. Thesis]*, Aeronautical and Astronautical Engineering, The Ohio State University, 2009.

[6] D. Perfiliev, *Optimization of wind blade design including its energetic characteristics [M.S. thesis]*, Faculty of Technology, Lappeenranta University of Technology, 2010.

[7] G. Kenway and J. R. R. A. Martins, "Aerostructural shape optimization of wind turbine blades considering site-specific winds," in *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (MAO '08)*, Victoria, Canada, September 2008.

[8] R. S. Amano and R. J. Malloy, "CFD analysis on aerodynamic design optimization of wind turbine rotor blades," *World Academy of Science, Engineering and Technology*, vol. 3, no. 12, pp. 64–69, 2009.

[9] M. Jureczko, M. Pawlak, and A. Mężyk, "Optimisation of wind turbine blades," *Journal of Materials Processing Technology*, vol. 167, no. 2-3, pp. 463–471, 2005.

[10] M. J. Clifton-Smith and D. H. Wood, "Further dual purpose evolutionary optimization of small wind turbine blades," *Journal of Physics: Conference Series*, vol. 75, no. 1, Article ID 012017, 2007.

[11] A. Kusiak and H. Zheng, "Optimization of wind turbine energy and power factor with an evolutionary computation algorithm," *Energy*, vol. 35, no. 3, pp. 1324–1332, 2010.

[12] K. Lee, K. Kim, D. Lee, K. Lee, and J. Park, "Two-step optimization for wind turbine blade with probability approach," *Journal of Solar Energy Engineering, Transactions of the ASME*, vol. 132, no. 3, pp. 0345031–0345035, 2010.

[13] K. Li, M. Smith, M. Vazquez, O. Polat, and I. H. Tuncer, "Aerodynamic shape optimization of wind turbine blades using a parallel genetic algorithm," *Procedia Engineering*, vol. 61, pp. 28–31, 2013.

[14] A. J. Delgado-Gutiérrez, *Numerical and experimental analysis of the furling system of a small wind turbine [M.S. thesis]*, Department of Physics, Engineering Division, Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), 2011, (Spanish).

[15] *International Electrotechnical Commission (IEC) 61400-2 International Standard: Wind Turbines—Part 2: Design Requirements for Small Wind Turbines*, 2nd edition, 2006, http://www.iec.ch.

[16] I. H. Abbott and A. E. Von Doenhoff, *Theory of Wing Sections*, Dover, New York, NY, USA, 1959.

[17] L. Loftin and H. A. Smith, "Aerodynamic characteristics of 15 NACA airfoil sections at seven reynolds numbers from 0.7E6 to 9E6," NACA TN 1945, 1949.

[18] M. S. Selig and B. D. McGranahan, "Wind tunnel aerodynamic tests of six airfoils for use on small wind turbines," *Journal of Solar Energy Engineering, Transactions of the ASME*, vol. 126, no. 4, pp. 986–1001, 2004.

[19] X. Mauclère, *Automatic 2D airfoil generation, evaluation and optimisation using MATLAB and XFOIL [M.S. thesis]*, Department of Mechanical Engineering, Section of Fluid Mechanics, Technical University of Denmark, Lyngby, Denmark, 2009.

[20] J. N. Eastman, W. E. Kenneth, and R. M. Pinkerton, "The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel," Tech. Rep. no. 460, NACA, 1933.

[21] V. Casas, F. Peña, and R. Duro, "Automatic design and optimization of wind turbine blades," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, p. 205, Sydney, Australia, November-December 2006.

[22] P. Bæk and P. Fuglsang, "Experimental detection of transition on wind turbine airfoils," in *Proceedings of the European Wind Energy Conference and Exhibition*, pp. 1628–1652, Marseille, France, March 2009.

[23] W. P. Wolfe and S. S. Ochs, "CFD calculations of S809 aerodynamic characteristics," in *Proceedings of the 35th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-97-0973, Reno, Nev, USA, 1997.

[24] J. Wata, M. Faizal, B. Talu, L. Vanawalu, P. Sotia, and M. R. Ahmed, "Studies on a low Reynolds number airfoil for small wind turbine applications," *Science China Technological Sciences*, vol. 54, no. 7, pp. 1684–1688, 2011.

[25] A. Vardar and I. Alibas, "Research on wind turbine rotor models using NACA profiles," *Renewable Energy*, vol. 33, no. 7, pp. 1721–1732, 2008.

[26] J. Yao, W. Yuan, J. Wang et al., "Numerical simulation of aerodynamic performance for two dimensional wind turbine airfoils," in *Proceedings of the 1st International Conference on Advances in Computational Modeling and Simulation (ACMS '11)*, vol. 31, pp. 80–86, Procedia Engineering, December 2012.

[27] M. Drela, "XFOIL: an analysis and design system for low Reynolds number airfoils," in *Proceedings on Low Reynolds Number Aerodynamics*, Notre Dame, Indiana, 1989.

[28] M. Drela and M. B. Giles, "Viscous- inviscid analysis of transonic and low reynolds number airfoils," *AIAA Journal*, vol. 25, no. 10, pp. 1347–1355, 1987.

[29] J. Martínez, *Aerodynamic and electromechanic model with experimental characterization for the prediction of the power curve of the bergey BWC XL.1 wind turbine [M.Sc. thesis]*, Department of Physics, Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), 2004, (Spanish).

[30] J. Martínez, L. Bernabini, O. Probst, and C. Rodríguez, "An improved BEM model for the power curve prediction of stall-regulated wind turbines," *Wind Energy*, vol. 8, no. 4, pp. 385–402, 2005.

[31] M. Døssing, H. A. Madsen, and C. Bak, "Aerodynamic optimization of wind turbine rotors using a blade element momentum method with corrections for wake rotation and expansion," *Wind Energy*, vol. 15, no. 4, pp. 563–574, 2012.

[32] G. Ausiello, P. Crescenzi, and M. Protasi, "Approximate solution of NP optimization problems," *Theoretical Computer Science*, vol. 150, no. 1, pp. 1–55, 1995.

[33] M. Gendreau and J. Potvin, *Handbook of Metaheuristics*, vol. 146 of *International Series in Operations Research & Management Science*, Springer, 2nd edition, 2010.

[34] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, Chichester, UK, 1989.

[35] P. J. Van Laarhoven and E. H. Aarts, *Simulated Annealing*, Springer, 1987.

[36] M. Valenzuela- Rendón, "The virtual gene genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '03)*, pp. 1457–1468, Chicago, Ill, USA, July 2003.

[37] Center for Research and Innovation in Aerospace Engineering (CIIIA), School of Electric and Mechanical Engineering (FIME), and Universidad Autónoma de Nuevo León (UANL), http://www.fime.uanl.mx/en/investigacion.php.

[38] A. Picard, R. S. Davis, M. Gläser, and K. Fujii, "Revised formula for the density of moist air (CIPM-2007)," *Metrologia*, vol. 45, no. 2, pp. 149–155, 2008.

[39] J. Martínez, A. Morales, O. Probst, A. Llamas, and C. Rodríguez, "Analysis and simulation of a wind-electric battery charging system," *International Journal of Energy Research*, vol. 30, no. 8, pp. 633–646, 2006.

[40] O. Probst, A. Gonzalez, M. Roidl, and A. Llamas, "A small wind-electric system based on an improved version of the ITDG axial flux permanent magnet generator," *Wind Engineering*, vol. 30, no. 5, pp. 385–400, 2006.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 287475, 9 pages
http://dx.doi.org/10.1155/2014/287475

Research Article

Energy-Aware Real-Time Task Scheduling for Heterogeneous Multiprocessors with Particle Swarm Optimization Algorithm

Weizhe Zhang,[1] Hucheng Xie,[1] Boran Cao,[1] and Albert M. K. Cheng[2]

[1] *School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China*
[2] *Department of Computer Science, University of Houston, Houston, TX 77204, USA*

Correspondence should be addressed to Weizhe Zhang; wzzhang@hit.edu.cn

Received 7 April 2014; Accepted 6 August 2014; Published 28 August 2014

Academic Editor: Yang Xu

Energy consumption in computer systems has become a more and more important issue. High energy consumption has already damaged the environment to some extent, especially in heterogeneous multiprocessors. In this paper, we first formulate and describe the energy-aware real-time task scheduling problem in heterogeneous multiprocessors. Then we propose a particle swarm optimization (PSO) based algorithm, which can successfully reduce the energy cost and the time for searching feasible solutions. Experimental results show that the PSO-based energy-aware metaheuristic uses 40%–50% less energy than the GA-based and SFLA-based algorithms and spends 10% less time than the SFLA-based algorithm in finding the solutions. Besides, it can also find 19% more feasible solutions than the SFLA-based algorithm.

1. Introduction

Multiple processing in heterogeneous computing platforms adapts to different types of computing needs. Using multiple processing platforms will improve the system performance and satisfy the increase in energy consumption. However, assigning real-time tasks to a multiprocessor implementation proves to be an NP-hard problem. The problems of real-time task allocation in a heterogeneous environment have been studied extensively in the existing references. However, most of the studies focus on the performance metrics of how to minimize the maximum utilization and these problems can be mapped to the traditional makespan problem [1].

Energy consumption has become a major problem in computer systems; the processor consumes most of the energy, especially in embedded systems, where the excessive energy consumption will cause serious pollution and waste of resources in the natural environment [2, 3]. Therefore, how to reduce processor energy consumption becomes a widespread concern. We need to focus on the problem from reducing the maximum utilization to energy consumption under the premise of meeting the specified task deadlines.

Although it is an NP-hard problem, there are many approximation algorithms for solving the problem of real-time task allocation in a heterogeneous processor environment, including traditional real-time task scheduling algorithms such as deadline-monotonic (DM) algorithm [4], rate-monotonic (RM) algorithm [5], least-laxity-first (LLF) algorithm [6], earliest-deadline-first (EDF) algorithm [5], and linear programming-based (LP) algorithm [7] and the swarm intelligence algorithms such as ant colony optimization (ACO) [8], genetic algorithm (GA) [9–11], and shuffled frog-leaping algorithm (SFLA) [12, 13]. In these studies, most algorithms do not consider energy consumption factors. Besides, the number of feasible solutions and energy saving are in conflict. Therefore, we need to find a new algorithm to solve this multiobjective optimization problem.

The heuristic algorithm in [14] is an adaptive algorithmic structure; it can be used to adapt to a series of relatively wide range of issues. Though many heuristic algorithms exist, the particle swarm optimization (PSO) algorithm emerges as a novel heuristic algorithm in recent years. This algorithm is inspired by the social behavior of a group of migratory birds that try to reach an unknown destination. Each bird is referred to as a particle. Each particle has a fitness value determined by the function to be optimized and a speed that determines their flight direction and distance; then the particle with the current optimal particle to search the optimal solution is chosen in the solution space. Compared with the genetic algorithm, the PSO algorithm has no processes such as reproduction, crossover, and mutation; it is only through simple operation for evolution, which is easy to achieve, and the efficiency is better.

The aim of this work is to propose a new algorithm to solve the problem of real-time task scheduling in a heterogeneous processor environment, under the premise of meeting all task deadlines to reduce energy consumption.

The main objective of the work is as follows.

(1) Formulate the real-time task scheduling problem based on energy awareness and add more constraints. Put the energy consumption as the utility into the constraint condition.

(2) Based on the PSO algorithm, propose one algorithm that can solve the problem of real-time task scheduling based on energy awareness, which can find as many feasible solutions as possible before the specified deadlines and minimize the energy consumption.

(3) Through a series of comprehensive experiments, make a comparison of the proposed algorithm with the existing traditional algorithms and the other heuristic algorithms, to improve the algorithm so as to achieve the purpose of the optimization.

This paper is organized as follows. Section 2 presents the state of the art of the task scheduling problem in multiprocessor platforms. Section 3 formulates the problem of real-time task scheduling in heterogeneous processors based on energy awareness. Overview of the PSO algorithm and the proposed energy-aware real-time task scheduling algorithm based on the PSO algorithm is introduced in Section 4. We analyze the performance and results of the proposed algorithm in Section 5. The final part gives the conclusion and summary and provides directions for future work.

2. Related Works

Baruah [15] has made a study on task scheduling in heterogeneous multiprocessor platforms, and some improvements have been made for the ACO heuristic algorithms and the improved algorithm performs very well in finding the feasible solutions under time constraint. Braun et al. [14] conclude 11 heuristic algorithms that can be applied to task scheduling in heterogeneous multiprocessor platforms to reduce the execution time. However, Braun et al. assume that each task on each machine has accurate execution time and has no time constraints. In the experimental results, the GA has a good performance. So far, there are available heuristic algorithms such as GA, ACO algorithm, SA algorithm, PSO algorithm, and SFLA algorithm [12, 13], and these algorithms have been applied to task scheduling in multiprocessor platforms.

Baruah [16] converts the task scheduling problem in multiprocessor platforms into an ILP problem and proposes an approximate polynomial time algorithm. However, the LP problem has a lot of feasible solutions; a polynomial time algorithm is not guaranteed to find the fundamental solution. Similarly, Leung and Whitehead [4] convert the task scheduling problems in multiprocessor platform, in which tasks can be divided and have priority to LP. However, they believe that each task can be arbitrarily segmented, but this assumption is limited in practice.

Baruah [17] puts forward a polynomial time algorithm for the task scheduling problem in multiprocessor platforms in which tasks are preemptive and transitive. Its purpose is to achieve task scheduling under a series of real-time tasks constraints in heterogeneous processor platforms. However, they ignore the communication overhead and that a task is dividable.

The task scheduling problem in multiprocessor platforms not only needs to solve the increasing number of the feasible solutions but also reduces the energy consumption of each found feasible solution. At present, there are a lot of researchers studying task scheduling in multiprocessor platforms to reduce energy consumption. But, in general, the PSO algorithm has not been used in these subjects.

Cheng et al. [8] propose an improved ACO algorithm in multiprocessor platforms task scheduling which can find sufficient feasible solutions, while satisfying the time constraints. But this algorithm is not PSO related and does not make a certain improvement in energy consumption. Baruah [17] in multiple processing scheduling applications describes a non-ACO algorithm. The algorithm can effectively reduce energy consumption and reduce scheduling time, but the algorithm needs a premise: all the tasks must have the same computation time. In addition, the algorithm is not PSO-based. Aydin and Yang [18] propose the worst-fit-decreasing algorithm in multiprocessor task scheduling to reduce energy consumption and meet the deadline. However, this heuristic algorithm is not a PSO-based algorithm. Zhu et al. [19] put forward the corresponding effective algorithm in multiprocessing task scheduling, but it is not PSO-based algorithm, either.

The evolutionary algorithms thick swarm intelligence optimization algorithm as the goal, such as ACO algorithm, evolution strategy, and GA, can solve the problem of multiobjective combinatorial optimization and obtain a better solution, but the algorithm is complex and has low efficiency. So, looking for a more effective task scheduling and allocation algorithm is very important.

Particle swarm optimization (PSO) algorithm [20, 21] is a new global optimization algorithm, the same with the other swarm intelligence algorithm; all belong to the group of intelligent evolutionary computation technology. Randomly initialize population and then evaluate it according to the fitness function, so as to determine whether to have further search. However, the PSO-based algorithm has no operation such as reproduction, crossover, and mutation, only works through simple arithmetic for evolution, and is simple and easy to achieve. As an important tool of optimization, the PSO-based algorithm can be applied in cloud computing and information retrieval [22, 23].

Table 1: The parameter of the PSO algorithm.

Number	Parameter	Description
1	CP	The current position of the particle
2	BP	The best position of the particle
3	V	The speed of the particle
4	C_1	The learning factor of the particle
5	C_2	The learning factor of the particle
6	ω	The inertia weight of the particle
7	Rand()	The random function between 0 and 1
8	Fitness	The quality evaluation standard of particle

3. Problem Formulation

Each task is assigned to a particular processor and does not exceed any of the computing capacity of the processor without exceeding the deadline of the task. In general, the computation time and deadline for each task are known. But for now, some real-time tasks are dynamically changed. A series of periodic tasks is assigned to the series of heterogeneous processor and does not exceed the deadline. The problem is an NP-hard problem. We solve this problem based on particle swarm optimization (PSO).

3.1. Heterogeneous Multiprocessors Platforms. HMP $= \{P_1, P_2, \ldots, P_m\}$ is a heterogeneous multiprocessor platform. P_j in each clock cycle executes only one command and determines speed according to the type of task. $S_{i,j}$ is the clock frequency and is the speed to perform a specific task T_i. $e_{i,j}$ refers to the execution time of T_i on the P_j, $e_{i,j} = c_i / s_{i,j}$, where c_i is the clock cycles needed for the execution of T_i task.

3.2. Periodic Task Set. PTS $= \{T_1, T_2, \ldots, T_n\}$ consists of n real-time tasks. T_i is made up of a binary group (e, p), where e presents WCET (it is estimated as the worst case execution time); p is the task period. T_i generates an infinite sequence of tasks; each task is at most e time units, and the interval is p time units. The deadline of each T_i is p time units after the arrival of T_i.

3.3. Real-Time Task Scheduling, Energy Utilization, and Energy Consumption. We build a task scheduling situation matrix $X_{n\times m}$ (see Table 2). Matrix element $x_{i\times j}$ indicates whether task T_i can be assigned to processor P_j. The value of element $x_{i\times j}$ is 0 or 1, respectively, which indicates that task T_i is not assigned to the processor P_j and task T_i is already assigned to processor P_j.

The energy consumption matrix in the real-time task scheduling problem on heterogeneous processors is presented by $U_{n\times m}$; its element $u_{i,j}$ is computed as $u_{i,j} = e_{i,j} / p_i$ which shows the energy consumption it takes to execute task T_i on processor P_j. $u_{i,j}$ is a real number whose range will be set $(0, 1) \cup +\infty$; if task T_i cannot run on processor P_j, then $u_{i,j}$ is set $+\infty$.

Table 2: The energy utility matrix of m processors with n tasks.

	P_1	P_2	P_3	$\cdots$	P_{m-1}	P_m
T_1	$u_{1,1}$	$u_{1,2}$	$u_{1,3}$	$\cdots$	$u_{1,m-1}$	$u_{1,m}$
T_2	$u_{2,1}$	$u_{2,2}$	$u_{2,3}$	$\cdots$	$u_{2,m-1}$	$u_{2,m}$
T_3	$u_{3,1}$	$u_{3,2}$	$u_{3,3}$	$\cdots$	$u_{3,m-1}$	$u_{3,m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
T_{n-1}	$u_{n-1,1}$	$u_{n-1,2}$	$u_{n-1,3}$	$\cdots$	$u_{n-1,m-1}$	$u_{n-1,m}$
T_n	$u_{n,1}$	$u_{n,2}$	$u_{n,3}$	$\cdots$	$u_{n,m-1}$	$u_{n,m}$

Energy consumption of T_i in processor P_j on each cycle is as follows:

$$E_{i,j} = \text{Power}_{i,j} \times e_{i,j} \approx \left(C_{ef} \times \frac{s_{i,j}^3}{k^2} \right) \times e_{i,j} = \frac{C_{ef}}{k^2} \times c_i \times s_{i,j}^2, \tag{1}$$

where C_{ef} and k are constants. Thus, $E_{i,j} \propto c_i \times s_{i,j}$, and the energy consumption is linear.

The total energy consumption on the m processors is

$$\text{Energy}_{n\times m} = \sum_{i=1}^{n} \sum_{j=1}^{m} E_{i,j} \times x_{i,j}. \tag{2}$$

Here we define the theoretical maximum energy consumption value as

$$\text{Max\_Energy}_{n\times m} = \sum_{i=1}^{n} \underset{j=1}{\overset{m}{\text{MAX}}} \left(E_{i,j} \right). \tag{3}$$

3.4. Constraint Model. On the basis of the defined energy consumption matrix $U_{n\times m}$, the constraint model of the real-time task scheduling problem in heterogeneous processor is given. The constraint model consists of the following three parts:

(1) $\sum_{j=1}^{m} x_{i,j} = 1$, $(i = 1, 2, 3, \ldots, n)$,

(2) $\sum_{i=1}^{n} x_{i,j} * u_{i,j} \le U$, $(j = 1, 2, 3, \ldots, m)$,

(3) $x_{i,j}$ is either 0 or 1, $(i = 1, 2, 3, \ldots, n;\ j = 1, 2, 3, \ldots, m)$,

where U represents the maximum amount of computation each processor allows and will be set to 1 in our experiments.

3.5. Calculation of the Fitness Function. The fitness function is defined as the ratio of actual energy consumption and theoretical energy consumption. By (1), we assume that C_{ef} and k are constants and are set as 1. The theoretical maximum energy consumption is calculated as MaxEC $= c_i \times s_{i,j}^2$. The actual energy consumption can be computed as dEC $= \sum_{i\le m, j\le n}$ energy$[i][j]$. Thus, the fitness function is defined as follows:

$$\text{fitness} = \frac{\text{dEC}}{\text{MaxEC}}. \tag{4}$$

3.6. Energy-Aware Real-Time Task Scheduling Problem for Heterogeneous Multiprocessors. Given HMP $= \{P_1, P_2, \ldots, P_m\}$ and PTS $= \{T_1, T_2, \ldots, T_n\}$, we name eRTSP energy-aware real-time tasks scheduling problem in heterogeneous processors. eRTSP has two conflict optimization aims. The first one is to look for each task assigned to a specific processor and makes the utilization of each processor that does not exceed its maximum utilization. The second one is the energy consumption, which is to find a feasible solution to minimize the energy consumption on the corresponding processor.

4. PSO Algorithm for Energy-Aware Real-Time Task Scheduling Problem

4.1. Introduction to the PSO Algorithm. The PSO algorithm [21] was first proposed by Eberhart and Shi. It is a kind of evolutionary computation theory. The PSO algorithm is inspired by a social behavior of a group of migrants trying to reach an unknown destination. In the PSO algorithm, each solution is a group of birds and each bird is said to be a particle. All particles have a fitness value which is determined by the function to be optimized and each particle has a speed which determines its flight direction and distance and then the particle searches the optimal solution in solution space with the current optimal particle. The PSO algorithm and GA are both based on the iterative method. A particle is similar to a chromosome in the GA. But unlike the GA, an evolutionary process does not generate new members from the parent member in the PSO algorithm but only changes its own social behavior according to the process of moving towards the destination.

In fact, the PSO algorithm imitates the communication of the birds when they are flying together. Each bird moves towards a certain direction; when in communication, it determines the best position. Therefore, each bird depends on the current position at a particular speed towards the best birds. Then, each bird forms its new location to view their search space and repeats the process until the bird reaches the desired destination. It is important to note that the process also involves the interaction and intelligence in the community, in order to learn from their own experience (local search) and from the surrounding particles experience (global search).

The PSO algorithm is initialized in the initial time for a group of random particles. The ith particle is presented as the position of an S-dimensional space as a point and S is the number of variables. In the entire process of the PSO algorithm, each particle i displays three variables: the current position of the particle CP_i, the best position of the previous iteration of the loop the particle has reached BP_i, and flight speed of the particle V_i. These three variables are represented with a component form as follows:

$$\begin{gathered} \mathrm{CP} = \left(cp_1, cp_2, \ldots, cp_i, \ldots, cp_n\right), \\ \mathrm{BP} = \left(bp_1, bp_2, \ldots, bp_i, \ldots, bp_n\right), \\ V = \left(v_1, v_2, \ldots, v_i, \ldots, v_n\right). \end{gathered} \tag{5}$$

In each time period, the best position CP_g of particles g is calculated as all of the best adaptations. Therefore, each particle updates its own speed V_i to catch up on the best particle g as follows:

$$\begin{aligned} V_i{}' = {} & \omega \times V_i + c_1 \times \mathrm{rand}\,() \times \left(\mathrm{BP}_i - \mathrm{CP}_i\right) \\ & + c_2 \times \mathrm{Rand}\,() \times \left(\mathrm{BP}_g - \mathrm{CP}_i\right). \end{aligned} \tag{6}$$

According to the above formula and making use of the new speed, we update the position of the particle as follows:

$$\mathrm{CP}_i' = \mathrm{CP}_i + V_i', \tag{7}$$

in which $-V_{\max} \le V_i \le V_{\max}$.

We called c_1, c_2 the learning factors which are two constants; rand() and Rand() are two random functions which range in $[0, 1]$; $V_{\max}$ is the maximum velocity limit of the particle; ω is an inertia weight used to affect the current speed. In the formula (6), the second component presents the thought of its current position and the best position. On the other hand, represented by the formula (1), the third component is the cooperation between the particles, comparing the current position of a particle and the best position.

4.2. Applying the PSO Algorithm to eRTSP

4.2.1. Building Energy Matrix and Time-Consuming Matrix. The eRTSP problem can be represented as a bipartite graph. There are two types of nodes: PTS and HMP. A task is mapped to a node of PTS, and a processor is mapped to a node of HMP. If and only if a task can be assigned to the corresponding processor and does not exceed the maximum computing power limit, there is an edge between the two nodes. This assignment consumption directly relates to the energy consumption of the task on the processor.

Therefore, in general, we construct an $n{\times}m$ energy matrix: n represents the n tasks, m represents the m processors, and $u_{i,j}$ is represented by the energy utilization of the task i in the jth processor. Each value of the matrix is set as $(0, 1) \cup +\infty$; if no tasks are assigned on the particular processors, we set the corresponding value of the element in the matrix $+\infty$. Now, we define the constraints: in each row there can only be an element to be visited; accumulated value of the energy of each column cannot exceed 1.

The same as energy consumption matrix, we can build a matrix recording the running time of a task in the corresponding processor. Each element in the matrix $dExeTime[i][j]$ of the execution time is $dExeTime[i][j] =$ *nCycles/nSpeed*.

4.2.2. The Update of the Velocity, Position, and Inertia Weight of the Particle. The velocity of the particles is the critical factor for the positions of the particles. The velocity of the particles will affect the overall convergence of the PSO algorithm and will affect the efficiency of the algorithm's global searching. We consider (6) as a speed profile. The particle's position updates present the next position of the task. As the particle position updates, we have mentioned formula (7) in the third section, $\mathrm{CP}_i' = \mathrm{CP}_i + V_i'$. When $V_i' > 0$, it indicates that it needs to adjust the number of the processor, and then, $\mathrm{CP}_i' = V_i'$; otherwise, the position of the particle remains unchanged; that is, $\mathrm{CP}_i' = \mathrm{CP}_i$.

The parameter ω in the PSO algorithm plays a balanced role in global searching and local searching. And over time, the number of iterations increases gradually while ω linearly reduces. The formula of updating ω is

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} \times t, \tag{8}$$

where t is the number of iterations and $t_{\max}$ is the total number of iterations.

4.2.3. Optimization of the Energy Consumption. When we find a feasible solution by the PSO algorithm, we often need to optimize the feasible solution to achieve the second objective: energy consumption target, that is, forthcoming a feasible solution with high energy consumption through a task assigned to other processor or exchanging their corresponding processor running two tasks to reduce the overall energy consumption.

In the initial state, for a processor if its utilization is greater than 1, we extract the task with maximal energy consumption in the processor, run this task in the processor with the lowest utilization, and compare the utilization of the processor to see whether it is greater than 1. If it is not greater than 1, then the corresponding coordinate of this task is updated.

Thereafter, in accordance with the calculated corresponding local and global optimum position of each task and from formula (2), the speed of the particles has to be updated. Subsequently, we check the speed of particles, in case utilization is less than the upper limit of the maximum utilization of each processor; if the speed V is greater than $V_{\max}$, V will be assigned to $V_{\max}$; if the speed is less than 0, then the speed is set as 0.

In the optimization, first we backup and then analyze the following three cases.

(1) Particle.v > 0 and Particle.v $\leq V_{\max}$. Let *Particle.x* equal *Particle.v* and calculate the corresponding utilization of the processor, in case guaranteed utilization is less than 1, recalculating fitness value. If the energy consumption ratio has been decreased, we modify the original solution and update the value of *Particle.x*. If there is no reduction of the energy consumption ratio, we do not change the original solution.

(2) Particle.x $> x_{\mathrm{up}}$. We will let the value of *Particle.x* be x_{up} and recalculate the corresponding processor utilization of x_{up}, in case guaranteed utilization is less than 1; we recalculate the fitness value and observe whether the energy consumption ratio has been decreased. If declined, we will alter the original plan; if not, we will not change the original plan for the assignment.

(3) Particle.x < 0. The general idea is the same with the second case; we will assign *Particle.x* to 0 and recalculate the utilization of their corresponding processor, in case of utilization is less than 1; we recalculate the fitness value and observe the ratio of the energy consumption to see whether it has been declined; we will change the original plan for the scheduling if so; if not reduced, we will not change the original plan for the assignment.

We assume that if the fitness value does not decrease or remain the same; we quit the iteration and return after iteration 1000 in the PSO algorithm.

4.2.4. PSO Algorithm for eRTSP. See Algorithm 1.

5. Experiment and Result Analysis

In this section, at first, as for the PSO algorithm, we want to determine its parameters in resolving eRTSP. After that, we solve the eRTSP problem with the PSO algorithm and analyze the comparison of the performance of the PSO algorithm, GA, and SFLA in eRTSP with the solution quality and energy consumption.

5.1. Environment of the Experiments

CPU: Intel Core 2 CPU 1.67 GHz.

Cache: 512 KB.

Memory: 2074492 KB.

Operating system: Windows 7.

Development platform: Visual Studio 2003.NET.

We will get the results from a large number of randomly generated problem sets with the PSO algorithm. There are a lot of different situations in problem sets and each issue is initialized as m processors and n tasks.

5.2. The Parameter of the PSO Algorithm. According to the PSO algorithm, there are three parameters ω, c_1, and c_2, which impact the performance of the PSO algorithm (see Table 1). ω denotes the inertia weight heavy; c_1 and c_2 denote the acceleration. The following experiments are set to determine the best combination of the three parameters. The results are shown in Table 4.

As seen from the results in Table 4, the results of the different parameters of the PSO algorithm running the same eRTSP problem are not the same; the parameters for $\omega = 1$, $c_1 = 2$, and $c_2 = 2$ in this group when solving eRTSP get the largest number of feasible solution and its running time is the shortest. Therefore, in the subsequent experiments, we will select the parameter in this group in solving the eRTSP and compare the performance with the GA and SFLA.

5.3. The Comparison of the Results among the PSO Algorithm, GA, and SFLA in eRTSP. To show a wider range of heterogeneous environments, the use of matrix values is varied. For a periodic task T_i, the definition of the task frequency is the average speed of execution of the tasks before deadline and is defined as c_i/p_i. In the PTS, the variance of the frequency of the task is defined as task heterogeneity. In the HMP, for

```
Input: set of HTM and PTS, $\omega$, $c_1$, $c_2$
Output: feasible solution
Initialize the inertia factor $\omega$, learning factor $c_1$, $c_2$
Calculate the eRTSP instances theoretical maximum energy consumption
Randomly generated the corresponding number of particle represents the corresponding number of tasks
While the current number of iterations is less than the set max number of iterations
Do
    Calculated for all processors, fitness fitness, that is, the ratio of the current energy consumption and
     maximum energy consumption;
    Calculated for each processor load does not reach 100%;
    If it reaches 100%
        The task is to re-arrange the largest processor utilization to its share of the minimum utilization
        will not exceed the maximum processor utilization processor;
    Else
        For each task
            calculate the local best position pbest according to a task;
            calculated the best position gbest according to a task;
        End each
    update the speed of the particle;
    Check whether the velocity is negative or exceeds the maximum;
    If the velocity of the particle is greater than $V_{\max}$
        Change speed of the particle to $V_{\max}$;
    If the velocity of the particle is less than 0
        Change the speed to 0;
    Energy Optimization;
    Best fitness of all particles is is gbest calculated as a processor;
    Update the inertia factor $\omega$;
    If the degree of fitness does not change more than 10 times,
        Then exit the iteration loop to obtain the final solution;
EndWhile
```

Algorithm 1: PSO algorithm for eRTSP.

Table 3: The parameter of GA and SFLA.

No.	GA		SFLA	
	Parameter	Value	Parameter	Value
1	Crossover rate	60%	Population size	200.00
2	Mutation rate	40%	Subpopulation size	20.00
3	Population size	200.00	Subpopulation iterations	10.00
4			Total iterations	100.00

Table 4: The value of the parameter of the PSO algorithm.

ω	c_1	c_2	Fes	Time (ms)
0.8	2	2	11	4.30
1.0	2	2	15	3.94
0.4	0	0	12	5.65
0.4	4	0	12	5.16
0.4	0	4	11	4.98
0.4	4	4	12	9.30
0.9	0	0	12	17.75
0.9	0	4	10	7.37
0.9	4	0	10	7.12
0.9	4	4	7	6.21

a given task, the variance of the processing time of each processor is defined as the processor heterogeneity.

The method of generation of the consumption utilization matrix of the task is as follows.

(1) Generate a vector C with n random elements in the range of $[100, 1000]$; its element C_i represents the implementation of clock cycles for each of the tasks T_i.

(2) Construct a vector containing n floating-point type elements T_B; the size of its elements is in the range $[1, \varphi_T]$. Said that the task of the heterogeneity. This vector can also reflect the frequency of the task T_i.

(3) For each column vector S_i, it contains m elements; the size of its elements $[\varphi_T, \varphi_T \times \varphi_B]$ indicates the degree of processor heterogeneity.

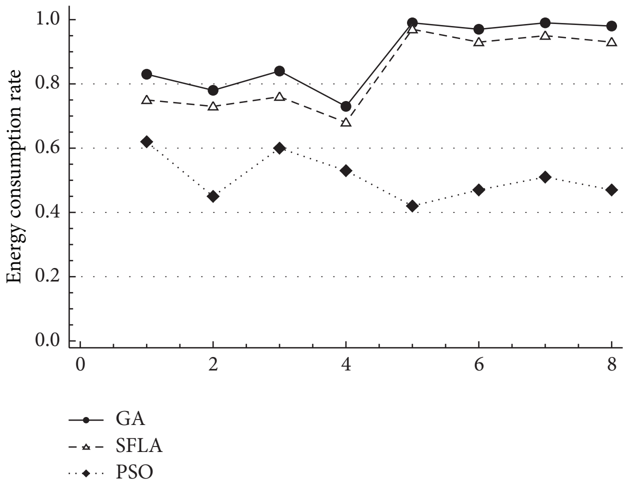


Figure 1: Energy consumption comparison among GA, SFLA, and PSO.

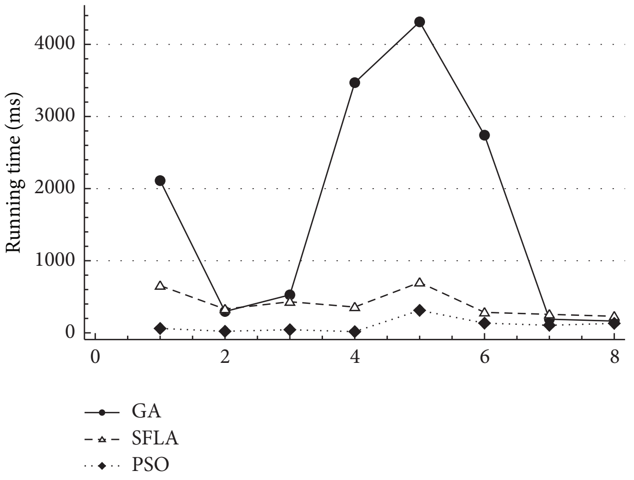


Figure 2: Running time comparison among GA, SFLA, and PSO.

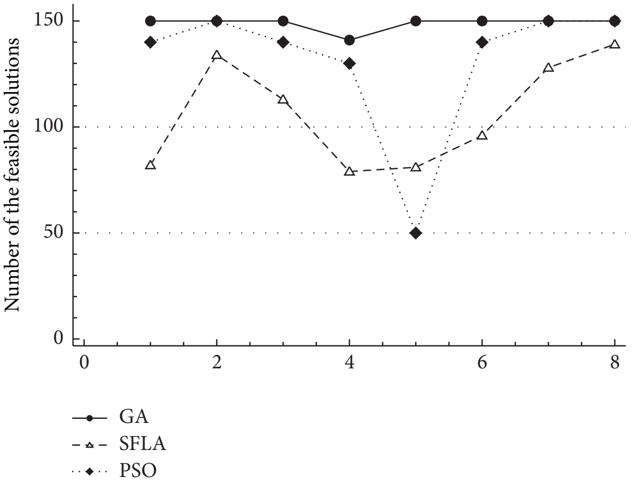


Figure 3: Comparison on searching the number of feasible solutions among GA, SFLA, and PSO algorithms.

Table 5: Eight utilization matrix scales of PSO's parameter test.

No.	Config.	Size
T1	C_HT_HP	U_{90*4}
T2	C_HT_LP	U_{50*8}
T3	C_LT_HP	U_{70*4}
T4	C_LT_LP	U_{40*8}
T5	IC_HT_HP	U_{140*5}
T6	IC_HT_HP	U_{50*8}
T7	IC_LT_HP	U_{90*4}
T8	IC_LT_LP	U_{50*8}

(4) Configure utilization of an $n \times m$ matrix, whose element is $\mathrm{TB}(i)/\mathrm{Si}(j)$. Accordingly, the size of the elements is affected by the degree of task and processor heterogeneity. The element size range is $[0, 1/\varphi_T \times \varphi_B]$.

In order to obtain the true and objective evaluation of the performance of each algorithm, the characteristics of the utilization matrix are task heterogeneity, processor heterogeneity, and consistency. Therefore, we generated a combination of eight kinds of experiments according to the above features: using the matrix of the high and low task heterogeneity, high and low processor heterogeneity, and being consistent or nonconsistent. High task heterogeneity is represented as 100; low task heterogeneity is expressed as 5. Highly heterogeneous processor is represented as 20; low heterogeneous processor is represented as 5. When the processor P_j performs any task shorter than the processor P_k, we use of matrix consistency. A consistent utility matrix is generated by sorting each vector, and P_0 has the fastest processing speed of all the processors and processor $P_{(m-1)}$ is the slowest. In contrast, nonconsistent utility matrix is that processor P_j processes fast on certain tasks than the processor P_k, but the processing speed is slow in the other tasks. It is an unsorted matrix randomly generated. Above 8 experimental category combinations are shown in Table 5.

From Figure 1, on the energy consumption aspect, the energy consumption of the GA and SFLA is higher than the PSO algorithm, wherein the PSO energy consumption using the different test dataset is about 40–50% of the GA and SLFA in energy consumption.

Figure 2 and Table 3 show that the three algorithms running time is different in the same environment of eRTSP; the running time of the GA is the longest, followed by the SFLA. The PSO algorithm is the fastest in 8 groups for the test in general. In particular, we focus on comparing the PSO algorithm and SLFA in the first problem set, finding that the PSO algorithm running time is 10% of the running time of the SFLA. Under consistency utility matrix conditions, the running time of SFLA and the PSO algorithm is essentially the same.

In Figure 3, we compare the three algorithms to find feasible solution volume. The number of feasible solutions found by SFLA and PSO algorithm is less than the GA (see Table 6). The PSO algorithm in the ability to find feasible

Table 6: Average runtime, number of feasible solutions, and energy consumption with GA, SFLA, and PSO algorithm.

No.	GA			SFLA			PSO		
	Avg.	Feas.	Energy	Avg.	Feas.	Energy	Avg.	Feas.	Energy
T1	2110.95	150	0.83	654.33	82	0.75	60.83	140	0.62
T2	296.07	150	0.78	329.54	134	0.73	21.62	150	0.45
T3	526.51	150	0.84	431.03	113	0.76	43.47	140	0.60
T4	3469.31	141	0.73	357.84	79	0.68	16.55	130	0.53
T5	4312.55	150	0.99	699.16	81	0.97	312.04	50	0.42
T6	2741.01	150	0.97	283.86	96	0.93	134.51	140	0.47
T7	189.20	150	0.99	256.37	128	0.95	104.81	150	0.51
T8	163.21	150	0.98	229.73	139	0.93	131.36	150	0.47

solution is slightly worse than the GA. Not all of feasible solutions in the GA in the fourth set of experiments are found; however, in other conditions all are found, but the PSO algorithm has a stronger ability to find the feasible solutions than the SFLA. The feasible solution number of the PSO algorithm is 50% more than that of the SFLA algorithm.

Therefore, from the above results we can get that energy consumption and running time of the PSO algorithm on the eRTSP are relatively small, compared with SFLA; GA and has certain advantages. The GA mainly uses the crossover and mutation method and the running time of looking for a feasible solution is much slower and the energy consumption is larger. Running time with SFLA is slower and the number of feasible solutions is less than the PSO algorithm, and the PSO algorithm can find a feasible solution in most cases. When operating time between SFLA and PSO algorithm is similar, energy consumption with the PSO algorithm uses lower energy consumption than SFLA. Therefore, considering the above several function tests, it can be said that the PSO algorithm outperforms GA and SFLA.

6. Conclusions

This paper has a formal description of the real-time task scheduling problem in a heterogeneous environment based on energy consumption and puts forward a new heuristic algorithm based on the particle swarm optimization algorithm to solve the problem. The proposed algorithm not only finds much more feasible solutions within the specified time but also optimizes the energy consumption. According to the results of extensive experiments, the PSO algorithm has a better performance in reducing energy consumption and running time and increasing the number of feasible solutions. The energy consumption is only 40–50% of GA and SLFA. In addition, in finding the feasible solution volume, the PSO algorithm finds a total of 19% more feasible solutions than SFLA in 7 out of 8 sets of test data and finds about 8% less feasible solutions than GA. At running time, the PSO algorithm is faster than GA and SFLA and about 10% faster than SFLA.

The current study in this paper focuses on independent, nonpreemptive, and periodic tasks, but many other factors are not taken into account in real-time task scheduling problem in a heterogeneous processor environment. In the future, we will reduce the constraint conditions and study the problem of real-time task scheduling with priority and communication between tasks.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant no. 61173145, the National Basic Research Program of China under Grant no. G2011CB302605, and the National High Technology Research and Development Program of China under Grant no. 2011AA010705. Albert M. K. Cheng is supported in part by the US National Science Foundation under Awards nos. 0720856 and 1219082.

References

[1] H. Aydin, R. Melhem, D. Mosse, and P. Mejia-Alvarez, “Dynamic and aggressive scheduling techniques for power-aware real-time systems,” in *Proceedings of the 22nd IEEE Real-Time Systems Symposium (RTSS ’01)*, pp. 95–105, London, UK, December 2001.

[2] V. Tiwari, D. Singh, S. Rajgopal, G. Mehta, R. Patel, and F. Baez, “Reducing power in high-performance microprocessors,” in *Proceedings of the 35th Design Automation Conference*, pp. 732–737, June 1998.

[3] D. M. Brooks, P. Bose, S. E. Schuster et al., “Power-aware microarchitecture: design and modeling challenges for next-generation microprocessors,” *IEEE Micro*, vol. 20, no. 6, pp. 26–44, 2000.

[4] J. Y. Leung and J. Whitehead, “On the complexity of fixed-priority scheduling of periodic, real-time tasks,” *Performance Evaluation*, vol. 2, no. 4, pp. 237–250, 1982.

[5] C. L. Liu and J. W. Layland, “Scheduling algorithms for multiprogramming in a hard-real-time environment,” *Journal of the Association for Computing Machinery*, vol. 20, no. 1, pp. 47–61, 1973.

[6] A. K. Mok, *Fundamental design problems of distributed systems for hard-real-time environments [Ph.D. thesis]*, Massachusetts Institute of Technology, Cambridge, Mass, USA, 1983.

[7] L. Puente-Maury, P. Mejía-Alvarez, and L. E. Leyva-Del-Foyo, "A binary integer linear programming-based approach for solving the allocation problem in multiprocessor partitioned scheduling," in *Proceedings of the 8th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE '11)*, Merida City, Mexico, October 2011.

[8] H. Chen, A. M. K. Cheng, and Y. Kuo, "Assigning real-time tasks to heterogeneous processors by applying ant colony optimization," *Journal of Parallel and Distributed Computing*, vol. 71, no. 1, pp. 132–142, 2011.

[9] G. Menghani, "A fast genetic algorithm based static heuristic for scheduling independent tasks on heterogeneous systems," in *Proceedings of the 1st International Conference on Parallel, Distributed and Grid Computing (PDGC '10)*, pp. 113–117, October 2010.

[10] W. Sun, "A novel genetic admission control for real-time multiprocessor systems," in *Proceedings of the International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT '09)*, pp. 130–137, December 2009.

[11] U. Manchon, C. Ho, S. Funk, and K. Rasheed, "GART: A genetic algorithm based real-time system scheduler," in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC '11)*, pp. 886–893, New Orleans, La, USA, June 2011.

[12] M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *Journal of Water Resources Planning and Management*, vol. 129, no. 3, pp. 210–225, 2003.

[13] L. Xue-hui, Y. Ye, and L. Xia, "Solving TSP with Shuffled Frog-Leaping Algorithm," in *Proceedings of the 8th International Conference on Intelligent Systems Design and Applications (ISDA '08)*, vol. 3, pp. 228–232, November 2008.

[14] T. D. Braun, H. J. Siegel, N. Beck et al., "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," *Journal of Parallel and Distributed Computing*, vol. 61, no. 6, pp. 810–837, 2001.

[15] S. Baruah, "Feasibility analysis of preemptive real-time systems upon heterogeneous multiprocessor platforms," in *Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS '04)*, pp. 37–46, December 2004.

[16] S. Baruah, "Task partitioning upon heterogeneous multiprocessor platforms," in *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '04)*, pp. 536–543, Toronto, Canada, May 2004.

[17] S. K. Baruah, "Partitioning real-time tasks among heterogeneous multiprocessors," in *Proceedings International Conference on Parallel Processing (ICPP '04)*, pp. 467–474, August 2004.

[18] H. Aydin and Q. Yang, "Energy-aware partitioning for multiprocessor real-time systems," in *Proceedings of the International Parallel and Distributed Processing Symposium*, 2003.

[19] D. Zhu, N. AbouGhazaleh, D. Mosse, and R. Melhem, "Power aware scheduling for and/or graphs in multi-processor real-time systems," in *Proceedings of the International Conference on Parallel Processing*, 2002.

[20] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 69–73, IEEE Press, Piscataway, NJ, USA, 1998.

[21] R. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Proceedings of the 7th Annual Conference on Evolutionary Programming*, pp. 611–618, Springer, Berlin, Geramanhy, 1998.

[22] W. Zhang, H. He, and J. Ye, "A two-level cache for distributed information retrieval in search engines," *The Scientific World Journal*, vol. 2013, Article ID 596724, 6 pages, 2013.

[23] W. Zhang, H. He, G. Chen, and J. Sun, "Multiple virtual machines resource scheduling for cloud computing," *Applied Mathematics and Information Sciences*, vol. 7, no. 5, pp. 2089–2096, 2013.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 605890, 7 pages
http://dx.doi.org/10.1155/2014/605890

Research Article

Simulation Analysis of the Long-Term Stability for Frozen Soil Roadbed

Xiaohui Liu,[1] Jianqing Jia,[1,2] and Yibo Zhang[1]

[1] *School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730000, China*
[2] *Cold and Arid Regions Environmental and Engineering Research Institute, Chinese Academy of Sciences, Lanzhou 730000, China*

Correspondence should be addressed to Jianqing Jia; jqjia@mail.lzjtu.cn

Received 9 April 2014; Accepted 6 July 2014; Published 13 August 2014

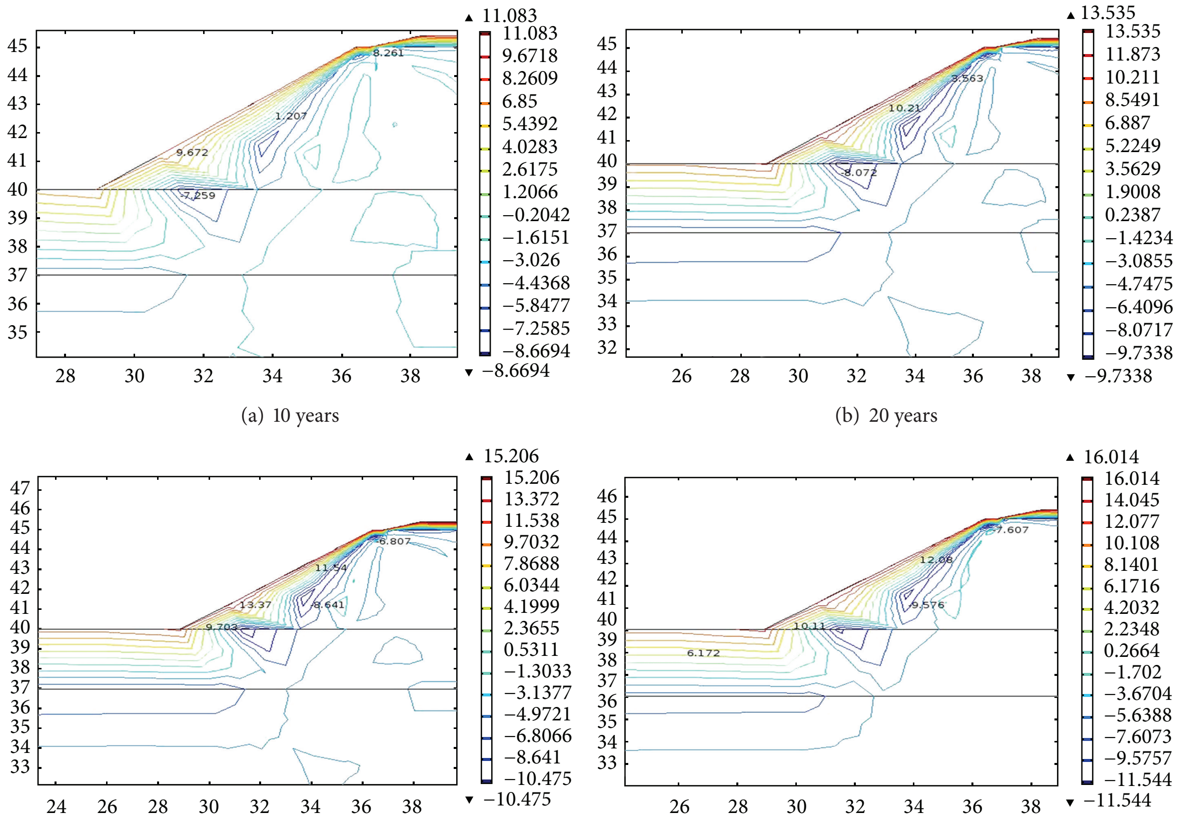
Academic Editor: Ker-Wei Yu

The global warming will lead to rising temperature in Tibetan plateau which will cause some trouble to the long-term stability of frozen soil roadbed. Of course, the temperature is the most important to stability analysis and study of frozen soil roadbed. In this paper, taking the frozen soil roadbed in Tibetan plateau as an example, the numerical simulation model is established. Firstly, the characteristics of temperature fields of frozen soil roadbed in the future 50 years are analyzed, and then the vertical and horizontal displacements without load and under dynamic load are analyzed.

1. Introduction

With the rapid development of society and economy in China, many transportation infrastructures in cold regions have been constructed in recent years. However, the engineering stability in cold regions has gained wide attention, especially the long-term stability. We all know that the temperature in cold region, which can be changed by many factors, such as construction and climate warming, is the most important to engineering stability. In Tibetan plateau, the temperature changes significantly with the global warming. According to the 4th IPCC's report, the global average temperature continues to increase, and the temperature has risen about 0.13°C (0.10~0.16°C) every ten years in the last 50 years which is almost twice that of the past 100 years. Compared with 1850–1899, the total temperature of 2001–2005 increased by 0.76°C (0.57~0.95°C), but the 5th IPPC's report (2013) has said that maybe the global warming is more serious than previous estimation. Assuming that carbon dioxide continues to increase, Qin et al. [1] predicted that the temperature in Tibet plateau will increase by 2.2~2.6°C [2, 3] according to the mode of Chinese regional climate.

In recent years, many scholars have studied the stability of frozen soil roadbed [4–8]. Sun et al. [9] analyzed the deformation characteristics of embankment of the Qinghai-Tibet railway, according to the monitored results of the ground temperature and deformation of 34 monitored sections from 2005 to 2011. Mao et al. [10] put forward the coupling calculation model of the heat-moisture-stress fields based on the control equation of the nonstationary temperature field, the finite element control equation of the moisture movement, and the two-dimensional numerical calculation model of the deformation and stress fields in the subgrade. They found that the temperature field, moisture field, and stress field of the permafrost roadbed were changing all the time, and the redistributing of stress caused by the change of heat and moisture is the key factor to the frost damages. Liu et al. [11] established the stochastic finite element equations for random temperature using the first-order perturbation technique taking into account the random thermal properties and boundary condition, based on heat transfer variational principle. Ma et al. [12] analyzed the deformation and the change of ground temperature of some embankment including the duct-ventilated embankment and the crushed rocks embankment and found that the embankment deformation are not only the behavior of settlement, but also influenced by the change of ground temperature. Zhu et al. [13] put forward a mathematics model and its control equation according to heat transfer, filtration theory, and mechanics of frozen soil. With numerical simulation model established, this paper takes the frozen soil roadbed in Tibetan Plateau as an example. The characteristics of temperature fields of frozen



(a) 10 years

(b) 20 years

(c) 30 years

(d) 50 years

Figure 1: The temperature and time.

soil roadbed in the future 50 years are analyzed, and the vertical and horizontal displacements without load and under dynamic load are analyzed, too.

2. Fundamental Theory

2.1. The Coupling Equations of Temperature Field and Moisture Field. Assuming that there is no dynamic load on roadbed, moisture migration and ice water phase transition are only considered. The heat transport equation can be written as follows [14–16].

In the freezing zone Ω_f is

$$c_f \frac{\partial T}{\partial t} = \operatorname{div}\left[\lambda_f \operatorname{grad}\left(T_f\right)\right] + L\rho_i \frac{\partial \theta_i}{\partial t}. \tag{1}$$

In the unfrozen zone Ω_μ is

$$C_\mu \frac{\partial T_\mu}{\partial t} = \operatorname{div}\left[\lambda_\mu \operatorname{grad}\left(T_\mu\right)\right], \tag{2}$$

where the symbols f and μ, respectively, express frozen state and unfrozen state, T_f, C_f, and λ_f are temperature, volumetric heat capacity, and thermal conductivity of soil in the freezing zone, L is the latent heat of ice water phase transition, θ_i is the ice volume, ρ_i is the ice density, and t is time. Parameters with subscript μ are the corresponding physical components in the unfrozen area Ω_μ, and (1) can be written as

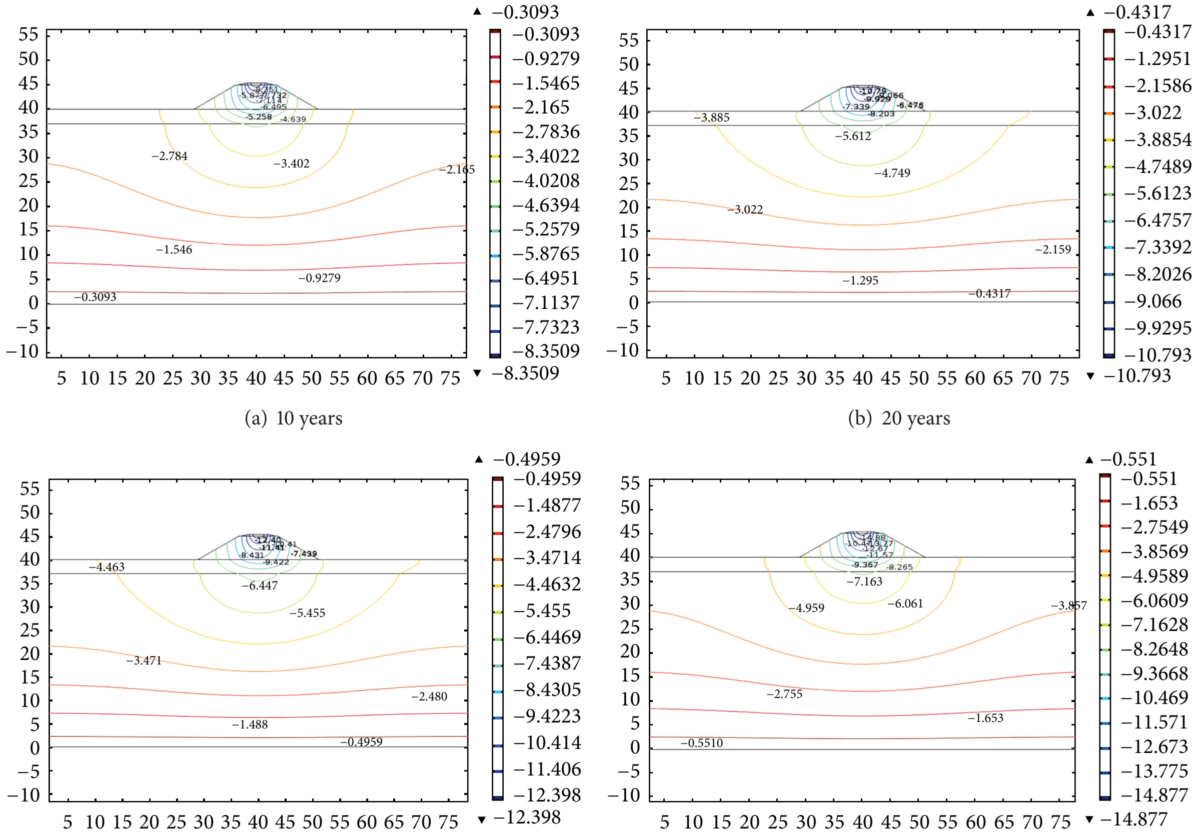
$$\begin{aligned}\left(C_f + L\rho_w \frac{\partial \theta_\mu}{\partial T}\right)\frac{\partial T}{\partial t} &= \operatorname{div}\left[\left(\lambda_f + \frac{kL^2\rho_w^2}{T_k}\right)\operatorname{grad}\left(T_f\right)\right] \\ &\quad + \frac{L\rho_w^2}{\rho_i}\operatorname{div}\left[k^*\operatorname{grad}\left(P_i\right)\right],\end{aligned} \tag{3}$$

where ρ_w is water density, P_i is ice pressure, T_k is absolute temperature, and k^* represents hydraulic conductivity of soil. Not considering the effect of thermal stress in frozen, so

$$C^* \frac{\partial T}{\partial t} = \operatorname{div}\left[\lambda^* \operatorname{grad}\left(T\right)\right], \tag{4}$$

where

$$\begin{aligned} C^* &= \begin{cases} C_f + L\rho_w \dfrac{\partial \theta_\mu}{\partial T} & \text{in } \Omega_f \\ C_\mu & \text{in } \Omega_\mu \end{cases} \\ \lambda^* &= \begin{cases} \lambda_f + \dfrac{kL^2\rho_w^2}{T_k} & \text{in } \Omega_f \\ \lambda_\mu & \text{in } \Omega_\mu. \end{cases}\end{aligned} \tag{5}$$



(a) 10 years

(b) 20 years

(c) 30 years

(d) 50 years

Figure 2: Vertical displacement.

2.2. Stress-Strain Equations

2.2.1. The Differential Equilibrium of Soil. The equilibrium differential equations of representative elemental volume in the soil under the train load can be represented as [14, 16]

$$[\partial]^T \{\sigma\} - \{f\} + \{c\} + \rho \{\ddot{\mu}\} = 0, \tag{6}$$

where $[\partial]$ is differential operator matrix, $[\partial] = \begin{bmatrix} \partial/\partial x & 0 & \partial/\partial y \\ 0 & \partial/\partial y & \partial/\partial x \end{bmatrix}^T$, $\{\sigma\}$ is stress, and $\{\sigma\} = \{\sigma_x \quad \sigma_y \quad \tau_{xy}\}^T$, and ρ is calculated damping force. Equation (6) can be expressed by

$$[C]\{\dot{\mu}\} = (\alpha [M] + \beta [K])\{\dot{\mu}\}, \tag{7}$$

where $[C]$ is damping matrix, $[M]$ is mass matrix, $[K]$ is stiffness matrix, $\{\mu\}$ is displacement vector, $\{\mu\} = \{\mu_x \quad \mu_y\}$, and α and β are damping constants; in this paper, $\alpha = 0.03$, $\beta = 0.01$.

2.2.2. Geometric Equation. The geometric equation can be written as [15]

$$\{\varepsilon\} = -[\partial]\{\mu\}, \tag{8}$$

where $\{\varepsilon\}$ is strain and $\{\varepsilon\} = \{\varepsilon_x \quad \varepsilon_y \quad \tau_{xy}\}^T$.

2.2.3. Physical Equation. Hardin and Drenvich [17] derived the hyperbolic equation under cycle loads which can be written as

$$\tau = \frac{G_0(T)\gamma}{1 + \gamma/\gamma_y(T)}, \tag{9}$$

where $G_0(T)$ is initial shear modulus related to temperature and $\gamma_y(T)$ is strain related to temperature which can be determined by experiment.

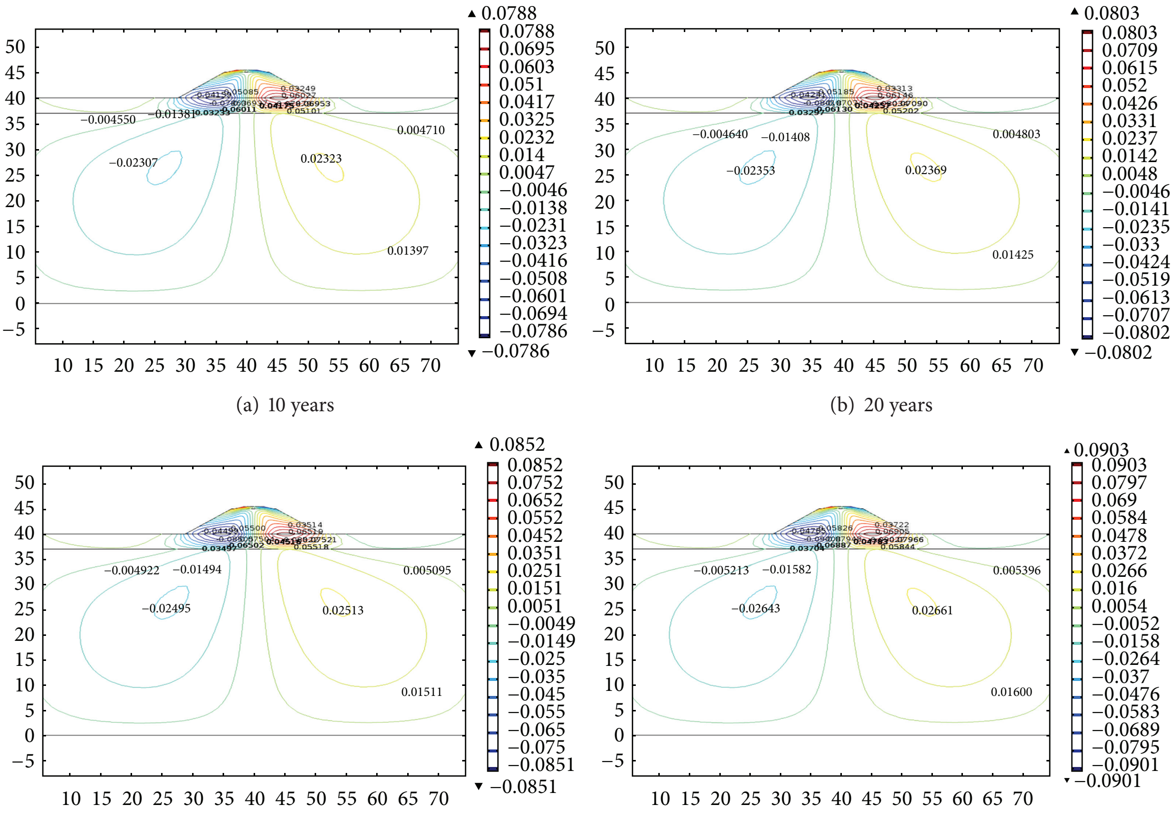
3. Simulation Analysis

3.1. Modeling. In this paper, taking one railway roadbed in Tibetan plateau as an example, the numerical simulation model is established. In the model, lithology from top to bottom is ballast, fill, subclay, and weathered mudstone, the slope ratio is 1 : 1.5, and the road width is 3.4 m [15].

3.2. Boundary Conditions. Considering that the mean annual air temperature will rise 2.6°C in Tibetan plateau in the future 50 years [14], the temperature is expressed as [15–19]

$$T_n = A + B\sin\left(\frac{2\pi}{8760}t_h + \frac{\pi}{2} + \alpha_0\right) + \frac{2.6t_h}{365 \times 24 \times 50}, \tag{10}$$

where A is the annual average temperature and B is the yearly variation temperature.



(a) 10 years

(b) 20 years

(c) 30 years

(d) 50 years

FIGURE 3: Horizontal displacement.

On the natural surface AB and IJ are

$$T_n = -1.0 + 12\sin\left(\frac{2\pi}{8760}t_h + \frac{\pi}{2} + \alpha_0\right) + \frac{2.6t_h}{365\times 24\times 50}. \tag{11}$$

On the roadbed slope BCDE and FGHI are

$$T_p = 1.2 + 13\sin\left(\frac{2\pi}{8760}t_h + \frac{\pi}{2} + \alpha_0\right) + \frac{2.6t_h}{365\times 24\times 50}. \tag{12}$$

At the top of the ballast, the temperature change of the EF is as follows:

$$T_s = 2.0 + 15\sin\left(\frac{2\pi}{8760}t_h + \frac{\pi}{2} + \alpha_0\right) + \frac{2.6t_h}{365\times 24\times 50}. \tag{13}$$

ANM and JKL can be assumed as thermal insulating boundary, and the geothermal heat flux through boundary LM is $q = 0.06\,\mathrm{W{\cdot}m^2}$.

As the mechanical properties of frozen soil are closely connected with temperature, studies have shown that the elastic modulus of permafrost, Poisson's ratio, and shear strength relationship with soil temperature reference strain relations can be expressed by [15, 16]

$$\begin{aligned} E &= \alpha_1 + b_1|T|^m, \\ \upsilon &= \alpha_2 + b_2\,|T|\,, \\ \gamma_r &= \alpha_3 + b_3\,|T|\,, \\ \varphi &= \alpha_4 + b_4\,|T|\,, \end{aligned} \tag{14}$$

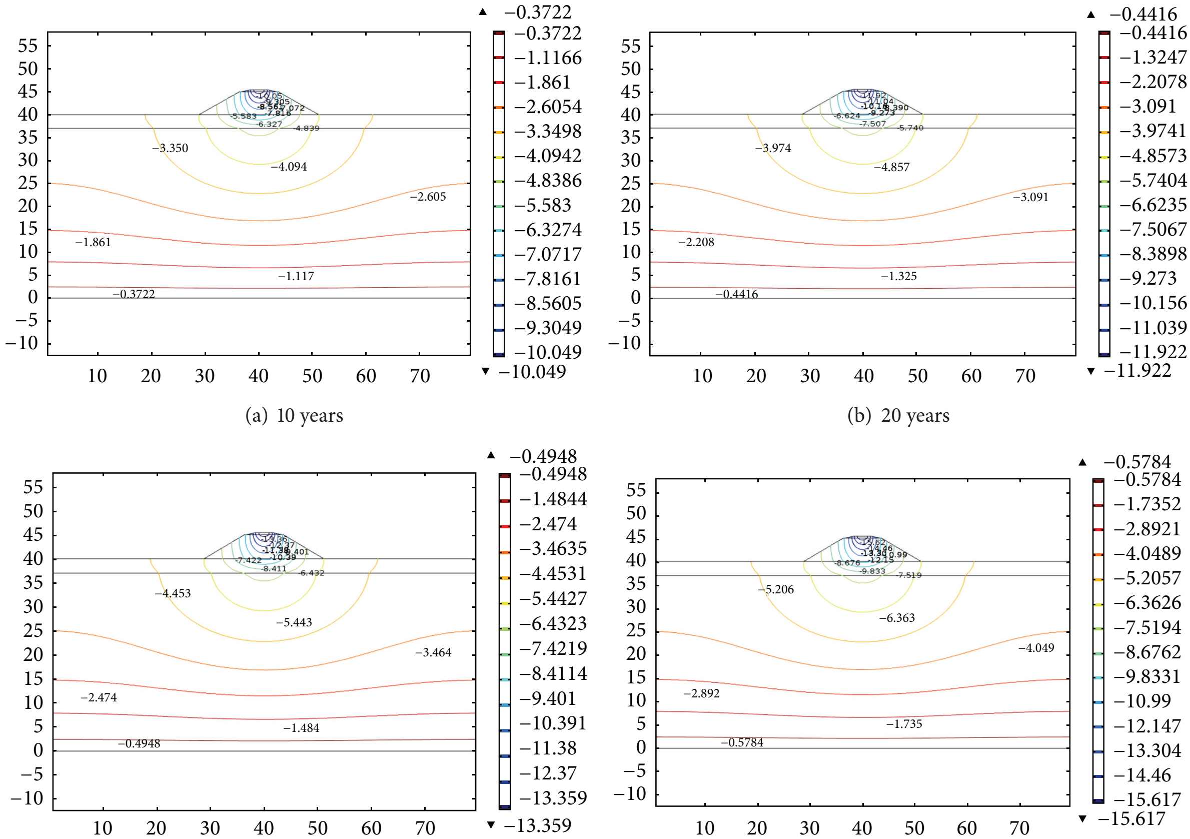
where α_i and b_i are the experimental coefficients, $b_i = 0$ when the soil temperature is higher than 0°C, T is soil temperature, m is nonlinear exponent which is less than 1, and usually $m = 0.6$.

In frozen soil, there is dynamical equilibrium relation between unfrozen water content and negative temperature [20, 21]. The temperature often effects a change of the soil permeability coefficient, and when the temperature decreases, the unfrozen water content decreased, and vice versa. So the unfrozen water content can be defined as

$$W_\mu = \alpha|T|^{-b}, \tag{15}$$

where W_μ is the unfrozen water content and α and b are both constants which are relative to the nature of the soil.

Assuming that the load loading on the roadbed is uniformly distributed load, the total time of train pass is 12 minutes a day and the time interval is equal, and the maximum



(a) 10 years

(b) 20 years

(c) 30 years

(d) 50 years

Figure 4: Vertical displacement.

train load is 70 KN/m. The boundary of ABCDEFG is free boundary, AM and GL are a roller bearing, and ML is fixed constraints boundary.

3.3. Results and Analysis

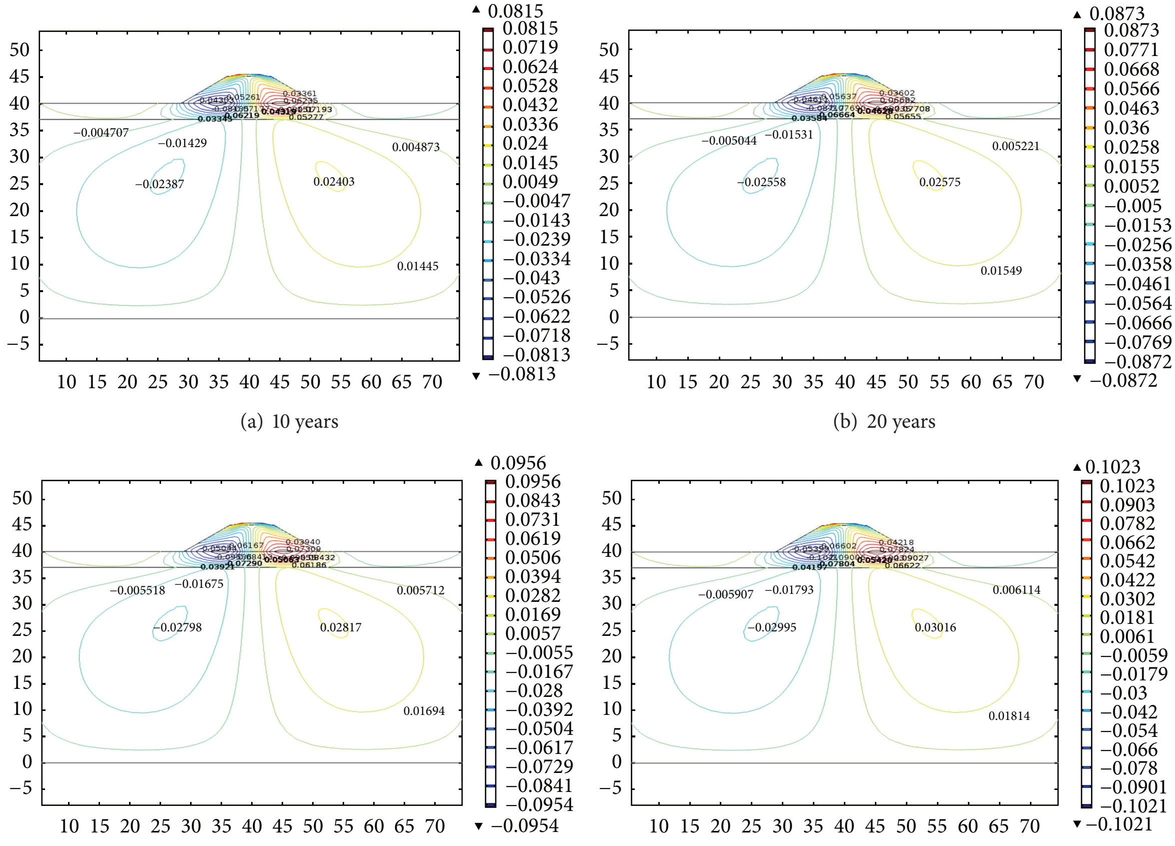
3.3.1. The Temperature Field. Using the boundary conditions from (11) to (13), the temperature fields' distribution in the next 50 years is shown in Figure 1.

In Figure 1 it can be known that the roadbed's temperature will increase as time goes by, and the temperature in the roadbed center is higher than that in the roadbed boundary. In 50 years, the maximum temperature will rise from 11.08°C to 16.01°C, and the temperature at roadbed center point (40, 45) increases from −9.80°C to −6.80°C. Overall, the temperature change from 5.0 m to 6.0 m in the roadbed of the natural surface is large.

3.3.2. Displacement without Load. The vertical displacement and horizontal displacement are, respectively, shown in Figures 2 and 3. From Figure 2, it can be known that the maximum vertical displacement is 10.80 cm after 20 years, and it will increase to 14.88 cm after 50 years. From Figure 3, we can know that the horizontal displacement of the toe of roadbed is larger than others as time goes by; the horizontal displacement of the toe is 8.02 cm after 20 years, and it will increase to 9.01 cm after 50 years.

3.3.3. The Displacement under Load. Assuming that the roadbed surface has the dynamic load that was loaded by the train, the vertical and horizontal displacements are, respectively, shown in Figures 4 and 5.

From the figures we can know that the vertical displacement of the roadbed under load is larger than that without load which is about 1 cm, and the vertical displacement of ballast center is the largest one. The largest vertical displacement of the roadbed surface is 10.80 cm without load 20 years later, and it is 11.92 cm under the dynamic load by the train. When there is no load, the largest vertical displacement of the roadbed surface is 14.88 cm after 50 years; however, it is 15.62 cm under the dynamic load that was loaded by train. It can be seen that the maximum horizontal displacement of the roadbed slope is continuing to increase as time goes by, and the largest horizontal displacement without load is about 0.5 cm larger than that with load. The largest horizontal displacement of the roadbed surface is 8.02 cm without load after 20 years, but it is 8.72 cm under the dynamic load by the train. After 50 years, the largest vertical displacement of the roadbed surface is 9.01 cm without load, and it is 10.21 cm under the dynamic load.



(a) 10 years

(b) 20 years

(c) 30 years

(d) 50 years

Figure 5: Horizontal displacement.

4. Conclusions

(1) The roadbed's temperature will increase year by year, and the temperature in the roadbed center is higher than that in the roadbed boundary. In 50 years, the maximum temperature will rise from 11.08°C to 16.01°C, and the temperature at roadbed center point (40, 45) increases from −9.80°C to −6.80°C. Overall, the temperature change in the roadbed from the natural surface 5.0 m to 6.0 m is large.

(2) When the roadbed surface has no load, the horizontal displacement of the toe of roadbed is larger than others as time goes by. The horizontal displacement of the toe is 8.02 cm after 20 years, and it will increase to 9.01 cm after 50 years. The maximum vertical displacement is 14.88 cm 50 years later.

(3) The vertical displacement of the roadbed under load is larger than that without load which is about 1 cm. The vertical displacement of ballast center is the largest. The largest vertical displacement of the roadbed surface is 14.88 cm without load after 50 years, and it is 15.62 cm under the dynamic load by the train. The maximum horizontal displacement of the roadbed slope is continuing to increase year by year, and the largest horizontal displacement without load is larger about 0.5 cm than that with load. The largest horizontal displacement of the roadbed surface is 9.01 cm without load after 50 years, and it is 10.21 cm under the dynamic load.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by National Natural Science Foundation of China (51104081), Postdoctoral Science Foundation of China (2012M521821), and the Natural Science Foundation of Gansu Province (1010RJZA063).

References

[1] D. H. Qin, Y. H. Ding, and S. W. Wang, "A study of environment change and its impacts in eastern China," *Earth Science Frontiers*, vol. 9, no. 2, pp. 321–328, 2002.

[2] Y. H. Ding, *Projection for the Future Environment in Western China*, Science Press, Beijing, China, 2002.

[3] IPCC, *Climate Change 2007: The Physical Science Basis: Contribution of Working Group I to the Fourth Assessment Report*

of the Intergovernmental Panel on Climate Change, Cambridge University Press, Cambridge, UK, 2007.

[4] M. Y. Zhang, Y. M. Lai, Z. H. Gao, and W. B. Yu, "Influence of boundary conditions on the cooling effect of crushed-rock embankment in permafrost regions of Qinghai-Tibetan Plateau," *Cold Regions Science and Technology*, vol. 44, no. 3, pp. 225–239, 2006.

[5] T. H. Wang, "Analysis of frost heave on subgrade in permafrost regions," *China Journal of Highway and Transport*, vol. 18, no. 2, pp. 1–5, 2005.

[6] G. D. Cheng, "A roadbed cooling approach for the construction of Qinghai-Tibet Railway," *Cold Regions Science and Technology*, vol. 42, no. 2, pp. 169–176, 2005.

[7] Y. Li, L. W. Han, and G. Q. Xu, "Research on stability of embankment in permafrost regions along Qinghai-Tibet Railway and its control," *Journal of Glaciology and Geocryology*, vol. 33, no. 4, pp. 880–883, 2011.

[8] V. G. Kondratyev and V. A. Pozin, *An Introduction to Monitoring System of Engineering-Geocryology for Railway under Construction*, Zabtrans Print Complex, Chita, Russia, 2000.

[9] Z. Z. Sun, W. Ma, H. M. Dang et al., "Characteristics and causes of embankment deformation for Qinghai-Tibet Railway in permafrost regions," *Rock and Soil Mechanics*, vol. 34, no. 9, pp. 2667–2671, 2013.

[10] X. S. Mao, N. Li, B. G. Wang, and C. S. Hu, "Coupling model and numerical simulation of moisture-heat-stress fields in permafrost embankment," *Journal of Chang'an University*, vol. 26, no. 4, pp. 18–62, 2006.

[11] Z. Q. Liu, Y. M. Lai, M. Y. Zhang, and X. F. Zhang, "Numerical analysis for random temperature fields of embankment in cold regions," *Science in China D: Earth Sciences*, vol. 50, no. 3, pp. 404–410, 2007.

[12] W. Ma, D. Liu, and Q. B. Wu, "Monitoring and analysis of embankment deformation in permafrost regions of Qinghai-Tibet Railway," *Rock and Soil Mechanics*, vol. 29, no. 3, pp. 571–579, 2008.

[13] Z. W. Zhu, J. G. Ning, and W. Ma, "Constitutive model and numerical analysis for the coupled problem of water, temperature and stress fields in the process of soil freeze-thaw," *Engineering Mechanics*, vol. 24, no. 5, pp. 138–144, 2007.

[14] W. D. An, W. Ma, Z. W. Wu et al., *Temperature Moisture and Stress Fields in Frozen Soil*, Lanzhou University Press, Lanzhou, China, 1989.

[15] S. Y. Li, Y. M. Lai, M. Y. Zhang, and Y. H. Dong, "Study on long-term stability of Qinghai-Tibet Railway embankment," *Cold Regions Science and Technology*, vol. 57, no. 2-3, pp. 139–147, 2009.

[16] Y. M. Lai, M. Y. Zhang, S. Y. Li et al., *Theory and Application of Cold Regions Engineering*, Science Press, Beijing, China, 2009.

[17] B. O. Hardin and V. P. Drenvich, "Shear modulus and damping in soil: design equation and curves," *Journal of the Soil Mechanics and Foundation Division*, vol. 98, no. 7, pp. 667–692, 1972.

[18] D. H. Qin, *The Comprehensive Evaluating Report on the Environment Evolvement in West China*, Science Press, Beijing, China, 2002.

[19] Z. Q. Liu, Y. M. Lai, X. F. Zhang, and M. Y. Zhang, "Random temperature fields of embankment in cold regions," *Cold Regions Science and Technology*, vol. 45, no. 2, pp. 76–82, 2006.

[20] R. H. Wang and D. W. Li, "Moisture-temperature coupling mathematical model in freezing soil and finite element numerical simulation," *Meitan Xuebao/Journal of China Coal Society*, vol. 31, no. 6, pp. 757–760, 2006.

[21] L. Ma, Q. Wang, and G. H. Yuan, "Analysis and simulation of moisture and heat translocation in seasonal freezing soil in Changchun," *Journal of Liaoning Technical University*, vol. 29, no. 1, pp. 52–55, 2010.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 954040, 8 pages
http://dx.doi.org/10.1155/2014/954040

Research Article

Pairwise Comparison and Distance Measure of Hesitant Fuzzy Linguistic Term Sets

Han-Chen Huang[1] and Xiaojun Yang[2]

[1] *Department of Tourism and MICE, Chung Hua University, Hsinchu 30012, Taiwan*
[2] *Luoyang Electronic Equipment Test Center, Luoyang, Henan 471003, China*

Correspondence should be addressed to Xiaojun Yang; yangxiaojun2007@gmail.com

Received 10 April 2014; Accepted 3 July 2014; Published 4 August 2014

Academic Editor: Ker-Wei Yu

A hesitant fuzzy linguistic term set (HFLTS), allowing experts using several possible linguistic terms to assess a qualitative linguistic variable, is very useful to express people's hesitancy in practical decision-making problems. Up to now, a little research has been done on the comparison and distance measure of HFLTSs. In this paper, we present a comparison method for HFLTSs based on pairwise comparisons of each linguistic term in the two HFLTSs. Then, a distance measure method based on the pairwise comparison matrix of HFLTSs is proposed, and we prove that this distance is equal to the distance of the average values of HFLTSs, which makes the distance measure much more simple. Finally, the pairwise comparison and distance measure methods are utilized to develop two multicriteria decision-making approaches under hesitant fuzzy linguistic environments. The results analysis shows that our methods in this paper are more reasonable.

1. Introduction

Since Zadeh introduced fuzzy sets [1] in 1965, several extensions of this concept have been developed, such as type-2 fuzzy sets [2, 3] and interval type-2 fuzzy sets [4], type-n fuzzy sets [5], intuitionistic fuzzy sets [6, 7] and interval-valued intuitionistic fuzzy sets [8], vague sets [9] (vague sets are intuitionistic fuzzy sets [10]), fuzzy multisets [11, 12], nonstationary fuzzy sets [13], Cloud models [14–18] (Cloud models are similar to nonstationary fuzzy sets and type-2 fuzzy sets), and hesitant fuzzy sets [19, 20]. In the real world, there are many situations in which problems must deal with qualitative aspects represented by vague and imprecise information. So, in these situations, often the experts are more accustomed to express their assessments using linguistic terms rather than numerical values. In [21–23], Zadeh introduced the concept of linguistic variable as "a variable whose values are not numbers but words or sentences in a natural or artificial language." Linguistic variable provides a means of approximate characterization of phenomena which are too complex or too ill defined to be amenable to description in conventional quantitative ways. Since then, fuzzy sets and linguistic variables have been widely used in describing linguistic information as they can efficiently represent people's qualitative cognition of an object or a concept [24]. Thus, linguistic approaches have been so far used successfully in a wide range of applications, such as information retrieval [25–28], data mining [29], clinical diagnosis [30, 31], and subjective evaluation [32–37], especially in decision-making [38–49]. Usually, linguistic terms (words) are represented by fuzzy sets [50], type-2 fuzzy sets [51], interval type-2 fuzzy sets [52–54], 2-tuple linguistic model [40, 55], and so forth. In these linguistic models, an expert generally provides a single linguistic term as an expression of his/her knowledge. However, just as Rodriguez et al. [56] pointed out, the expert may think of several terms at the same time or look for a more complex linguistic term that is not defined in the linguistic term set to express his/her opinion. In order to cope with this situation, they recently introduced the concept of hesitant fuzzy linguistic term sets (HFLTSs) [56] under the idea of hesitant fuzzy sets introduced in [19, 20].

Similarly to a hesitant fuzzy set which permits the membership having a set of possible values, an HFLTS allows an expert hesitating among several values for a linguistic variable. For example, when people assess a qualitative criterion, they prefer to use a linguistic one such as "between medium and very high" which contains several linguistic terms {*medium*, *high*, *veryhigh*}, rather than a single linguistic term. In practical decision-making process, uncertainty and hesitancy are usually unavoidable problems. The HFLTSs can deal with such uncertainty and hesitancy more objectively, and thus it is very necessary to develop some theories about HFLTSs.

Comparisons and distance measures used for measuring the deviations of different arguments are fundamentally important in a variety of applications. In the existing literature, there are a number of studies on distance measures for intuitionistic fuzzy sets [57–60], interval-valued intuitionistic fuzzy sets [61], hesitant fuzzy sets [62, 63], linguistic values [64, 65], and so forth. Nevertheless, an HFLTS is a linguistic term subset, and the comparison among these elements is not simple. In [56], Rodriguez et al. introduced the concept of envelope for an HFLTS and then ranked HFLTSs using the preference degree method of interval values [66]. But, because an HFLTS is a set of discrete linguistic terms, it may seem problematical using the preference degree method for continuous interval to compare these discrete terms of HFLTSs. Up to now, just a few research has been done on the distance measure of HFLTSs [67]. Consequently, it is very necessary to develop some comparison methods and distance measure methods for HFLTSs. In [67], to calculate the distance of two HFLTSs, Liao et al. extend the shorter HFLTS by adding any value in it until it has the same length of the longer one according to the decision-maker's preferences and actual situations. In this paper, we present a new comparison method of HFLTSs based on pairwise comparisons of each linguistic term in the two HFLTSs. Then, a distance measure method based on the pairwise comparison matrix of HFLTSs is proposed without adding any value. Finally, we utilize the comparison method and distance measure method to develop some approaches to solve the multicriteria decision-making problems under hesitant fuzzy linguistic environments.

The rest of the paper is organized as follows. In Section 2, the concepts of hesitant fuzzy sets and HFLTSs are introduced; also the defects of the previous comparison method for HFLTSs are analyzed according to an example. Section 3 describes the comparison and distance measure of HFLTSs based on the proposed pairwise comparison method. In Section 4, a multicriteria decision-making problem is shown to illustrate the detailed processes and effectiveness of two ranking methods which are based on the comparisons and distance measures of HFLTSs, respectively. Finally, Section 5 draws our conclusions and presents suggestions for future research.

2. Preliminaries

2.1. Hesitant Fuzzy Sets. Hesitant fuzzy sets (HFSs) were first introduced by Torra [19] and Torra and Narukawa [20]. The motivation is that when determining the membership degree of an element into a set, the difficulty is not because we have a margin of error (such as an interval) but because we have several possible values.

Definition 1 (see [19]). Let X be a fixed set; a hesitant fuzzy set (HFS) on X is in terms of a function h that when applied to X returns a subset of $[0, 1]$.

To be easily understood, Zhu et al. [68] represented the HFS as the following mathematical symbol:

$$E = \{\langle x, h(x)\rangle \mid x \in X\}, \tag{1}$$

where $h(x)$ is a set of some values in $[0, 1]$, denoting the possible membership degrees of the element $x \in X$ to the set E. Liao et al. [67] called $h(x)$ a hesitant fuzzy element (HFE).

Example 2. Let $h = \{0.2, 0.3, 0.4\}$; then h is an HFE.

Definition 3 (see [69]). For an HFE h, the score function of h is defined as

$$s(h) = \frac{1}{\#h} \sum_{\gamma \in h} \gamma, \tag{2}$$

where $\#h$ is the number of the elements in h.

For two HFEs h_1 and h_2, if $s(h_1) > s(h_2)$, then h_1 is superior to h_2, denoted by $h_1 \succ h_2$; if $s(h_1) = s(h_2)$, then h_1 is indifferent with h_2, denoted by $h_1 \sim h_2$.

Example 4. Assume that we have three HFEs, $h_1 = \{0.2, 0.3, 0.4\}$, $h_2 = \{0.2, 0.35, 0.5\}$, and $h_3 = \{0.3, 0.4\}$; then according to the score function of HFE, (2), and Definition 3, we have $s(h_1) = (0.2 + 0.3 + 0.4)/3 = 0.3$, $s(h_2) = (0.2 + 0.35 + 0.5)/3 = 0.35$, and $s(h_3) = (0.3 + 0.4)/2 = 0.35$. Thus, $s(h_2) = s(h_3) > s(h_1)$; that is, the ranking is $h_2 \sim h_3 \succ h_1$.

The concept of HFS is very useful to express people's hesitancy in daily life. So, since it was introduced, more and more decision-making theories and methods under hesitant fuzzy environment have been developed [56, 62, 63, 67–73].

2.2. Hesitant Fuzzy Linguistic Term Sets. Similarly to the HFS, an expert may hesitate among several linguistic terms, such as "between medium and very high" or "lower than medium," to assess a qualitative linguistic variable. To deal with such situations, Rodriguez et al. [56] introduced the concept of hesitant fuzzy linguistic term sets (HFLTSs).

Definition 5 (see [56]). Suppose that $S = \{s_0, \ldots, s_g\}$ is a finite and totally ordered discrete linguistic term set, where s_i represents a possible value for a linguistic variable. An HFLTS, H_S, is defined as an ordered finite subset of the consecutive linguistic terms of S.

It is required that the linguistic term set S should satisfy the following characteristics:

(1) the set is ordered: $s_i > s_j$, if and only if $i > j$;

(2) there is a negation operator: $\mathrm{Neg}(s_i) = s_{g-i}$.

Example 6. Let S be a linguistic term set, $S = \{s_0 :$ n (nothing), $s_1 :$ vl (very low), $s_2 :$ l (low), $s_3 :$ m (medium), $s_4 :$ h (high), $s_5 :$ vh (very high), and $s_6 :$ p (perfect)$\}$; two different HFLTSs might be $H_S^1 = \{s_1 : \text{vl}, s_2 : \text{l}, s_3 : \text{m}\}$ and $H_S^2 = \{s_3 : \text{m}, s_4 : \text{h}\}$.

Definition 7. One defines the number of linguistic terms in the HFLTS H_S as the cardinality of H_S, denoted by $|H_S|$. In Example 6, $|H_S^1| = 3$ and $|H_S^2| = 2$.

Definition 8 (see [56]). The lower bound H_{S-} and upper bound H_{S+} of the HFLTS H_S are defined as $H_{S-} = \min\{s_i \mid s_i \in H_S\}$ and $H_{S+} = \max\{s_i \mid s_i \in H_S\}$.

Definition 9 (see [56]). The envelope of the HFLTS H_S, $\text{env}(H_S)$, is defined as the linguistic interval $[H_{S-}, H_{s+}] = [\text{Ind}(H_{S-}), \text{Ind}(H_{S+})]$, where Ind provides the index of the linguistic term; that is, $\text{Ind}(s_i) = i$. In Example 6, $\text{env}(H_S^1) = [s_1, s_3] = [1, 3]$ and $\text{env}(H_S^2) = [s_3, s_4] = [3, 4]$.

Based on the definition of envelope, Rodriguez et al. [56] compare two HFLTSs using the comparison method between two numerical intervals introduced by Wang et al. [66].

Definition 10 (see [66]). Letting $A = [a_1, a_2]$ and $B = [b_1, b_2]$ be two intervals, the preference degree of A over B (or $A > B$) is defined as

$$P(A > B) = \frac{\max(0, a_2 - b_1) - \max(0, a_1 - b_2)}{(a_2 - a_1) + (b_2 - b_1)} \quad (3)$$

and the preference degree of B over A (or $B > A$) is defined as

$$P(B > A) = \frac{\max(0, b_2 - a_1) - \max(0, b_1 - a_2)}{(a_2 - a_1) + (b_2 - b_1)}. \quad (4)$$

Example 11. Let $H_S^1 = \{s_1, s_2, s_3\}$, $H_S^2 = \{s_3, s_4\}$, and $H_S^3 = \{s_5, s_6\}$ be three different HFLTSs on S. According to Definition 9, we have $\text{env}(H_S^2) = [1, 3]$, $\text{env}(H_S^2) = [3, 4]$, and $\text{env}(H_S^3) = [5, 6]$. The preference degrees calculated by Definition 10, (3), and (4) are

$$\begin{aligned} P(H_S^1 > H_S^2) &= 0, \qquad P(H_S^2 > H_S^1) = 1; \\ P(H_S^1 > H_S^3) &= 0, \qquad P(H_S^3 > H_S^1) = 1; \\ P(H_S^2 > H_S^3) &= 0, \qquad P(H_S^3 > H_S^2) = 1. \end{aligned} \quad (5)$$

From Example 11 mentioned above, it can be observed that when we compare two HFLTSs using the preference degree method, there exist two defects as follows.

(1) The result $P(H_S^2 > H_S^1) = 1$ indicates that H_S^2 is absolutely superior to H_S^1. In fact, both H_S^1 and H_S^2 contain the linguistic term s_3. It means that the value of a linguistic variable may be equal in these two cases. Thus, it is unreasonable to say that H_S^2 is absolutely superior to H_S^1.

(2) The result $P(H_S^3 > H_S^1) = P(H_S^2 > H_S^1) = 1$, meaning that when compared with H_S^1, the two HFLTSs H_S^2 and H_S^3 are identical. In fact, H_S^3 is more superior to H_S^1 compared to H_S^2 to H_S^1. Thus, using the preference degree method to compare HFLTSs may result in losing some important information.

Based on the analysis mentioned above, we think that it is not suitable to compare discrete linguistic terms in HFLTSs using the comparison method for continuous numerical intervals. By the definition of an HFLTS, we know that every linguistic term in it is a possible value of the linguistic information. And noting that, the two HFLTSs for comparing may have different lengths. So, when comparing two HFLTSs, it needs pairwise comparisons of each linguistic term in them.

3. Comparison and Distance Measure of HFLTSs

3.1. Distance between Two Single Linguistic Terms. Let $s_i, s_j \in S$ be two linguistic terms. Xu [64] defined the deviation measure between s_i and s_j as follows:

$$d(s_i, s_j) = \frac{|i - j|}{T}, \quad (6)$$

where T is the cardinality of S; that is, $T = |S|$.

If only one preestablished linguistic term set S is used in a decision-making model, we can simply consider [49, 65]: $d(s_i, s_j) = |i - j| = s_{|i-j|}$.

Definition 12. Letting $s_i, s_j \in S$ be two single linguistic terms, then we call

$$d(s_i, s_j) = s_i - s_j = i - j \quad (7)$$

the distance between s_i and s_j.

The distance measure between s_i and s_j has a definite physical implication and reflects the relative position and distance between s_i and s_j. If $d(s_i, s_j) = 0$, then $s_i = s_j$. If $d(s_i, s_j) > 0$, then $s_i > s_j$. If $d(s_i, s_j) < 0$, then $s_i < s_j$.

Theorem 13. *Letting $s_i, s_j, s_k \in S$ be three linguistic terms, then*

(1) $d(s_i, s_j) = -d(s_j, s_i)$;

(2) $(|S| - 1) \leq d(s_i, s_j) \leq (|S| - 1)$;

(3) $d(s_i, s_k) = d(s_i, s_j) + d(s_j, s_k)$.

Proof. They are straightforward and thus omitted. □

3.2. Comparison of HFLTSs. The comparison of HFLTSs is necessary in many problems, such as ranking and selection. However, an HFLTS is a linguistic term subset which contains several linguistic terms, and the comparison among HFLTSs is not simple. Here, a new comparison method of HFLTSs, which is based on pairwise comparisons of each linguistic term in the two HFLTSs, is put forward.

Definition 14. Letting H_S^1 and H_S^2 be two HFLTSs on S, then one defines the pairwise comparison matrix between H_S^1 and H_S^2 as follows:

$$C(H_S^1, H_S^2) = [d(s_i, s_j)]_{|H_S^1| \times |H_S^2|}, \quad s_i \in H_S^1, \ s_j \in H_S^2. \quad (8)$$

Remark 15. The number of linguistic terms in the two HFLTSs, H_S^1 and H_S^2, may be unequal; that is, $|H_S^1| \neq |H_S^2|$. To deal with such situations, usually it is necessary to extend the shorter one by adding the stated value several times in it [62, 63], while our pairwise comparison method does not require this step.

Remark 16. From Definition 14, we have $[C(H_S^1, H_S^2)] = -[C(H_S^2, H_S^1)]^T$, where T is the transpose operator of matrix.

Example 17. Let $H_S^1 = \{s_1, s_2, s_3\}$ and $H_S^2 = \{s_2, s_3, s_4, s_5\}$ be two HFLTSs on S. According to Definition 14, the comparison matrix C between H_S^1 and H_S^2 is

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cccc} s_2 & s_3 & s_4 & s_5 \end{array} \\ \begin{bmatrix} -1 & -2 & -3 & -4 \\ 0 & -1 & -2 & -3 \\ 1 & 0 & -1 & -2 \end{bmatrix} \end{array}. \tag{9}$$

Definition 18. Letting $C = C(H_S^1, H_S^2)$ be the pairwise comparison matrix between H_S^1 and H_S^2, the preference relations of H_S^1 and H_S^2 are defined as follows:

$$\begin{aligned} p\left(H_S^1 > H_S^2\right) &= \frac{\left|\sum_{C_{mn}>0} C_{mn}\right|}{\#\{C_{mn} = 0\} + \sum \left|C_{mn}\right|}, \\ p\left(H_S^1 = H_S^2\right) &= \frac{\#\{C_{mn} = 0\}}{\#\{C_{mn} = 0\} + \sum \left|C_{mn}\right|}, \\ p\left(H_S^1 < H_S^2\right) &= \frac{\left|\sum_{C_{mn}<0} C_{mn}\right|}{\#\{C_{mn} = 0\} + \sum \left|C_{mn}\right|}. \end{aligned} \tag{10}$$

It is obvious that $p(H_S^1 > H_S^2) + p(H_S^1 = H_S^2) + p(H_S^1 < H_S^2) = 1$. We say that H_S^1 is superior to H_S^2 with the degree of $p(H_S^1 > H_S^2)$, denoted by $H_S^1 \succ^{p(H_S^1 > H_S^2)} H_S^2$; H_S^1 is equal to H_S^2 with the degree of $p(H_S^1 = H_S^2)$, denoted by $H_S^1 \sim^{p(H_S^1 = H_S^2)} H_S^2$; and H_S^1 is inferior to H_S^2 with the degree of $p(H_S^1 < H_S^2)$, denoted by $H_S^1 \prec^{p(H_S^1 < H_S^2)} H_S^2$.

Considering Example 17, by Definition 18, (10), the preference relations of H_S^1 and H_S^2 were calculated as $p(H_S^1 > H_S^2) = 1/22$, $p(H_S^1 = H_S^2) = 2/22$, and $p(H_S^1 < H_S^2) = 19/22$. Thus, the comparison results are $H_S^1 \succ^{1/22} H_S^2$, $H_S^1 \sim^{2/22} H_S^2$, and $H_S^1 \prec^{19/22} H_S^2$.

3.3. Distance Measure of HFLTSs

Definition 19. Letting $C = C(H_S^1, H_S^2)$ be the pairwise comparison matrix between H_S^1 and H_S^2, the distance between H_S^1 and H_S^2 is defined as the average value of the pairwise comparison matrix:

$$d\left(H_S^1, H_S^2\right) = \frac{1}{\left|H_S^1\right| \times \left|H_S^2\right|} \sum_{m=1}^{|H_S^1|} \sum_{n=1}^{|H_S^2|} C_{mn}. \tag{11}$$

Considering Example 17, one has $d(H_S^1, H_S^2) = (-18)/(3 \times 4) = -1.5$.

To preserve all the given information, the discrete linguistic term set S is extended to a continuous term set $\overline{S} = \{s_\alpha \mid \alpha \in [-q, q]\}$, where q is a sufficiently large positive number. If $s_\alpha \in S$, then we call s_α an original linguistic term; otherwise, we call s_α a virtual linguistic term.

Remark 20. In general, the decision-maker uses the original linguistic terms to express his/her qualitative opinions, and the virtual linguistic terms can only appear in operations.

Definition 21. The average value of an HFLTS H_S is defined as

$$\operatorname{Aver}\left(H_S\right) = \frac{1}{\left|H_S\right|} \sum_{s_i \in H_S} s_i = \frac{1}{\left|H_S\right|} \sum_{s_i \in H_S} \operatorname{Ind}\left(s_i\right). \tag{12}$$

This definition is similar to the score function of an HFE, Definition 3.

Considering Example 17, we have $\operatorname{Aver}(H_S^1) = s_{(1+2+3)/3} = s_2 = 2$, and $\operatorname{Aver}(H_S^2) = s_{(2+3+4+5)/4} = s_{3.5} = 3.5$.

Theorem 22. *Letting H_S be an HFLTS on S, then*

$$0 \leq H_{s-} \leq Aver\left(H_S\right) \leq H_{s+} \leq (|S| - 1). \tag{13}$$

Proof. It is straightforward and thus omitted. □

Theorem 23. *Letting H_S^1 and H_S^2 be two HFLTSs on S, the distance between H_S^1 and H_S^2 defined by the average value of their pairwise comparison matrix is equal to the distance of the two average values of H_S^1 and H_S^2; that is, the distance between H_S^1 and H_S^2 can be easily obtained by*

$$d\left(H_S^1, H_S^2\right) = Aver\left(H_S^1\right) - Aver\left(H_S^2\right). \tag{14}$$

Proof. From Definitions 19 and 14, we have

$$\begin{aligned} d\left(H_S^1, H_S^2\right) &= \frac{1}{\left|H_S^1\right| \times \left|H_S^2\right|} \sum_{m=1}^{|H_S^1|} \sum_{n=1}^{|H_S^2|} C_{mn} \\ &= \frac{1}{\left|H_S^1\right| \times \left|H_S^2\right|} \sum_{s_i \in H_S^1} \sum_{s_j \in H_S^2} d\left(s_i, s_j\right) \\ &= \frac{1}{\left|H_S^1\right| \times \left|H_S^2\right|} \sum_{s_i \in H_S^1} \sum_{s_j \in H_S^2} \left(s_i - s_j\right) \\ &= \frac{1}{\left|H_S^1\right| \times \left|H_S^2\right|} \left(\sum_{s_i \in H_S^1} \sum_{s_j \in H_S^2} s_i - \sum_{s_i \in H_S^1} \sum_{s_j \in H_S^2} s_j \right) \\ &= \frac{1}{\left|H_S^1\right| \times \left|H_S^2\right|} \left(\left|H_S^2\right| \times \sum_{s_i \in H_S^1} s_i - \left|H_S^1\right| \times \sum_{s_j \in H_S^2} s_j \right) \\ &= \frac{1}{\left|H_S^1\right|} \sum_{s_i \in H_S^1} s_i - \frac{1}{\left|H_S^2\right|} \sum_{s_j \in H_S^2} s_j \\ &= \operatorname{Aver}\left(H_S^1\right) - \operatorname{Aver}\left(H_S^2\right), \end{aligned} \tag{15}$$

which completes the proof of Theorem 23. □

Considering Example 17, we have $d(H_S^1, H_S^2) = \text{Aver}(H_S^1) - \text{Aver}(H_S^2) = 2 - 3.5 = -1.5$.

By Theorem 23, we can easily obtain the following corollary.

Corollary 24. *Letting H_S^1, H_S^2 and H_S^3 be three HFLTSs on S, then*

(1) $(|S| - 1) \leq d(H_S^1, H_S^2) \leq (|S| - 1)$;

(2) $d(H_S^1, H_S^2) = -d(H_S^2, H_S^1)$;

(3) $d(H_S^1, H_S^3) = d(H_S^1, H_S^2) + d(H_S^2, H_S^3)$.

Proof. They are straightforward and thus omitted. □

If $d(H_S^1, H_S^2) > 0$ (or $\text{Aver}(H_S^1) > \text{Aver}(H_S^2)$), then we say that H_S^1 is superior to H_S^2 with the distance of $d(H_S^1, H_S^2)$, denoted by $H_S^1 \overset{d(H_S^1,H_S^2)}{\succ} H_S^2$; if $d(H_S^1, H_S^2) = 0$ (or $\text{Aver}(H_S^1) = \text{Aver}(H_S^2)$), then we say that H_S^1 is indifferent to H_S^2, denoted by $H_S^1 \sim H_S^2$; if $d(H_S^1, H_S^2) < 0$ (or $\text{Aver}(H_S^1) < \text{Aver}(H_S^2)$), then we say that H_S^1 is inferior to H_S^2 with the distance of $d(H_S^1, H_S^2)$, denoted by $H_S^1 \overset{d(H_S^1,H_S^2)}{\prec} H_S^2$.

4. Multicriteria Decision-Making Models Based on Comparisons and Distance Measures of HFLTSs

In this section, two new methods are presented for ranking and choice from a set of alternatives in the framework of multicriteria decision-making using linguistic information. One is based on the comparisons and preference relations of HFLTSs and the other is based on the distance measure of HFLTSs. We adopt Example 5 in [56] (Example 25 in our paper) to illustrate the detailed processes of the two methods.

Example 25 ([see [56]). Let $X = \{x_1, x_2, x_3\}$ be a set of alternatives, $C = \{c_1, c_2, c_3\}$ a set of criteria defined for each alternative, and $S = \{s_0$: n (nothing), s_1 : vl (very low), s_2 : l (low), s_3 : m (medium), s_4 : h (high), s_5 : vh (very high), s_6 : p (perfect)$\}$ the linguistic term set that is used to generate the linguistic expressions. The assessments that are provided in such a problem are shown in Table 1 and they are transformed into HFLTSs as shown in Table 2.

4.1. Multicriteria Decision-Making Based on the Comparisons of HFLTSs

Step 1. Considering each criterion c_i $(i = 1, 2, 3)$, calculate the preference degrees between all the alternatives x_j $(j = 1, 2, 3)$.

Considering criterion c_1, $H_S^{x_1} = \{s_1, s_2, s_3\}$, $H_S^{x_2} = \{s_2, s_3\}$ and $H_S^{x_3} = \{s_4, s_5, s_6\}$, so the preference degrees about criterion c_1 calculated using the comparison method of HFLTSs as described in Section 3.2 are $pc_1(x_1 > x_2) = 1/7$, $pc_1(x_1 = x_2) = 2/7$, $pc_1(x_1 < x_2) = 4/7$; $pc_1(x_1 > x_3) = 0/27$, $pc_1(x_1 = x_3) = 0/27$, $pc_1(x_1 < x_3) = 27/27$; $pc_1(x_2 > x_3) = 0/15$, $pc_1(x_2 = x_3) = 0/15$, $pc_1(x_2 < x_3) = 15/15$.

Table 1: Assessments that are provided for the decision problem.

	c_1	c_2	c_3
x_1	Between vl and m	Between h and vh	h
x_2	Between l and m	m	Lower than l
x_3	Greater than h	Between vl and l	Greater than h

Table 2: Assessments transformed into HFLTSs.

	c_1	c_2	c_3
x_1	$\{s_1, s_2, s_3\}$	$\{s_4, s_5\}$	$\{s_4\}$
x_2	$\{s_2, s_3\}$	$\{s_3\}$	$\{s_0, s_1, s_2\}$
x_3	$\{s_4, s_5, s_6\}$	$\{s_1, s_2\}$	$\{s_4, s_5, s_6\}$

Considering criterion c_2, $H_S^{x_1} = \{s_4, s_5\}$, $H_S^{x_2} = \{s_3\}$, and $H_S^{x_3} = \{s_1, s_2\}$, so the preference degrees about criterion c_2 calculated using the comparison method of HFLTSs as described in Section 3.2 are $pc_2(x_1 > x_2) = 3/3$, $pc_2(x_1 = x_2) = 0/3$, $pc_2(x_1 < x_2) = 0/3$; $pc_2(x_1 > x_3) = 12/12$, $pc_2(x_1 = x_3) = 0/12$, $pc_2(x_1 < x_3) = 0/12$; $pc_2(x_2 > x_3) = 3/3$, $pc_2(x_2 = x_3) = 0/3$, $pc_2(x_2 < x_3) = 0/3$.

Considering criterion c_3, $H_S^{x_1} = \{s_4\}$, $H_S^{x_2} = \{s_0, s_1, s_2\}$, and $H_S^{x_3} = \{s_4, s_5, s_6\}$, so the preference degrees about criterion c_3 calculated using the comparison method of HFLTSs as described in Section 3.2 are $pc_3(x_1 > x_2) = 9/9$, $pc_3(x_1 = x_2) = 0/9$, $pc_3(x_1 < x_2) = 0/9$; $pc_3(x_1 > x_3) = 0/4$, $pc_3(x_1 = x_3) = 1/4$, $pc_3(x_1 < x_3) = 3/4$; $pc_3(x_2 > x_3) = 0/36$, $pc_3(x_2 = x_3) = 0/36$, $pc_3(x_2 < x_3) = 36/36$.

Step 2. Aggregate the preference relations using the weighted average method: $p(x_j > x_k) = \text{sum}(w_i \times pc_i(x_j > x_k))$, $p(x_j = x_k) = \text{sum}(w_i \times pc_i(x_j = x_k))$, and $p(x_j < x_k) = \text{sum}(w_i \times pc_i(x_j < x_k))$, where w_i is the weight of criterion c_i, and $\text{sum}(w_i) = 1$. In this paper, $w_i = 1/3$, $i = 1, 2, 3$. Thus, the final preference relations are $(x_1 > x_2) = 15/21$, $p(x_1 = x_2) = 2/21$, $p(x_1 < x_2) = 4/21$; $p(x_1 > x_3) = 1/3$, $p(x_1 = x_3) = 1/12$, $p(x_1 < x_3) = 7/12$; $p(x_2 > x_3) = 1/3$, $p(x_2 = x_3) = 0$, $p(x_2 < x_3) = 2/3$.

Step 3. Rank the alternatives using the nondominance choice degree method as described in [56]. From the results of Step 2, it can be easily obtained that

$$P_D^S = \begin{pmatrix} - & \frac{11}{21} & 0 \\ 0 & - & 0 \\ \frac{1}{4} & \frac{1}{3} & - \end{pmatrix}. \tag{16}$$

Thus, $\text{NDD}_1 = \min\{(1 - 0), (1 - 1/4)\} = 3/4$, $\text{NDD}_2 = \min\{(1 - 11/21), (1 - 1/3)\} = 10/21$, and $\text{NDD}_3 = \min\{(1 - 0), (1 - 0)\} = 1$. Finally, the ranking of alternatives is $x_3 > x_1 > x_2$.

Table 3: Average values of the assessments.

	c_1	c_2	c_3
x_1	2	4.5	4
x_2	2.5	3	1
x_3	5	1.5	5

Table 4: Aggregation results of each alternative.

	x_1	x_2	x_3
Aggregation result	3.5	2.2	3.8

4.2. Multicriteria Decision-Making Based on the Distance Measures of HFLTSs

Step 1. Considering each criterion c_i $(i = 1, 2, 3)$, calculate the average values of HFLTSs for all the alternatives x_j $(j = 1, 2, 3)$. The results are shown in Table 3.

Step 2. Aggregate the average values using the weighted average method. The results are shown in Table 4.

Step 3. Rank the alternatives using the distance measure method. Thus, the ranking of alternatives is $x_3 \overset{0.3}{\succ} x_1 \overset{1.3}{\succ} x_2$.

4.3. Results Analysis. In [56], the ranking of alternatives is $x_1 \succ x_3 \succ x_2$, while both methods in this paper are $x_3 \succ x_1 \succ x_2$. Note that the practical decision-making problem is quite different from other applications where well-established measures can be used to quantify the performance for validation. In decision-making, usually there is no ground truth data or quantitative measures to assess the performance of a method [37]. This is why "plausibility" is used rather than "validation." Here, we analyze the original assessments about each criterion of alternatives x_1 and x_3. Considering criterion c_1, the original assessments of x_1 and x_3 are "between vl and m" and "greater than h," respectively, so it is obviously $x_3 \succ x_1$ about criterion c_1. Considering criterion c_2, the original assessments of x_1 and x_3 are "between h and vh" and "between vl and l," respectively, so this time $x_1 \succ x_3$. Considering criterion c_3, the original assessments of x_1 and x_3 are "h" and "greater than h," respectively, so $x_3 \succ x_1$ again. Summarily, $x_3 \succ x_1$ occurs twice, while $x_1 \succ x_3$ only once. Thus, we believe that our result is more plausible.

5. Conclusion

The comparison and distance measure of HFLTSs are fundamentally important in many decision-making problems under hesitant fuzzy linguistic environments. From an example, we found that there existed two defects when comparing HFLTSs using the previous preference degree method. By analyzing the definition of an HFLTS, a new comparison method based on pairwise comparisons of each linguistic term in the two HFLTSs has been put forward. This comparison method does not need the assumption that the values in all HFLTSs are arranged in an increasing order and two HFLTSs have the same length when comparing them. Then, we have defined a distance measure method between HFLTSs based on pairwise comparisons. Further, we have proved that this distance is equal to the distance of the average values of HFLTSs, which makes the distance measure much simpler. Finally, two new methods for multicriteria decision-making in which experts provide their assessments by HFLTSs have been proposed. The encouraging results demonstrate that our methods in this paper are more reasonable.

In the future, the application of HFLTSs to group decision-making problems will be explored. We will also investigate how to obtain the weights of criteria under hesitant fuzzy linguistic environments.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[2] M. Mizumoto and K. Tanaka, "Some properties of fuzzy sets of type 2," *Information and Control*, vol. 31, no. 4, pp. 312–340, 1976.

[3] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, 2002.

[4] J. M. Mendel and H. Wu, "Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: part 1, forward problems," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 781–792, 2006.

[5] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, NY, USA, 1980.

[6] K. T. Atanassov, "Intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 87–96, 1986.

[7] K. T. Atanassov, "More on intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 33, no. 1, pp. 37–45, 1989.

[8] K. T. Atanassov and G. Gargov, "Interval valued intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 31, no. 3, pp. 343–349, 1989.

[9] W. Gau and D. J. Buehrer, "Vague sets," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 2, pp. 610–614, 1993.

[10] H. Bustince and P. Burillo, "Vague sets are intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 79, no. 3, pp. 403–405, 1996.

[11] R. R. Yager, "On the theory of bags," *International Journal of General Systems*, vol. 13, no. 1, pp. 23–37, 1986.

[12] S. Miyamoto, "Remarks on basics of fuzzy sets and fuzzy multisets," *Fuzzy Sets and Systems*, vol. 156, no. 3, pp. 427–431, 2005.

[13] J. M. Garibaldi, M. Jaroszewski, and S. Musikasuwan, "Nonstationary fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 4, pp. 1072–1086, 2008.

[14] D. Li, J. Han, X. Shi, and M. C. Chan, "Knowledge representation and discovery based on linguistic atoms," *Knowledge-Based Systems*, vol. 10, no. 7, pp. 431–440, 1998.

[15] D. Li, C. Liu, and W. Gan, "A new cognitive model: cloud model," *International Journal of Intelligent Systems*, vol. 24, no. 3, pp. 357–375, 2009.

[16] X. Yang, L. Zeng, F. Luo, and S. Wang, "Cloud hierarchical analysis," *Journal of Information & Computational Science*, vol. 7, no. 12, pp. 2468–2477, 2010.

[17] X. Yang, L. Zeng, and R. Zhang, "Cloud Delphi method," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 20, no. 1, pp. 77–97, 2012.

[18] X. Yang, L. Yan, and L. Zeng, "How to handle uncertainties in AHP: the Cloud Delphi hierarchical analysis," *Information Sciences*, vol. 222, pp. 384–404, 2013.

[19] V. Torra, "Hesitant fuzzy sets," *International Journal of Intelligent Systems*, vol. 25, no. 6, pp. 529–539, 2010.

[20] V. Torra and Y. Narukawa, "On hesitant fuzzy sets and decision," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1378–1382, Jeju Island, Korea, August 2009.

[21] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Information Sciences*, vol. 3, pp. 199–249, 1975.

[22] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—II," *Information Sciences*, vol. 4, pp. 301–357, 1975.

[23] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—III," *Information Sciences*, vol. 9, no. 1, pp. 43–80, 1975.

[24] J. Ma, D. Ruan, Y. Xu, and G. Zhang, "A fuzzy-set approach to treat determinacy and consistency of linguistic terms in multi-criteria decision making," *International Journal of Approximate Reasoning*, vol. 44, no. 2, pp. 165–181, 2007.

[25] G. Bordogna and G. Pasi, "A fuzzy linguistic approach generalizing boolean information retrieval: a model and its evaluation," *Journal of the American Society for Information Science*, vol. 44, no. 2, pp. 70–82, 1993.

[26] G. Bordogna and G. Pasi, "An ordinal information retrieval model," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, supplement 1, 2001.

[27] J. Kacprzyk and S. Zadrozny, "Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools," *Information Sciences*, vol. 173, no. 4, pp. 281–304, 2005.

[28] E. Herrera-Viedma and A. G. López-Herrera, "A model of an information retrieval system with unbalanced fuzzy linguistic information," *International Journal of Intelligent Systems*, vol. 22, no. 11, pp. 1197–1214, 2007.

[29] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*, Springer, Berlin, Germany, 2004.

[30] R. Degani and G. Bortolan, "The problem of linguistic approximation in clinical decision making," *International Journal of Approximate Reasoning*, vol. 2, no. 2, pp. 143–162, 1988.

[31] E. Sanchez, "Truth-qualification and fuzzy relations in natural languages, application to medical diagnosis," *Fuzzy Sets and Systems*, vol. 84, no. 2, pp. 155–167, 1996.

[32] H. Lee, "Applying fuzzy set theory to evaluate the rate of aggregative risk in software development," *Fuzzy Sets and Systems*, vol. 79, no. 3, pp. 323–336, 1996.

[33] L. Martínez, "Sensory evaluation based on linguistic decision analysis," *International Journal of Approximate Reasoning*, vol. 44, no. 2, pp. 148–164, 2007.

[34] L. Martínez, J. Liu, D. Ruan, and J. Yang, "Dealing with heterogeneous information in engineering evaluation processes," *Information Sciences*, vol. 177, no. 7, pp. 1533–1542, 2007.

[35] J. Lu, Y. Zhu, X. Zeng, L. Koehl, J. Ma, and G. Zhang, "A linguistic multi-criteria group decision support system for fabric hand evaluation," *Fuzzy Optimization and Decision Making*, vol. 8, no. 4, pp. 395–413, 2009.

[36] R. de Andrés, M. Espinilla, and L. Martínez, "An extended hierarchical linguistic model for managing integral evaluation," *International Journal of Computational Intelligence Systems*, vol. 3, no. 4, pp. 486–500, 2010.

[37] D. Wu and J. M. Mendel, "Computing with words for hierarchical decision making applied to evaluating a weapon system," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 441–460, 2010.

[38] F. Herrera and E. Herrera-Viedma, "Choice functions and mechanisms for linguistic preference relations," *European Journal of Operational Research*, vol. 120, no. 1, pp. 144–161, 2000.

[39] F. Herrera, E. Herrera-Viedma, and L. Martínez, "A fusion approach for managing multi-granularity linguistic term sets in decision making," *Fuzzy Sets and Systems*, vol. 114, no. 1, pp. 43–58, 2000.

[40] F. Herrera and L. Martínez, "A 2-tuple fuzzy linguistic representation model for computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 6, pp. 746–752, 2000.

[41] B. Arfi, "Fuzzy decision making in politics: a linguistic fuzzy-set approach," *Political Analysis*, vol. 13, no. 1, pp. 23–56, 2005.

[42] V. Huynh and Y. Nakamori, "A satisfactory-oriented approach to multiexpert decision-making with linguistic assessments," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 35, no. 2, pp. 184–196, 2005.

[43] D. Ben-Arieh and Z. Chen, "Linguistic group decision-making: opinion aggregation and measures of consensus," *Fuzzy Optimization and Decision Making*, vol. 5, no. 4, pp. 371–386, 2006.

[44] J. L. García-Lapresta, "A general class of simple majority decision rules based on linguistic opinions," *Information Sciences*, vol. 176, no. 4, pp. 352–365, 2006.

[45] J. Wang and J. Hao, "Fuzzy linguistic PERT," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 2, pp. 133–144, 2007.

[46] T. C. Wang and Y. H. Chen, "Applying fuzzy linguistic preference relations to the improvement of consistency of fuzzy AHP," *Information Sciences*, vol. 178, no. 19, pp. 3755–3765, 2008.

[47] Y. Dong, Y. Xu, and S. Yu, "Linguistic multiperson decision making based on the use of multiple preference relations," *Fuzzy Sets and Systems*, vol. 160, no. 5, pp. 603–623, 2009.

[48] P. D. Liu, "A novel method for hybrid multiple attribute decision making," *Knowledge-Based Systems*, vol. 22, no. 5, pp. 388–391, 2009.

[49] Y. Dong, W. Hong, Y. Xu, and S. Yu, "Selecting the individual numerical scale and prioritization method in the analytic hierarchy process: a 2-Tuple fuzzy linguistic approach," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 13–25, 2011.

[50] L. A. Zadeh, "Fuzzy logic = computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 2, pp. 103–111, 1996.

[51] J. M. Mendel, "Fuzzy sets for words: a new beginning," in *Proceedings of the IEEE International conference on Fuzzy Systems*, pp. 37–42, St. Louis, Mo, USA, May 2003.

[52] J. M. Mendel, "Computing with words and its relationships with fuzzistics," *Information Sciences*, vol. 177, no. 4, pp. 988–1006, 2007.

[53] F. Liu and J. M. Mendel, "Encoding words into interval type-2 fuzzy sets using an interval approach," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1503–1521, 2008.

[54] S. Coupland, J. M. Mendel, and D. Wu, "Enhanced interval approach for encoding words into interval type-2 fuzzy sets and convergence of the word FOUs," in *Proceedings of the 6th IEEE World Congress on Computational Intelligence (WCCI '10)*, pp. 1–8, Barcelona, Spain, July 2010.

[55] L. Martínez and F. Herrera, "An overview on the 2-tuple linguistic model for computing with words in decision making: extensions, applications and challenges," *Information Sciences*, vol. 207, pp. 1–18, 2012.

[56] R. M. Rodriguez, L. Martinez, and F. Herrera, "Hesitant fuzzy linguistic term sets for decision making," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 109–119, 2012.

[57] E. Szmidt and J. Kacprzyk, "Distances between intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 114, no. 3, pp. 505–518, 2000.

[58] P. Grzegorzewski, "Distances between intuitionistic fuzzy sets and/or interval-valued fuzzy sets based on the Hausdorff metric," *Fuzzy Sets and Systems*, vol. 148, no. 2, pp. 319–328, 2004.

[59] W. Wang and X. Xin, "Distance measure between intuitionistic fuzzy sets," *Pattern Recognition Letters*, vol. 26, no. 13, pp. 2063–2069, 2005.

[60] Z. S. Xu and J. Chen, "An overview of distance and similarity measures of intuitionistic fuzzy sets," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 16, no. 4, pp. 529–555, 2008.

[61] Z. S. Xu, "A method based on distance measure for interval-valued intuitionistic fuzzy group decision making," *Information Sciences*, vol. 180, no. 1, pp. 181–190, 2010.

[62] Z. Xu and M. Xia, "Distance and similarity measures for hesitant fuzzy sets," *Information Sciences*, vol. 181, no. 11, pp. 2128–2138, 2011.

[63] Z. Xu and M. Xia, "On distance and correlation measures of hesitant fuzzy information," *International Journal of Intelligent Systems*, vol. 26, no. 5, pp. 410–425, 2011.

[64] Z. S. Xu, "Deviation measures of linguistic preference relations in group decision making," *Omega*, vol. 33, no. 3, pp. 249–254, 2005.

[65] Y. Xu and H. Wang, "Distance measure for linguistic decision making," *Systems Engineering Procedia*, vol. 1, pp. 450–456, 2011.

[66] Y. M. Wang, J. B. Yang, and D. L. Xu, "A preference aggregation method through the estimation of utility intervals," *Computers and Operations Research*, vol. 32, no. 8, pp. 2027–2049, 2005.

[67] H. Liao, Z. Xu, and X. Zeng, "Distance and similarity measures for hesitant fuzzy linguistic term sets and their application in multi-criteria decision making," *Information Sciences*, vol. 271, pp. 125–142, 2014.

[68] B. Zhu, Z. Xu, and M. Xia, "Hesitant fuzzy geometric Bonferroni means," *Information Sciences*, vol. 205, pp. 72–85, 2012.

[69] M. Xia and Z. Xu, "Hesitant fuzzy information aggregation in decision making," *International Journal of Approximate Reasoning*, vol. 52, no. 3, pp. 395–407, 2011.

[70] X. Gu, Y. Wang, and B. Yang, "A method for hesitant fuzzy multiple attribute decision making and its application to risk investment," *Journal of Convergence Information Technology*, vol. 6, no. 6, pp. 282–287, 2011.

[71] M. Xia, Z. Xu, and N. Chen, "Some hesitant fuzzy aggregation operators with their application in group decision making," *Group Decision and Negotiation*, vol. 22, no. 2, pp. 259–279, 2013.

[72] G. Wei, "Hesitant fuzzy prioritized operators and their application to multiple attribute decision making," *Knowledge-Based Systems*, vol. 31, pp. 176–182, 2012.

[73] B. Zhu, Z. Xu, and M. Xia, "Dual hesitant fuzzy sets," *Journal of Applied Mathematics*, vol. 2012, Article ID 879629, 13 pages, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 789234, 18 pages
http://dx.doi.org/10.1155/2014/789234

Research Article

Expert System for Competences Evaluation 360° Feedback Using Fuzzy Logic

Alberto Alfonso Aguilar Lasserre,[1] Marina Violeta Lafarja Solabac,[1] Roberto Hernandez-Torres,[1] Rubén Posada-Gomez,[1] Ulises Juárez-Martínez,[1] and Gregorio Fernández Lambert[2]

[1] *Division of Research and Postgraduate Studies, Instituto Tecnológico de Orizaba, Avenida Instituto Tecnológico 852, Colonia Emiliano Zapata, 94300 Orizaba,VER, Mexico*
[2] *Division of Research and Postgraduate Studies, Instituto Tecnológico Superior de Misantla, Km. 1.8 Carretera a Loma del Cojolite, 93821 Misantla, VER, Mexico*

Correspondence should be addressed to Alberto Alfonso Aguilar Lasserre; aaguilar@itorizaba.edu.mx

Received 11 April 2014; Revised 24 June 2014; Accepted 25 June 2014; Published 24 July 2014

Academic Editor: Jer-Guang Hsieh

Performance evaluation (PE) is a process that estimates the employee overall performance during a given period, and it is a common function carried out inside modern companies. PE is important because it is an instrument that encourages employees, organizational areas, and the whole company to have an appropriate behavior and continuous improvement. In addition, PE is useful in decision making about personnel allocation, productivity bonuses, incentives, promotions, disciplinary measures, and dismissals. There are many performance evaluation methods; however, none is universal and common to all companies. This paper proposes an expert performance evaluation system based on a fuzzy logic model, with competences 360° feedback oriented to human behavior. This model uses linguistic labels and adjustable numerical values to represent ambiguous concepts, such as imprecision and subjectivity. The model was validated in the administrative department of a real Mexican manufacturing company, where final results and conclusions show the fuzzy logic method advantages in comparison with traditional 360° performance evaluation methodologies.

1. Introduction

Nowadays, labor competences and competence evaluation represent a real challenge for organizations, which emerged in order to assign the right man to the right job. This evaluation method is based on questionnaires that involve fixed scales with specific values, such as 100%, 75%, 50%, 25%, and 0%. This kind of evaluation reduces the evaluator opportunity to express points of view and causes a rigid evaluation. Fuzzy set theory appears as an important tool to include inaccurate judgments inherent in personnel evaluation process. According to Butkiewicz [1] Fuzzy Logic is a very good tool for decision problems, especially when nonprecise or partially precise description is available. Although it can be applied with success in management problems, fuzzy logic is not common in this area.

Fuzzy Logic is an artificial intelligence (AI) technique [2]. AI comes with the purpose of developing models and programs of the intelligent behavior. One of the approaches of the AI is Logic, with the main objective of formalization of natural reasoning.

Fuzzy logic has two main components: membership functions and fuzzy rules. Using them it is possible to move a qualitative to a quantitative description, for example, to represent linguistic expressions as mathematic expressions. This is very useful when it is necessary to model the expertise of a human expert.

Fuzzy membership functions express the certainty than an element of the universe belongs to a fuzzy set. It represents the degree of truth as an extension of the valuation. Degrees of truth are very often confused with probabilities but they are conceptually different because fuzzy truth represents membership in vaguely defined sets, and not likelihood of an event. These membership functions can take different shapes according to expertise and preferences of the designer.

In these membership functions the x-axis represents the universe of discourse, and the y-axis represents the degrees of membership in the $[0,1]$ interval. Most commonly used functions are triangular, trapezoidal, Gaussian, singleton, Gamma, and so forth. Membership functions can be expressed as a discrete or continuous function. In other words, $\mu_A(X)$ is a membership function of a set A, according to the elements of the universe.

Fuzzy sets are classes of objects with grades of membership. Each set is characterized by a membership function, which assigns a grade of membership to each object based on a characteristic.

When the universe of discourse is continuous and finite, commonly used notation to represent set A is

$$A = \int_x \frac{\mu_{A(x)}}{x}, \tag{1}$$

where

$$0 \le \mu_A(x) \le 1. \tag{2}$$

When the universe of discourse is discrete and finite, fuzzy set A is commonly represented as

$$A = \sum_{i=1}^{n} \frac{\mu_A(x_i)}{x_i} = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \cdots + \frac{\mu_A(x_n)}{x_n}. \tag{3}$$

For instance, if fuzzy set A contains the elements x_1, x_2, x_3, x_4, and x_5 with membership degrees of 0, 0.5, 1, 0.5, and 0, respectively, the fuzzy set is expressed as

$$A = \frac{0}{x_1} + \frac{0.5}{x_2} + \frac{1}{x_3} + \frac{0.5}{x_4} + \frac{0}{x_5}. \tag{4}$$

A fuzzy set in a discrete and finite universe of discourse can be represented too as a set of ordered pairs of x and its membership degree in A as

$$A = \{(x, \mu_A(x)) \mid x \in X\}, \tag{5}$$

which results in

$$A = \{(x_1, 0), (x_2, 0.5), (x_3, 1), (x_4, 0.5), (x_5, 0)\}. \tag{6}$$

Geometry of fuzzy sets involves three elements: domain, range, and mapping. In this example, geometry of the fuzzy set A is

domain: $\{x_1, x_2, x_3, x_4, x_5\}$,

range: $[0,1]$,

mapping: $\mu A(x) \rightarrow [0,1]; \{0, 0.5, 1, 0.5, 0\}$.

Graphically, fuzzy set A can be expressed as shown in Figure 1.

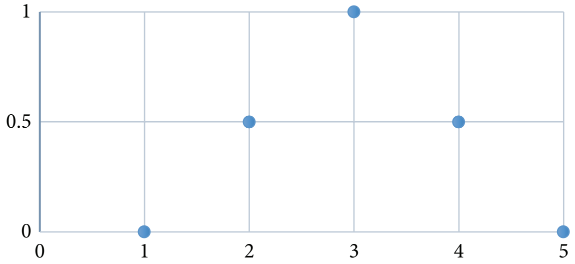


Figure 1: Graphical representation of a fuzzy set.

As in classic logic, fuzzy logic uses three basic operations in fuzzy sets: union, intersection, and complement. However, fuzzy sets have certain characteristics that make them different from classic sets. Fuzzy sets have elements with variable membership degrees, which means that an element of the universe of discourse can belong to one or more fuzzy sets, with different membership degrees.

The first operation on fuzzy sets is *intersection*. It is the degree of membership that two fuzzy sets share, that is, is the smallest degree of membership of each element in the fuzzy sets. Intersection of two fuzzy sets A and B is a fuzzy set $A \cap B$ in the universe of discourse X, whose function is given by

$$\mu_{A\cap B}(x) = \min[\mu_A(x), \mu_B(x)] = \underset{x}{\wedge}[\mu_A(x), \mu_B(x)], \tag{7}$$

where

$A \cap B$ represents the intersection of the fuzzy sets A and B;

$\wedge$ represents the minimum operator.

The operation *union* results as the biggest degree of membership of each element in the fuzzy sets, that is, the highest value of the fuzzy values. Union of two fuzzy sets A and B is a fuzzy set $A \cup B$ in the universe of discourse X, whose function is given by

$$\mu_{A\cup B}(x) = \max[\mu_A(x), \mu_B(x)] = \underset{x}{\vee}[\mu_A(x), \mu_B(x)], \tag{8}$$

where

$A \cup B$ represents the intersection of the fuzzy sets A and B;

$\vee$ represents the maximum operator.

The logic operation *complement* results as the degree of membership that the fuzzy set needs to reach the unit. The complementary set A of a fuzzy set A is that whose function is given by

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x). \tag{9}$$

Functions that define operations of intersection and union can be generalized using the triangular norm (called T-norm) and the triangular conorm (called T-conorm or S-norm), respectively.

A *t-norm operator* is a function of two elements $T(\cdot,\cdot)$ that satisfies the following.

Boundary Conditions. This condition implies the generalization of the classic sets:

$$T(a,0)=0,\qquad T(a,1)=a. \tag{10}$$

Monotonicity. This condition implies that a decrease in the degree of membership for the set A or B will not produce an increase in the degree of membership of the intersection of the sets A and B:

$$T(a,b)\le T(c,d)\quad \text{if } a\le c,\ b\le d. \tag{11}$$

Commutative Property. This property indicates that the operator is indifferent to the order of the fuzzy sets that are combined:

$$T(a,b)=T(b,a). \tag{12}$$

Associative Property. This property allows calculating the intersection of any number of fuzzy sets, grouped in pairs, regardless of the order of couples:

$$T(a,T(b,c))=T(T(a,b),c). \tag{13}$$

In the same way, operator T-conorm (S-norm) is a function of two elements $S(\cdot,\cdot)$ that satisfies the following.

Boundary Conditions

$$S(a,1)=1,\qquad S(a,0)=a. \tag{14}$$

Monotonicity

$$S(a,b)\le S(c,d)\quad \text{if } a\le\ c,\ b\le\ d. \tag{15}$$

Commutative

$$S(a,b)=S(b,a). \tag{16}$$

Associative

$$S(a,S(b,c))=S(S(a,b),c). \tag{17}$$

Some interesting contributions of fuzzy logic as a technique to model subjective viewpoints are found in [3], where the authors firstly considered decision making problems using fuzzy logic. Probably, the first attempt to apply fuzzy logic to personnel evaluation was proposed in [4, 5]. Another approach can be found in [6]. Cannavacciuolo et al. [4] presented the application of fuzzy set theory to a personnel evaluation procedure. Effectiveness of fuzzy concepts and methods depends on the approach used for the analysis of organizational issues. Fuzzy set theory allows them to model the weak signals existing in evaluation processes and highlights part of the tacit knowledge involved in individual judgments. Usually, researchers, consultants, and managers use a rather qualitative approach to organizational problems. However the natural language is the preferred instrument to describe the organizational conditions because the shades of meaning and the ambiguity of verbal statements allow the company actors to manage diverging opinions, tensions, and conflicts.

These approaches are detailed in [7–11].

On the other hand, the logical-mathematical models tend to represent a world of certainty and coherence where doubts, contradictions, divergences, polysemy, conflicts, and ambiguities are usually typified, dissolved, degraded, and linearized. Within this same conceptual framework, mathematicians, computer scientists, A.I. researchers, and engineers, in search of formal coherence, quantifiable variables, and efficient algorithms, usually tend to use fuzzy set theory without considering complexity and ambiguity in organizational situations, for example, [3, 12–18].

There seems to be a growing trend towards the use of systematic procedures in personnel selection. For instance, Karsak [19] introduced a method that integrates decision makers linguistic assessments about subjective factors such as excellence in oral communication skills, personality, leadership, and quantitative factors such as aptitude test score within multiple objective programming frameworks. The importance level of each goal is considered by applying the composition operator to the membership function of the goal and the membership function corresponding to its fuzzy priority defined by linguistic variables.

Kolarik et al. [20] present an online approach to monitoring human performance in terms of conditional reliability when a task is performed. Unlike traditional human reliability analysis, this approach develops a dynamic model that can cope with constantly changing conditions that affect operator performance. A fuzzy knowledge-based assessment approach is developed in order to deal with uncertainty and subjectivity associated with human performance assessment.

Podofellini et al. [21] assess the influence of the failure of the operators to perform one task on the failure probabilities of subsequent tasks with an approach called technique for human error rate prediction (THERP) and a fuzzy expert system (FES).

Other works include the use of fuzzy logic to evolve an optimal and accurate judgment according to the human thinking model and also to mitigate the commonly occurred biases in human recruitment and selection procedures, as seen in [22].

García et al. [23] propose, through the use of tools based on fuzzy logic, the evaluation of the impact of training in companies, by applying the reasoning characteristic of fuzzy logic, with the aim of complementing and extending the classical logic.

Tosti and Addison [24] refer that a poorly designed 360° feedback system can do more harm than good. The use of commercial 360° software is not always an option due to specific requirements of each organization. In this way, it may be that people who design 360° programs are well versed in assessment and measurement technology and woefully lacking in their understanding of feedback technology. Reliability of the 360 degree feedback is supported by the number and hierarchy of raters, as referred to in [25], assuming that personal qualities are developmental goals. Therefore, software has been designed specifically for this study evaluation.

This paper is organized as follows. Section 1 presents an introduction to the study. Section 2 shows some fundamental concepts about competences, performance evaluation 360° feedback, and competence evaluation methodology; fuzzy logic basis and a proposed fuzzy logic model are shown too. After that, the application of both systems (traditional 360° feedback system and fuzzy logic 360° feedback expert system) at the administrative area of a real manufacturing company in the state of Veracruz, Mexico, is shown. Section 3 shows clear and concise results while discussion about the significance of the results is developed. Finally, Section 4 describes the main conclusions of the study.

2. Expert System for Competences Evaluation 360° Feedback Using Fuzzy Logic

2.1. Competences. Personnel appraisal is considered as performance evaluation, and it is based on formal evaluation programs with reasonable information amount about employees and their job performance.

The literature describes several evaluation methods, each with its own advantages and drawbacks, and there is no ideal or universal method for all people, positions, organizations, and situations. The choice will depend on many other aspects such as

(i) position,

(ii) characteristics to be measured,

(iii) organizational culture,

(iv) objectives, achieved or to be achieved,

(v) circumstantial elements.

Performance evaluation methods are classified according to the feature they measure characteristics, behaviors, or outcomes, as referred to in [1]. Behavior methods enable the evaluator to identify how far the employee performance is away from a specific scale. These methods describe what actions should be exhibited during the position performance. It is mainly used to provide development-oriented feedback. According to Gomez-Mejia et al. [26], the main advantage in the performance measure adopting a behavior-based approach is that criteria or performance standards are concrete. Behavior scales give employees specific behavior examples that can make them successful (or avoid their success) in their work. If an employee knew the required skills for the position and the corresponding aperture in degrees, it could verify, analyze, and control its own behavior according to the requirements.

A competence is an underlying characteristic in the employee related to an effectiveness standard and superior performance in a job or situation, as discussed in [27].

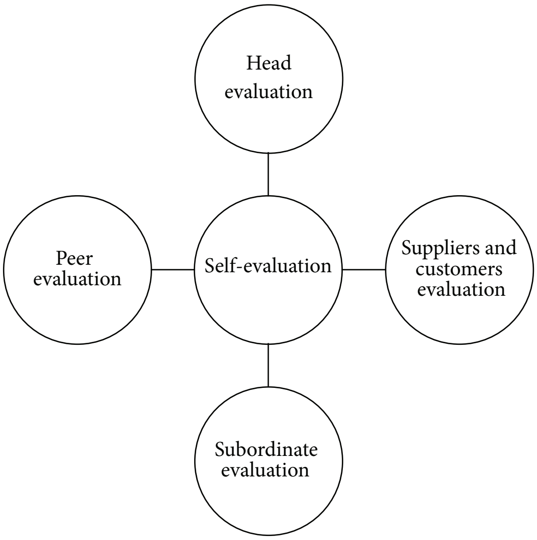


Figure 2: 360° evaluation scheme.

According to Levy-Leboyer [28], individual skills and company competences are closely related. Company competences are constituted by the integration and coordination‘of individual skills; however, these competences require an integration and coordination of knowledge and personal qualities. Individual competences are an individual property. Company competences are developed by individuals, but they belong to the company.

2.2. 360° Performance Evaluation. 360° performance evaluation is a sophisticated scheme that allows the employee to be evaluated by its surrounding bosses, peers, and subordinates (see Figure 2). A scheme may include suppliers or customers.

360° performance evaluation can potentially bring a globalized diagnosis about the employee performance, allowing the evaluator to compare different opinions about the level of competence expected in the evaluated person and then to take decisions about how to increase the level of compliance of this competences.

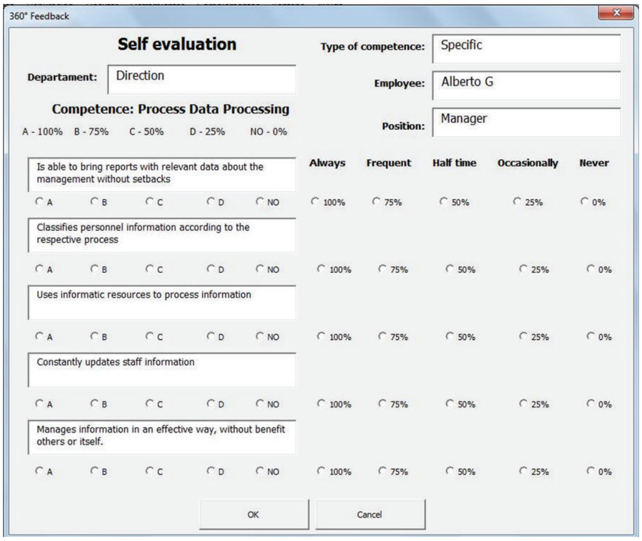
Alles [29] proposes 7 points for 360° evaluation.

(1) Identify cardinal and individual competences. If the company has implemented a performance evaluation system, the competences will be the same. Eventually, it is possible to use a reduced number of competences when using the 360° evaluation system.

(2) Design the tool. Questionnaires typically constitute the process support (see Figure 3).

(3) Select evaluators: superiors, partners, internal customers in other areas, customers, and external suppliers. Customers can be included or not. It is important to emphasize the fact that assessments are anonymous and that evaluators are chosen by the evaluating person.

(4) Launch the evaluation process with stakeholders and evaluators.



360° Feedback

Self evaluation

Departament: Direction

Type of competence: Specific

Employee: Alberto G

Position: Manager

Competence: Process Data Processing

A - 100% B - 75% C - 50% D - 25% NO - 0%

					Always	Frequent	Half time	Occasionally	Never
Is able to bring reports with relevant data about the management without setbacks									
○ A	○ B	○ C	○ D	○ NO	○ 100%	○ 75%	○ 50%	○ 25%	○ 0%
Classifies personnel information according to the respective process									
○ A	○ B	○ C	○ D	○ NO	○ 100%	○ 75%	○ 50%	○ 25%	○ 0%
Uses informatic resources to process information									
○ A	○ B	○ C	○ D	○ NO	○ 100%	○ 75%	○ 50%	○ 25%	○ 0%
Constantly updates staff information									
○ A	○ B	○ C	○ D	○ NO	○ 100%	○ 75%	○ 50%	○ 25%	○ 0%
Manages information in an effective way, without benefit others or itself.									
○ A	○ B	○ C	○ D	○ NO	○ 100%	○ 75%	○ 50%	○ 25%	○ 0%

Figure 3: Questionnaire example.

(5) Data processing: most of the time data are processed by external consultants to preserve information confidentiality.

(6) Communicate 360° evaluation results to concerned people.

(7) The company will receive a consolidated report. This report will be received only by the employee.

In this work, 360 degree feedback methodology proposed by Alles [29] is applied under two different approaches: traditional 360° feedback and fuzzy logic 360° feedback. Likewise, there are some other substantial differences. (1) Data processing is performed by two software applications, each designed for its specific process. Both applications are able to select evaluators randomly and present questionnaires. From this point, there is another difference: (2) first application performs the evaluation in the traditional 360 degree feedback; the second application is an expert system that uses fuzzy logic into the questionnaires to perform evaluation. The third substantial difference, hence, is that (3) expert system does not require a human expert, except when they are designed. No external consultant is necessary anymore, since expert system achieves this objective too.

2.3. Fuzzy Logic Basis. Fuzzy logic is the mapping from an input measurement space to an output measurement space using linguistic variables. It gives the ability to model imprecision by incorporating qualitative components into a quantitative analysis.

Fuzzy logic systems have a narrow relationship with fuzzy logic concepts such as fuzzy sets and linguistic variables. The most popular fuzzy logic systems are Mamdani and Takagi-Sugeno.

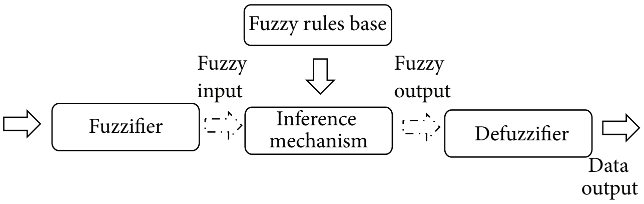


Figure 4: Mamdani general fuzzy logic system.

Mamdani fuzzy systems use 4 components (see Figure 4).

(i) Fuzzifier: Mamdani system inputs are typically numeric values, coming from some kind of sensor or being results of a process; to be able to operate this value, Mamdani systems translate this value into a special value that can be operated by the inference mechanisms. This translation is done by the fuzzifier, which converts numeric values into fuzzy values that represent the level of pertinence of the different variables of the system to the fuzzy sets.

(ii) Fuzzy inference mechanism: once the fuzzifier has translated the fuzzy values, these have to be processed to generate a fuzzy output. Inference mechanism task is to take fuzzy values and generate a fuzzy output based on a fuzzy rules base.

(iii) Fuzzy rules base is the way in which Mamdani fuzzy systems have to represent expertise and linguistic knowledge to solve the issue. It is a set of IF-THEN sentences, containing two parts each: antecedent and conclusion. In a Mamdani fuzzy system, antecedent and conclusion are given by linguistic expressions.

(iv) Defuzzifier: inference system output is a fuzzy output, so it cannot be interpreted by an external element which only could operate numeric data. To make it possible to operate this data, output is translated to numeric format, and this task is done by the defuzzifier, using one of different procedures such as gravity center or averaged centers.

Fuzzy Logic uses certain essential components to achieve its purpose.

Imprecision. Often the same term is used to describe imprecision and uncertainty in only slightly related areas of measurement. Imprecision in measurement is associated with a lack of knowledge. Imprecision as a probability form is associated with uncertainty about the future event occurrence. Imprecision in description, the imprecision type addressed by fuzzy logic, is connected with intrinsic or built-in imprecision that belongs to the event itself.

Fuzzy logic addresses the issues associated with an intrinsic imprecision rather than those directly concerned with measuring devices failures in the measurements accuracy.

Intrinsic imprecision is associated with a phenomenon properties description and not with properties measurement using some external device.

Ambiguity. There are close semantic relationships between the ambiguity idea and fuzziness; in fact, some fuzzy states can be highly ambiguous. Ambiguity connotes the property to have several but plausible and reasonable interpretations. These interpretations can have different belief states. Ambiguity in meaning is a common occurrence in natural languages.

Likelihood and Ambiguity. Fundamentally, the basic confusion between fuzzy logic and probability arises from the idea that they measure the same kind of uncertainty. In strictly semantic, as well as mechanistic, the two forms of uncertainty are different. Propositions in probability address the likelihood of an outcome for some discrete event. The event outcome either happens or does not happen. Propositions in fuzzy logic concern the degree to which an event occurred. While a probability outcome happens unequivocally, a fuzzy event occurrence may involve some degree of ambiguity or uncertainty.

Fuzzy Sets Components. Gregory [30] indicates that the fuzzy logic has two main components: membership functions and fuzzy rules. When using these components it is possible to move the experiences and human preferences from a qualitative description to a quantitative description.

Membership fuzzy functions can take different figures and forms, according the designer experiences and preferences. Typical functions are triangular, trapezoidal, S, Gamma, Gaussian, and exponential. On the other hand, the fuzzy rules are written as IF-THEN couples and reported in tabular form.

The four basic ways in which the fuzzy rules can be achieved are expert experiences and engineering knowledge, human behaviors, models based on a fuzzy system, and learning processes. These methods are not necessarily mutually exclusive.

Membership Functions. In classical set theory, something is completely included or not. This situation can be described by assigning a value of one to all the elements included in the set and the value of zero to the ones not included in it. The function that assigns these values is called "membership function." The fuzzy sets allow to describe the degree of membership of the object to the concept given by the labels, and allow to assign values between zero and one to the membership function (see Figure 5).

Mathematical Features of Fuzzy Sets. Main characteristics of the fuzzy sets are height, support, cutoff level-α, and nucleus.

Height. It is the highest degree of membership of the elements of the set; that is,

$$\text{Height}\,(A) = \max\left\{h \mid h = \mu_A(x), x \in X\right\}. \qquad (18)$$

When the height of a fuzzy set is equal to 1 it is said that it is a normalized fuzzy set.

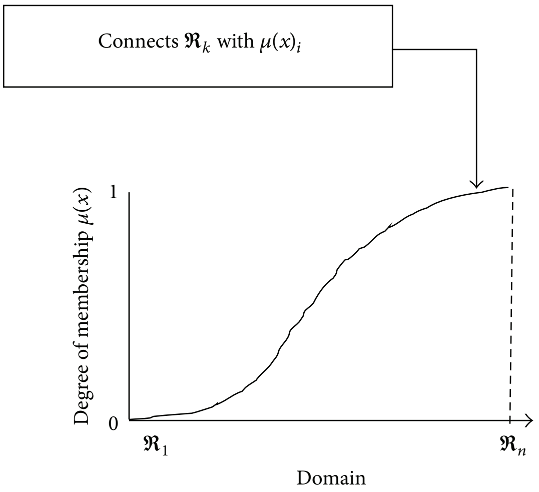


Figure 5: General structures in a fuzzy set.

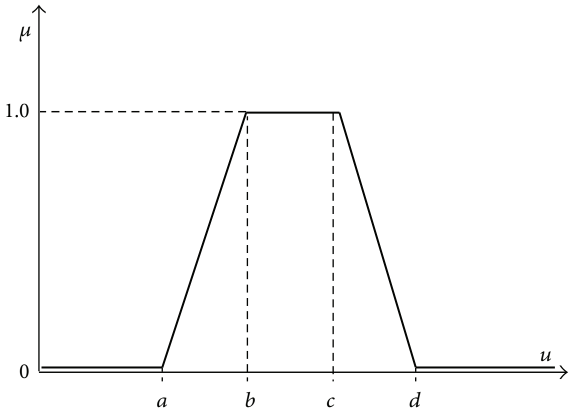


Figure 6: Trapezoidal type membership function.

Support. It is the number of elements whose degree of membership is not zero; that is,

$$\text{Sup}\,(A) = \left\{x \mid \mu_A(x) > O, x \in X\right\}. \qquad (19)$$

Cutoff Level-α. It is the set of elements of X with a minimum degree α; that is,

$$A_\alpha = \left\{x \mid \mu_A(x) \geq \alpha, x \in X\right\}. \qquad (20)$$

Nucleus. It is the set of elements of X with a degree of membership equal to 1; that is,

$$\text{Nucleus}\,(A) = \left\{x \mid \mu_A(x) = 1, x \in X\right\}. \qquad (21)$$

Inclusion Functions in Fuzzy Sets. There are standard families for inclusion functions; the most frequent ones are trapezoidal, singleton, triangular, S, exponential, and π type.

Trapezoidal functions are defined by four points: a, b, c, and d (see Figure 6). This function is zero for values lower than "a" and higher than "d" and one between "b" and "c" and takes values in range $[0, 1]$ between "a" and "b" and between "*c*" and "*d*." It is used in simple fuzzy systems, since it allows defining a fuzzy set with little information and computing membership function values in a simple way.

This function is common for microprocessor based systems since it can be encoded in a similar format as *S* functions, π functions, and triangular and singleton functions (e.g., if points b and c are combined the result is a triangular function). Trapezoidal function is defined as follows (see (22)):

$$S(u;a,b,c,d)=\begin{cases}0, & u<a,\\ \left(\dfrac{u-a}{b-a}\right), & a\le u\le b,\\ 1, & b\le u\le c,\\ \left(\dfrac{d-u}{d-c}\right), & c\le u\le d,\\ 0, & u>d.\end{cases} \quad (22)$$

Trapezoidal functions are suitable to model properties in a range of values, stages, or levels (e.g., young, adult, elder, etc.). Modeling a triangular function can be done through the $b = c$ simplification. For an *S* function and singleton types (but not soft), $c = d = \max(U)$ and $a = b = c = d$ transformations can be applied, respectively.

Triangular function (T) can be defined as indicated in

$$T(u;a,b,c)=\begin{cases}0, & u<a,\\ \dfrac{u-a}{b-a}, & a\le u\le b,\\ \dfrac{c-a}{c-b}, & b\le u\le c,\\ 0, & u>c.\end{cases} \quad (23)$$

T functions (see Figure 7) are appropriate for modeling properties with an inclusion value different from zero and for a narrow range of values around point *b*.

Linguistic Variables. Linguistic variables take values from natural language, for example, much, little, positive, negative, and so forth. These words are considered as labels within the fuzzy set theory.

Even though linguistic variables aim to assign labels as variable values taken from natural language words, they will be able to assign numerical values too. Then, in the expression "temperature is cold," the variable "temperature" must be seen as a linguistic variable, since the value cold is assigned as a fuzzy set. However, this variable can also take numerical values such as "temperature is 4°C."

Fuzzy Rules. Fuzzy rules combine one or more input fuzzy sets, called premises, and associate them with an outcome fuzzy set, called consequence. The fuzzy set premise is associated using AND, OR, and so forth operators.

Defuzzification Process. There are two common methods of defuzzification process: gravity center (centroid) and maximum output. As is shown in Figure 8, both techniques produce different results [31].

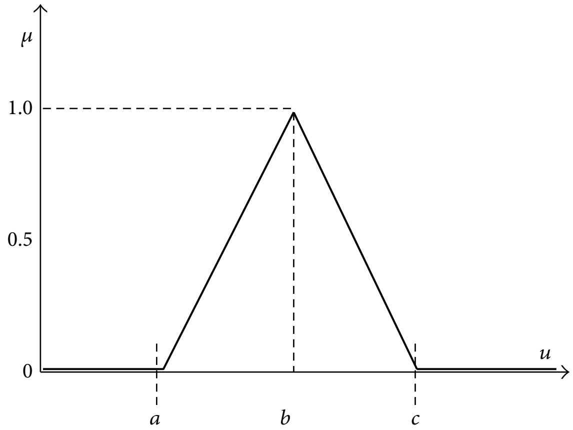


FIGURE 7: Type T (triangular) function.

Both techniques produce reasonable results when they are applied in specific fuzzy models. Gravity center is the most common method, because it combines evidence about rules and response fields are pondered by the total true degree. Gravity center is, essentially, the weighted average of the output membership function.

Centroid Computation. Centroid technique finds the balance point solution in fuzzy zone using the weighted average in the fuzzy region. Arithmetically, the procedure is formulated by

$$\mathfrak{R}=\frac{\sum_{i=0}^{n} d_i\cdot\mu_A(d_i)}{\sum_{i=0}^{n}\mu_A(d_i)}, \quad (24)$$

where d is the ith domain value and $\mu(d)$ is the true membership value at this point. Centroid or defuzzification with moments finds a point that represents the fuzzy set gravity center.

2.4. Fuzzy Logic Personnel Evaluation Model. A competence performance evaluation involves subjective viewpoints and evaluator preferences are reflected at the evaluation moment. Very often, evaluators express their perceptions in natural language terms. Nevertheless, as mentioned above, evaluation questionnaires constitute the performance evaluation support, and they are based on punctual values that do not reflect or approximate real viewpoints.

Therefore, it is desirable to have a flexible evaluation tool that facilitates imprecision, ambiguity, and subjectivity handling. In this purpose, fuzzy sets allow a suitable treatment.

The fuzzy logic model proposed is constituted as follows.

Linguistic Variables. Three linguistic variables have been defined, "Scale," "Frequency," and "Required Level."

"Scale" variable refers to the percentage (assigned by the evaluator) indicating how well the employee behavior matches the competence definition.

"Frequency" refers to the percentage obtained when the evaluator answers a question and ”rethinks” his/her evaluation determining the number of times the behavior is manifested.

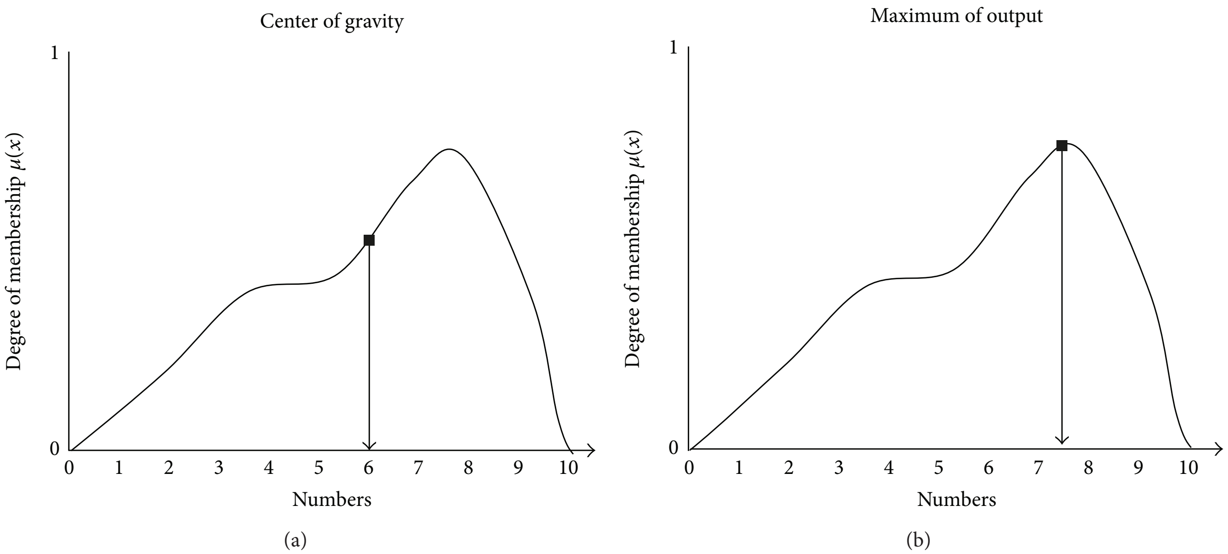


FIGURE 8: Defuzzification common methods.

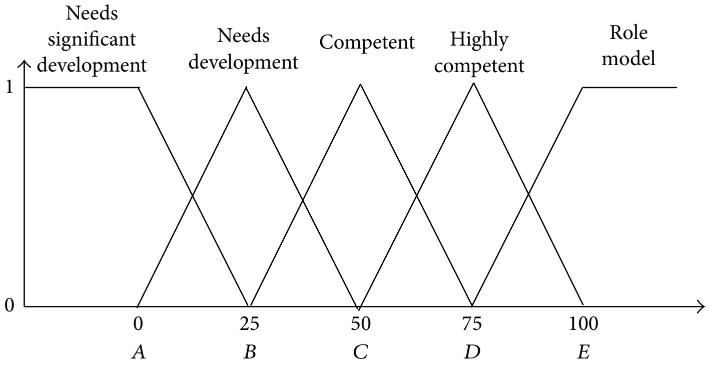


FIGURE 9: Fuzzy sets for linguistic variable "Scale."

"Required Level" refers the percentage expected by the organization; the individual must cover conduct according the competence definition.

Output Variables. They refer to the fit conduct qualification corresponding to the competence definition given in percentage. Under this consideration there are three cases, "needs to improve," "satisfies," and "exceeds."

Fuzzy Sets Representation. This model uses triangular inclusion functions to represent linguistic variables; the output variable was modeled with trapezoidal functions.

According to the expert, five fuzzy sets define the variable "Scale" possible values (see Figure 9), which are

(i) "needs significant development,"

(ii) "needs development,"

(iii) "competent,"

(iv) "highly competent,"

(v) "role model."

Likewise, there are four fuzzy sets to define the variable "Frequency" possible values (see Figure 10), which are

(i) "occasionally,"

(ii) "half time,"

(iii) "frequent,"

(iv) "always."

The "Required Level" variable was modeled through three possible fuzzy sets (see Figure 11), called

(i) "low,"

(ii) "average,"

(iii) "high."

The output variable has three cases which allow defining fuzzy sets (see Figure 12); these cases are

(i) "requires improvement,"

(ii) "complies,"

(iii) "exceeds."

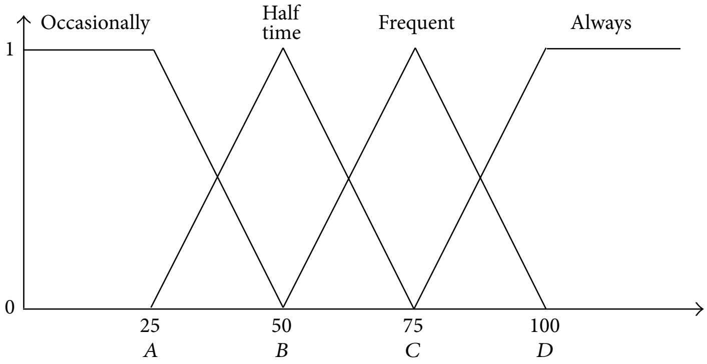


Figure 10: Fuzzy sets for linguistic variable "Frequency."

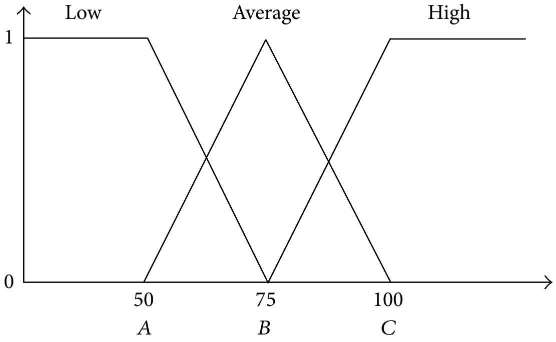


Figure 11: Fuzzy sets for linguistic variable "Required Level."

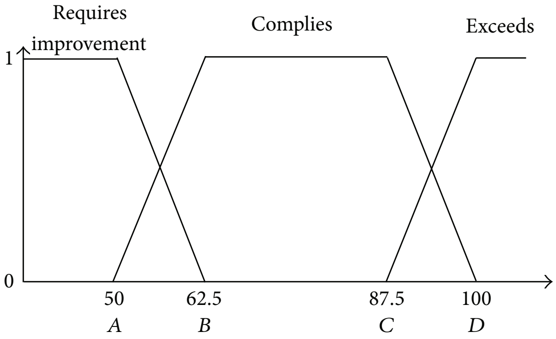


Figure 12: Fit qualification.

Fuzzy Rules. Fuzzy rules must consider all the combinations among input variables sets. Each combination will be associated with an output variable fuzzy set. Sixty fuzzy rules were created in this case, according to the opinion of an expert, who rigorously analyzed each set of inputs, determining the level of performance produced by each rule (see Table 1).

Fuzzification. The fuzzification process includes membership functions calculation for input variables and, then, uses the minimum-maximum criterion for variables activation.

Defuzzification. The defuzzification process will allow getting the final fit qualification; the proposed method is the gravity center (centroid) including the following four steps:

(1) divide total area in partial areas,

(2) calculate partial areas value,

(3) calculate each partial area centroid,

(4) calculate total centroid.

2.5. Computational Experiments. This section describes the use of the methodology using both traditional 360° feedback and 360° fuzzy logic feedback.

2.5.1. Application of the Expert System in a Real Company. The use of both methods was carried out in the administrative area of a Mexican manufacturing company, located in the state of Veracruz, Mexico, where administrative procedures require staff to be evaluated based on their performance annually. A branch of the organization chart, consisting of five different but related positions was selected to perform the 360° feedback evaluation process (see Figure 13).

2.5.2. Traditional 360° Methodology

(1) Identify cardinal and individual competences. To evaluate a position we will consider five cardinal competences and eight specific competences (see Table 2).

(2) Design the tool. This evaluation includes four questionnaires and each questionnaire includes five questions.

(3) Select an evaluator. Evaluation includes self-evaluation, boss evaluation, and peers evaluation. To choose evaluators and launch the process we will design and use the software; in this sense, the selection procedure is random.

(4) Execute the evaluation process with concerned people and evaluators. As an example, Table 3 shows self-evaluation values for cardinal competences.

Table 1: Fuzzy rules summary.

									SCALE											
	Needs significant development				Needs development				Competent				Highly competent				Role model			
	Frequency				Frequency				Frequency				Frequency				Frequency			
	Occasional	Half time	Frequent	Always	Occasional	Half time	Frequent	Always	Occasional	Half time	Frequent	Always	Occasional	Half time	Frequent	Always	Occasional	Half time	Frequent	Always
Required level																				
Low	C (1)	C (4)	I (7)	I (10)	C (13)	C (16)	I (19)	I (22)	C (25)	C (28)	E (31)	E (34)	C (37)	C (40)	E (43)	E (46)	E (49)	C (52)	E (55)	E (58)
Average	C (2)	I (5)	I (8)	I (11)	C (14)	I (17)	I (20)	I (23)	I (26)	C (29)	C (32)	C (35)	C (38)	C (41)	C (44)	E (47)	C (50)	C (53)	C (56)	E (59)
High	I (3)	I (6)	I (9)	I (12)	I (15)	I (18)	I (21)	I (24)	I (27)	I (30)	C (33)	C (36)	I (39)	I (42)	C (45)	C (48)	I (51)	I (54)	C (57)	C (60)

I: requires improvement.
C: complies.
E: exceeds.

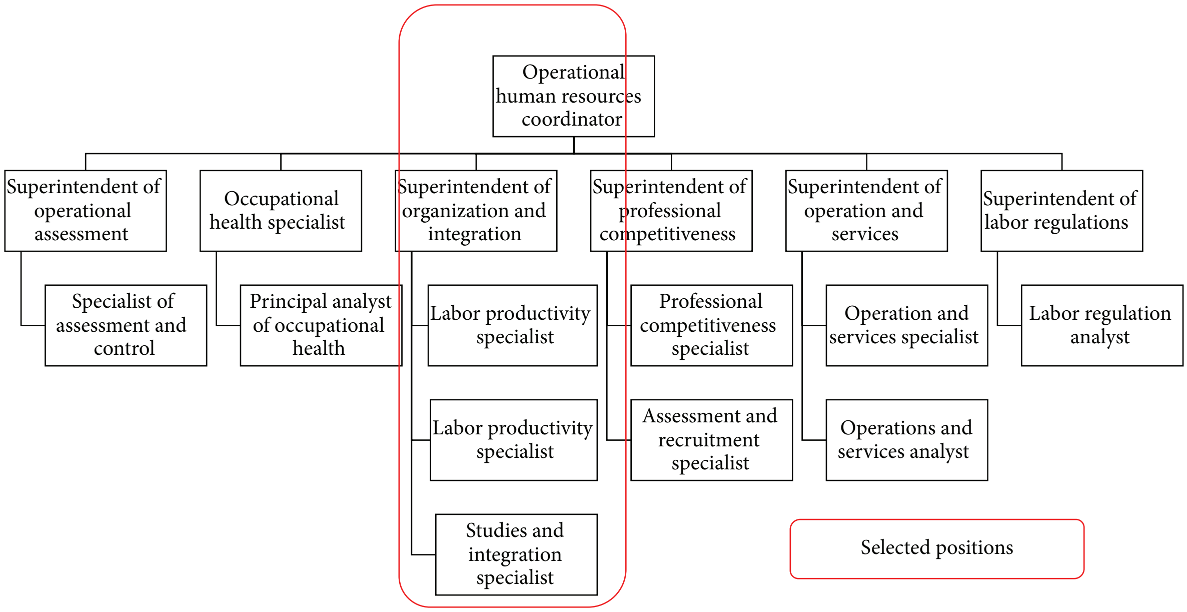

Operational human resources coordinator
Superintendent of operational assessment
Specialist of assessment and control
Occupational health specialist
Principal analyst of occupational health
Superintendent of organization and integration
Labor productivity specialist
Labor productivity specialist
Studies and integration specialist
Superintendent of professional competitiveness
Professional competitiveness specialist
Assessment and recruitment specialist
Superintendent of operation and services
Operation and services specialist
Operations and services analyst
Superintendent of labor regulations
Labor regulation analyst
Selected positions

Figure 13: Organizational chart.

Table 2: Cardinal and specific competences.

Cardinal competences	Specific competences
(1) Work quality	(1) Planning and organization skills
(2) Tolerance to work under pressure	(2) Customer orientation
(3) Communication skills	(3) Productivity
(4) Contact modalities	(4) Technical credibility
(5) Analytical skills	(5) Innovation
	(6) Empowerment
	(7) Collaboration
	(8) Commitment level-personal discipline-productivity

(5) Data processing: software specifically designed for data processing to obtain the competence level scale value and frequency value was used; then we will average the results for each question (see Table 4). Finally, averages are compared with the Required Level. The same procedure will be used for specific competences treatment.

(6) Reports will be delivered only to the evaluated person. Comments and graphs will be printed in two reports: the first one will include cardinal competences and will be given to the company, while the second report will include specific competences and will be given to the employee.

(7) Communicate the 360° evaluation results to the concerned people.

2.5.3. Fuzzy Logic Model Implementation. (1) Choose an organizational department.

(2) Choose a post in the selected department.

(3) Choose a required competence in the post.

(4) Select one question in the questionnaire that evaluates the competence.

(5) Evaluator assigns values for input variables "Scale" and "Frequency," for example,

Scale = 62%
Frequency = 84%
The Required Level assigned by the company = 75%.

(6) Fuzzification process: compute the variable membership values corresponding to the variable Scale:

$$\begin{gathered}
\mu_{\text{COMPETENT}}(\text{SCALE}) \\
= \begin{cases}
0; \quad W \leq B' & 0, \\
1 - \dfrac{C' - W}{C' - B'}; \quad B' < W \leq C' & 0, \\
1 - \dfrac{W - C'}{D' - C'}; \quad C' < W < D' & 0.52, \\
0; \quad D' \leq W & 0,
\end{cases} \\
\mu_{\text{HIGHLY-COMPETENT}}(\text{SCALE}) \\
= \begin{cases}
0; \quad W \leq C' & 0, \\
1 - \dfrac{D' - W}{D' - C'}; \quad C' < W \leq D' & 0.48, \\
1 - \dfrac{W - D'}{E' - D'}; \quad D' < W < E' & 0, \\
0; \quad E' \leq W & 0.
\end{cases}
\end{gathered} \tag{25}$$

Table 3: Self-evaluation values, cardinal competences.

(a)

	360° evaluation							Self-evaluation		
					Cardinal competencies					
	Work quality	Required level: 100	Tolerance to work under pressure	Required level: 75	Communication	Required level: 75	Contact modalities	Required level: 100	Analytic skills	Required level: 75
	Scale	Frequency	Scale	Frequency	Scale	Frequency	Scale	Frequency	Scale	Frequency
Ques. 1	75	100	100	100	100	25	75	100	75	25
Ques. 2	75	75	75	50	100	100	100	100	100	75
Ques. 3	100	50	100	100	75	100	100	75	100	75
Ques. 4	100	100	75	75	50	100	75	100	50	50
Ques. 5	75	100	50	25	25	75	50	25	100	100

(b)

Where	
Scale	Frequency
100%: A	100%: always
75%: B	75%: frequent
50%: C	50%: half time
25%: D	25%: occasional
0%: NO	0%: never

TABLE 4: Self-evaluation, cardinal competences treatment.

(a)

	360° evaluation						Self-evaluation								
	Cardinal competencies														
	Work quality	Required level: 100		Pressure tolerance	Required level: 75		Communication	Required level: 75		Contact modalities	Required level: 100		Analytic skills	Required level: 75	
	Scale	Frequency	Product	Scale	Frequency	Product	Scale	Frequency	Product	Scale	Frequency	Product	Scale	Frequency	Product
Ques. 1	75	100	75	100	100	100	100	25	25	75	100	75	75	25	18.75
Ques. 2	75	75	56.25	75	50	37.5	100	100	100	100	100	100	100	75	75
Ques. 3	100	50	50	100	100	100	75	100	75	100	75	75	100	75	75
Ques. 4	100	100	100	75	75	56.25	50	100	50	75	100	75	50	50	25
Ques. 5	75	100	75	50	25	12.5	25	75	18.75	50	25	12.5	100	100	100
	Average		71.25			61.25			53.75			67.5			58.75

(b)

Where	
Scale	Frequency
100%: A	100%: always
75%: B	75%: frequent
50%: C	50%: half time
25%: D	25%: occasional
0%: NO	0%: never

Compute the variable membership values corresponding to the variable Frequency:

$$\mu_{\text{FREQUENT}}\,(\text{FREQUENCY}) = \begin{cases} 0; \quad W \le B & 0, \\ 1 - \dfrac{C-W}{C-B}; \quad B < W \le C & 0, \\ 1 - \dfrac{W-C}{D-C}; \quad C < W < D & 0.64, \\ 0; \quad D \le W & 0. \end{cases} \tag{26}$$

$$\mu_{\text{ALWAYS}}\,(\text{FREQUENCY}) = \begin{cases} 0; \quad W \le C & 0, \\ 1 - \dfrac{D-W}{D-C}; \quad C < W \le D & 0.36, \\ 1; \quad D \le W & 0. \end{cases}$$

Compute the variable membership values corresponding to the variable Required Level:

$$\mu_{\text{AVERAGE}}\,(\text{REQUIREDLEVEL}) = \begin{cases} 0; \quad W \le A'' & 0, \\ 1 - \dfrac{B''-W}{B''-A''}; \quad A'' < W \le B'' & 1, \\ 1 - \dfrac{W-B''}{C''-B''}; \quad B'' < W < C'' & 0, \\ 0; \quad C'' \le W & 0. \end{cases} \tag{27}$$

The fuzzification process continues with variable activation, performed according to the min.-max. criterion (see (28), (29), (30), and (31)); this procedure allows identifying the membership values that will appear in the defuzzification process.

Rule 32 is

$$\begin{aligned} &\mu_{C\cap F\cap M}\,(62, 84, 75) \\ &\quad = \min\{\mu_C\,(62), \mu_F\,(84), \mu_M\,(75)\} \\ &\quad = \min\{0.52, 0.64, 1.00\} = 0.52. \end{aligned} \tag{28}$$

Rule 35 is

$$\begin{aligned} &\mu_{C\cap S\cap M}\,(62, 84, 75) \\ &\quad = \min\{\mu_C\,(62), \mu_S\,(84), \mu_M\,(75)\} \\ &\quad = \min\{0.52, 0.36, 1.00\} = 0.36. \end{aligned} \tag{29}$$

Rule 44 is

$$\begin{aligned} &\mu_{A\cap F\cap M}\,(62, 84, 75) \\ &\quad = \min\{\mu_A\,(62), \mu_F\,(84), \mu_M\,(75)\} \\ &\quad = \min\{0.48, 0.64, 1.00\} = 0.48. \end{aligned} \tag{30}$$

Table 5: Membership degree for activated variables.

Activated variable	Set, output variable	Membership degree
32	Requires improvement	0.52
35	Complies	0.36
44	Complies	0.48
47	Complies	0.36

Table 6: Membership degree for activated variables.

Activated variable	Fuzzy set	Membership degree
—	Exceeds	0
35, 44, 47	Complies	0.48
32	Requires improvement	0.52

Rule 47 is

$$\begin{aligned} &\mu_{A\cap S\cap M}\,(62, 84, 75) \\ &\quad = \min\{\mu_A\,(62), \mu_S\,(84), \mu_M\,(75)\} \\ &\quad = \min\{0.48, 0.36, 1.00\} = 0.36. \end{aligned} \tag{31}$$

The rules 32, 35, 44, and 47 are the activated variables; they have correspondence with a set in the output variable (see Table 5).

Three activated variables are in the "Obey" set; it is necessary to use the min.-max. criterion to select one value (see (32)):

$$\begin{aligned} \mu_C &= (62, 84, 75) \\ &= \vee\{\wedge\{\mu_C\,(62), \mu_S\,(84), \mu_M\,(75)\}, \\ &\qquad \wedge\{\mu_{AC}\,(62), \mu_F\,(84), \mu_M\,(75)\}, \\ &\qquad \wedge\{\mu_{AC}\,(62), \mu_S\,(84), \mu_M\,(75)\}\} \\ &= 0.48. \end{aligned} \tag{32}$$

The last membership degrees and activated variables are summarized as follows (see Table 6).

(7) Defuzzification process: this procedure includes five steps.

Divide the total area in partial areas (see Figure 14).

Compute the partial area values (see (33)):

$$\begin{gathered} (A_{\text{I}}) = \frac{[0.52]^2\,(62.5 - 50)}{2} = 1.69, \\ (A_{\text{II}}) = [0.52]\,[1 - 0.52]\,(62.5 - 50) = 3.12, \\ (A_{\text{III}}) = \frac{[0.48]\,(62.5 - 50)}{2} = 3, \\ (A_{\text{IV}}) = [0.48]\,(87.5 - 62.5) = 12, \\ (A_{\text{V}}) = [0.48]\,[1 - 0.48]\,(100 - 87.5) = 3.12, \\ (A_{\text{VI}}) = \frac{[0.48]^2\,(100 - 87.5)}{2} = 1.44. \end{gathered} \tag{33}$$

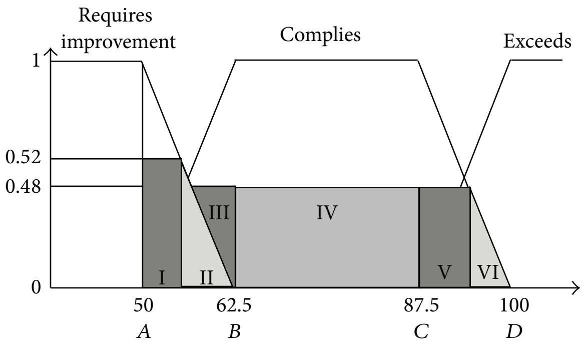


Figure 14: Partial areas.

Compute each partial area centroid (see (34)):

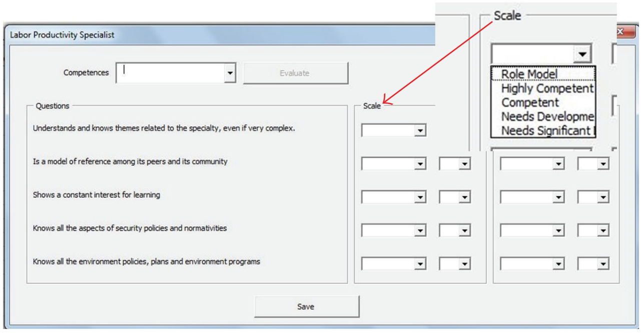
$$\begin{aligned}
\text{Centroid } (A_{\text{I}}) &= 50 + \frac{(62.5 - 50)(1 - 0.52)}{2} = 53, \\
\text{Centroid } (A_{\text{II}}) &= 62.5 - \frac{2(62.5 - 50)[0.52]}{3} = 58.1666, \\
\text{Centroid } (A_{\text{III}}) &= 62.5 - \frac{(62.5 - 50)[0.48]}{3} = 60.5, \\
\text{Centroid } (A_{\text{IV}}) &= \frac{62.5 + 87.5}{2} = 75, \\
\text{Centroid } (A_{\text{V}}) &= 87.5 + \frac{(100 - 87.5)(1 - 0.48)}{2} = 90.75, \\
\text{Centroid } (A_{\text{VI}}) &= 100 - \frac{2(100 - 87.5)[0.48]}{3} = 96.
\end{aligned} \tag{34}$$

Calculate the total centroid (see (35)):

$$\begin{aligned}
\text{CeT} = {} & ((1.69)(53) + (3.12)(58.16) + (3)(60.5) \\
& + (12)(75) + (3.12)(90.75) + (1.44)(96)) \\
& \times (1.69 + 3.12 + 3 + 12 + 3.12 + 1.44)^{-1} \\
= {} & \frac{1773.93}{24.37} = 72.79.
\end{aligned} \tag{35}$$

(8) Competences questionnaires have more than one question. In this sense, each questionnaire will have as many fuzzy treatments as questions. At the end of the process, an average qualification for each competence will be obtained.

(9) Evaluation software and evaluation report. Intelligent evaluation systems have questionnaires that emulate human thought since it gives the option to answer questions using linguistic labels (see Figure 15). Once the evaluator has selected a label, it selects a numerical value. Final stage includes an evaluation report,which includes a summarized table with average qualifications for each competence.



Understands and knows themes related to the specialty, even if very complex.

Is a model of reference among its peers and its community

Shows a constant interest for learning

Knows all the aspects of security policies and normativities

Knows all the environment policies, plans and environment programs

Figure 15: Evaluation questionnaire.

Table 7: Cardinal competences and required level.

Cardinal competences	Required level
Work quality	100%
Pressure tolerance	75%
Communication	75%
Contact modalities	100%
Analytic skill	75%

3. Results

3.1. 360° Methodology Application. A complete application includes four-position analysis (see Figure 13); five cardinal competences were developed using the job description (see Table 7). The first position has nine specific competences, while the second position has ten specific competences (see Table 8), the third position has eight competences, and finally the fourth position has ten competences. Required Levels were established with the company.

Applying the complete traditional methodology 360° to the first position, we concluded that the company must improve "work quality" and "contact modalities," since these competences obtained the major difference with the Required Level (see Table 9).

The individual report includes specific competences summary.

Table 8: Specific competences and required level.

Specific competences			
Position level I	Required level	Position level II	Required level
(1) Work team	100%	(1) Planning and organization skills	100%
(2) Negotiation	100%	(2) Negotiation	50%
(3) Personnel development	75%	(3) Initiative autonomy	100%
(4) Leadership	100%	(4) Leadership	100%
(5) Frankness-trustworthy-integrity	50%	(5) Frankness-trustworthy-integrity	50%
(6) Commitment	75%	(6) Empowerment	100%
(7) Collaboration	50%	(7) Strategic thinking	100%
(8) Coaching	75%	(8) Commitment level-personal discipline-productivity	100%
(9) Decision making	100%	(9) Making decisions	100%
		(10) Human Resources strategic development	50%

Table 9: Cardinal competences report.

	Cardinal competences				
	Work quality	Pressure tolerance	Communication	Contact modalities	Analytic skill
Self-evaluation	71.25	61.25	53.75	67.5	58.75
Peer	21.25	16.25	13.75	41.25	51.25
Head	25.00	43.75	17.50	15.00	16.25
360° weighted	39.17	40.42	28.33	41.25	42.08
Required level	100.00	75.00	75.00	100.00	75.00
Difference: required level, 360° weighted	60.83*	34.58	46.67	58.75*	32.92

*Competences with major difference.

Table 10: Specific competences report.

	Specific competences							
	Planning and organization ability	Client orientation	Productivity	Credibility technique	Innovation	Empowerment	Collaboration	Commitment level-personal discipline-productivity
Self-evaluation	63.75	43.75	60.00	26.25	72.50	25.00	37.50	22.50
Peer	48.75	61.25	56.25	48.75	41.25	37.50	63.75	62.50
Boss	12.50	18.75	17.50	13.75	6.25	5.00	15.00	13.75
360° weighted	41.67	41.25	44.58	29.58	40.00	22.50	38.75	32.92
Required level	75.00	50.00	100.00	75.00	100.00	50.00	100.00	100.00
Difference: required level, 360° weighted	33.33	8.75	55.42	45.42	60.00	27.50	61.25*	67.08*

*Competences with major difference.

Applying the complete traditional methodology 360° to the same post, we concluded that an employee must improve "collaboration" and "commitment level-personal discipline-productivity," since these competences obtained the biggest difference with the Required Level (see Table 10).

3.2. Fuzzy Logic Model Application. Applying the full fuzzy logic model evaluation to the first position, we concluded that the company must improve "communication" and "contact modalities," since these competences obtained the lower final qualification.

Employee must focus on "credibility technique" and "commitment level-personal discipline-productivity" (see Figure 16).

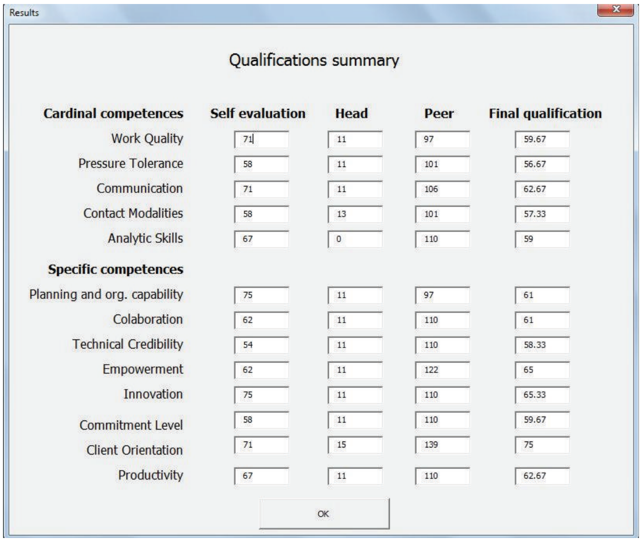
4. Discussion

Traditional methodology 360° indicates competences that the company and the employee must care. Fuzzy logic 360° methodology indicates the competences on which the company and the employees must focus their efforts but in addition gives a qualification. This qualification can be arranged and the lowest value will indicate which competences must be immediately attended.

According to Table 11, the traditional 360° and fuzzy logic methodologies conclude that the company must attend the cardinal competence "contact modalities" and the employee must focus on the specific competence "commitment level-personal discipline-productivity." However, fuzzy logic

Table 11: Comparison results.

Competences classification	Traditional 360° Methodology	Fuzzy Logic 360° Expert system
Cardinal	∗Work quality ∗**Contact modalities**	∗Communication: 56.67 ∗**Contact modalities:** 57.33
Specific	∗Collaboration ∗**Commitment level-personal discipline-productivity**	∗Technical Credibility: 58.33 ∗**Commitment level-personal discipline-productivity:** 59.67



Qualifications summary

Cardinal competences	Self evaluation	Head	Peer	Final qualification
Work Quality	71	11	97	59.67
Pressure Tolerance	58	11	101	56.67
Communication	71	11	106	62.67
Contact Modalities	58	13	101	57.33
Analytic Skills	67	0	110	59
Specific competences				
Planning and org. capability	75	11	97	61
Colaboration	62	11	110	61
Technical Credibility	54	11	110	58.33
Empowerment	62	11	122	65
Innovation	75	11	110	65.33
Commitment Level	58	11	110	59.67
Client Orientation	71	15	139	75
Productivity	67	11	110	62.67

Figure 16: Qualifications summary.

method indicates that "contact modalities" must be attended after "communication," and "commitment level-personal discipline-productivity" must be attended after "technical credibility."

The traditional 360° system involves only two factors, "Scale" and "Frequency," and these are compared with the third factor "Required Level." On the other hand, fuzzy logic model can involve three factors like input variables.

Evaluation questionnaires of the traditional 360° methodology are filled with five possible percentages (100, 75, 50, 25, and 0). On the other hand, questionnaires in fuzzy logic expert system have adjustable values scales, allowing improved flexibility for the evaluator.

Thus, the main advantage in fuzzy logic expert system is the human thinking simulation, assigning labels as a qualification, and allowing subjective and ambiguity treatment.

5. Conclusions

Competences performance evaluation 360° is a complete system since it involves different viewpoints to appraise personnel performance. It allows better interpretations, since the evaluation responsibility falls in different evaluators. This kind of evaluation facilitates the competence concept comprehension, and Required Levels are assimilated. Nevertheless, it is a rigid system because the questionnaire filling procedure is strict, since these were designed to assign fixed values.

Fuzzy logic competences evaluation expert system includes complex analysis, due to identification and modeling input and output variables. However, its main advantage is the ambiguity and subjectivity handling, since the evaluator can assign words to stand a qualification. This system is flexible because numerical adjustable values can be assigned to behaviors. Graphical interpretation helps to obtain suitable feedback. Even more, final processing reports can be obtained easier and faster. Therefore, it represents an excellent tool for competences monitoring, given the importance that the staff appraisal process has for human resource management, in the areas of recruitment and selection, job evaluation, identification of training needs, and so forth, and the value that results from having nonsubjective and bias-free assessments.

The application of an expert system to performance evaluation in a Mexican manufacturing company allows knowing its effectiveness against traditional techniques.

This work brings innovative contributions to soft computing and human resources management solutions, finding new ways to apply artificial intelligence techniques by means of computer applications to processes that typically were performed by humans.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] B. S. Butkiewicz, "Selection of staff for enterprise using fuzzy logic," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, Warsaw University of Technology, Warsaw, Poland, 2002.

[2] E. Gómez, *Modelo difuso de planeación agregada [Doctoral thesis]*, Centro de Ingeniería y Desarrollo Industrial, Querétaro, México, 2009.

[3] L. A. Zadeh, "Making computers think like people," *IEEE Spectrum*, vol. 21, no. 8, pp. 26–32, 1984.

[4] A. Cannavacciuolo, G. Capaldo, A. Ventre, and G. Zollo, "Linking the fuzzy set theory to organizational routines. A study in personnel evaluation in a large company," in *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, pp. 667–672, San Francisco, Calif, USA, April 1993.

[5] A. Cannavacciuolo, G. Capaldo, A. Ventre, and G. Zollo, "An approach to the evaluation of human resourses by using fuzzy set theory," in *Proceedings of the 3rd IEEE Conference on Fuzzy Systems*, Orlando, Fla, USA, 1994.

[6] J. Stahl, "A method for personnel selection in concurrent engineering using fuzzy sets," in *Fuzzy Sets in Engineering Design and Configuration*, H.-J. Sebastian and E. K. Antonsson, Eds., Kluwer Academic Publishers, Boston, Mass, USA, 1996.

[7] C. Argyris and D. A. Schon, *Organizational Learning. A Theory of Action Perspective*, Addison-Wesley, Reading, Mass, USA, 1978.

[8] K. S. Cameron, "Effectiveness as paradox: consensus and conflict in conceptions of organizational effectiveness," *Journal of Management Science*, vol. 32, no. 5, pp. 539–553, 1986.

[9] D. J. Orton and K. E. Weik, "Loosely coupled system: a reconceptualization," *Academy of Management Review*, vol. 15, no. 2, pp. 203–223, 1990.

[10] C. R. Schwenk, "Cognitive simplification processes in strategic decision-making," *Strategic Management Journal*, vol. 5, pp. 111–128, 1984.

[11] K. E. Weick, "Educational organizations as loosely coupled systems," *Administrative Science Quarterly*, vol. 21, pp. 1–19, 1976.

[12] C. Carisson, "On the relevance of fuzzy sets in management science methodology," in *Fuzzy Sets: Theory and Applications to Policy Analysis and information System*, P. P. Wang and S. K. Chang, Eds., Plenum Press, New York, NY, USA, 1980.

[13] B. R. Gaines, L. A. Zadeh, and H. Zimmermann, "Fuzzy sets and decision analysis—a perspective," in *Fuzzy sets and Decision Analysis*, H. J. Zimmermann and etal, Eds., North-Holland, Amsterdam, Netherlands, 1984.

[14] R. M. Tong and P. P. Bonissone, "Linguistic solutions to fuzzy decision problems," in *Studies in the Management Science*, H. J. Zimmermann, Ed., pp. 323–324, 1984.

[15] R. R. Yager, "Fuzzy thinking as quick and efficient," *Cybernetica*, vol. 23, no. 4, pp. 265–298, 1980.

[16] P. L. Yu, "Dissolution of fuzziness for better decision-perspective and techniques," in *Fuzzy Sets and Decision Analysis*, J. H. Zimmermann et al., Ed., North-Holland, Amsterdam, The Netherlands, 1984.

[17] S. French, "Fuzzy decision analysis: some criticism," in *Fuzzy sets and Decision Analysis*, H. J. Zimmermann, B. R. Gaines, and L. Zadeh, Eds., Amsterdam, The Netherlands, North Holland Publisher, 1984.

[18] L. A. Zadeh, "Outline of new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, pp. 28–44, 1973.

[19] E. E. Karsak, "A fuzzy multiple objective programming approach for personnel selection," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 2007–2012, Nashville, Tenn, USA, October 2000.

[20] W. J. Kolarik, J. C. Woldstad, S. Lu, and H. Lu, "Human performance reliability: on-line assessment using fuzzy logic," *IIE Transactions*, vol. 36, no. 5, pp. 457–467, 2004.

[21] L. Podofellini, V. Dang, E. Zio et al., "Using expert models in human reliability analysis—a dependence assessment method based on fuzzy logic," *Risk Analysis*, vol. 30, no. 8, pp. 1277–1297, 2010.

[22] K. C. Mittal, A. K. Goel, and P. Mohindru, "Fuzzy multi-criteria decision making (MCDM) in human resource selection procedure—a case study of Indian IT industry," *BVIMR Management Edge*, vol. 6, no. 1, pp. 89–97, 2013.

[23] R. E. García, G. Felix Benjamin, and R. Bello Perez, "Impact assessment of training with fuzzy logic," *Ingeniare. Revista Chilena de Ingeniería*, vol. 22, no. 1, pp. 41–52, 2014.

[24] D. T. Tosti and R. M. Addison, "360-degree feedback: going around in circles?" *Performance Improvement*, vol. 48, no. 3, pp. 36–39, 2009.

[25] R. Hensel, F. Meijers, R. van der Leeden, and J. Kessels, "360 degree feedback: how many raters are needed for reliable ratings on the capacity to develop competences, with personal qualities as developmental goals?" *International Journal of Human Resource Management*, vol. 21, no. 15, pp. 2813–2830, 2010.

[26] L. R. Gomez-Mejia, D. B. Balkin, and R. L. Cardy, *Managing Human Resources*, Prentice Hall, Saddle River, NJ, USA, 2nd edition, 1998.

[27] M. Spencer Lyle and M. Spencer Signe, *Competence at Work: Models for Superior Performance*, 1993.

[28] C. Levy-Leboyer, "Evaluación del personal: los métodos a elegir," Ediciones Díaz de Santos, 1992.

[29] M. A. Alles, *Gestión por Competencias El diccionario*, Granica, Buenos Aires, Argentina, 2nd edition, 2005.

[30] C. Gregory, *Dynamic Economics: Optimization by the Lagrange Method*, Oxford University Press, 1997.

[31] E. Cox, *The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems*, Ap. Professional, 1994.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 238168, 11 pages
http://dx.doi.org/10.1155/2014/238168

Research Article

OrclassWeb: A Tool Based on the Classification Methodology ORCLASS from Verbal Decision Analysis Framework

Thais Cristina Sampaio Machado, Plácido Rogerio Pinheiro, and Isabelle Tamanini

Graduate Program in Applied Informatics, University of Fortaleza (UNIFOR), Avenue Washington Soares 1321, Bl J Sl 30, 60811-905 Fortaleza, CE, Brazil

Correspondence should be addressed to Plácido Rogerio Pinheiro; placidrp@uol.com.br

Received 11 April 2014; Revised 10 July 2014; Accepted 11 July 2014; Published 24 July 2014

Academic Editor: Ker-Wei Yu

The decision making is present in every activity of the human world, either in simple day-by-day problems or in complex situations inside of an organization. Sometimes emotions and reasons become hard to separate; therefore decision support methods were created to help decision makers to make complex decisions, and Decision Support Systems (DSS) were created to aid the application of such methods. The paper presents the development of a new tool, which reproduces the procedure to apply the Verbal Decision Analysis (VDA) methodology ORCLASS. The tool, called OrclassWeb, is software that supports the process of the mentioned DSS method and the paper provides proof of concepts, that which presents its reliability with ORCLASS.

1. Introduction

Present in every human activity, the decision may be defined because of a process of choice, given an identified problem or when the decision maker faces an opportunity of creation, optimization, or improvement in an environment. Aiming to solve complex problems, difficultly to be resolved by human process capacity in a qualitative way, Decision Support Systems (DSS) are methodologies that help a decision maker to rank or classify multiattribute alternatives, according to [1].

On the context of Multiple Criteria Decision Analysis (MCDA) classification approaches, we have several methods to be considered. The MCDA classification approaches can be subdivided into two types [2], described as follows.

(i) Parameter-Based Methods. The approach works with constructing a formal model of the decision maker's behavior based on some parameters, and this approach can be subdivided into *utility function-based* and *outranking relation-based* models [2]. These methods can also be categorized further into direct and indirect. In this area, methods such as ELECTRE TRI, PROAFTN, and UTADIS can be cited, among various others. The work presented in [3] proposes a new classification method based on ECLECTRE IV and on a customized genetic algorithm, which are employed in order to select the prototypes (alternatives that serve as class representatives related to a given problem) and calibrate the control parameters automatically.

(ii) Parameter-Free Methods. This approach is characterized for containing decision rule methods. It assumes that the Decision Maker's behavior is too complex to be mapped as parameters, and thus, proposes that it can be described in a symbolic or judgmental form by means of decision rules [2]. The idea was originated based on machine learning techniques, such as expert systems. Among the approaches that belong to the model, we have the Verbal Decision Analysis (VDA) methods and methods based on fuzzy logic. The VDA framework is structured on the acknowledgment that most of the decision making problems can be verbally described. Among the VDA framework, the first methods proposed, and also the most known, were ORCLASS, PACOM, and ZAPROS [1] and their characteristics and applications can be found in [4–9].

According to [2], methods on MCDA are usually not capable of working with verbal information of preferences, and procedures are applied in order to convert these preferences to a numerical form. However, there might be losses of

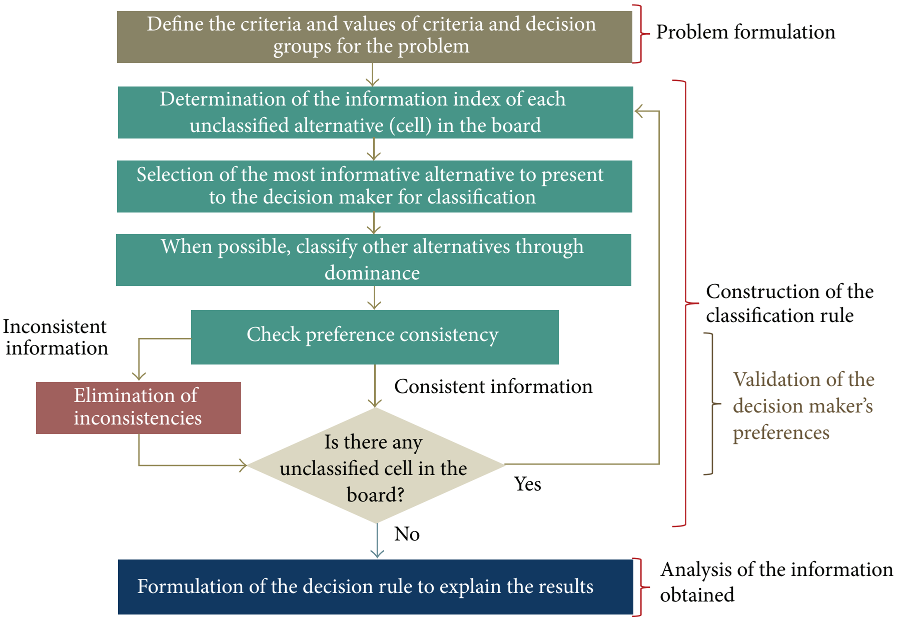

Define the criteria and values of criteria and decision groups for the problem
Problem formulation
Determination of the information index of each unclassified alternative (cell) in the board
Selection of the most informative alternative to present to the decision maker for classification
When possible, classify other alternatives through dominance
Check preference consistency
Inconsistent information
Elimination of inconsistencies
Consistent information
Is there any unclassified cell in the board?
Yes
No
Construction of the classification rule
Validation of the decision maker's preferences
Formulation of the decision rule to explain the results
Analysis of the information obtained

Figure 1: Procedure to apply the ORCLASS method.

information when trying to assign numerical values to verbal measurements. Thus, procedures that are able to work with verbal information of preferences are more consistent and stable [10]. Thus, this work focuses on the Verbal Decision Analysis methods [1] which are categorized as nonclassical approaches of MCDA, in particular, the ORCLASS method.

Regarding the choice of the multicriteria approaches, an important point is to evaluate the problem, the objects of decision and information available. According to [11], the choice of the method should be the result of an evaluation of the chosen parameters, the type and accuracy of the data, the way of thinking of the decision maker, and his/her knowledge of the problem. It is noteworthy that the direct consequence of the choice between various methods is that the results can be uneven and even contradictory. Even as, it does not complicate the evaluation, since the differences are more related to diversity that results contradictions and there are some criteria to validate the chosen method [11].

In the issue in question, the application of method ORCLASS came to acceptance by the decision maker, which meant that the issues that were being presented to the decision maker made sense to him, and he had confidence in answering them. Beyond this point, extolled the need to assess the acceptance of the data, its properties used by the method, and the results supported the decision process. Secondary issues such as the existence of tools were also observed, as those would allow greater integration with the problem addressed. Still in the view of the authors, they agree with [12], which stressed that the methodology for multicriteria decision has several methods that can be applied in various problems.

This way, objecting to solve complex problems and provide better utilization of ORCLASS, the new tool OrclassWeb was developed to reproduce the mentioned methodology automatically.

2. ORCLASS Method—Overview and Structure

The ORCLASS method (Ordinal Classification) is focused in classifying multicriteria alternatives, which are the options for solving the problem and can be characterized according to the criteria provided. The method differs from other Verbal Decision Analysis methods (such as ZAPROS and PACOM) because it does not aim at ordering alternatives but at classifying all possible multicriteria alternatives, which are given by the Cartesian product of the criteria values defined for the problem. The method aims at categorizing the alternatives into a small number of decision classes or groups, which are preordered according to the decision maker's preferences [1].

2.1. ORCLASS Structure. In fact, many different methods for solving multicriteria classification problems are widely known. The ordinary classification method ORCLASS was one of the first methods designed to solve these kinds of problems. Moreover, recent methods appeared, such as SAC and CYCLE [13]. ORCLASS methodology aims at classifying the alternatives in a given set: the decision maker needs these alternatives to be categorized into a small number of decision classes or groups, usually two. The first group covers the most preferable alternatives, and the less preferable alternatives belong to the second one. However, more than two classes might be defined for a problem. According to [8], Figure 1 presents a flowchart with steps to apply the method ORCLASS. In accordance with the scheme described in Figure 1, the application of the method can be divided into three stages: Problem Formulation, Structuring of the Classification Rule, and Analysis of the Information Obtained.

2.2. Formal Statement of the Problem. The methodology follows the same problem formulation proposed in [1].

Given. (1) $K = 1, 2, \dots, N$, representing a set of N criteria.

(2) n_q represents the number of possible values on the scale of qth criterion ($q \in K$); for the ill-structured problems, as in this case, usually $n_q \leq 4$.

(3) $X_q = \{x_{iq}\}$ represents a set of values to the qth criterion, which is this criterion scale; $|X_q| = n_q$ ($q \in K$); the values of the scale are ranked from best to worst, and this order does not depend on the values of other scales.

(4) $Y = X_1 * X_2 * \cdots * X_n$ represents a set of vectors y_i (every possible alternative: hypothetical alternatives + real alternatives), such that $y_i = (y_{i1}; y_{i2}; \dots; y_{iQ})$, and $y_i \in Y$, $y_{iq} \in X_q$, and $Q = |Y|$, such that $|Y| = \prod_{q=1}^{Q} n_q$.

(5) $A = \{a_i\} \in Y$, $i = 1, 2, \dots, t$, such that the set of t vectors represents the description of the real alternatives.

(6) L is number of ordered decision classes.

Required. The multicriteria alternatives classification based on the decision maker's preferences (judgment) to build a reflection.

2.3. Method's Application. At first, in the problem's formulation stage, the set of criteria, criteria values, and the decision groups are defined. The criteria values must be sorted in ascending order of preference (from most preferable to least).

Then, the construction of the classification rule will be carried out based on the decision maker's preferences. In order to present the method's application in a comprehensive way, we use the same concepts presented in [1], on which a classification task is presented as a set of boards.

Each cell is composed by a combination of values for each criterion defined to the problem, which represents a possible alternative to the problem. Let us consider a problem containing the criteria A, B, and C, the criteria values: A = {A1, A2, A3}, B = {B1, B2, B3}, C = {C1, C2, C3}, and two decision classes (I and II), so that, according to the decision maker, the first class is preferable to the second one. For the given problem, we can say the following.

(i) An alternative composed by the best characteristics ([A1, B1, C1]) will always belong to Class I.

(ii) An alternative composed by the worst characteristics ([A3, B3, C3]) will always belong to Class II.

As defined in [1] if a possible alternative composed by the criterion values [A1, B1, C3] is presented to the decision maker for judgment, for example, and supposing that the decision maker classifies the alternative in the first group, then we can infer that a better alternative as [A1, B1, C2] certainly belongs to the first group, since the last one is naturally more preferable than the previous. This response would fill two cells of the classification board. However, if the decision maker judges that the alternative from the example [A1, B1, C3] is classified into the second group, then the worst alternatives also will be classified as group 2. This way, the alternatives [A1, B2, C3], [A1, B3, C3], [A2, B1, C3], [A2, B2, C3], [A2, B3, C3], [A3, B1, C3], and [A3, B2, C3] will belong to the second decision group, since they are naturally less preferable than the previous. This response would fill eight cells of the classification board.

TABLE 1: Classification boards composed by the amount of generated information.

(a) C1

	B1	B2	B3
A1	I	1 + 17	2 + 8
A2	1 + 17	3 + 11	5 + 5
A3	2 + 8	5 + 5	8 + 2

(b) C2

	B1	B2	B3
A1	1 + 17	3 + 11	5 + 5
A2	3 + 11	7 + 7	11 + 3
A3	5 + 5	11 + 3	17 + 1

(c) C3

	B1	B2	B3
A1	2 + 8	5 + 5	8 + 2
A2	5 + 5	11 + 3	17 + 1
A3	8 + 2	17 + 1	II

The additional cells are classified based on dominance relations when the decision maker classifies a determined cell of the table. Moreover, Table 1 presents the classification board for illustrating and better visualization of the amount of cells that could be additionally classified based on the decision maker's preferences. The left number on each cell represents the amount of cells that would be additionally classified in case that cell belongs to class I, and the right number represents the number of additionally classified cells in case that cell belongs to class II. These numbers represent the *informative index* of each cell.

Only the combinations that, when classified, will reflect this information to a subset of the possible alternatives will be presented to the decision maker for classification, and only the cell with the most informative index will be considered. The one that presents the smallest difference between the indexes and the maximum sum of these indexes will give the most informative cell. In case of a tie between the alternatives, one can randomly choose one of them. In conclusion, the most informative cell is given by [A2, B2, C2] (which enables seven new classifications regardless of which class it belongs to), and it is the best option to be presented to the decision maker for classification. Supposing that cell [A2, B2, C2] was classified as class I, Table 2 presents the new informative indexes after this classification.

In addition, when the problem contains only two classes, it is necessary to check the consistency of the decision maker's answers. On the detection of inconsistent information, these should be presented to the decision maker for analysis and correction. The checking can be made by asking the decision

Table 2: Classification boards composed by the amount of generated information once the first classification is made.

(a) C1

	B1	B2	B3
A1	I	I	1 + 8
A2	I	I	2 + 5
A3	1 + 8	2 + 5	5 + 2

(b) C2

	B1	B2	B3
A1	I	I	2 + 5
A2	I	I	4 + 3
A3	2 + 5	4 + 3	10 + 1

(c) C3

	B1	B2	B3
A1	1 + 8	2 + 5	5 + 2
A2	2 + 5	4 + 3	10 + 1
A3	5 + 2	10 + 1	II

maker to classify an already classified cell on the board. The classification should respect the dominance relations, so that an alternative with better criteria values should not be classified into a group that is considered not preferable from the decision maker's point of view, once, at the same time, the group that is considered better contains alternatives with worse criteria values. In problems with more than two classes, the check of consistency is automatically performed within the classification task. More information about the ORCLASS method is available in [1].

2.4. Tools Based on Verbal Decision Analysis. There are a few tools for automating Verbal Decision Analysis (VDA) methodologies. For methods that aim at the ordering of alternatives, for example, we can cite the one described in detail by Dimitriadi and Larichev [14], which illustrates an implementation of a different version of ZAPROS-III in Visual C++ 6.0, not applying Formal Index of Quality (FIQ) for alternatives comparing process. Still for ordering of alternatives purposes, we have UniComBOS [15], which applies its own method to determine the order of given alternatives, implementing a new procedure for comparison and choice of multicriteria alternatives. As for the ZAPROS-III method, Tamanini [8] implements a tool focused on treatment of incomparability. The work reduces problems of incomparability and improves the method ZAPROS-III [16, 17], generating the new approach methodology ZAPROS III-*i*.

In the alternatives classification scenario, a tool for the method SAC (or CLARA) is presented in [18]. Different from the ORCLASS method, it aims at classifying the alternatives that are relevant to the problem being approached. It also differs from the ORCLASS one on the form that it calculates the information index of the alternatives: it considers a variation measurement of the number of indirectly classified alternatives. Some recommendations for the variance values are presented in [2]. When the variance value is 0, the information index of the alternatives is calculated on the same way of the ORCLASS method.

Thus, the motivation of creating a new tool to implement the ORCLASS method is justified by the fact that, for research purposes, the authors wanted to have a final classification rule that would apply to any alternative that is described based on the established criteria values. This way, the multicriteria models would be more generic and the alternatives could be changed without impacting on the final results.

3. A Tool for the ORCLASS Method—The OrclassWeb Tool

In order to facilitate the decision making process using ORCLASS and perform it consistently, observing its rules and aiming at making it accessible, a tool developed in platform Java Web that applies the methodology is presented [19]. The tool was built in a web environment in Platform Java 1.6 using JSF 2 and runs in server Tomcat 6. OrclassWeb tool was proposed to automate the comparison process of alternatives and to provide the decision maker a concrete result for the problem, according to ORCLASS definition. OrclassWeb was developed divided into four stages:

(1) criteria and criteria values definition,

(2) alternatives definition,

(3) construction of the classification rule,

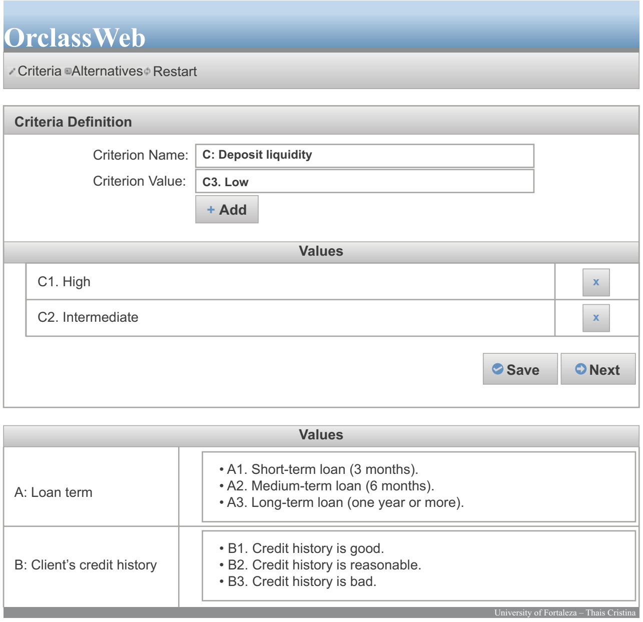
(4) presentation of results obtained.

The manual application of the system ORCLASS is usually made with a maximum of three criteria and three criteria values for each one, because the complexity of the application increases exponentially with the growth of these numbers. This way, the main advantage of OrclassWeb is that the tool processes the complexity of the application, which means that the user can apply ORCLASS with a greater number of criteria and criteria values. OrclassWeb was developed adapting the rules to identify the most informative cell, after applying the rules defined by [1]:

> After the identification of the most informative index according to the rules, the tool verifies among all the others alternatives, which one presents the highest values for both indexes; being this the most informative cell of the board.

In conclusion, the adaptation was necessary to increase the method's comparison capacity, without giving away the adherence to the ORCLASS method. The tool's interfaces and its features are presented in Sections 3.1 and 3.2, which describe the stages of the ORCLASS application. For research purposes, the tool can be reached at http://www2.unifor.br/OrclassWeb.

3.1. OrclassWeb: Interfaces and Availability. First of all, at the beginning of the application it is necessary to define the criteria and criteria values associated that the alternatives will be compared against, presented in the problem. Figure 2



Criteria Definition

Criterion Name: C: Deposit liquidity

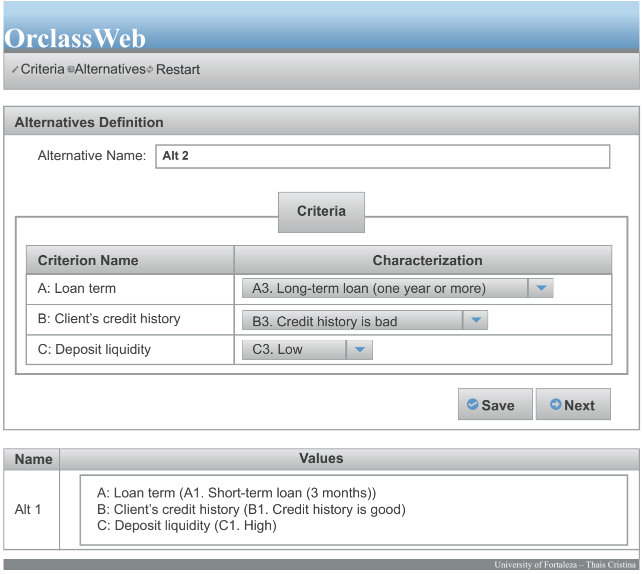
Criterion Value: C3. Low

Values	
C1. High	x
C2. Intermediate	x

Values	
A: Loan term	• A1. Short-term loan (3 months). • A2. Medium-term loan (6 months). • A3. Long-term loan (one year or more).
B: Client's credit history	• B1. Credit history is good. • B2. Credit history is reasonable. • B3. Credit history is bad.

University of Fortaleza – Thais Cristina

Figure 2: Criteria definition interface: who is worth the loan?



Alternatives Definition

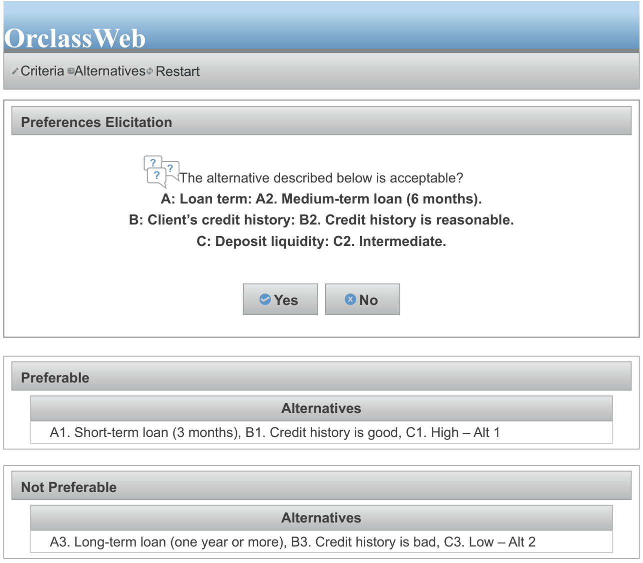
Alternative Name: Alt 2

Criterion Name	Characterization
A: Loan term	A3. Long-term loan (one year or more)
B: Client's credit history	B3. Credit history is bad
C: Deposit liquidity	C3. Low

Name	Values
Alt 1	A: Loan term (A1. Short-term loan (3 months)) B: Client's credit history (B1. Credit history is good) C: Deposit liquidity (C1. High)

University of Fortaleza – Thais Cristina

Figure 3: Alternatives definition interface: who is worth the loan?



Preferences Elicitation

The alternative described below is acceptable?

A: Loan term: A2. Medium-term loan (6 months).

B: Client's credit history: B2. Credit history is reasonable.

C: Deposit liquidity: C2. Intermediate.

Preferable

Alternatives
A1. Short-term loan (3 months), B1. Credit history is good, C1. High – Alt 1

Not Preferable

Alternatives
A3. Long-term loan (one year or more), B3. Credit history is bad, C3. Low – Alt 2

Figure 4: Preferences elicitation interface: who is worth the loan?

presents the tool's interface for defining criteria and criteria values. This is the stage named "Problem Formulation" from the procedure. The second stage of OrclassWeb is the definition of the alternatives. This is the step which the decision maker states the alternatives and characterize them according to the criteria values. Figure 3 presents the tool's interface for defining alternatives. This is the phase named "Construction of the Classification Rule" of the procedure.

The next step of OrclassWeb is the elicitation of preferences stage. The interface for elicitation of preferences presents questions that can be easily answered by the decision maker in order to obtain the scale of preferences. According to the decision maker's responses, OrclassWeb initiates the division of alternatives into the classes/groups. Figure 4 presents the tool's interface for capturing preferences. The current step also belongs to "Construction of the Classification Rule" phase from the procedure. As the last step for applying the method, at the end of the comparison process OrclassWeb presents an interface composed by the groups and the related alternatives preferred for each one in order to allow a complete and more detailed analysis of the results obtained. Table 3 presents the tool's interface for the analysis results. This interface provides the visualization of "Analysis of Information Obtained" stage from procedure.

OrclassWeb is a tool developed for web, not a Desktop application. It means that the system can be accessed everywhere, via Internet, through the website.

3.2. OrclassWeb: Reliability and Exemplification of Use. Functional tests were elaborated and executed aiming at testing the OrclassWeb and stating it as a reliable and effective tool. The tests were done by applying ORCLASS system manually and were also applied through the tool. The results are compared in order to identify that OrclassWeb attends to all the requirements imposed by the methodology. As example of applications that used the tool to aid in the method's application, we have the applications presented in [20, 21].

For the following section, we show the tool's application to the problem presented in [1]. The application occurs exactly as defined in the procedure for the classification methodology ORCLASS. After that, we provide an original multicriteria problem to exemplify the use of the tool. The results obtained with OrclassWeb will be compared to the ones obtained through the manual application of the method to problems previously published.

3.2.1. Application of the Tool to the Multicriteria Problem: Who Is Worth the Loan? The first problem stated used to

Table 3: Results presentation interface: who is worth the loan?

Final result
Preferable
A1. Short-term loan (3 months). B1. Credit history is good. C1. High—Alt 1
A2. Medium-term loan (6 months). B2. Credit history is reasonable. C2. Intermediate.
A1. Short-term loan (3 months). B2. Credit history is reasonable. C2. Intermediate.
A2. Medium-term loan (6 months). B1. Credit history is good. C2. Intermediate.
A2. Medium-term loan (6 months). B2. Credit history is reasonable. C1. High.
A1. Short-term loan (3 months). B1. Credit history is good. C2. Intermediate.
A1. Short-term loan (3 months). B2. Credit history is reasonable. C1. High.
A2. Medium-term loan (6 months). B1. Credit history is good. C1. High.
A3. Long-term loan (one year or more). B1. Credit history is good. C2. Intermediate.
A3. Long-term loan (one year or more). B1. Credit history is good. C1. High.
A1. Short-term loan (3 months). B3. Credit history is bad. C1. High.
Not preferable
A3. Long-term loan (one year or more). B3. Credit history is bad. C3. Low—Alt 2
A3. Long-term loan (one year or more). B2. Credit history is reasonable. C2. Intermediate.
A3. Long-term loan (one year or more). B3. Credit history is bad. C2. Intermediate.
A3. Long-term loan (one year or more). B2. Credit history is reasonable. C3. Low.
A1. Short-term loan (3 months). B3. Credit history is bad. C3. Low.
A2. Medium-term loan (6 months). B3. Credit history is bad. C3. Low.
A2. Medium-term loan (6 months). B3. Credit history is bad. C1. High.
A3. Long-term loan (one year or more). B3. Credit history is bad. C1. High.
A2. Medium-term loan (6 months). B3. Credit history is bad. C2. Intermediate.
A2. Medium-term loan (6 months). B1. Credit history is good. C3. Low.
A3. Long-term loan (one year or more). B1. Credit history is good. C3. Low.
A2. Medium-term loan (6 months). B2. Credit history is reasonable. C3. Low.
A3. Long-term loan (one year or more). B2. Credit history is reasonable. C1. High.
A1. Short-term loan (3 months). B2. Credit history is reasonable. C3. Low.
A1. Short-term loan (3 months). B1. Credit history is good. C3. Low.
A1. Short-term loan (3 months). B3. Credit history is bad. C2. Intermediate.

exemplify the tool's use is the evaluation of loan applications defined in [1]. This problem was used originally by [1] to explain ORCLASS method application procedure. The situation consists in a group of businesses submitting a loan application to a commercial bank. In what cases the loan should be rejected or accepted? The bank is the decision maker and provides a policy for granting loans. The decision maker is the person to whom the decision will be taken, and the structuring of the classification rule in the OrclassWeb was done based on the rule already defined on the original problem resolution. Table 4 presents the list of criteria and criteria values, which will be base to apply the methodology. The criteria values are described from the naturally most preferable to the least preferable one.

Criteria and criteria values are required by the tool. The criteria values need to be entered in order, according to their preference degrees, through the "Criteria Definition Interface," shown in Figure 2.

The alternatives for the application will be hypothetical situations, since the problem definition in [1] does not describe the alternatives. Table 5 presents the list of alternatives for this case.

Table 4: Criteria and associated values: who is worth the loan?

Criteria	Criteria values
A: Loan term	A1. Short-term loan (3 months). A2. Medium-term loan (6 months). A3. Long-term loan (one year or more).
B: Client's credit history	B1. Credit history is good. B2. Credit history is reasonable. B3. Credit history is bad.
C: Deposit liquidity	C1. High. C2. Intermediate. C3. Low.

Alternatives are also required objects by the tool, and they should have a name and be characterized based on criteria values. The "Alternatives Definition Interface" of OrclassWeb can be visualized in Figure 3.

Table 5: Identification of alternatives board: who is worth the loan?

ID	Alternatives
Alt 1	Represents the best alternative
Alt 2	Represents the worst alternative

Table 6: Result matrix: who is worth the loan?

(a) C1

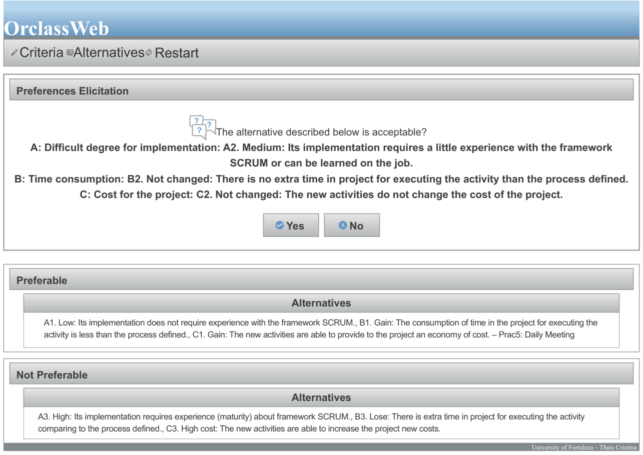
	B1	B2	B3
A1	I	I	I
A2	I	I	II
A3	I	II	II

(b) C2

	B1	B2	B3
A1	I	I	II
A2	I	I	II
A3	I	II	II

(c) C3

	B1	B2	B3
A1	II	II	II
A2	II	II	II
A3	II	II	II



Preferences Elicitation

The alternative described below is acceptable?

A: Difficult degree for implementation: A2. Medium: Its implementation requires a little experience with the framework SCRUM or can be learned on the job.

B: Time consumption: B2. Not changed: There is no extra time in project for executing the activity than the process defined.

C: Cost for the project: C2. Not changed: The new activities do not change the cost of the project.

Preferable

Alternatives

A1. Low: Its implementation does not require experience with the framework SCRUM., B1. Gain: The consumption of time in the project for executing the activity is less than the process defined., C1. Gain: The new activities are able to provide to the project an economy of cost. – Prac5: Daily Meeting

Not Preferable

Alternatives

A3. High: Its implementation requires experience (maturity) about framework SCRUM., B3. Lose: There is extra time in project for executing the activity comparing to the process defined., C3. High cost: The new activities are able to increase the project new costs.

Figure 5: Preferences elicitation interface: selecting practices of Framework SCRUM.

After all the criteria and alternatives are defined, the user's preferences will be accessed through the "Preferences Elicitation Interface," as presented in Figure 4. Note that the user is not required to have a deep knowledge of the method to be able to use the tool. The information indexes are used at this part of the process, following the rules presented in [1].

The tool conducts the entire process of elicitation of preferences. The process can be applied manually using a board that will be filled according to the decision maker preferences. For both cases, the logic involved is the same: the application of ORCLASS is made calculating the most informative index, which will provide the next question to be posed to the decision maker. Afterwards, the most informative cell from the board is exactly the next question presented to the user of OrclassWeb. Comparing both application techniques, we notice that the same is obtained and can be visualized in Tables 3 and 6.

The use of the tool facilitates the process since the user does not need to calculate the information indexes manually. For problems having three or more criteria, this task can be very stressful and time consuming. Also, no formal knowledge about the method's application is required once using the tool, since one just needs to follow the steps and answer the required questions.

3.2.2. Application of the Tool to the Multicriteria Problem: Selecting Practices of Framework SCRUM. Considering that agile methodologies, in particular the Framework SCRUM [10, 21–24], are more and more common in development software companies and noticing that the mentioned companies cannot always apply every approach of the framework, the research applies the OrclassWeb tool aiming at selecting the approaches to be applied in the company, considering the elicitation of preferences of a decision maker. The application consists in the classification of SCRUM approaches in groups or classes. The first group was chosen to support the practices of Framework SCRUM which should be selected after the application of ORCLASS, to be implanted by the organizations, and the second class will support the set of approaches that should not be implanted by the organization which desire to implement part of SCRUM. Table 7 presents the list of criteria and criteria values, which will be used to evaluate the set of alternatives. These were required to be entered into the tool according to its preference degrees (following the order shown in the table).

The alternatives for the application will be SCRUM approaches and they are listed in Table 8. The alternatives were entered into the tool through the "Alternatives Definition Interface."

The process of elicitation of preferences is conducted by OrclassWeb, tool in parallel to the application made manually, through "Preferences Elicitation Interface," as presented in Figure 5.

The tool conducts the entire process of elicitation of preferences, and, manually, the process can be applied using a table, which will be filled according to the decision maker preferences. Manually, the application of ORCLASS is made calculating the most informative index, which will provide the next question to be posed to the decision maker. Afterwards, the most informative cell from the board is exactly the next question presented to the user of OrclassWeb. The conclusion of elicitation of preferences is reached in both tools: OrclassWeb and manually. The result is the same and can be visualized in Tables 9 and 10.

Comparing the result obtained in both situations, the first and the second group are composed by the SCRUM approaches as described below.

(i) Class 1 is composed by the following SCRUM approaches:

Table 7: Criteria and associated values: selecting practices of Framework SCRUM.

Criteria	Criteria values
A: Difficult degree for implementation	A1. Low: its implementation does not require experience with the framework SCRUM.
	A2. Medium: its implementation requires a little experience with the framework SCRUM or can be learned on the job.
	A3. High: its implementation requires experience (maturity) about framework SCRUM.
B: Time consumption	B1. Gain: the consumption of time in the project for executing the activity is less than the process defined.
	B2. Not changed: there is no extra time in project for executing the activity compared to the process defined.
	B3. Lose: there is extra time in project for executing the activity comparing to the process defined.
C: Cost for the project	C1. Gain: the new activities are able to provide to the project an economy of cost.
	C2. Not changed: the new activities do not change the cost of the project.
	C3. High cost: the new activities are able to increase the project new costs.

Table 8: Identification of alternatives board: selecting practices of Framework SCRUM.

ID	Alternatives
Prac1	Sprints (or iterations) with 1 to 4 weeks
Prac2	A product backlog and a sprint backlog creation and prioritization
Prac3	Planning meeting—part 1
Prac4	Planning meeting—part 2
Prac5	Daily meeting
Prac6	Burn down chart and visible activities board
Prac7	Sprint review
Prac8	Sprint retrospective
Prac9	Release planning

Table 9: Result matrix: selecting practices of Framework SCRUM.

(a) C1

	B1	B2	B3
A1	I	I	I
A2	I	I	I
A3	II	II	II

(b) C2

	B1	B2	B3
A1	I	I	I
A2	I	I	II
A3	II	II	II

(c) C3

	B1	B2	B3
A1	I	II	II
A2	II	II	II
A3	II	II	II

(a) Prac1: sprints (or iterations) with 1 to 4 weeks,

(b) Prac5: daily meeting,

(c) Prac6: burn down chart and visible activities board,

(d) Prac7: sprint review,

(e) Prac8: sprint retrospective,

(f) Prac9: release planning.

(ii) Class 2 is composed by SCRUM approaches:

(a) Prac2: a product backlog and a sprint backlog creation and prioritization,

(b) Prac3: planning meeting—part 1,

(c) Prac4: planning meeting—part 2.

4. Conclusions and Future Works

The decision may be defined as a result of a process of choice, given an identified problem or when the decision maker faces an opportunity of creation, optimization, or improvement in an environment. Several factors, which might be measurable, usually influence on the decision making process. It means that the decision is made according to the decision maker preferences.

Observing that the decision maker is capable of manifesting his preferences and interests, it is sufficient to solve simple problems. If the decision maker needs to solve complex problems covering many alternatives and much information that cannot be measured and easily compared, there are some methodologies to be applied in order to support the decision making process.

The main contribution described in this work was OrclassWeb tool. The software was developed and examples of the tool usage and functioning were provided. The tool is available for anyone through a website for research. In conclusion, the developed tool OrclassWeb is proved as truly reliable and strong enough to support the methodology of Verbal Decision Analysis and perform it consistently. Other applications using the SCRUM where the OrclassWeb was applied can be found in [20, 25, 26].

As future works, other tools can be developed in order to reproduce other DSS methodologies not provided automatically yet. Furthermore, more research can be done applying the OrclassWeb tool with more than three criteria and criteria values in known published problems. There is also the intention to refine the software in order to provide more

Table 10: Results presentation interface: selecting practices of Framework SCRUM.

Final result
Preferable
A1. Low: its implementation does not require experience with the framework SCRUM. B1. Gain: the consumption of time in the project for executing the activity is less than the process defined. C1. Gain: the new activities are able to provide to the project an economy of cost—Prac5
A2. Medium: its implementation requires a little experience with the framework SCRUM or can be learned on the job. B2. Not changed: there is no extra time in project for executing the activity compared to the process defined. C2. Not changed: the new activities do not change the cost of the project—Prac6
A1. Low: its implementation does not require experience with the framework SCRUM. B2. Not changed: there is no extra time in project for executing the activity compared to the process defined. C2. Not changed: the new activities do not change the cost of the project.
A2. Medium: its implementation requires a little experience with the framework SCRUM or can be learned on the job. B1. Gain: the consumption of time in the project for executing the activity is less than the process defined. C2. Not changed: the new activities do not change the cost of the project.
A2. Medium: its implementation requires a little experience with the framework SCRUM or can be learned on the job. B2. Not changed: there is no extra time in project for executing the activity than the process defined. C1. Gain: the new activities are able to provide to the project an economy of cost.
A1. Low: its implementation does not require experience with the framework SCRUM. B1. Gain: the consumption of time in the project for executing the activity is less than the process defined. C2. Not changed: the new activities do not change the cost of the project.
A1. Low: its implementation does not require experience with the framework SCRUM. B2. Not changed: there is no extra time in project for executing the activity compared to the process defined. C1. Gain: the new activities are able to provide to the project an economy of cost.
A2. Medium: its implementation requires a little experience with the framework SCRUM or can be learned on the job. B1. Gain: the consumption of time in the project for executing the activity is less than the process defined. C1. Gain: the new activities are able to provide to the project an economy of cost—Prac1
A2. Medium: its implementation requires a little experience with the framework SCRUM or can be learned on the job. B3. Lose: there is extra time in project for executing the activity comparing to the process defined. C1. Gain: the new activities are able to provide to the project an economy of cost.
A1. Low: its implementation does not require experience with the framework SCRUM. B3. Lose: there is extra time in project for executing the activity comparing to the process defined. C1. Gain: the new activities are able to provide to the project an economy of cost.
A1. Low: its implementation does not require experience with the framework SCRUM. B1. Gain: the consumption of time in the project for executing the activity is less than the process defined. C3. High cost: the new activities are able to increase the project new costs—Prac8
A1. Low: its implementation does not require experience with the framework SCRUM. B3. Lose: there is extra time in project for executing the activity comparing to the process defined. C2. Not changed: the new activities do not change the cost of the project.
Not preferable
A3. High: its implementation requires experience (maturity) about framework SCRUM. B3. Lose: there is extra time in project for executing the activity comparing to the process defined. C3. High cost: the new activities are able to increase the project new costs—Prac4
A3. High: its implementation requires experience (maturity) about framework SCRUM. B2. Not changed: there is no extra time in project for executing the activity compared to the process defined. C2. Not changed: the new activities do not change the cost of the project.
A3. High: its implementation requires experience (maturity) about framework SCRUM. B3. Lose: there is extra time in project for executing the activity comparing to the process defined. C2. Not changed: the new activities do not change the cost of the project.
A3. High: its implementation requires experience (maturity) about framework SCRUM. B2. Not changed: there is no extra time in project for executing the activity compared to the process defined. C3. High cost: the new activities are able to increase the project new costs.
A1. Low: its implementation does not require experience with the framework SCRUM. B3. Lose: there is extra time in project for executing the activity comparing to the process defined. C3. High cost: the new activities are able to increase the project new costs.
A2. Medium: its implementation requires a little experience with the framework SCRUM or can be learned on the job. B3. Lose: there is extra time in project for executing the activity comparing to the process defined. C3. High cost: the new activities are able to increase the project new costs.

Table 10: Continued.

Final result
A2. Medium: its implementation requires a little experience with the framework SCRUM or can be learned on the job. B1. Gain: the consumption of time in the project for executing the activity is less than the process defined. C3. High cost: the new activities are able to increase the project new costs.
A3. High: its implementation requires experience (maturity) about framework SCRUM. B1. Gain: the consumption of time in the project for executing the activity is less than the process defined. C3. High cost: the new activities are able to increase the project new costs.
A2. Medium: its implementation requires a little experience with the framework SCRUM or can be learned on the job. B2. Not changed: there is no extra time in project for executing the activity compared to the process defined. C3. High cost: the new activities are able to increase the project new costs.
A3. High: its implementation requires experience (maturity) about framework SCRUM. B2. Not changed: there is no extra time in project for executing the activity compared to the process defined. C1. Gain: the new activities are able to provide to the project an economy of cost.
A3. High: its implementation requires experience (maturity) about framework SCRUM. B3. Lose: there is extra time in project for executing the activity comparing to the process defined. C1. Gain: the new activities are able to provide to the project an economy of cost.
A3. High: its implementation requires experience (maturity) about framework SCRUM. B1. Gain: the consumption of time in the project for executing the activity is less than the process defined. C2. Not changed: the new activities do not change the cost of the project.
A2. Medium: its implementation requires a little experience with the framework SCRUM or can be learned on the job. B3. Lose: there is extra time in project for executing the activity comparing to the process defined. C2. Not changed: the new activities do not change the cost of the project.
A3. High: its implementation requires experience (maturity) about framework SCRUM. B1. Gain: the consumption of time in the project for executing the activity is less than the process defined. C1. Gain: the new activities are able to provide to the project an economy of cost.
A1. Low: its implementation does not require experience with the framework SCRUM. B2. Not changed: there is no extra time in project for executing the activity compared to the process defined. C3. High cost: the new activities are able to increase the project new costs—Prac3

treatment of incorrect insertions, storage of results, visual presentation of results, and much more.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The second and the third authors are thankful to the National Council for Scientific and Technological Development (CNPq) and the first author is thankful to Foundation of Support to the Scientific and Technological Development of the State of Ceara (FUNCAP) for the support received on this project.

References

[1] O. I. Larichev and H. M. Moshkovich, *Verbal Decision Analysis for Unstructured Problems*, Kluwer Academic, Dodrecht, The Netherlands, 1997.

[2] I. Yevseyeva, *Solving Classification Problems with Multicriteria Decision Aiding Approaches*, Jyväskylä University Printing House, Jyväskyl, Finland, 2007.

[3] A. T. Brasil Filho, P. R. Pinheiro, and A. Coelho, "Towards the early diagnosis of Alzheimer s disease via a multicriteria classification model," in *Evolutionary Multi-Criterion Optimization*, vol. 5467 of *Lecture Notes in Computer Science*, pp. 393–406, 2009.

[4] T. C. S. Machado, A. C. Menezes, I. Tamanini, and P. R. Pinheiro, "A hybrid model in the selection of prototypes for educational tools: an applicability in verbal decision analysis," in *Proceedings of the IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MDCM '11)*, 2011.

[5] T. C. S. Machado, A. C. Menezes, L. F. R. Pinheiro, I. Tamanini, and P. R. Pinheiro, "Applying verbal decision analysis in selecting prototypes for educational tools," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS '10)*, pp. 531–535, Xiamen, China, October 2010.

[6] T. C. S. Machado, P. R. Pinheiro, A. B. Albuquerque, and M. M. L. de Lima, "Applying verbal decision analysis in selecting specific practices of CMMI," in *International Conference on Rough Sets and Knowledge Technology (RSKT)*, vol. 7414 of *Lecture Notes in Computer Science*, pp. 215–221, Springer, Berlin, Germany, 2012.

[7] T. C. S. Machado, P. R. Pinheiro, and I. Tamanini, "Towards the selection of prototypes for educational tools: a hybrid model in the verbal decision analysis," *International Journal of Information Science, Scientific & Academic Publishing*, vol. 2, no. 3, pp. 23–32, 2012.

[8] I. Tamanini, *Improving the ZAPROS method considering the incomparability cases [M.S. thesis]*, Graduate Program in Applied Informatics, University of Fortaleza, 2010.

[9] I. Tamanini, A. K. de Castro, P. R. Pinheiro, and M. C. D. Pinheiro, "Verbal decision analysis applied on the optimization

of alzheimer's disease diagnosis: a case study based on neuroimaging," *Advances in Experimental Medicine and Biology*, vol. 696, pp. 555–564, 2011.

[10] K. Schwaber, "Agile Project Management with Scrum," Microsoft, 2004.

[11] L. F. A. M. Gomes, H. Moshkovich, and A. Torres, "Marketing decisions in small businesses: how verbal decision analysis can help," *International Journal of Management and Decision Making*, vol. 11, no. 1, pp. 19–36, 2010.

[12] V. M. Ozernoy, "Choosing the best multiple criteria decision making method," *INFOR*, vol. 30, no. 2, pp. 159–171, 1992.

[13] L. Ustinovich and D. Kochin, "Verbal decision analysis methods for determining the efficiency of investments in construction," *Foundations of Civil and Environmental Engineering*, vol. 5, pp. 35–46, 2004.

[14] G. G. Dimitriadi and O. I. Larichev, "Decision support system and the ZAPROS-III method for ranking the multiattribute alternatives with verbal quality estimates," *Automation and Remote Control*, vol. 66, no. 8, pp. 1322–1335, 2005.

[15] I. Ashikhmin and E. Furems, "UniComBOS—intelligent decision support system for multi-criteria comparison and choice," *Journal of Multi-Criteria Decision Analysis*, vol. 13, no. 2-3, pp. 147–157, 2005.

[16] O. I. Larichev, "Ranking multicriteria alternatives: the method ZAPROS III," *European Journal of Operational Research*, vol. 131, no. 3, pp. 550–558, 2001.

[17] O. Larichev, "Method ZAPROS for multicriteria alternatives ranking and the problem of incomparability," *Informatica*, vol. 12, no. 1, pp. 89–100, 2001.

[18] O. I. Larichev, A. V. Kortnev, and D. Y. Kochin, "Decision support system for classification of a finite set of multicriteria alternatives," *Decision Support Systems*, vol. 33, no. 1, pp. 13–21, 2002.

[19] T. C. S. Machado, *Towards aided by multicriteria support methods and software development: a hybrid model of verbal decision analysis for selecting approaches of project management [M.S. thesis]*, Graduate Program in Applied Informatics, University of Fortaleza, 2012.

[20] T. C. S. Machado, P. R. Pinheiro, and I. Tamanini, "Project management aided by verbal decision analysis approaches: a case study for the selection of the best SCRUM practices," *International Transactions in Operational Research*, 2014.

[21] T. C. S. Machado, P. R. Pinheiro, and I. Tamanini, "Dealing the selection of project management through hybrid model of verbal decision analysis," *Procedia Computer Science*, vol. 17, pp. 332–339, 2013.

[22] K. Beck, M. Beedle, A. van Bennekum et al., "Manifesto for Agile Software Development," 2001, http://agilemanifesto.org/.

[23] K. Schwaber, *SCRUM Guide*, SCRUM Alliance, 2009.

[24] J. Sutherland and K. Schwaber, *The SCRUM Papers: Nuts, Bolts, and Origins of an Agile Process*, 2007.

[25] T. C. S. Machado, P. R. Pinheiro, and H. F. Landim, "A hybrid model in the selection of practices of framework SCRUM: an applicability in verbal decision analysis," in *Proceedings of the World Summit on the Knowledge Society (WSKS '11)*, Mykonos, Greece, 2011.

[26] T. C. S. Machado, P. R. Pinheiro, M. L. De Lima Marcelo, and H. F. Landim, "Towards a verbal decision analysis on the selecting practices of framework SCRUM," *Lecture Notes in Computer Science*, vol. 7030, pp. 585–594, 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 545191, 12 pages
http://dx.doi.org/10.1155/2014/545191

Research Article

Improving the Bin Packing Heuristic through Grammatical Evolution Based on Swarm Intelligence

Marco Aurelio Sotelo-Figueroa,[1] Héctor José Puga Soberanes,[1] Juan Martín Carpio,[1] Héctor J. Fraire Huacuja,[2] Laura Cruz Reyes,[2] and Jorge Alberto Soria-Alcaraz[1]

[1] *División de Estudios de Posgrado e Investigación, Instituto Tecnológico de León, León 37290, GTO, Mexico*
[2] *División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Ciudad Madero, Ciudad Madero 89440, TAMPS, Mexico*

Correspondence should be addressed to Marco Aurelio Sotelo-Figueroa; marco.sotelo@itleon.edu.mx

Received 11 April 2014; Revised 22 June 2014; Accepted 23 June 2014; Published 24 July 2014

Academic Editor: Ker-Wei Yu

In recent years Grammatical Evolution (GE) has been used as a representation of Genetic Programming (GP) which has been applied to many optimization problems such as symbolic regression, classification, Boolean functions, constructed problems, and algorithmic problems. GE can use a diversity of searching strategies including Swarm Intelligence (SI). Particle Swarm Optimisation (PSO) is an algorithm of SI that has two main problems: premature convergence and poor diversity. Particle Evolutionary Swarm Optimization (PESO) is a recent and novel algorithm which is also part of SI. PESO uses two perturbations to avoid PSO's problems. In this paper we propose using PESO and PSO in the frame of GE as strategies to generate heuristics that solve the Bin Packing Problem (BPP); it is possible however to apply this methodology to other kinds of problems using another Grammar designed for that problem. A comparison between PESO, PSO, and BPP's heuristics is performed through the nonparametric Friedman test. The main contribution of this paper is proposing a Grammar to generate online and offline heuristics depending on the test instance trying to improve the heuristics generated by other grammars and humans; it also proposes a way to implement different algorithms as search strategies in GE like PESO to obtain better results than those obtained by PSO.

1. Introduction

The methodology development to solve a specific problem is a process that entails the problem study and the analysis instances from such problem. There are many problems [1] for which there are no methodologies that can provide the exact solution, because the size of the problem search space makes it intractable in time, and it makes it necessary to search and improve methodologies that can give a solution in a finite time. There are methodologies based on Artificial Intelligence which do not yield exact solutions; those methodologies, however, provide an approximation, and among those we can find the following methodologies.

Heuristics are defined as "a type of strategy that dramatically limits the search for solutions" [2, 3]. One important characteristic of heuristics is that they can obtain a result for an instance problem in polynomialtime [1], although heuristics are developed for a specific instance problem.

Metaheuristics are defined as "a master strategy that guides and modifies other heuristics to obtain solutions generally better that the ones obtained with a local search optimization" [4]. The metaheuristics can work over several instances of a given problem or various problems, but it is necessary to adapt the metaheuristics to work with each problem.

It has been shown that the metaheuristic Genetic Programming [5] can generate a heuristic that can be applied to an instance problem [6]. There also exist metaheuristics that are based on Genetic Programming's paradigm [7] such as *Grammatical Differential Evolution* [8], *Grammatical Swarm* [9], *Particle Swarm Programming* [10], and *Geometric Differential Evolution* [11].

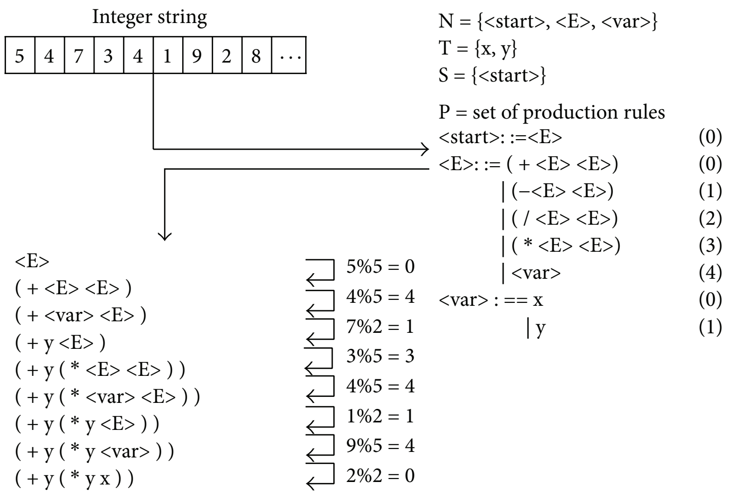


Figure 1: An example of a transformation from genotype to phenotype using a BNF Grammar. It begins with the start symbol, if the production rule for this symbol is only one rule, then the production rule replaces the start symbol, and the process begins choosing the production rules based on the current genotype. It takes each genotype and the nonterminal symbol from the left to perform the next production using (1) until all the genotypes are mapped or there are not more nonterminals in the phenotype.

The Bin Packing Problem (BPP) has been widely studied because of its many Industrial Applications, like wood and glass cutting, packing in transportation and warehousing [12], and job scheduling on uniform processors [13, 14]. This is an NP-Hard Problem [1] and due to its complexity many heuristics have been developed attempting to give an approximation [15–19]. Some metaheuristics have also been applied to try to obtain better results than those obtained by heuristics [20–22]. Some exact algorithms have been developed [23–25]; however, given the nature of the problem the time reported by these algorithms grows up and depending on the instance the time may grow up exponentially.

The contribution of this paper is to propose a generic methodology to generate heuristics using GE with search strategies. It has been shown that is possible to use this methodology to generate BPP heuristics by using PESO and PSO as search strategies; it was also shown that the heuristics generated with the proposed Grammar have better performance than the BPP's classical heuristics, which were designed by an expert in Operational Research. Those results were obtained by comparing the results obtained by the GE and the BPP heuristics by means of Friedman nonparametric test [26].

The GE is described in Section 2, including the PSO and PESO. Section 3 describes the Bin Packing Problem, the state-of-the-art heuristics, the instances used, and the fitness function. We describe the experiments performed in Section 4. Finally, general conclusions about the present work are presented in Section 5, including future perspectives of this work.

2. Grammatical Evolution

Grammatical Evolution (*GE*) [7] is a grammar-based form of Genetic Programming (*GP*) [27]. GE joins the principles of molecular biology, which are used by GP, and the power of formal grammars. Unlike GP, GE adopts a population of lineal genotypic integer strings, or binary strings, witch are transformed into functional phenotypic through a genotype-to-phenotype mapping process [28]; this process is also known as *Indirect Representation* [29]. The genotype strings evolve with no knowledge of their phenotypic equivalent, only using the fitness measure.

The transformation is governed through a Backus Naur Form grammar (*BNF*), which is made up of the tuple N, T, P, S, where N is the set of all nonterminal symbols, T is the set of terminals, P is the set of production rules that map $N \rightarrow T$, and S is the initial start symbol where $S \in N$. There are a number of production rules that can be applied to a nonterminal; an "|" (or) symbol separates the options.

Even though the GE uses the Genetic Algorithm (*GA*) [7, 28, 30] as a search strategy it is possible to use another search strategy like the Particle Swarm Optimization, called Grammatical Swarm (*GS*) [8].

In GE each individual is mapped into a program using the BNF, using (1) proposed in [28] to choose the next production based on the nonterminal symbol. An example of the mapping process employed by GE is shown in Figure 1. Consider

$$\text{Rule} = c \% r, \quad (1)$$

where c is the codon value and r is the number of production rules available for the current nonterminal.

The GE can use different search strategies; our proposed model is shown in Figure 2. This model includes the problem instance and the search strategy as an input. In [28] the search strategy is part of the process; however it can be seen as an additional element that can be chosen to work with GE. The GE will generate a solution through the search strategy selected and it will be evaluated in the objective function using the problem instance.

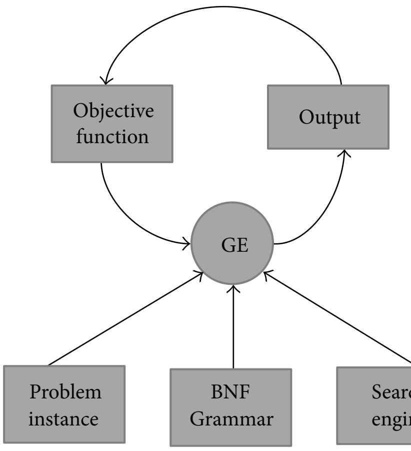


Figure 2: GE's methodology used in the present work: the presented methodology can be used with different search strategies.

2.1. Particle Swarm Optimization. Particle Swarm Optimization (PSO) [31–35] is a metaheuristic bioinspired in the flock of birds or school of fish. It was developed by Kennedy and Eberthart based on a concept called social metaphor. This metaheuristic simulates a society where all individuals contribute with their knowledge to obtain a better solution. There are three factors that influence the change of status or behavior of an individual.

(i) The knowledge of the environment or adaptation: it is related to the importance given to the experience of the individual.

(ii) His experience or local memory: it is related to the importance given to the best result found by the individual.

(iii) The experience of their neighbors or global memory: this is related to how important is the best result obtained by their neighbors or other individuals.

In this metaheuristic each individual is considered as a particle and moves through a multidimensional space that represents the social space; the search space depends on the dimension of space which in turn depends on the variables used to represent the problem.

For the update of each particle we use the velocity vector which tells how fast the particle will move in each of the dimensions; the method for updating the speed of PSO is given by (2), and its position is updated by (3). Algorithm 1 shows the complete PSO algorithm:

$$v_i = wv_i + \phi_1 \left(x_i - B_{\text{global}}\right) + \phi_2 \left(x_i - B_{\text{local}}\right), \qquad (2)$$

$$x_i = x_i + v_i, \qquad (3)$$

where

(i) v_i is the velocity of the ith particle,

(ii) w is adjustment factor to the environment,

(iii) ϕ_1 is the memory coefficient in the neighborhood,

(iv) ϕ_2 is the coefficient memory,

(v) x_i is the position of the ith particle,

(vi) B_{global} is the best position found so far by all particles,

(vii) B_{local} is the best position found by the ith particle.

2.2. Particle Evolutionary Swarm Optimization. Particle Evolutionary Swarm Optimization (*PESO*) [36–38] is based on PSO but introduces two perturbations in order to avoid two problems observed in PSO [39]:

(i) premature convergence,

(ii) poor diversity.

Algorithm 2 shows the PESO Algorithm with two perturbations, Algorithms 3 and 4. The C-Perturbation has the advantage of keeping the self-organization potential of the flock as no separate probability distribution needs to be computed; meanwhile the M-Perturbation helps keeping diversity into the population.

3. Bin Packing Problem

The Bin Packing Problem (*BPP*) [40] can be described as follows: given n items that need to be packed in the lowest possible number of bins, each item has a weight w_j, where j is the element; the max capacity of the bins c is also available. The objective is to minimize the bins used to pack all the items, given that each item is assigned only to one bin, and the sum of all the items in the bin can not exceed the bin's size.

This problem has been widely studied, including the following:

(i) proposing new theorems [41, 42],

(ii) developing new heuristic algorithms based on Operational Research concepts [18, 43],

(iii) characterizing the problem instances [44–46],

(iv) implementing metaheuristics [20, 47–49].

This problem has been shown to be an NP-Hard optimization problem [1]. A mathematical definition of the BPP is as follows:

Minimize

$$z = \sum_{i=1}^{n} y_i, \qquad (4)$$

subject to the following constraints and conditions:

$$\begin{gathered} \sum_{j=1}^{n} w_j x_{ij} \le c y_i, \quad i \in N = \{1, \dots, n\}, \\ \sum_{i=1}^{n} x_{ij} = 1, \quad j \in N, \\ y_i \in \{0,1\}, \quad i \in N, \\ x_{ij} \in \{0,1\}, \quad i \in N, \ j \in N, \end{gathered} \qquad (5)$$

```
Require: $w$ adaptation to environment coefficient, $\phi_1$ neighborhood memory
    coefficient, $\phi_2$ memory coefficient, $n$ swarm size.
Start the swarm particles.
Start the velocity vector for each particle in the swarm.
while stopping criterion not met do
    for $i = 1$ to $n$ do
        If the $i$-particle's fitness is better than the local best then replace the
            local best with the $i$-particle.
        If the $i$-particle's fitness is better than the global best then replace the
            global best with the $i$-particle.
        Update the velocity vector by (2).
        Update the particle's position with the velocity vector by (3).
    end for
end while
```

Algorithm 1: PSO Algorithm.

```
Require: $w$ adaptation to environment coefficient, $\phi_1$ neighborhood memory
    coefficient, $\phi_2$ memory coefficient, $n$ swarm size.
Start the swarm particles.
Start the velocity vector for each particle in the swarm.
while stopping criterion not met do
    for $i = 1$ to $n$ do
        If the $i$-particle's fitness is better than the local best then replace the
            local best with the $i$-particle.
        If the $i$-particle's fitness is better than the global best then replace the
            global best with the $i$-particle.
        Update the velocity vector by (2).
        Update the particle's position with the velocity vector by (3).
        Apply the C-Perturbation
        Apply the M-Perturbation
    end for
end while
```

Algorithm 2: PESO Algorithm.

```
for all Particles do
    Generate $r$ uniformly between 0 and 1.
    Generate $p1$, $p2$ and $p3$ as random numbers between 1 and the number
        of particles.
    Generate the $i$-new particle using the following equation and applying it
        to each particle dimension: $new_i = p1 + r\left(p2 - p3\right)$.
end for
for all Particles do
    If the $i$-new particle is better that the $i$-particle then replace the $i$-particle
        with the $i$-new particle.
end for
```

Algorithm 3: C-Perturbation.

where

(i) w_j is weight of the j item,

(ii) y_i is binary variable that shows if the bin i has items,

(iii) x_{ij} indicates whether the j item is into the i bin,

(iv) n is number of available bins,

(v) c is capacity of each bin.

The algorithms for the BPP instances can be classified as *online* or *offline* [46]. We have algorithms considered *online* if

```
for all Particles do
    for all Dimension do
        Generate $r$ uniformly between 0 and 1.
        if $r \leq 1/dimension$ then
            $new_{id} = random(LowerBound, UpperBound)$
        else
            $new_{id} = Particle_d$
        end if
    end for
end for
for all Particles do
    If the $i$-new particle is better that the $i$-particle then replace the $i$-particle
    with the $i$-new particle.
end for
```

Algorithm 4: M-Perturbation.

we do not know the items before starting the packing process and *offline* if we know all the items before starting. In this research we worked with both algorithms.

3.1. Tests Instances. Beasley [50] proposed a collection of test data sets, known as *OR-Library* and maintained by the Beasley University, which were studied by Falkenauer [21]. This collection contains a variety of test data sets for a variety of Operational Research problems, including the BPP in several dimensions. For the one-dimensional BPP case the collection contains eight data sets that can be classified in two classes.

(i) *Unifor.* The data sets from binpack1 to binpack4 consist of items of sizes uniformly distributed in $(20, 100)$ to be packed into bins of size 150. The number of bins in the current known solution was found by [21].

(ii) *Triplets.* The data sets from binpack5 to binpack8 consist of items from $(24, 50)$ to be packed into bins of size 100. The number of bins can be obtained dividing the size of the data set by three.

Scholl et al. [23] proposed another collection of data sets; only 1184 problems were solved optimally. Alvim et al. [51] reported the optimal solutions for the remaining 26 problems. The collection contains three data sets.

(i) *Set 1.* It has 720 instances with items drawn from a uniform distribution on three intervals $[1, 100]$, $[20, 100]$, and $[30, 100]$. The bin capacity is $C = 100$, 120, and 150 and $n = 50$, 100, 200, and 500.

(ii) *Set 2.* It has 480 instances with $C = 1000$ and $n = 50$, 100, 200, and 500. Each bin has an average of 3–9 items.

(iii) *Set 3.* It has 10 instances with $C = 100{,}000$, $n = 200$, and items are drawn from a uniform distribution on $[20000, 35000]$. Set 3 is considered the most difficult of the three sets.

3.2. Classic Heuristics. Heuristics have been used to solve the BPP, obtaining good results. Reference [18] shows the following heuristics as Classical Heuristics; these heuristics can be used as *online heuristics* if the items need to be packed as they come in or *offline heuristics* if the items can be sorted before starting the packing process.

(i) *Best Fit* [17] puts the piece in the fullest bin that has room for it and opens a new bin if the piece does not fit in any existing bin.

(ii) *Worst Fit* [18] puts the piece in the emptiest bin that has room for it and opens a new bin if the piece does not fit in any existing bin.

(iii) *Almost Worst Fit* [18] puts the piece in the second emptiest bin if that bin has room for it and opens a new bin if the piece does not fit in any open bin.

(iv) *Next Fit* [15] puts the piece in the right-most bin and opens a new bin if there is not enough room for it.

(v) *First Fit* [15] puts the piece in the left-most bin that has room for it and opens a new bin if it does not fit in any open bin.

Even though there are some heuristics having better performance than the heuristics shown in the present section [16, 19, 42, 52, 53], such heuristics have been the result of research of lower and upper bounds to determine the minimal number of bins.

3.3. Fitness Measure. There are many *Fitness Measures* used to discern the results obtained by heuristics and metaheuristics algorithms. In [54] two fitness measures are shown: the first measure (see (6)) tries to find the difference between the used bins and the theorical upper bound on the bins needed; the second (see (7)) was proposed in [47] and rewards full or

```
Require: SS search strategy, FF Fitness Function, BNF-G Grammar, IS
      Instances Set.
for all Instance Set into Instances Set do
    Select randomly an Instance from the Instance Set
    Start an initial population.
    while stopping criterion not met do
        Apply the mapping process using the Grammar BNF-G, as seen in
        the Figure 1, to obtain an heuristic by each element into the population.
        Calculate the fitness value, using FF, for each element into the population
        applying the heuristic generated to the instance selected.
        Apply the search strategy to optimize the elements into the population.
    end while
    Apply the found heuristic to all instances from the Instance Set.
end for
return Heuristic for each instance set.
```

Algorithm 5: Proposed approach.

almost full bins; the objective is to fill each bin, minimizing the free space:

$$\text{Fitness} = B - \frac{\sum_{i=1}^{n} w_i}{C}, \tag{6}$$

$$\text{Fitness} = 1 - \left(\frac{\sum_{i=1}^{n}\left(\left(\sum_{j=1}^{m} w_j x_{ij}\right)/C\right)^{2}}{n}\right), \tag{7}$$

where

(i) B is number of bins used,

(ii) n is number of containers,

(iii) m is number of pieces,

(iv) w_j is jth's piece size,

(v) consider

$$x_{ij} = \begin{cases} 1 & \text{if the piece } j \text{ is in the container } i \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

(vi) C is bin capacity.

4. Grammar Design and Testing

Algorithm 5 shows the proposed approach; this approach allows the use of different fitness functions and search strategies to generate heuristics automatically.

To improve the Bin Packing Heuristics it was necessary to design a grammar that represents the Bin Packing Problem. In [55] Grammar 1 is shown to be based on heuristic elements taken by [6]; however the results obtained in [1] give 10% of solutions that can not be applied to the instance and for this reason this approach does not need to be included to be compared against the results obtained.

Table 1: PESO and PSO parameters.

Parameter	Value
Population size	50
w	1.0
ϕ_1	0.8
ϕ_2	0.5
Function calls	1500

That Grammar has been improved in the Grammar 2 [56] to obtain similar results to those obtained by the BestFit heuristic. However this grammar cannot be applied to Bin Packing offline Problems because it does not sort pieces. Grammar 3 is proposed to improve the results obtained by Grammar 2, given that it can generate heuristics online and offline:

$$\begin{gathered}
\langle \text{inicio} \rangle \models (\langle \text{expr} \rangle) <= (\langle \text{expr} \rangle) \\
\langle \text{expr} \rangle \models (\langle \text{expr} \rangle \, \langle \text{op} \rangle \, \langle \text{expr} \rangle) \mid \langle \text{var} \rangle \mid \text{abs}\,(\langle \text{expr2} \rangle) \\
\langle \text{expr2} \rangle \models (\langle \text{expr2} \rangle \, \langle \text{op} \rangle \, \langle \text{expr2} \rangle) \mid \langle \text{var} \rangle \\
\langle \text{var} \rangle \models F \mid C \mid S \\
\langle \text{op} \rangle \models + \mid * \mid - \mid /.
\end{gathered} \tag{9}$$

Grammar 1. Grammar based on FirstFit Heuristic was proposed in [55] and we use the Heuristic Components shown in [57]

$$\begin{gathered}
\langle \text{inicio} \rangle \models \langle \text{exprs} \rangle \cdot (\langle \text{expr} \rangle) <= (\langle \text{expr} \rangle) \\
\langle \text{exprs} \rangle \models \text{Sort}\,(\langle \text{exprk} \rangle, \langle \text{order} \rangle) \mid \lambda \\
\langle \text{exprk} \rangle \models \text{Bin} \mid \text{Content} \\
\langle \text{order} \rangle \models \text{Asc} \mid \text{Des}
\end{gathered}$$

Table 2: Results obtained by each heuristic using (2).

Instance	Fitness function	MTP algorithm	Online heuristics					Offline heuristics				
			BestFit	FirstFit	NextFit	WorstFit	Almost WorstFit	BestFit	FirstFit	NextFit	Almost WorstFit	WorstFit
bin1data	Equation (7)	64.307365	316.106050	316.106020	316.106020	316.106000	321.880250	68.090770	68.167305	314.939820	86.119570	76.842740
	Bins used	78378	101097	101097	101097	101097	101705	78660	78661	101097	79314	78843
	Leftover bins	—	22719	22719	22719	22719	23327	282	283	22719	465	936
bin2data	Equation (7)	24.725332	93.025230	93.388504	109.447590	97.613075	115.927666	44.560112	44.561455	110.053760	67.597020	47.655037
	Bins used	20246	23085	23094	23518	23158	23580	20994	20994	23615	21446	21030
	Leftover bins	—	2839	2848	3272	2912	3334	748	748	3369	784	1200
bin3data	Equation (7)	0.390077	1.885825	1.885825	2.586886	2.094760	2.253316	1.390289	1.390289	2.699653	1.518808	1.397016
	Bins used	562	613	613	642	622	631	596	596	650	603	596
	Leftover bins	—	51	51	80	60	69	34	34	88	34	41
binpack1	Equation (7)	0.421557	2.425894	2.604965	7.941016	5.089604	5.365787	0.913949	0.914034	9.504668	1.479180	1.233465
	Bins used	981	1038	1044	1279	1131	1147	995	995	1372	1016	1003
	Leftover bins	—	57	63	298	150	166	14	14	391	22	35
binpack2	Equation (7)	0.180042	2.259154	2.396851	7.989118	4.916397	4.992059	0.705970	0.706004	9.459408	1.043219	0.846457
	Bins used	2032	2154	2162	2669	2342	2353	2062	2062	2851	2087	2068
	Leftover bins	—	122	130	637	310	321	30	30	819	36	55
binpack3	Equation (7)	0.097509	2.014334	2.133326	7.994689	4.725924	4.769628	0.591541	0.591543	9.412333	0.746746	0.666438
	Bins used	4024	4240	4255	5300	4614	4626	4078	4078	5647	4100	4085
	Leftover bins	—	216	231	1276	590	602	54	54	1623	61	76
binpack4	Equation (7)	0.047260	1.838669	1.935710	7.961214	4.597339	4.622161	0.495512	0.495522	9.404397	0.622414	0.572282
	Bins used	8011	8407	8430	10548	9154	9167	8108	8108	11253	8141	8123
	Leftover bins	—	396	419	2537	1143	1156	97	97	3242	112	130
binpack5	Equation (7)	0.000000	0.000000	0.000000	0.000000	0.000000	1.324353	4.830872	4.830872	6.420693	5.453772	4.848756
	Bins used	400	400	400	400	400	420	464	464	491	479	464
	Leftover bins	—	0	0	0	0	20	64	64	91	64	79
binpack6	Equation (7)	0.000000	0.000000	0.000000	0.000000	0.000000	0.682040	4.553619	4.553619	6.197923	4.994128	4.557339
	Bins used	800	800	800	800	800	820	916	916	971	936	916
	Leftover bins	—	0	0	0	0	20	116	116	171	116	136
binpack7	Equation (7)	0.000000	0.000000	0.000000	0.000000	0.000000	0.334901	4.556952	4.556953	6.078939	4.776067	4.558652
	Bins used	1660	1660	1660	1660	1660	1680	1900	1900	2002	1919	1900
	Leftover bins	—	0	0	0	0	20	240	240	342	240	259
binpack8	Equation (7)	0.000000	0.000000	0.000000	0.000000	0.000000	0.167147	4.400591	4.400588	6.059548	4.527734	4.400708
	Bins used	3340	3340	3340	3340	3340	3360	3801	3801	4024	3823	3801
	Leftover bins	—	0	0	0	0	20	461	461	684	461	483
hard28	Equation (7)	0.167276	0.655480	0.655350	13.133352	1.547310	1.927801	0.655480	0.655350	13.133352	1.927801	1.547310
	Bins used	1972	1995	1995	2755	2024	2050	1995	1995	2755	2050	2024
	Leftover bins	—	23	23	783	52	78	23	23	783	52	78
	Remaining bins		26423	26484	31602	27936	29133	2163	2164	34322	2447	3508

Table 3: Example of heuristics obtained for each instance set using Grammar 3.

Instance	Heuristic generated
dasaset1	$\text{Sort(Elements, Des)} \cdot \text{Sort(Bin, Des)} \cdot ((F + S)) \leq (\text{abs}(C))$
dataset2	$\text{Sort(Elements, Des)} \cdot \text{Sort(Cont, Des)} \cdot (\text{abs}(S)) \leq (\text{abs}((C - F)))$
dataset3	$\text{Sort(Elements, Des)} \cdot \text{Sort(Cont, Des)} \cdot (S) \leq ((C - F))$
binpack1	$\text{Sort(Content, Des)} \cdot (\text{abs}(F)) \leq ((C- \text{abs(S)}))$
binpack2	$\text{Sort(Content, Des)} \cdot ((F + S)) \leq (C)$
binpack3	$\text{Sort(Content, Des)} \cdot (F) \leq (\text{abs}((C - S)))$
binpack4	$\text{Sort(Content, Asc)} \cdot (S) \leq ((C - F))$
binpack5	$((S + F)) \leq (C)$
binpack6	$(F) \leq ((\text{abs}(C) -S))$
binpack7	$(\text{abs}(F)) \leq (\text{abs}((S - C)))$
binpack8	$(\text{abs}((S + F))) \leq (C)$
hard28	$\text{Sort(Cont, Des)} \cdot (F) \leq (\text{abs}((C - S)))$

$$\begin{gathered}
\langle \text{expr} \rangle \vDash (\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle) \mid \langle \text{var} \rangle \mid \text{abs}\,(\langle \text{expr2} \rangle) \\
\langle \text{expr2} \rangle \vDash (\langle \text{expr2} \rangle \langle \text{op} \rangle \langle \text{expr2} \rangle) \mid \langle \text{var} \rangle \\
\langle \text{var} \rangle \vDash F \mid C \mid S \\
\langle \text{op} \rangle \vDash + \mid * \mid - \mid /.
\end{gathered} \tag{10}$$

Grammar 2. Grammar proposed in [56] was based on Best-Fist Heuristic:

$$\begin{gathered}
\langle \text{begin} \rangle \vDash \langle \text{exproff} \rangle \langle \text{exprsort} \rangle (\langle \text{expr} \rangle) <= (\langle \text{expr} \rangle) \\
\langle \text{Exproff} \rangle \vDash \text{Sort (Elements, } \langle \text{order} \rangle) \mid \lambda \\
\langle \text{exprsort} \rangle \vDash \text{Sort}\,(\langle \text{exprkind} \rangle, \langle \text{order} \rangle) \mid \lambda \\
\langle \text{exprkind} \rangle \vDash \text{Bins} \mid \text{SumElements} \\
\langle \text{order} \rangle \vDash \text{Asc} \mid \text{Des} \\
\langle \text{expr} \rangle \vDash (\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle) \mid \langle \text{var} \rangle \mid \text{abs}\,(\langle \text{expr2} \rangle) \\
\langle \text{expr2} \rangle \vDash (\langle \text{expr2} \rangle \langle \text{op} \rangle \langle \text{expr2} \rangle) \mid \langle \text{var} \rangle \\
\langle \text{var} \rangle \vDash F \mid C \mid S \\
\langle \text{op} \rangle \vDash + \mid * \mid - \mid /.
\end{gathered} \tag{11}$$

Grammar 3. Grammar proposal to generate heuristics online and offline is based on Grammar 2 , where

(i) S is size of the current piece,

(ii) C is bin capacity,

(iii) F is sum of the pieces already in the bin,

(iv) Elements sorts the elements,

(v) Bin sorts the bins based on the bin number,

(vi) Cont sorts the bins based on the bin contents,

(vii) Asc sorts in ascending order,

(viii) Des sorts in descending order.

In order to generate the heuristics Grammar 3 was used. The search strategies applied to the GE were PESO and PSO. The number of function calls was taken from [56], where it was explained that this number is only 10% from the number of function calls used by [6]. To obtain the parameters shown in Table 1 a fine-tuning process was applied based on Covering Arrays (CA) [58]; in this case the CA was generated using the Covering Array Library (CAS) [59] from The National Institute of Standards and Technology (NIST) (http://csrc.nist.gov/groups/SNS/acts/index.html).

In order to generate the heuristics, one instance from each set was used. Once the heuristic was obtained for each instance set, it was applied to all the sets to obtain the heuristic's fitness. The instance sets used were detailed in Section 3.1. 33 experiments were performed independently and the median was used to compare the results against those obtained with the heuristics described in Section 3. The comparison was implemented through the nonparametric test of Friedman [26, 60]; this nonparametric test used a post hoc analysis to discern the performance between the experiments and gives a ranking of them.

The method to apply the heuristics generated by Grammar 3 for an instance set is described below.

(i) For each instance in the instance set the generated heuristic will be applied.

(ii) The generated heuristic has the option to sort the items before starting the packing process, to treat the instances like offline instances.

(iii) The next part of the generated heuristic says how to sort the bins; many heuristics require sorting the bins before packing an item.

(iv) The last part, the inequality, determines the rule to pack an item.

Sometimes the generated heuristic does not have items ordered and it makes the heuristic work like an online heuristic. If it does not have the bins ordered all the items will be packed into the bins in the order they were created.

Table 4: Results obtained by each heuristic over the instance set.

Instance	Fitness function		PSO			PESO	
		Grammar 1	Grammar 2	Grammar 3	Grammar 1	Grammar 2	Grammar 3
bin1data	Equation (7)	316.106020	316.106050	68.167305	316.106020	316.106050	68.090770
	Bins used	101097	101097	101097	101097	101097	78660
	Leftover bins	22719	22719	22719	22719	22719	282
bin2data	Equation (7)	93.388510	93.025230	44.561455	93.388510	93.007030	44.560112
	Bins used	23094	23156	23583	23097	23156	20994
	Leftover bins	2848	2910	3337	2851	2910	748
bin3data	Equation (7)	1.885825	1.885825	1.390289	1.885825	1.885825	1.390289
	Bins used	613	622	650	613	622	596
	Leftover bins	51	60	88	51	60	34
binpack1	Equation (7)	2.604965	2.425894	0.914034	2.604965	2.425894	0.913949
	Bins used	1044	1131	1372	1044	1131	995
	Leftover bins	63	150	391	63	150	14
binpack2	Equation (7)	2.396851	2.259154	0.706004	2.396851	2.259154	0.705970
	Bins used	2162	2342	2851	2162	2342	2062
	Leftover bins	130	310	819	130	310	30
binpack3	Equation (7)	2.133326	2.014334	0.591543	2.133326	2.014334	0.591541
	Bins used	4255	4614	5647	4255	4614	4078
	Leftover bins	231	590	1623	231	590	54
binpack4	Equation (7)	1.935710	1.838669	0.495522	1.935710	1.838669	0.495512
	Bins used	8430	9154	11253	8430	9154	8108
	Leftover bins	419	1143	3242	419	1143	97
binpack5	Equation (7)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Bins used	400	400	400	400	400	400
	Leftover bins	0	0	0	0	0	0
binpack6	Equation (7)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Bins used	800	800	800	800	800	800
	Leftover bins	0	0	0	0	0	0
binpack7	Equation (7)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Bins used	1660	1660	1660	1660	1660	1660
	Leftover bins	0	0	0	0	0	0
binpack8	Equation (7)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Bins used	3340	3340	3340	3340	3340	3340
	Leftover bins	0	0	0	0	0	0
hard28	Equation (7)	0.655350	0.655480	0.655350	0.655350	0.655480	0.655480
	Bins used	1995	2024	2755	1995	2024	1995
	Leftover bins	23	52	783	23	52	23
Remaining bins		26484	27934	33002	26487	27934	1282

5. Results

In Table 2 the results obtained with online and offline heuristics (described in Section 3.2) are shown. Results obtained by an exact algorithm were included, the MTP algorithm [40], and results from the fitness function from Section 3.3 are shown as well with the number of bins used. A row was added where the difference of containers regarding the optimal is shown. These results were obtained by applying the heuristics to each instance; all the results from an instance set were added.

Table 3 shows examples of heuristics generated using the proposed Grammar with GE for each instance set; some heuristics can be reduced but this is not part of the present work.

The results obtained by the PSO and PESO with the Grammars are shown in Table 4; these results are the median from 33 individual experiments. Using the results obtained by the heuristics and the GE with PESO and PSO the Friedman nonparametric test was performed to discern the results. The value obtained by the Friedman nonparametric test is 85.789215 and the P value 6.763090E-11; this means that the

Table 5: Rankings of the algorithms.

Algorithm	Ranking Friedman
Exact	2.666667
PESO-Grammar 3	4.375000
PSO-Grammar 3	4.791667
BestFit-Offline	6.916667
FirstFit-Offline	7.250000
PESO-Grammar 2	8.666667
BestFit	8.791667
PSO-Grammar 2	8.791667
FirstFit	8.958333
PESO-Grammar 1	9.083333
PSO-Grammar 1	9.083333
WorstFit-Offline	9.541667
AlmostWorstFit-Offline	10.625000
WorstFit	10.791667
NextFit	12.250000
AlmostWorstFit	14.291667
NextFit-Offline	16.125000

tested heuristics have different performance. Due to this it was necessary to apply a post hoc procedure to obtain the Heuristics Ranking shown in Table 5.

Both Tables 2 and 4 have an extra row at the bottom with the total remaining bins. The results obtained by PESO using Grammar 3 show that this heuristic which has been deployed automatically has less bins than the other classic heuristics.

6. Conclusions and Future Works

In the present work a Grammar was proposed to generate online and offline heuristics in order to improve heuristics generated by other grammars and by humans. It also was proposed using PESO as a search strategy based on Swarm Intelligence to avoid the problems observed in PSO.

Through the results obtained in Section 5, it was concluded that it is possible to generate good heuristics with the proposed Grammar. Additionally it can be seen that the quality of these heuristics strongly depends on the grammar used to evolve.

The grammar proposed in the present work shows that is possible to generate heuristics with better performance that the well-known BestFit, FirstFit, NextFit, WorstFit, and Almost WorstFit heuristics from Section 3.2 regardless of heuristics being online or offline. While the heuristics are designed to work with all the instances sets, the GE adjusts heuristics automatically to work with one instance set and it makes it possible for GE to generate offline or online heuristics. The GE can generate as many heuristics as instances sets that have been working and try to adapt the best heuristic that can be generated with the used Grammar.

The results obtained by PESO are better than those obtained by PSO by using Grammars 2 and 3, but with Grammar 1 PESO and PSO have the same performance.

The current investigation is based on the one-dimensional bin packing problem but this methodology can be used to solve other problems, due to the generality of the approach. It is necessary to apply heuristic generation to other problems and investigate if the GE with PESO as search strategy gives better results than the GP or GE with other search strategies.

It will be necessary to find a methodology to choose the instance or instances for the training process as well as to determine if the instances are the same or to classify the instances in groups with the same features to generate only one heuristic by group.

It will also be necessary to research other metaheuristics that do not need the parameter tuning because the metaheuristics shown in the present paper were tuned using Covering Arrays.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors want to thank to the Instituto Tecnológico de León (ITL) for the support provided for this research. Additionally the authors want to aknowledge the generous support from the Consejo Nacional de Ciencia y Tecnológia (CONACyT) from Mexico for this research project.

References

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, NY, USA, 1979.

[2] E. A. Feigenbaum and J. Feldman, *Computers and Thought*, AAAI Press, 1963.

[3] M. H. J. Romanycia and F. J. Pelletier, "What is a heuristic?" *Computa tional Intelligence*, vol. 1, no. 1, pp. 47–58, 1985.

[4] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.

[5] J. R. Koza, " Hierarchical genetic algorithms operating on populations of computer programs," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pp. 768–774, San Mateo, Calif, USA, 1989.

[6] E. K. Burke, M. Hyde, and G. Kendall, "Evolving bin packing heuristics with genetic programming," in *Parallel Problem Solving from Nature-PPSN IX*, T. Runarsson, H.-G. Beyer, J. Merelo-Guervós, L. Whitley, and X. Yao, Eds., vol. 4193 of *Lecture Notes in Comput er Science*, pp. 860–869, Springer, Berlin, Germany, 2006.

[7] C. Ryan, J. Collins, and M. O'Neill, "Grammatical evolution: evolving programs for an arbitrary language," in *Proceedings of the 1st European Workshop on Genetic Programming*, vol. 1391 of *Lecture Notes in Com puter Science*, pp. 83–95, Springer, 1998.

[8] M. O'Neill and A. Brabazon, "Grammatical differential evolution," in *Proceedings of the International Conference on Artificial Intelligence (ICAI '06)*, CSEA Press, Las Vegas, Nev, USA, 2006.

[9] M. O'Neill and A. Brabazon, "Grammatical swarm: the generation of programs by social programming," *Natural Computing*, vol. 5, no. 4, pp. 443–462, 2006.

[10] J. Togelius, R. de Nardi, and A. Moraglio, "Geometric PSO + GP = particle swarm programming," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 3594–3600, Hong Kong, June 2008.

[11] A. Moraglio and S. Silva, "Geometric di ff ere ntial evolution on the space of genetic programs," in *Genetic Programming*, A. Esparcia-Alcazar, A. Ekart, S. Silva, S. Dignum, and A. Uyar, Eds., vol. 6021 of *Lecture Notes in Computer Science*, pp. 171–183, Springer, Berlin, Germany, 2010.

[12] A. Lodi, S. Martello, and D. Vigo, "Recent advances on two-dimensional bin packing problems," *Discrete Applied Mathematics*, vol. 123, no. 1–3, pp. 379–396, 2002.

[13] H. van de Vel and S. Shijie, "Application of the bin-packing technique to job scheduling on uniform processors," *The Journal of the Operational Research Society*, vol. 42, no. 2, pp. 169–172, 1991.

[14] B. T. Han, G. Diehr, and J. S. Cook, "Multiple-type, two-dimensional bin packing problems: applications and algorithms," *Annals of Operations Research*, vol. 50, no. 1, pp. 239–261, 1994.

[15] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham, "Worst-case performance bounds for simple one-dimensional packing algorithms," *SIAM Journal on Computing*, vol. 3, pp. 299–325, 1974.

[16] A. C. C. Yao, "New algorithms for bin packing," *Journal of the Association for Computing Machinery*, vol. 27, no. 2, pp. 207–227, 1980.

[17] W. T. Rhee and M. Talagrand, "On-line bin packing with items of random size," *Mathematics of Operations Research*, vol. 18, no. 2, pp. 438–445, 1993.

[18] E. Coffman Jr., G. Galambos, S. Martello, and D. Vigo, *Bin Packing Approximation Algorithms: Combinatorial Analysis*, Kluwer Academic Publishers, 1998.

[19] K. Fleszar and K. S. Hindi, "New heuristics for one-dimensional bin-packing," *Computers and Operations Research*, vol. 29, no. 7, pp. 821–839, 2002.

[20] T. Kämpke, "Simulated annealing: use of a new tool in bin packing," *Annals of Operations Research*, vol. 16, no. 1, pp. 327–332, 1988.

[21] E. Falkenauer, "A hybrid grouping genetic algorithm for bin packing," *Journal of Heuristics*, vol. 2, pp. 5–30, 1996.

[22] A. Ponee-Pérez, A. Pérez-Garcia, and V. Ayala-Ramirez, "Bin-packing using genetic algorithms," in *Proceedings of the 15th International Conference on Electronics, Communications and Computers (CONIELECOMP '05)*, pp. 311–314, IEEE Computer Society, Los Alamitos, Calif, USA, March 2005.

[23] A. Scholl, R. Klein, and C. Jürgens, "Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem," *Computers and Operations Research*, vol. 24, no. 7, pp. 627–645, 1997.

[24] J. M. V. de Carvalho, "Exact solution of bin-packing problems using column generation and branch-and-bound," *Annals of Operations Research*, vol. 86, pp. 629–659, 1999.

[25] J. Puchinger and G. Raidl, "Combining metaheuristics and exact algorithms in combinatorial optimization: a survey and classification," in *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, J. Mira and J. Alvarez, Eds., vol. 3562 of *Lecture Notes in Computer Science*, pp. 41–53, Springer, Berlin, Germany, 2005.

[26] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[27] J. R. Koza and R. Poli, "Genetic programming," in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, E. K. Burke and and G. Kendall, Eds., pp. 127–164, Kluwer, Boston, Mass, USA, 2005.

[28] I. Dempsey, M. O'Neill, and A. Brabazon, "Foundations in grammatical," in *Foundations in Grammatical Evolution for Dynamic Environments*, vol. 194, Springer, New York, NY, USA, 2009.

[29] H. lan Fang, P. Ross, and D. Corne, "A promising genetic algorithm approach to job -shop scheduling, rescheduling, and open-shop scheduling problems," in *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 375–382, Morgan Kaufmann, Burlington, Mass, USA, 1993.

[30] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[31] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.

[32] C. Maurice, *Particle Swarm Optimization*, Wiley, Estados Unidos, USA, 2006.

[33] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[34] M. F. Tasgetiren, P. N. Suganthan, and Q. Pan, "A discrete particle swarm optimization algorithm for the generalized traveling salesman problem," in *Proceedings of the 9th Annual Genetic and Evolutionary Computation Conference (GECCO '07)*, pp. 158–167, New York, NY, USA, July 2007.

[35] T. Gong and A. L. Tuson, "Binary particle swarm optimization: a forma analysis approach," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation (GECCO '07)*, p. 172, ACM, New York, NY, USA, July 2007.

[36] A. E. M. Zavala, A. H. Aguirre, and E. R. Villa Diharce, "Constrained optimization via Particle Evolutionary Swarm Optimization algorithm (PESO)," in *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO '05)*, pp. 209–216, New York, NY, USA, June 2005.

[37] A. E. M. Zavala, A. H. Aguirre, and E. R. Villa Diharce, "Particle evolutionary swarm optimization algorithm (PESO)," in *Proceedings of the 6th Mexican International Conference on Computer Science (ENC '05)*, pp. 282–289, Puebla, Mexico, September 2005.

[38] A. E. Muñoz-Zavala, A. Hernández-Aguirre, E. R. Villa-Diharce, and S. Botello-Rionda, "PESO+ for constrained optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 231–238, July 2006.

[39] G. T. Pulido and C. A. C. Coello, "A constraint-handling mechanism for particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, vol. 2, pp. 1396–1403, Portland, Ore, USA, June 2004.

[40] S. Martello and P. Toth, *Knapsack Problems. Algorithms and Computer Implementations*, John Wiley & Sons, New York, NY, USA, 1990.

[41] E. G. Coffman Jr., C. Courcoubetis, M. R. Garey, P. W. Shor, and R. R. Weber, "Bin packing with discrete item sizes. I. Perfect packing theorems and the average case behavior of optimal packings," *SIAM Journal on Discrete Mathematics*, vol. 13, no. 3, pp. 384–402, 2000.

[42] T. G. Crainic, G. Perboli, M. Pezzuto, and R. Tadei, "New bin packing fast lower bounds," *Computers and Operations Research*, vol. 34, no. 11, pp. 3439–3457, 2007.

[43] J. Coffman, G. Galambos, S. Martello, and D. Vigo, "Bin packing approximation algorithms: combinatorial analysis," in *Handbook of Combinatorial Optimization*, pp. 151–207, Kluwer Academic, 1999.

[44] S. P. Fekete and J. Schepers, "New classes of fast lower bounds for bin packing problems," *Mathematical Programming*, vol. 91, no. 1, pp. 11–31, 2001.

[45] S. S. Seiden, R. van Stee, and L. Epstein, "New bounds for variable-sized online bin packing," *SIAM Journal on Computing*, vol. 32, no. 2, pp. 455–469, 2003.

[46] E. G. Coffman Jr. and J. Csirik, "A classification scheme for bin packing theory," *Acta Cybernetica*, vol. 18, no. 1, pp. 47–60, 2007.

[47] E. Falkenauer and A. Delchambre, "A genetic algorithm for bin packing and line balancing," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1186–1192, May 1992.

[48] A. Lodi, S. Martello, and D. Vigo, "Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems," *INFORMS Journal on Computing*, vol. 11, no. 4, pp. 345–357, 1999.

[49] E. Hopper and B. C. H. Turton, "A review of the application of meta-heuristic algorithms to 2D strip packing problems," *Artificial Intelligence Review*, vol. 16, no. 4, pp. 257–300, 2001.

[50] J. E. Beasley, "OR-Library: distributing test problems by electronic mail," *Journal of the Operational Research Society*, vol. 41, no. 11, pp. 1069–1072, 1990.

[51] A. C. F. Alvim, C. C. Ribeiro, F. Glover, and D. J. Aloise, "A hybrid improvement heuristic for the one-dimensional bin packing problem," *Journal of Heuristics*, vol. 10, no. 2, pp. 205–229, 2004.

[52] C. D. T. Suárez, E. P. Gonzlez, and M. V. Rendón, "A heuristic algorithm for the offline one-dimensional bin packing problem inspired by the point Jacobi matrix iterative method," in *Proceedings of the 5th Mexican International Conference on Artificial Intelligence (MICAI '06)*, pp. 281–286, Mexico City, Mexico, November 2006.

[53] S. Tam, H. Tam, L. Tam, and T. Zhang, "A new optimization method, the algorithm of changes, for Bin Packing Problem," in *Proceedings of the IEEE 5th International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA '10)*, pp. 994–999, September 2010.

[54] M. Hyde, *A genetic programming hyper-heuristic approach to automated packing [Ph.D. thesis]*, University of Nottingham, 2010.

[55] M. A. Sotelo-Figueroa, H. J. Puga Soberanes, J. Martin Carpio et al., "Evolving bin packing heuristic using micro-differential evolution with indirect representation," in *Recent Advances on Hybrid Intelligent Systems*, vol. 451 of *Studies in Computational Intelligence*, pp. 349–359, Springer, Berlin, Germany, 2013.

[56] M. Sotelo-Figueroa, H. Puga Soberanes, J. Martin Carpio, H. Fraire Huacuja, L. Cruz Reyes, and J. Soria-Alcaraz, "Evolving and reusing bin packing heuristic through grammatical differential evolution," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC '13)*, pp. 92–98, Fargo, ND, USA, August 2013.

[57] E. K. Burke and G. Kendall, *Search Method ologies: Introductory Tutorials in Optimization and Decision Support Techniques*, Springer, New York, NY, USA, 2006.

[58] A. Rodriguez-Cristerna, J. Torres-Jimenez, I. Rivera-Islas, C. Hernandez-Morales, H. Romero-Monsivais, and A. Jose-Garcia, "A mutation-selection algorithm for the problem of minimum brauer chains," in *Advances in Soft Computing*, I. Batyrshin and G. Sidorov, Eds., vol. 7095 of *Lecture Notes in Computer Science*, pp. 107–118, Springer, Berlin, Germany, 2011.

[59] R. N. Kacker, D. Richard Kuhn, Y. Lei, and J. F. Lawrence, "Combinatorial testing for software: an adaptation of design of experiments," *Measurement*, vol. 46, no. 9, pp. 3745–3752, 2013.

[60] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2nd edition, 2000.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 713427, 8 pages
http://dx.doi.org/10.1155/2014/713427

Research Article

A Hybrid Optimized Weighted Minimum Spanning Tree for the Shortest Intrapath Selection in Wireless Sensor Network

Matheswaran Saravanan[1] and Muthusamy Madheswaran[2]

[1] *Department of Computer Science and Engineering, VMKV Engineering College, Salem, Tamil Nadu 636308, India*
[2] *Department of Electronics and Communication Engineering, Mahendra Engineering College, Namakkal, Tamil Nadu 637503, India*

Correspondence should be addressed to Matheswaran Saravanan; saravanan.vmkv@rediffmail.com and Muthusamy Madheswaran; madheswaran.dr@gmail.com

Received 11 April 2014; Revised 22 June 2014; Accepted 22 June 2014; Published 22 July 2014

Academic Editor: Jer-Guang Hsieh

Wireless sensor network (WSN) consists of sensor nodes that need energy efficient routing techniques as they have limited battery power, computing, and storage resources. WSN routing protocols should enable reliable multihop communication with energy constraints. Clustering is an effective way to reduce overheads and when this is aided by effective resource allocation, it results in reduced energy consumption. In this work, a novel hybrid evolutionary algorithm called Bee Algorithm-Simulated Annealing Weighted Minimal Spanning Tree (BASA-WMST) routing is proposed in which randomly deployed sensor nodes are split into the best possible number of independent clusters with cluster head and optimal route. The former gathers data from sensors belonging to the cluster, forwarding them to the sink. The shortest intrapath selection for the cluster is selected using Weighted Minimum Spanning Tree (WMST). The proposed algorithm computes the distance-based Minimum Spanning Tree (MST) of the weighted graph for the multihop network. The weights are dynamically changed based on the energy level of each sensor during route selection and optimized using the proposed bee algorithm simulated annealing algorithm.

1. Introduction

Wireless sensor network (WSN) is a cooperative collection of sensor nodes, each having processing capability. Routing in WSN is different from conventional fixed network routing by several ways. WSNs are infrastructureless, have unreliable wireless links, contain sensor nodes that might fail, and its routing protocols face rigorous energy saving requirements [1]. WSN is a distributed real-time system and many routing algorithms have been proposed in literature [2–4]. In the earlier research for distributed systems, it was assumed that wired systems had unlimited power. They had user interfaces, had fixed resources, treated each system node as important, and were location independent. WSNs in contrast are wireless systems with limited power, are constrained in energy consumption, and are real time, with dynamically varying resources [5, 6]. Routing in WSN utilizing minimal energy has been proposed in the literature [7–11]. The power management solutions, at the software level, aim at reducing communications as broadcasting or listening to messages uses up energy. Minimizing message numbers cuts costs and a good MAC protocol ensures reduced collisions and retries. Better routing minimizes the number of messages sent by the use of short paths and congestion avoidance. Factors like efficient neighbor detection, localization, time synchronization, flooding, and query dissemination reduce the number of messages and increase the life of the network. There are varied solutions for scheduling sleep/wake-up patterns [12, 13] with most trying to keep up minimum nodes labeled sentries. The latter provides sensing coverage and allows others to sleep.

The clustering process divides a network into interconnected substructures, called clusters, with each cluster having many Sensor Node (SN) led by a Cluster Head (CH) which is the coordinator in this substructure [14] as seen in Figure 1. The CH is also a temporary base station which keeps in touch with other CHs. Nodes have four possible states: normal, isolated, cluster head, and gateway. Basically, nodes are in an

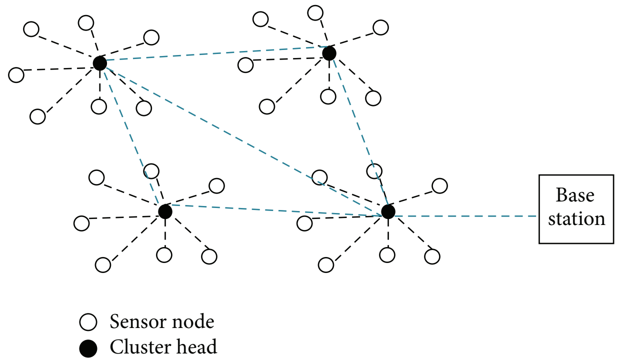


Figure 1: Block diagram of WSN deployment with cluster heads.

isolated state with each maintaining the neighbor table where neighbour node information is stored. Electing the CH is the basic step in clustering. Clustering is widely applied in WSN for managing power efficiently [15–17].

A clustering architecture in WSN environment enables features such as network scalability, communication overhead reduction, and fault tolerance. Cluster formation benefits routing as the cluster head and cluster gateways are responsible for the intercluster routing, thus, restricting, creating, and spreading of routing information. Local changes such as nodes changing cluster are updated only in the corresponding clusters and no update is required by the whole network. This significantly reduces information stored by each mobile node.

The major problem in WSNs is experienced by the sensor nodes nearest to the base station that require transmitting more number of packets than that of faraway nodes. Generally clustering algorithms use two methods for extending the lifetime. In the first method, the cluster heads with high residual energy are selected, and in the second method, to distribute the energy utilization between nodes in all clusters, the cluster heads are rotated periodically [18]. To choose the cluster heads, various techniques have been proposed in literature.

Other than clustering technique, tree based routing has been popularly used in WSN due to its energy efficiency [19–21]. Tree based techniques use the concept of selecting a root node before data transmission. A tree-like hierarchical path of nodes is constructed to connect the nodes. The WSN nodes construct a tree which can either be Minimum Spanning Tree (MST) or be optimal tree through which data is transmitted till it reaches the sink. The root node uses tree traversal algorithm to gather data about the children nodes. There are three methods in WSN tree routing protocols [19]. While the first does not create clusters the other two mix the clustering strategy with a tree routing algorithm. The latter strategy reduces low latency, while the tree routing algorithm improves energy efficiency [22, 23]. In cluster based routing strategy, the sensor nodes in the network divide into several clusters. Each cluster then chooses a cluster head randomly or by cluster head election algorithm. The cluster head is responsible for gathering the data from the sensor nodes in its cluster. The aggregated data collected by the cluster head is then transmitted to the sink. Generally, a MST is created to connect the nodes and this reduces energy consumption either by transmitting packets through smaller distances or by reducing number of packets transmitted or by both techniques.

A modified Kruskal's Minimum Spanning Tree (MST) search algorithm based on distributed search by hierarchical clusters was proposed to search the network for small balanced weight routing spanning trees [5]. The proposed technique provided spanning trees with low maximum degree and larger diameter to balance energy consumption in WSN's routing. Based on energy matrix transmission the results proved that this approach extended WSN functional life by more than three times with respect to sensor transmission energy.

An energy efficient spanning tree (EESR) was proposed for multihop routing to increase the lifetime of the network [24]. The EESR provides location of the sensor nodes and base station and produces a sequence of routing paths consisting of suitable number of rounds. The results obtained by simulation reveal that the EESR method outperforms the other existing methods in relation to increasing the lifetime of the network.

A trajectory clustering technique was proposed for the purpose of selecting the cluster heads [25]. In this algorithm, the cluster heads are selected on the basis of traffic and they are periodically rotated. The cluster heads are selected using the trajectory based clustering technique and thus network lifetime is extended.

Guangyan et al. [26] proposed Dynamic Minimal Spanning Tree Routing Protocol (DMSTRP), an innovative cluster-based routing protocol, that enhanced Base Station Controlled Dynamic Clustering Protocol (BCDCP) by means of initiating MSTs rather than clubbing for the purpose of connecting nodes in clusters. When compared to LEACH and BCDCP, the DMSTRP performed well even in large network in terms of network lifetime and delay.

A distributed topology control technique was proposed in [27] to enhance energy efficiency and reduce radio interference in WSNs. Each network node makes local decisions about transmission power. These decisions conclude in a network topology preserving global connectivity. The fundamental control technique is the novel Smart Boundary Yao Gabriel Graph (SBYaoGG) and optimization ensures that all network links are symmetric and energy efficient. This technique was effective as compared to other approaches to topology control.

A combined algorithm (COM), a generalization of the MST and SPT, was proposed in [28] which dealt with the issue of executing the operation of Data Aggregation enhanced Convergecast (DAC) in an energy and latency efficient manner. The valuable portion of the total data gathered is approximated by assuming that each and every node in the network consists of a data item and a known application dependent data compression factor.

Multiple Cluster Heads Routing Protocol (MCHRP) [29] was proposed to address cluster head overload. This method improved LEACH by incorporating a decision function which is based on the cluster head's remaining energy, location, and frequency. The decision function selects the main cluster heads and the alternative cluster heads used for data acquisition, data fusion, and data transmission.

Cluster-based and Tree-based Power Efficient Data Collection and Aggregation (CTPEDCDA) protocol by Wang et al. [30] was based on clustering and MST to minimize energy consumption in WSN. MSTs are built by connecting the cluster heads to improve the transmission routing mechanism. Chhabra and Sharma [31] improved the power consumption by improving the first node death. This method combined both the cluster-based and the tree-based protocol to improve evenness of dissipated network energy.

Kumrai et al. [32] proposed evolutionary algorithm heuristically that optimizes the sensing coverage area and the installation cost in WSN by considering the sensor network connectivity as a constraint. The algorithm uses a population of individuals, each of which represents a set of wireless sensor nodes types and positions and evolves them via the proposed genetic operators. The proposed mutation and constraint-domination operators were designed to quickly seek the optimal solutions that meet the WSN installation requirements. Simulation result shows that the sensing coverage and installation cost were improved.

Karimi et al. [33] proposed two algorithms such as GP-Leach and HS-Leach. The energy consumption was improved by partitioning the network and using evolutionary algorithms for optimized cluster head selection considering WSN nodes position information and residual energy. The simulation results performed in MATLAB show that the proposed algorithms were more efficient and they increased the lifetime of network.

In this work, a Weighted Minimum Spanning Tree, Bee Algorithm-Simulated Annealing (BASA-WMST) algorithm is proposed. Cluster heads are selected based on the proposed optimization technique and WMST is used to find the shortest intrapath selection within the cluster. The proposed method computes the distance-based minimum spanning tree of the weighted graph for the multihop network. During route selection, the weights are dynamically adjusted based on the mobility, energy level, and distance of each sensor. Section 2 presents the problem formulation, Section 3 deals with the proposed methodology in detail, Section 4 shows the experimental results, and Section 5 concludes the paper.

2. Problem Formulation

Tree based routing has the advantage of lower control packet overheads but suffers from approximation error compared to cluster based routing. Cluster based routing provides better energy savings compared to tree based techniques. In this work it is proposed to combine the features of cluster based routing for cluster formation and cluster head selection and use minimum spanning tree for intracluster communication. Ideal clusters are formed when the network parameters like energy spent, lifetime, Packet Delivery Ratio, and end to end delay are optimized. Since most of the network parameters are additive in nature the optimization problem is NP hard. Several metaheuristic techniques including genetic algorithm have been proposed in literature. In this paper, bee algorithm in combination with simulated annealing was chosen due to its faster convergence and its capability to avoid local minima problem.

WSN network can be considered as a connected undirected graph represented by $G = (V, E)$, where V is the vertices made up of $(v_1, \ldots, v_n)$ nodes and E is the edges represented as $(e_{1,2}, e_{1,3}, \ldots, e_{i,j}, \ldots, e_{n-1,n})$ the connection between nodes [34]. In this work the normalized values of mobility, delay, and remaining energy are represented on the edges. Each edge may then be defined by the attributes represented in positive real numbers and denoted by $w_{i,j} = w^1_{i,j}, w^2_{i,j}, \ldots, w^m_{i,j}$.

Let $x = x_{1,2}, x_{1,3}, \ldots, x_{i,j}, \ldots, x_{n-1,n}$ be defined as the connectivity between node i and j:

$$x_{i,j} = \begin{cases} 1 & \text{if } e_{i,j} = 1 \text{ and is selected} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The proposed technique can be formulated as in

$$\begin{aligned} \min f_1(x) &= \sum w^1_{i,j} x_{i,j} \\ \min f_2(x) &= \sum w^2_{i,j} x_{i,j} \\ &\vdots \\ \min f_m(x) &= \sum w^m_{i,j} x_{i,j}, \end{aligned} \tag{2}$$

where $f_i(x)$ is the objective to be minimized for the problem, $i = 1, \ldots, n-1$; $j = 1, \ldots, n$ subject to $x \in X$. These objectives either can be formulated as a multiobjective function or can be represented as in

$$\min f_i(x) = \alpha \sum w^1_{i,j} x_{i,j} + \beta \sum w^2_{i,j} x_{i,j} + \cdots, \tag{3}$$

where $\alpha + \beta + \cdots = 1$.

Since node mobility, delay, and remaining energy are used as the edge in the graph, the objective function can be formulated as in

$$\begin{aligned} \min f_i(x) = {} & \alpha(\min(\text{mobility})) + \beta(\min(\text{delay})) \\ & + \gamma(\min(1 - \text{remaining\_energy})). \end{aligned} \tag{4}$$

The following assumptions are made for the sensor network.

(1) Nodes are dispersed randomly.

(2) The energy of sensor nodes is limited and uniform initially.

(3) Nodes are location unaware.

(4) The transmitting power of the nodes varies depending on the distance to the receiver.

(5) Approximate distance is estimated based on the received signal strength.

3. Methodology

The node energy model is based on [35]. The energy dissipated to transmit n bit is given in

$$E_{\text{diss\_Tx}} = n\left(\text{Energy\_dissipated\_tranmitter\_electronics} + \left(\text{Energy\_dissipated\_transmitter\_amplifier} * \text{distance\_squared}\right)\right). \quad (5)$$

The energy dissipated to receive n bit is given in

$$E_{\text{diss\_Rx}} = n\left(\text{Energy\_dissipated\_receiver\_electronics}\right). \quad (6)$$

Power consumed for a given time period t can be computed by dividing the dissipated energy by time and is given by

$$\frac{E_{\text{diss\_Rx}} + E_{\text{diss\_Tx}}}{t}. \quad (7)$$

The mobility of a node is estimated using the Free Space Path Loss (FSPL) model. The relation between FSPL, frequency of radio signal, and distance between the transmitter and receiver is given by

$$\text{FSPL (dB)} = 20\log(d) + 20\log(f) + k, \quad (8)$$

where d is the distance, f is the frequency, and log is the logarithm to base 10. K is a constant and is equal to 32.44 when frequency is measured in Mhz and distance is measured in Kilometer. Another method to compute the FSPL is using the fade margin and it is given by

$$\text{FSPL} = \text{Energy\_dissipated\_Tx\_electronics} + \text{Energy\_dissipated\_Tx\_amplifier} + \text{Energy\_dissipated\_Rx\_electronics}. \quad (9)$$

Using the two FSPL equations (8) and (9), the distance can be computed by

$$d = \frac{10\left(\text{Free Space Path Loss} - 32.44 - 20\log(f)\right)}{20}. \quad (10)$$

To find the distance travelled by nodes i and j with respect to each other during time n, the distance between the nodes is computed at time t and $t = n$ if high mobility increases the reclustering process and increases the energy consumption. The objective is to form clusters based on low mobility which leads to lower energy consumption and lower delays due to lower link breakages. The mobility of the node can be computed by

$$m = \frac{d_t - d_{t+n}}{D}\left\{\begin{array}{l} > 0.5 \text{ implies high mobility} \\ \leq 0.5 \text{ implies normal mobility} \\ 0 \text{ implies no mobility} \\ < 0 \text{ implies nodes converging} \end{array}\right\}. \quad (11)$$

Table 1: Information maintained in the neighborhood table.

Parameter	Description
V_i	Node i
V_j	A neighbor node in cluster range of V_i
RE_{V_i}	Residual energy of V_i
Dis_{V_j}	Distance between V_i and V_j
RE_{V_j}	Residual energy of V_j
Ech_Msg	Elect cluster head message
Crt_Msg	Create tree message

Each node stores in its neighborhood table the information about its neighbors, as shown in Table 1. Each node broadcasts the Ech_Msg, at the beginning of each round which contains residual energies, within radio range r. All nodes within the cluster range of one node are considered as the neighbors of this node. On receiving the Ech_Msg nodes update the neighborhood table.

The flow chart of the proposed technique is shown in Figure 2. Bee algorithm-simulated annealing algorithm is proposed to avoid the local minima problem faced by bee algorithm and to select the best cluster heads by forming ideal clusters. Clustering is achieved by dividing arbitrarily organized sensors into the best possible number of self-determining clusters with cluster head and optimal route to form the initial population. The edge weights between nodes are computed and the objective function is computed. These initial solutions become the initial food source in the proposed BASA algorithm. Once the initial population is found, the bee algorithm is initiated. Each node broadcasts its ID along with its weight W_i to the neighboring nodes and stores the weights W_j of the other nodes within its transmission range.

Bee's algorithm is a population-based search algorithm inspired by bees foraging behaviour [36]. The algorithm starts with search space being populated by worker bees being placed randomly at the location of the initial food source. The fitness of sites visited by worker bees is evaluated and bees with the best fitness continue to be worker bees. Bees which have visited sites with lower fitness value are delegated to onlooker bees. The location of food source with the best fitness becomes the new search location for better solutions. Effectively a cluster with CH can either add new nodes to increase the cluster or remove some nodes to decrease the cluster. Similarly, the CH can be rotated within the cluster. This is achieved by searching neighborhoods of selected sites by assigning scout bees to search near the best e sites. Neighborhood searches of the best e sites are made detailed by recruiting more bees other than the selected bees to follow them. This differential recruitment along with scouting is a key bee's algorithm operation.

The probability p_i of selecting a food source i can be determined by using

$$p_i = \frac{\text{fit}_i}{\sum_{n=1}^{\text{SN}} \text{fit}_n}, \quad (12)$$

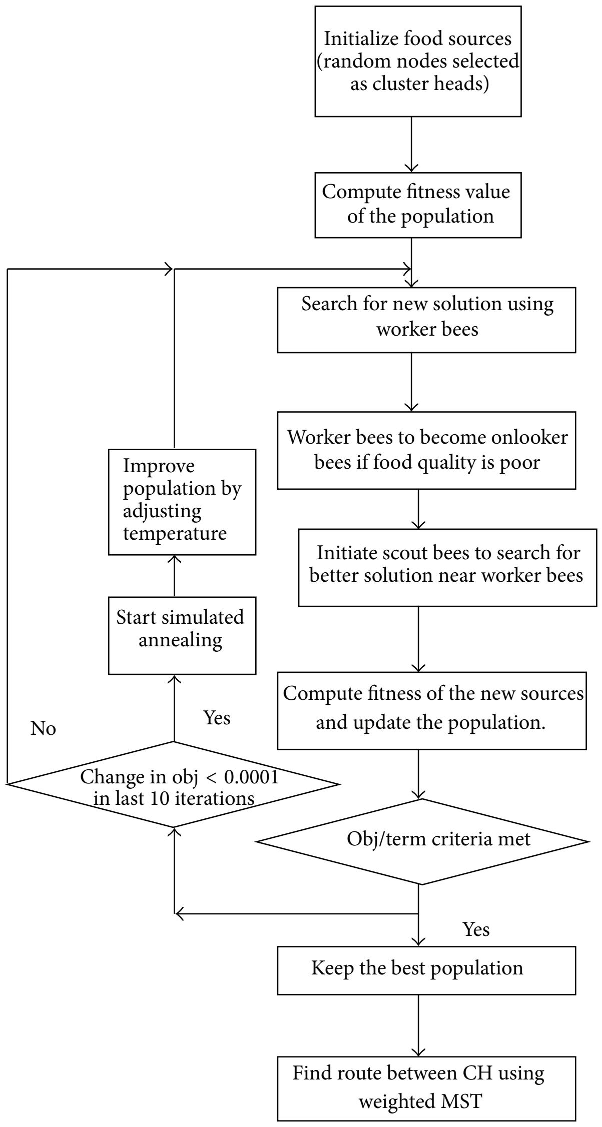

Initialize food sources (random nodes selected as cluster heads)
Compute fitness value of the population
Search for new solution using worker bees
Worker bees to become onlooker bees if food quality is poor
Initiate scout bees to search for better solution near worker bees
Compute fitness of the new sources and update the population.
Obj/term criteria met
Yes
Keep the best population
Find route between CH using weighted MST
Change in obj < 0.0001 in last 10 iterations
Yes
No
Start simulated annealing
Improve population by adjusting temperature

Figure 2: Flowchart of the proposed technique.

and the fitness is given as in

$$\text{fitness}(i) = \begin{cases} 1 + \text{abs}(f(i)), & f(i) < 0 \\ \dfrac{1}{(f(i))} + 1, & f(i) \geq 0, \end{cases} \quad (13)$$

where fit_i is the fitness value of ith solution which represents the nectar amount at food source at ith position and SN is the number of employed bees and also a number of food sources.

The process is iterated till the termination criteria are reached or till the improvement in the fitness does not increase by more than 0.001 in the last 10 iterations. If there is no improvement in the solution, the algorithm could have struck in the local minima. Simulated annealing is starting to climb out of the local minima problem.

The simulated annealing (SA) was introduced in 1983 which is based on the ideas formulated in the early 1950s [37]. Simulated annealing is a relatively straight forward algorithm which includes metropolis Monte Carlo method. The metropolis Monte Carlo algorithm is well suited for simulated annealing, since only energetically feasible states can be sampled at any given temperature. Therefore the simulated annealing algorithm starts at a high temperature with simulation of metropolis Monte Carlo algorithm. The temperature is slowly reduced such that the search space becomes smaller for the metropolis simulation, and when the temperature is low then the system has hopefully settled into the most favorable state. Simulated annealing can also be used for searching the optimum solution of the problems by properly determining the initial (high) and final (low) effective temperatures which are used in place of kT (where k is a Boltzmann's constant) in the acceptance checking and deciding what constitutes a Monte Carlo step [38].

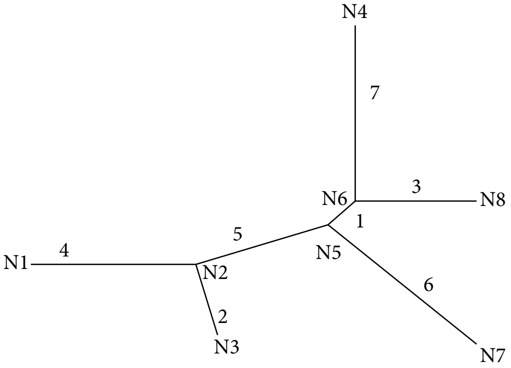


Figure 3: A tree connecting eight vertices.

Simulated annealing is a probabilistic method [39] to find global minimum of a cost function that can have several local minima. Simulated annealing emulates the physical process wherein a solid is slowly cooled so that when eventually its structure is frozen, this happens at a minimum energy configuration. Simulated annealing to compute the probability of acceptance:

$$P_a = \begin{cases} e^{-k\Delta E/T} & \Delta E > 0 \\ 1 & \Delta E \leq 0, \end{cases} \quad (14)$$

where ΔE is the difference between the solution error after it has perturbed and the solution error before it was perturbed, T is the current temperature, and k is a suitable constant.

On identifying the potential cluster heads, the MST algorithm is used for tree construction to find the intracluster routes. Suppose that n points are given in different dimensions, then a tree spanning to these points is a set of straight line segments joining pairs of points [40], so that (1) there are no closed loops, (2) a line visits each point at least once, and (3) the tree is connected.

Figure 3 shows an example of a tree of integer segment lengths. If, for example, vertices N3 and N7 are joined, a closed loop is formed and the result would not be a tree. The length of a tree is the sum of its segments lengths. When a set of n points and the lengths of all $(n/2)$ segments are given, a spanning tree of minimum length (MST) is required. The MST is computed using reaching the base station.

Table 2: The parameters of the network simulation.

Parameters	Values
Initial energy of nodes E_{init}	0.6 J
Amplification coefficient of the free space model E_{fs}	10 pJ-m^2/b
Amplification coefficient of the multipath transmission model E_{amp}	0.0025 pJ-m^2/b
Circuit loss E_{elec}	50 nJ/b
Data packet length	8000 b
Control packet length	120 b
Initial number of worker bees	20
Food source-the initial search space for cluster head (random)	20
Encoding	Binary with
Termination condition	Fitness value <0.0001

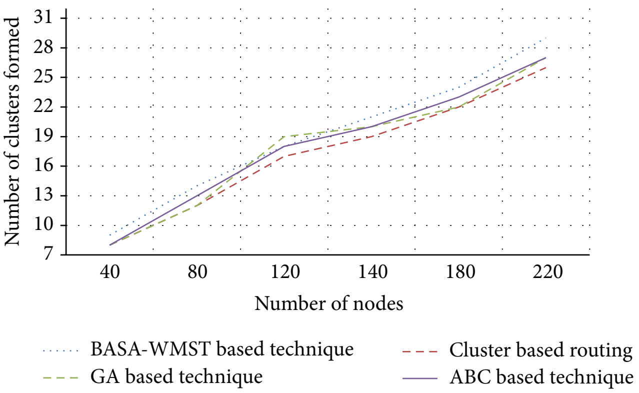


Figure 4: Number of clusters formed.

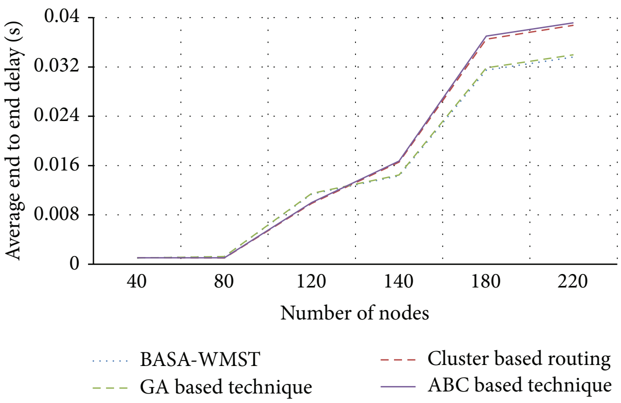


Figure 5: Average end to end delay.

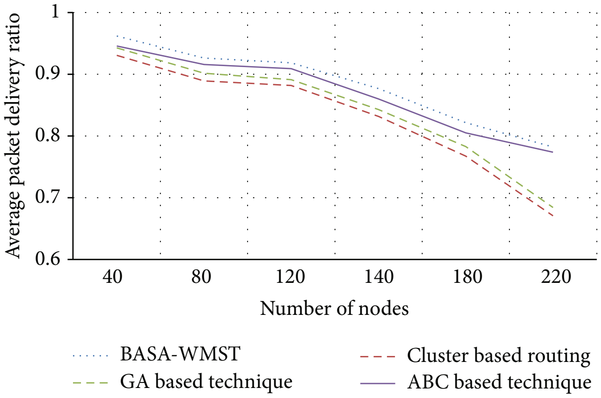


Figure 6: Average Packet Delivery Ratio.

With traditional MST algorithms, construction cost of a minimum spanning tree is $O(m \log n)$, where m is the number of graph edges and n is the number of vertices [41]. The weight of a tree edge is computed by Euclidean distance between two end points. The average weight w of MST edges based on remaining energy and delay is computed. Any edge with a weight $w > w_{\text{avg}}$ is removed leading to a set of disjoint subtrees $t = \{t_1, t_2, \ldots t_i\}$.

The so formed routes are optimal from the spatial perspective since the cluster heads are uniformly distributed over the imperfectly formed wireless sensor network.

4. Experimental Set-Up and Results

Experiments were conducted with different number of mobile sensor nodes, spread over an area of 1000 m by 1000 m with the Base Station being stationary at location (500, 500). The simulation parameters and bee algorithm parameters are shown in Table 2.

Experiments were conducted to simulate the proposed technique and are compared to cluster based routing, GA based cluster formation, and ABC based cluster optimization. Figure 4 shows the number of clusters formed. The proposed BASA WMST technique increases the average number of clusters across by 10.41% compared to cluster based routing for varying number of nodes in the network. ABC produces better clusters compared to GA.

Figure 5 shows the average end to end delay obtained in the network for different number of nodes. As the number of nodes increases, the performance improvement of the proposed technique is in par with GA based technique showing a small marginal average improvement of 1.27% compared to GA based technique.

However, both GA and proposed BASA-WMST show significant decrease in end to end delay as the number of nodes is increased. End to end delay decreased over 14% when compared to cluster based technique. This becomes significant for WSN used in streaming applications. Figure 6 shows the Packet Delivery Ratio obtained under different number of nodes.

The average PDR in the proposed BASA WMST improved by 6.49% when compared to cluster based routing for various numbers of nodes in the network. Compared to GA the PDR improvement was significant by 4.98% and by 1.47% compared to ABC based technique. Figure 7 shows the life time of the network. ABC and the proposed technique significantly improve the life of the network compared to GA based technique.

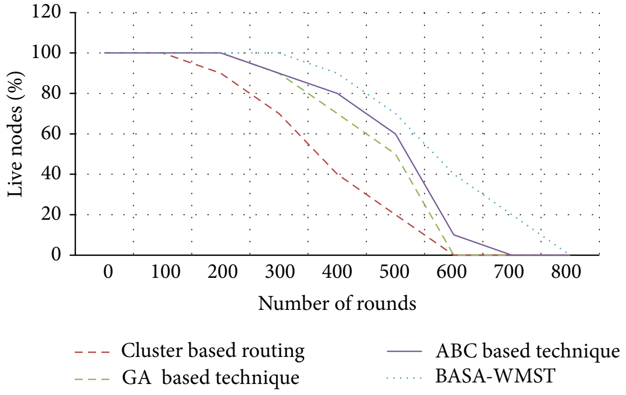


Figure 7: Network life time.

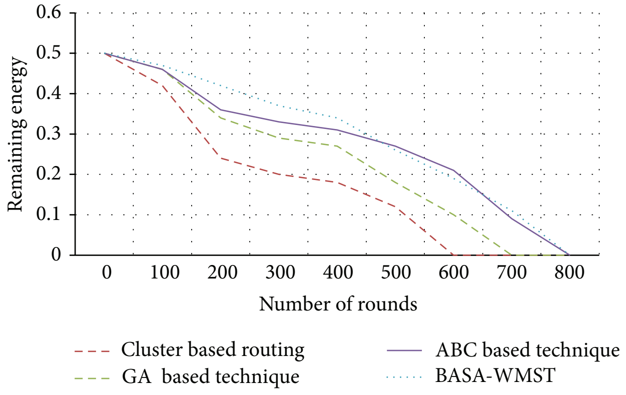


Figure 8: Remaining energy.

Figure 8 shows the average remaining energy in the nodes. The energy savings are significant in the proposed technique compared to GA based technique. Average energy to the tune of 21.67% is observed compared to GA.

5. Conclusion

WSN routing protocols must perform efficiently under mobility and energy constraints. In this paper, clustering is achieved through a hybrid algorithm that divides arbitrarily organized sensors into the best possible number of self-determining clusters with cluster head and optimal route to base station using a novel optimization BASA-WMST. Bee algorithm is incorporated to increase the information exchange among bees and SA is used to escape local optima. Intracluster route is selected from the optimal trees based on weights described. The proposed routing was simulated and compared with conventional cluster based routing and other optimization techniques showing improvements in the QOS.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] S. C. Misra and I. Woungang, *Guide to Wireless Sensor Networks*, vol. 7, Springer, New York, NY, USA, 2009.

[2] C. K. Toh, A. N. Le, and Y. Z. Cho, "Load balanced routing protocols for ad hoc mobile wireless networks," *IEEE Communications Magazine*, vol. 47, no. 8, pp. 78–84, 2009.

[3] J. Le, J. C. Lui, and D. M. Chiu, "DCAR: distributed coding-aware routing in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 596–608, 2010.

[4] N. S. Moayedian and S. J. Golestani, "Optimal scheduling and routing in wireless networks: a new approach," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '09)*, pp. 1–6, IEEE, April 2009.

[5] A. Gagarin, S. Hussain, and L. T. Yang, "Distributed search for balanced energy consumption spanning trees in wireless sensor networks," in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (WAINA '09)*, pp. 1037–1042, Bradford, UK, May 2009.

[6] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[7] T. Watteyne, A. Molinaro, M. G. Richichi, and M. Dohler, "From MANET to IETF ROLL standardization: a paradigm shift in WSN routing protocols," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 4, pp. 688–707, 2011.

[8] B. Priya and S. S. Manohar, "EE-MAC: energy efficient hybrid MAC for WSN," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 526383, 9 pages, 2013.

[9] E. Ahvar, S. Ahvar, G. M. Lee, and N. Crespi, "An energy-aware routing protocol for query-based applications in wireless sensor networks," *The Scientific World Journal*, vol. 2014, Article ID 359897, 9 pages, 2014.

[10] X. Zhu, L. Shen, and T. P. Yum, "Hausdorff clustering and minimum energy routing for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 990–997, 2009.

[11] T. W. Kuo and M. J. Tsai, "On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: NP-completeness and approximation algorithms," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 2591–2595, IEEE, March 2012.

[12] O. Khader, A. Willig, and A. Wolisz, "Distributed wakeup scheduling scheme for supporting periodic traffic in WSNs," in *Proceedings of the Wireless Conference*, pp. 287–292, IEEE, May 2009.

[13] U. Jang, S. Lee, and S. Yoo, "Optimal wake-up scheduling of data gathering trees for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 72, no. 4, pp. 536–546, 2012.

[14] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, vol. 3, pp. 1713–1723, IEEE Societies, April 2003.

[15] T. Kwon and M. Gerla, "Clustering with power control," in *Proceedings of the IEEE Military Communications Conference (MILCOM '99)*, vol. 2, pp. 1424–1428, November 1999.

[16] D. De, A. Sen, and M. D. Gupta, "Cluster based energy efficient lifetime improvement mechanism for WSN with multiple mobile sink and single static sink," in *Proceedings of the 3rd International Conference on Computer and Communication Technology (ICCCT '12)*, pp. 197–199, IEEE, November 2012.

[17] M. J. Reddy, P. S. Prakash, and P. C. Reddy, “Homogeneous and heterogeneous energy schemes for hierarchical cluster based routing protocols in WSN: a survey,” in *Proceedings of the 3rd International Conference on Trends in Information, Telecommunication and Computing*, pp. 591–595, Springer, New York, NY, USA, 2013.

[18] G. Ma and Z. Tao, “A hybrid energy- and time-driven cluster head rotation strategy for distributed wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 109307, 13 pages, 2013.

[19] G. Zheng and Z. Hu, “Tree routing protocol with location-based uniformly clustering strategy in WSNs,” *Journal of Networks*, vol. 5, no. 11, pp. 1373–1380, 2010.

[20] J. Zhang, Y. Xie, D. Liu, and Z. Zhang, “OCTBR: optimized clustering tree based routing protocol for wireless sensor networks,” in *Internet of Things*, pp. 192–199, Springer, Berlin, Germany, 2012.

[21] X. Wang and H. Qian, “Constructing a 6LoWPAN wireless sensor network based on a cluster tree,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 1398–1405, 2012.

[22] S. S. Satapathy and N. Sarma, “TREEPSI: tree based energy efficient protocol for sensor information,” in *Proceeding of the IFIP International Conference on Wireless and Optical Communications Networks*, Bangalore, India, April 2006.

[23] D. Singhal, S. Barjatiya, and G. Ramamurthy, “A novel network architecture for cognitive wireless sensor network,” in *Proceedings of the International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN ’11)*, pp. 76–80, IEEE, July 2011.

[24] S. Hussain and O. Islam, “An energy efficient spanning tree based multi-hop routing in wireless sensor networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’07)*, pp. 4383–4388, Kowloon, Hong Kong, 2007.

[25] M. Hazarath, I. Lucio, and C. Luca, “CAST—a novel trajectory clustering and visualization tool for spatio temporal data,” in *Proceedings of the 1st International conference on Intelligent Human Computer Interaction (IHCI ’09)*, pp. 169–175, Springer, 2009.

[26] H. Guangyan, L. Xiaowei, and H. Jing, “Dynamic minimal spanning tree routing protocol for large wireless sensor networks,” in *Proceedings of the 1st IEEE Conference on Industrial Electronics and Applications (ICIEA ’06)*, pp. 1–5, Singapore, May 2006.

[27] T. M. Chiwewe and G. P. Hancke, “A distributed topology control technique for low interference and energy efficiency in wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 11–19, 2011.

[28] S. Upadhyayula and S. K. S. Gupta, “Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (DAC) in wireless sensor networks,” *Ad Hoc Networks*, vol. 5, no. 5, pp. 626–648, 2007.

[29] D. Tang, X. Liu, Y. Jiao, and Q. Yue, “A load balanced multiple Cluster-heads routing protocol for wireless sensor networks,” in *Proceedings of the IEEE 13th International Conference on Communication Technology (ICCT ’11)*, pp. 656–660, Jinan, China, September 2011.

[30] W. Wang, B. Wang, Z. Liu, L. Guo, and W. Xiong, “A cluster-based and tree-based power efficient data collection and aggregation protocol for wireless sensor networks,” *Information Technology Journal*, vol. 10, no. 3, pp. 557–564, 2011.

[31] G. S. Chhabra and D. Sharma, “Cluster-tree based data gathering in wireless sensor network,” *International Journal of Soft Computing and Engineering*, vol. 1, no. 1, pp. 27–31, 2011.

[32] T. Kumrai, P. Champrasert, and R. Kuawattanaphan, “Heterogeneous wireless sensor network (WSN) installation using novel genetic operators in a multiobjective optimization evolutionary algorithm,” in *Proceedings of the 9th International Conference on Natural Computation (ICNC ’13)*, pp. 606–611, IEEE, 2013.

[33] M. Karimi, H. R. Naji, and S. Golestani, “Optimizing cluster-head selection in wireless sensor networks using genetic algorithm and harmony search algorithm,” in *Proceedings of the 20th Iranian Conference on Electrical Engineering (ICEE ’12)*, pp. 706–710, IEEE, Tehran, Iran, May 2012.

[34] W. Guo, B. Zhang, G. Chen, X. Wang, and N. Xiong, “A PSO-optimized minimum spanning tree-based topology control scheme for wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 985410, 14 pages, 2013.

[35] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS ’00)*, p. 10, January 2000.

[36] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, “The bees algorithm-a novel tool for complex optimisation problems,” in *Proceedings of the 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS ’06)*, pp. 454–459, 2006.

[37] M. Kumar and D. R. Gupta, “A comparative study using simulated annealing and fast output sampling feedback technique based pss design for single machine infinite bus system modeling,” *International Journal of Engineering Research and Applications*, vol. 2, no. 2, pp. 223–228, 2012.

[38] A. Tamilarasi, “An enhanced genetic algorithm with simulated a nnealing for job-shop scheduling,” *International Journal of Engineering, Science and Technology*, vol. 2, no. 1, pp. 144–151, 2010.

[39] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[40] J. C. Gower and G. J. S. Ross, “Minimum spanning trees and single linkage cluster analysis,” *Applied Statistics*, vol. 18, pp. 54–64, 1969.

[41] D. R. Edla and P. K. Jana, “Minimum spanning tree based clustering using partitional approach,” in *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA ’13)*, pp. 237–244, Springer, Berlin, Germany, 2013.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 827509, 12 pages
http://dx.doi.org/10.1155/2014/827509

Research Article

Multiple Memory Structure Bit Reversal Algorithm Based on Recursive Patterns of Bit Reversal Permutation

K. K. L. B. Adikaram,[1,2,3] M. A. Hussein,[1] M. Effenberger,[2] and T. Becker[1]

[1] *Group of Bio-Process Analysis Technology, Technische Universität München, Weihenstephaner Steig 20, 85354 Freising, Germany*
[2] *Institut für Landtechnik und Tierhaltung, Vöttinger Straße 36, 85354 Freising, Germany*
[3] *Computer Unit, Faculty of Agriculture, University of Ruhuna, Mapalana, 81100 Kamburupitiya, Sri Lanka*

Correspondence should be addressed to K. K. L. B. Adikaram; lasantha@daad-alumni.de

Received 6 April 2014; Revised 4 June 2014; Accepted 5 June 2014; Published 17 July 2014

Academic Editor: Ker-Wei Yu

With the increasing demand for online/inline data processing efficient Fourier analysis becomes more and more relevant. Due to the fact that the bit reversal process requires considerable processing time of the Fast Fourier Transform (FFT) algorithm, it is vital to optimize the bit reversal algorithm (BRA). This paper is to introduce an efficient BRA with multiple memory structures. In 2009, Elster showed the relation between the first and the second halves of the bit reversal permutation (BRP) and stated that it may cause serious impact on cache performance of the computer, if implemented. We found exceptions, especially when the said index mapping was implemented with multiple one-dimensional memory structures instead of multidimensional or one-dimensional memory structure. Also we found a new index mapping, even after the recursive splitting of BRP into equal sized slots. The four-array and the four-vector versions of BRA with new index mapping reported 34% and 16% improvement in performance in relation to similar versions of Linear BRA of Elster which uses single one-dimensional memory structure.

1. Introduction

The efficiency of a bit reversal algorithm (BRA) plays a critical role in the Fast Fourier Transform (FFT) process because it contributes 10% to 50% of total FFT process time [1]. Therefore, it is vital to optimize the BRA to achieve an efficient FFT algorithm. In 2009, Elster showed the relation between the first and the second halves of the BRP [2], but did not implement it. Elster stated that implementation of this relation may cause serious impact on cache performance of modern computers. As Elster stated, use of a two-dimensional memory structure to implement this relation reduced the efficiency of the bit reversal permutation (BRP). In contrast, the efficiency of the BRA increased when a one-dimensional memory structure was used for index mapping. When two equal sided one-dimensional memory structures were used, the performance was even much better than with single one-dimensional memory structure. Also, it was found out that bit reversal permutation can be further split into equal size blocks recursively, up to maximum of $\log_2(n)$ times, where n is the number of samples and $n = 2^k$; $k \in Z^+$. These two findings motivate us to introduce a BRA which is capable of using 2^k $(k = 2, 3, \ldots, \log 2(n))$ equal-sized $(n/2^k)$ one-dimensional memory structures.

In 1965 Cooley and Tukey introduced the FFT algorithm, which is an efficient algorithm to compute the Discrete Fourier Transformation (DFT) and its inverse [3]. FFT is a fast algorithm that has replaced the process of DFT, which had been used frequently in the fields of signal and image processing [4–7]. The structure of the FFT algorithm published by Cooley and Tukey known as radix-2 algorithm [3] is the most popular one [5]. There are several other algorithm structures as radix-4, radix-8, radix-16, mixed-radix, and split-radix [8].

To apply FFT to a certain signal, there are basically two major requirements. The first requirement is $n = b^N$, where n is the number of samples of the signal, $N \in Z^+$, and b is the selected radix structure, for example, $b = 2$, 4, 8, and 16 for radix-2, radix-4, radix-8, and radix-16, respectively. The second requirement is that the input (or output) samples must be arranged according to a certain order to obtain the correct output [3, 5, 8, 9]. The BRA is used to create the order of input or output permutation according to the required order. The BRA, used in most FFT algorithms, including the original Cooley-Tukey algorithm [3], is known as bit reversal method (BRM). The BRM is an operation, for exchanging two elements $x(k)$ and $x(\widetilde{k})$ of an array of length m as shown in (1) and (2), respectively, where a_j are either 0 or 1 and b is the relevant base 2, 4, 8, or 16 depending on the selected radix structure:

$$k = \sum_{j=0}^{m-1} a_j b^j, \qquad (1)$$

$$\widetilde{k} = \sum_{j=0}^{m-1} a_j b^{m-1-j}. \qquad (2)$$

All the later algorithms for creating BRP were named as BRA (bit reversal algorithm), though they used other techniques like patterns of BRP instead of bit reversing techniques.

During the last decades, many publications addressed new BRAs [10] by improving the already existing original BRA (BRM) or using totally different approaches. In 1996, Karp compared the performance of 30 different algorithms [10] against uniprocessor systems (computer system with a single central processing unit) with different memory systems. Karp found that the performance of a BRA depended on the memory architecture of the machine and the way of accessing the memory. Karp stated two hardware facts that influence the BRA, namely, the memory architecture and the cache size of the machine. According to Karp, a machine with hierarchical memory is slower than a machine with vector memory (computers with vector processor), and algorithms do not perform well when array size is larger than the cache size. Also Karp pointed out four features of an algorithm that influence the BRA, namely, memory access technique, data reading sequence, size of the index of memory, and type of arithmetic operations. According to Karp, an algorithm that uses efficient memory access techniques is the fastest among algorithms with exactly the same number of arithmetic operations. Algorithms are faster if (i) they require only a single pass over the data, (ii) they use short indexes, and (iii) they operate with addition instead of multiplication.

Karp especially mentioned that the algorithm published by Elster [11] in 1989 was different from other algorithms, because it used a pattern of BRP rather than improving decimal to binary and binary to decimal conversion. According to the findings of Karp, Elster's "Linear Bit Reversal Algorithm" (LBRA) performs much better in most of the cases. The publication of Elster (1989) [11] consists of two algorithms to achieve BRP. One algorithm used a pattern of BRP and the other one used bit shifting operations. Both algorithms are interesting because they eliminate the conventional bit reversing mechanism, which need more computing time. The algorithm by Rubio et al. (BRA-Ru) of 2002 [12] is another approach that uses an existing pattern of BRP. However, the pattern described in Rubio's algorithm is different from the pattern described in Elster's [11]. In 2009, Elster and Meyer published an improved version of "Linear Register-Level Bit Reversal" which was published in 1989 [11] as "Elster's Bit Reversal" (EBR) algorithm. Elster mentioned it is possible to generate the second half of the BRP by incrementing the relevant element of the first half by one. Also, Elster mentioned there can be a serious impact on cache performance of the computer if the said pattern (Figure 1) is used.

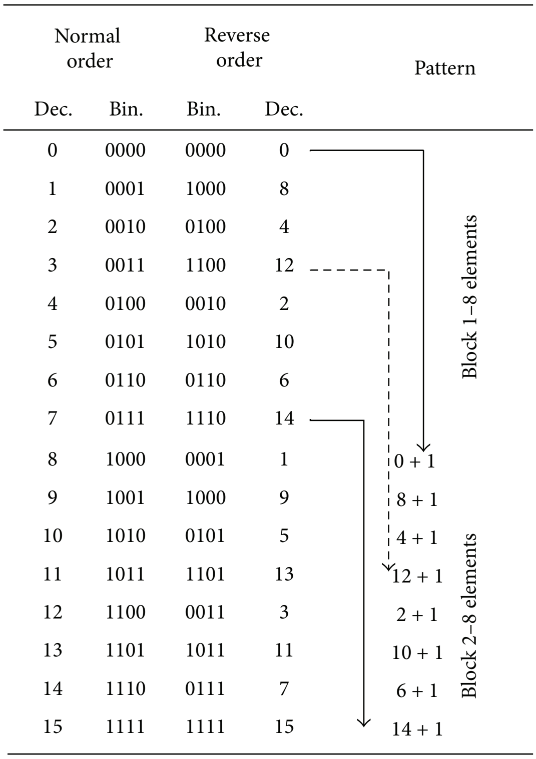
Programming languages provide different data structures [13] which handle memory in different ways. In addition, the performance of the memory depends on machine architecture and operating system [14]. Therefore, the efficiency of the memory is the resultant output of the performances of hardware, operating system, programming language, and selected data structure.

Based on the physical arrangement of memory elements, there are two common ways of allocating memory for a series of variables: "slot of continuous memory elements" and "collection of noncontinuous memory elements," commonly known as "stack" and "heap" [15]. In most programming languages, the term *array* is used to refer to a "slot of continuous memory elements." Arrays are the simplest and most common type of data structure [16, 17] and, due to continuous physical arrangement of memory elements, provide faster access than "collection of noncontinuous memory elements" memory types. However, with the development of programming languages, different types of data structures were introduced with very similar names to the standard names like array, stack, and heap. The names of new data structures sometimes did not agree with the commonly accepted meaning. "Stack," "Array," and "ArrayList" provided by Microsoft Visual C++ (VC++) [18, 19] are good examples. According to the commonly accepted meaning they should be a "slot of continuous memory elements," but they are in fact a "collection of noncontinuous memory elements." Therefore, it is not a good practice to determine the performance of a certain data structure just by looking at its name. To overcome this ambiguous situation, we use "slot of continuous memory elements" to refer to "primitive array" (or array) type memory structures.

Due to the very flexible nature, vector is the most common one among the different types of data structures [14]. Vector was introduced with C++, which is one of the most common and powerful programming languages which has been used since 1984 [20]. However, as most of other data structures, the term "*vector*" is used to refer to memory in computers with processor architecture called "vector processor." In this paper the term vector is used to refer to the vector data structure that is used in the C++ interface.

Index mapping is a technique that can be used to improve the efficiency of an algorithm by reducing the arithmetic load of the algorithm [8]. If $n : [0, N-1]$ and N is not prime, N can be defined as $N = \prod_{i=0}^{n_i-1} N_i$, where $n_i : [0, N_i - 1] < N$. This allows the usage of small ranges of n_i instead of large range of n and maps a function $f(n)$ into a multidimensional function $f'(n_1, n_2, \ldots, n_i)$.

There are two common methods of implementing index mapping: one-dimensional or multidimensional memory



Normal order		Reverse order		Pattern
Dec.	Bin.	Bin.	Dec.	
0	0000	0000	0	
1	0001	1000	8	
2	0010	0100	4	
3	0011	1100	12	
4	0100	0010	2	
5	0101	1010	10	
6	0110	0110	6	
7	0111	1110	14	
8	1000	0001	1	0 + 1
9	1001	1000	9	8 + 1
10	1010	0101	5	4 + 1
11	1011	1101	13	12 + 1
12	1100	0011	3	2 + 1
13	1101	1011	11	10 + 1
14	1110	0111	7	6 + 1
15	1111	1111	15	14 + 1

Figure 1: Relation between first and second halves of the BRP for $n = 16$.

structures. In addition, it is also possible to implement the index mapping using several equal size one-dimensional memory structures. However, this option is not popular as it is uncomfortable for programming. The performance of modern computers is highly dependent on the effectiveness of the cache memory of the CPU [4]. To achieve the best performance of a series of memory elements, the best practice is to maintain sequential access [4]. Otherwise, the effectiveness of the cache memory of the central processing unit (CPU) will be reduced. Index mapping with multidimensional data structures violates sequential access due to switching between columns/rows and thus reduces the effectiveness of the cache memory. Therefore, it is generally accepted that the use of a multidimensional data structure reduces computer performance [4].

In this paper an efficient BRA is introduced to create the BRP based on multiple memory structures and recursive patterns of BRP. The findings of this paper show that the combination of multiple one-dimensional memory structures, index mapping, and the recessive pattern of BRP can be used to improve the efficiency of BRA. These findings are very important to the field of signal processing as well as any field that is involved in index mapping techniques.

2. Material and Methods

2.1. New Algorithm (BRA-Split). Elster stated that it is possible to generate the second half of the BRP by incrementing items in the first half by one [2] (Figure 1), without changing the order and the total number of calculations of the algorithm. Due to the recursive pattern of BRP, it can be further divided into equal size blocks by splitting each block recursively (Maximum $\log_2 N$ times). After splitting s times, BRP is divided into 2^s equal blocks each containing $n/2^s$ elements. The relation between the elements in blocks is given as follows:

$$B\left(2^m + j\right)[i] = B\left(j\right)[i] + 2^{S-m-1}, \qquad (3)$$

where $B(2^m + j)[i]$ is ith element of block $B(2^m + j)$ and $m = 0, 1, \ldots, s-1$, $j = 1, \ldots, 2^m$.

Table 1 shows the relationship between elements in blocks according to the index mapping shown in (3), after splitting BRP one time and two times for $n = 16$. Depending on the requirement, the number of splitting can be increased.

2.2. Evaluation Process of New Algorithm. To evaluate algorithms, we used Windows 7 and Visual C++ 2012 on a PC with multicore CPU (4 cores, 8 logical processors) and 12 GB memory. Detailed specifications of the PC and the software are given in Table 2. To eliminate limits of memory and address space related to the selected platform, the compiler option *"/LARGEADDRESSAWARE"* was set [21] and platform was set to "x64." All other options of the operating system and the compiler were kept unchanged.

The new algorithm was implemented using single one-dimensional memory structure and the most common multidimensional memory structure. Furthermore, the new BRA was implemented using several equal size one-dimensional memory structures (multiple memory structure).

TABLE 1: Relation between elements of 16-element BRP (n = 16) for split = 1 and split = 2.

		Split (S) = 1				Split (S = 2)			
Normal order	Reverse order	Blocks $(1, \ldots, 2^s)$	Index of the block i	Reverse order calculation	Calculation method $m = 0, \ldots, s-1$ $(m = 0)$ $j = 1, \ldots, 2^m$ $m = 0$: $j = 1$ $i = 1, \ldots, n/2^{s-1}$ $i = 1, \ldots, 7$	Blocks $(1, \ldots, 2^s)$	Index of the block i	Reverse order calculation	Calculation method $m = 0, \ldots, s-1$ $(m = 0, 1)$ $j = 1, \ldots, 2^m$ $m = 0$: $j = 1, m = 1$: $j = 1, 2$ $i = 1, \ldots, n/2^{s-1}$ $i = 1, 2, 3$
0	**0**	1	0	*0*	*Initialized*	1	0	*0*	*Initialized*
1	**8**		1	**8** = 8 + *0*	Use Elster's Linear Bit Reversal method		1	**8** = 8 + *0*	Use Elster's Linear Bit Reversal method
2	**4**		2	**4** = 4 + *0*			2	**4** = 4 + *0*	
3	**12**		3	**12** = 12 + *0*			3	**12** = 12 + *0*	
4	**2**		4	**2** = 2 + *0*		2	0	*2*	*Initialized*
5	**10**		5	**10** = 10 + *0*			1	**10** = 8 + *2*	$B(2^m + j)[i] = B(j)[i] + 2^{s-m-1}$
6	**6**		6	**6** = 6 + *0*			2	**6** = 4 + *2*	For $m = 0$, $j = 1$, $i = 1, 2, 3$ $B(2^0 + 1)[i] = B(1)[i] + 2^{2-0-1}$
7	**14**		7	**14** = 14 + *0*			3	**14** = 12 + *2*	$B(2)[i] = B(1)[i] + 2$
8	**1**	2	0	*1*	*Initialized*	3	0	*1*	*Initialized*
9	**9**		1	**9** = 8 + *1*	$B(2^m + j)[i] = B(j)[i] + 2^{s-m-1}$ For $m = 0, j = 1, i = 1, \ldots, 7$ $B(2^0 + 1)[i] = B(1)[i] + 2^{2-0-1}$ $B(2)[i] = B(1)[i] + 1$		1	**9** = 8 + *1*	$B(2^m + j)[i] = B(j)[i] + 2^{s-m-1}$
10	**5**		2	**5** = 4 + *1*			2	**5** = 4 + *1*	For $m = 1, j = 1, i = 1, 2, 3$ $B(2^1 + 1)[i] = B(1)[i] + 2^{2-1-1}$
11	**13**		3	**13** = 12 + *1*			3	**13** = 12 + *1*	$B(3)[i] = B(1)[i] + 1$
12	**3**		4	**3** = 2 + *1*		4	0	3	*Initialized*
13	**11**		5	**11** = 10 + *1*			1	**11** = 10 + *1*	$B(2^m + j)[i] = B(j)[i] + 2^{s-m-1}$
14	**7**		6	**7** = 6 + *1*			2	**7** = 6 + *1*	For $m = 1, j = 2, i = 1, 2, 3$ $B(2^1 + 2)[i] = B(2)[i] + 2^{2-1-1}$
15	**15**		7	**15** = 14 + *1*			3	**15** = 14 + *1*	$B(4)[i] = B(2)[i] + 2$

Table 2: Hardware and software specifications of the PC.

Specifications	
Processor	Intel Core i7 CPU 870 @ 2.93 GHz (4 cores, 8 threads)
RAM	12 GB, DDR3-1333, 2 channels
Memory bandwidth	21 GB/s
L1, L2, and L3 cache	4 × 64 KB, 4 × 256 KB, and 8 MB shared
L1, L2, and L3 cache line size	64 bit
Brand and type	Fujitsu, Celsius
BIOS settings	Default (hyper threading enabled)
OS and service pack	Windows 7 professional with service pack 1
System type	64 bit operating system
OS settings	Default
Visual Studio 2012	Version 11.0.50727.1 RTMREL
.NET Framework	Version 4.5.50709

The next task was to identify a suitable data structure from different types of available data structures. We considered several common techniques as summarized in Table 3. Data structure 1 mentioned in Table 3 is not supporting dynamic memory allocation (need to mention the size of the array when array is being declared). For general bit reversal algorithm, it is a must to have dynamic memory allocation to cater different sizes of samples. Even after setting the compiler option *"/LARGEADDRESSAWARE"* [21], data structures 3 and 4 mentioned in Table 3 were not supported for accessing memory greater than 2 GB. Therefore, structures 1, 3, and 4 were rejected and memory structures 2 (array) and 5 (vector) were used to create all one-dimensional memory structures. The same versions of array and vector were used to create multidimensional memory structures.

The new algorithm mentioned in Section 2.1 was implemented using C++ in 24 types of memory structures as shown in Table 4. The performance of these algorithms was evaluated considering the "clocks per element" (CPE) consumed by each algorithm. To retrieve this value, first, average CPE for each sample size of 2^n where $n : [21, 31]$ (11 sample sizes) were calculated after executing each algorithm 100 times. This gave 11 CPE representing each sample size. Finally, the combined averageof CPE was calculated for each algorithm by averaging those 11 values along with "combined standard deviation." The combined average of CPE was considered as the CPE for each algorithm. The built-in "clock" function of C++ was used to calculate the clocks. Combined standard deviation was calculated using the following:

$$\sigma_c = \left(\frac{\sum_{s=1}^{k} \left(n_s \sigma_s^2 + n_s \left(\overline{X_s} - \overline{X_c} \right)^2 \right)}{\sum_{s=1}^{k} n_s} \right)^{1/2}, \qquad (4)$$

where $\overline{X_c} = \sum_{s=1}^{k} n_s \overline{X_s} / \sum_{s=1}^{k} n_s$, s is the number of samples, n_s is number of samples in each sample, and σ_s is the standard deviation of each sample.

Algorithms 1, 2, and 3 illustrate the implementation of new BRA with single one-dimensional memory structure, multidimensional memory structure, and multiple memory structures, respectively. The algorithm illustrated in Algorithm 1 (BRA_Split_1_1A) was implemented using primitive array for split = 1. The algorithm BRM_Split_2_4A (Algorithm 2) was implemented using vectors for split = 2. The algorithm BRM_Split_2_4A (Algorithm 3) was implemented using primitive array for split = 2. A sample permutation filling sequence of algorithms with single one-dimensional memory structures is illustrated in Figure 2. Figure 3 illustrates a sample permutation filling sequence of both multidimensional and multiple memory structures.

Secondly, arithmetic operations per element (OPPE) were calculated for each algorithm. Arithmetic operations within each algorithm were located in three regions of the code: inner FOR loop, outer FOR loop, and outside of the loops. Then, the total number of operations (OP) can be defined as

$$\mathrm{OP} = K_1 * C_1 + K_2 * C_2 + C_3, \qquad (5)$$

where C_1, C_2, and C_3 are the number of operations in inner FOR loop, outer FOR loop, and outside of the loops. K_1 and K_2 are the number of iterations of outer loop and inner loop. Equation (5) can be represented as

$$\mathrm{OP} = \sum_{t=0}^{\log_2(\mathrm{NS})-s-1} 2^t * C_1 + \sum_{t=0}^{\log_2(\mathrm{NS})-s-1} C_2 + C_3, \qquad (6)$$

where NS is the number of samples and s is the number of splits.

The main contribution to calculations comes from the inner loop. Comparing with the contribution of operations in the inner loop, the contribution of operations in rest of the code is very small. For example, consider the algorithm BRA_Split_1_1A shown in Algorithm 1. As sample size is 2^{31}, $K_1 * C_1 \approx 2.15 * 10^9 * C_1$ and $K_2 * C_2 + C_3 \approx 1000$, where $C_1 > 1$. Therefore, only the operations of inner loop were considered for evaluation. Then (6) can be simplified as

$$\mathrm{OP} = \sum_{t=0}^{\log_2(\mathrm{NS})-s-1} 2^t * C_1. \qquad (7)$$

The "operations per element" (OPPE) can be defined as

$$\mathrm{OPPE} = \frac{\left(\sum_{t=0}^{\log_2(\mathrm{NS})-s-1} 2^t * C_1 \right)}{\mathrm{NS}}. \qquad (8)$$

For FFT always $\mathrm{NS} = 2^k; k \in Z^+$.

Then from (7)

$$\mathrm{OPPE} = \frac{\left(\sum_{t=0}^{k-s-1} 2^t * C_1 \right)}{2^k}, \qquad (9)$$

$$\sum_{t=0}^{k} 2^t + 1 = 2^{k+1}. \qquad (10)$$

Table 3: Common memory allocating methods that are used in Visual C++.

Number	Name	Syntax	Nature of memory layout
1	Array	int BRP[1000]	Slot of continuous memory elements
2	Array	int* BRP = new int[n]	Slot of continuous memory elements
3	Array	array⟨int⟩^BRP = gcnew array⟨int⟩(n)	Collection of noncontinuous memory elements
4	ArrayList	ArrayList ^BRP = gcnew ArrayList()	Collection of noncontinuous memory elements
5	Vector	std::vector⟨int⟩ BRP(n)	Collection of noncontinuous memory elements

Table 4: Different versions of new BRA implemented with different data structures.

Split (s)	Data structure type	Algorithm		
		Single memory structure	Multidimensional memory structure	Multiple memory structure
1	Array (A)	BRA_Split_1_1A	BRA_Split_1_2DA	BRA_Split_1_2A
1	Vector (V)	BRA_Split_1_1V	BRA_Split_1_2DV	BRA_Split_1_2V
2	Array (A)	BRA_Split_2_1A	BRA_Split_2_4DA	BRA_Split_2_4A
2	Vector (V)	BRA_Split_2_1V	BRA_Split_2_4DV	BRA_Split_2_4V
3	Array (A)	BRA_Split_3_1A	BRA_Split_3_8DA	BRA_Split_3_8A
3	Vector (V)	BRA_Split_3_1V	BRA_Split_3_8DV	BRA_Split_3_8V
4	Array (A)	BRA_Split_4_1A	BRA_Split_4_16DA	BRA_Split_4_16A
4	Vector (V)	BRA_Split_4_1V	BRA_Split_4_16DV	BRA_Split_4_16V

Naming convention for algorithms: "BRA_Split" + <*Number of splits*> + <*nature of memory structure*>. xA, xV: x number of arrays and x number of vectors. xDA, xDV: single x-dimensional array and single x-dimensional vector.

```
void mf_ BRM_Split_1_1A (unsigned int ui_NS, int ui_log2NS)
{
unsigned int ui_N
unsigned int ui_EB;
unsigned int ui_t;
unsigned int ui_L;
unsigned int ui_DL;

ui_N = ui_NS;   // Number of samples
ui_t = ui_log2NS – 1;
ui_EB = ui_N/2;
ui_L = 1;
unsigned int* BRP = new unsigned int[ui_N];
BRP[0] = 0;
BRP[ui_EB] = 1;
for (unsigned int q = 0; q < ui_t; q++)
  {
      ui_DL = ui_L + ui_L;
      ui_N = ui_N/2;
      for (unsigned int j = ui_L; j < ui_DL; j++)
      {
         BRP[j] = BRP[j – ui_L] + ui_N;
         BRP[ui_EB + j] = BRP[j] + 1;
      }
ui_L = ui_L + ui_L;
  }
delete[] BRP;
}
```

Algorithm 1: C++ implementation of new BRA with single array for split = 1 (BRA_Split_1_1A).

```
Void mf_ BRM_Split_2_4DV (unsigned int ui_NS, int ui_log2NS)
 {
  unsigned int ui_N;
  unsigned int ui_EB;
  unsigned int ui_t;
  unsigned int ui_L;
  unsigned int ui_DL;

  ui_N = ui_NS;   // Number of samples
  ui_t = ui_log2NS − 2;
  ui_EB = ui_N/4;
  ui_L = 1;

  std::vector<std::vector<unsigned int>> BRP(4,
std::vector<unsigned int>(ui_EB));
BRP[0][0] = 0;
BRP[1][0] = 2;
BRP[2][0] = 1;
BRP[3][0] = 3;
for (unsigned int q = 0; q < ui_t; q++)
 {
   ui_DL = ui_L + ui_L;
   ui_N = ui_N/2;
   for (unsigned int j = ui_L; j < ui_DL; j++)
    {
      BRP[0][j] = BRP[0][j − ui_L] + ui_N;
      BRP[1][j] = BRP[0][j] + 2;
      BRP[2][j] = BRP[0][j] + 1;
      BRP[3][j] = BRP[1][j] + 1;
    }
   ui_L = ui_L + ui_L;
} }
```

Algorithm 2: C++ implementation of new BRA with four arrays for split = 2 (BRM_Split_2_4A).

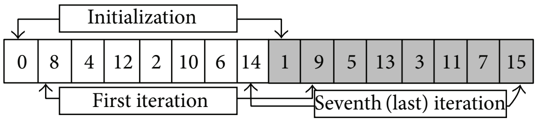


Figure 2: Permutation filling sequence of new BRA with single memory structure for $n = 16$ and split =1 (BRA_Split_1_1A).

For large k, $2^k + 1 \approx 2^k$. Then,

$$\sum_{t=0}^{k} 2^t = 2^{k+1}. \quad (11)$$

Because the considered sample size is 2^{21} to 2^{31}, the value k can be considered as large. Then, from (9)

$$\text{OPPE} = \frac{C_1 * \left(2^{k-s}\right)}{2^k}, \quad (12)$$

$$\text{OPPE} = \frac{C_1}{2^s}. \quad (13)$$

According to (13), OPPE is $f(C_1, s)$. The value C_1 (operations in inner loop) and the value s (number of splits) are a constant for a certain algorithm.

To evaluate the performance of new BRA, we selected three algorithms (LBRA, EBR, and BRA-Ru) which used a pattern instead of conventional bit reversing method. The performance of vector and array versions of the best version of new BRA was compared with the relevant versions of selected algorithms.

Table 5: The number of operations in the inner loop (C_1) of each algorithm for different splits (s). (The total number of operations in each algorithm ≈ C_1.)

Memory structure type	Value of C_1			
	$s = 1$	$s = 2$	$s = 3$	$s = 4$
Single one-dimension (array and vector)	8	15	30	61
Multidimension (array and vector)	7	11	19	35
Multiple one-dimension (array and vector)	7	11	19	35

3. Results and Discussion

Our objective was to introduce BRA using recursive pattern of the BRP that we identified. We used multiple memory structures, which is a feasible yet unpopular technique to implement index mapping. According to Table 5, the numbers of operations in all the array and vector versions of both

```
Void mf_ BRM_Split_2_4A (unsigned int ui_NS, int ui_log2NS)
{
 unsigned int ui_N
 unsigned int ui_EB;
 unsigned int ui_t;
 unsigned int ui_L;
 unsigned int ui_DL;

 ui_N = ui_NS;   // Number of samples
 ui_t = ui_log2NS − 2;
 ui_EB = ui_N/4;
 ui_L = 1;
 unsigned int* BRP1 = new unsigned int[ui_EB];
 unsigned int* BRP2 = new unsigned int[ui_EB];
 unsigned int* BRP3 = new unsigned int[ui_EB];
 unsigned int* BRP4 = new unsigned int[ui_EB];

 BRP1[0] = 0;
 BRP2[0] = 2;
 BRP3[0] = 1;
 BRP4[0] = 3;
 for (unsigned int q = 0; q < ui_t; q++)
 {
   ui_DL = ui_L + ui_L;
   ui_N = ui_N/2;
   for (unsigned int j = ui_L; j < ui_DL; j++)
   {
     BRP1[j] = BRP1[j − ui_L]+ ui_N;
     BRP2[j] = BRP1[j] + 2;
     BRP3[j] = BRP1[j] + 1;
     BRP4[j] = BRP2[j] + 1;
   }
   ui_L = ui_L + ui_L;
 }
 delete[] BRO1;
 delete[] BRO2;
 delete[] BRO3;
 delete[] BRO4;
 }
```

ALGORITHM 3: C++ implementation of new BRA with four one-dimensional arrays for split = 2 (BRM_Split_2_4A).

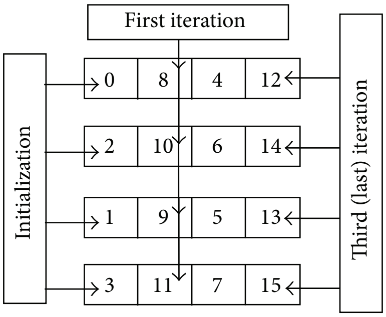


FIGURE 3: Permutation filling sequence of 4 individual and single 4-dimensional memory structure for $n = 16$.

multidimensional and multiple memory structures are the same. Also, Figure 5 shows continuous decrement of OPPE when the number of splits increases. Then, the algorithm with the highest number of splits and the lowest number of operations is the one which is expected to be most efficient. However, results in relation with CPE (Figure 5) show that the new algorithm with four memory structures of array is the fastest and most consistent in the selected range. Two, four, eight, and sixteen multiple array implementations of new BRA reported 25%, 34%, 33%, and 18% higher efficiency,

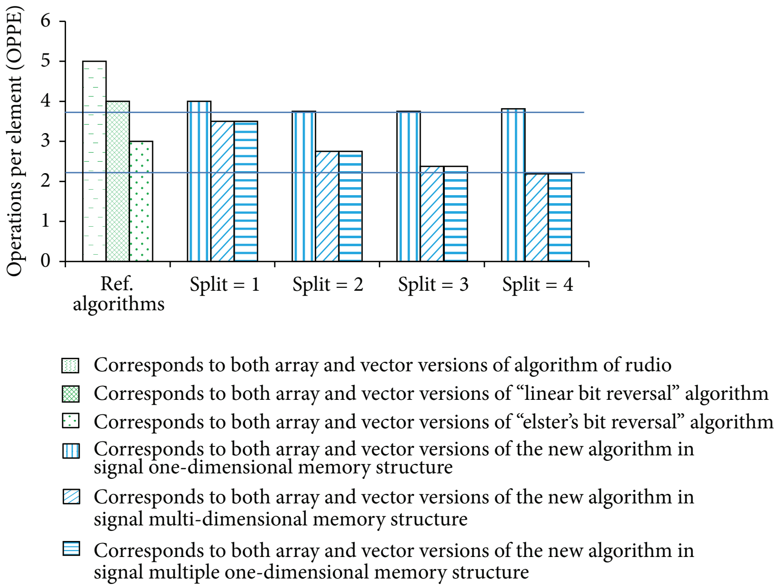


Figure 4: Operations per element versus reference algorithms and new algorithm with different *s* (splits), where dashed column corresponds to both array and vector versions of algorithm of Rudio, cross lines column corresponds to both array and vector versions of "Linear Bit Reversal" algorithm, dotted column corresponds to both array and vector versions of "Elster's Bit Reversal" algorithm, vertical lines column corresponds to both array and vector versions of the new algorithm in single one-dimensional memory structure, inclined lines column corresponds to both array and vector versions of the new algorithm in single multidimensional memory structure, and horizontal lines column corresponds to both array and vector versions of the new algorithm in multiple one-dimensional memory structures.

respectively, in relation to the array version of LBRA. The algorithm with eight memory structures has nearly the same CPE as the four-array and four-vector versions, but is less consistent. On the other hand, the four-vector implementation of the new algorithm is the fastest and most consistent among all vector versions. Two, four, eight, and sixteen multiple vector implementations of new BRA reported 13%, 16%, and 16% higher and 23% lower efficiency, respectively, in relation to the vector version of LBRA. This result proves that at a certain point, multiple memory structure gives the best performances in the considered domain. Also, usage of multiple memory structures of primitive array is a good option for implementing index mapping of BRP compared to multidimensional or single one-dimensional memory structures.

Due to the flexible nature of the vector, it is commonly used for implementing algorithms. According to Figure 4 there is no difference in OPPE between array and vector versions. However, our results in Figure 5 show that the vector versions of BRA always required more CPE (44%–142%) than the array version. The structure of vector gives priority to generality and flexibility rather than to execution speed, memory economy, cache efficiency, and code size [22]. Therefore, vector is not a good option with respect to efficiency.

The results in Table 5 and Figure 4 show that there is no difference between the number of calculations and OPPE for equal versions of algorithms with multidimensional and multiple memory structure. Structure and type of calculations are the same for both types. The only difference is the nature of the memory structure: multidimension or multiple one-dimension. When CPE is considered, it shows 19%–79% performance efficiency from algorithms with multiple one-dimension memory structures. The reason for that observation is that the memory access of multidimensional memory structure is less efficient than one-dimensional memory structure [22].

We agree with the statement of Elster about index mapping of BRP [2] and the generally accepted fact (the usage of multidimensional memory structures reduces the performance of a computer) [4] only with respect to multidimensional memory structures of vector. Our results show that even with multidimensional arrays there are situations where the new BRA performs better than the same type of one-dimensional memory structure. The four, eight, and sixteen dimensional array versions of new BRA perform 8%, 10%, and 2% in relation to one-dimensional array version of new BRA. Some results in relation to single one-dimensional memory structure implementation of new BRA are also not in agreement with the general accepted idea. For example sample size = 2^{31}, the two-dimensional vector version of new BRA (BRA_Split_1_2DV) reported $5.42E-05$ CPE which is 389% higher in relation to average CPE of sample size range of 2^{21} to 2^{30}. Also, the inconsistency was very high. Therefore, we excluded values related to sample size = 2^{31} for the two-dimensional vector version.

We observed very high memory utilization with the two-dimensional vector version, especially with sample size = 2^{31}. Windows task manager showed that the memory utilization of all the considered algorithms was nearly the same for all sample sizes except for multidimension versions of vector. The multidimensional version of vector initially

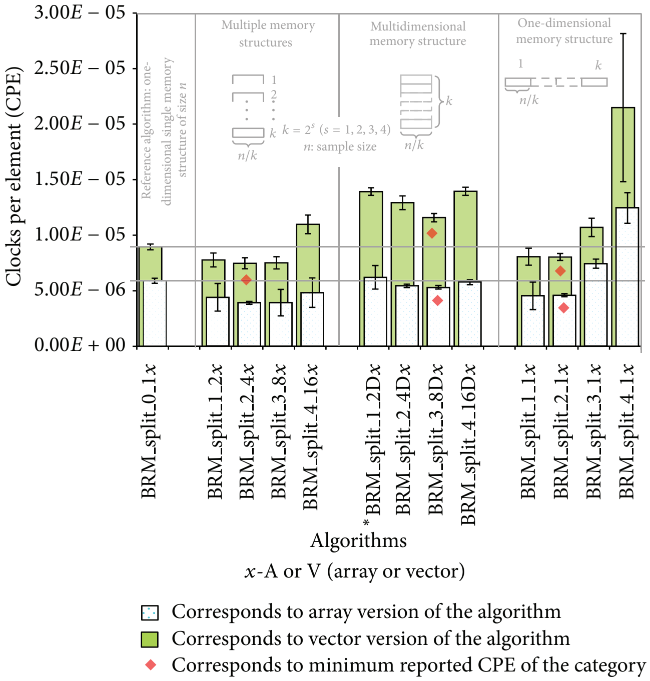


Figure 5: Clocks per element (combined average) versus algorithm for the sample size range from 2^{21} to 2^{31}, where blue dotted column corresponds to array version of the algorithm, green column corresponds to vector version of the algorithm, and red rhombus corresponds to minimum reported CPE of the category. *For vector version, sample size 2^{31} was excluded, because at sample size 2^{31} it showed huge deviation due to memory limitation of the machine.

utilizes higher memory and drops down to normal value. The results in relation to sample size = 2^{30} showed that the extra memory requirement of two dimension vector was higher than that of the four-dimensional vector. Based upon that it can be predicted that BRA_Split_1_2DV needs an extra 3 GB (total 13 GB) for normal execution at sample size = 2^{31}, but the total memory barrier of 12 GB of the machine slows the process down. The most likely reason for this observation is the influence of memory pooling mechanism. When defining a memory structure it is possible to allocate the required amount of memory. This is known as memory pooling [23]. In all the memory structures used in algorithms discussed in this paper we used memory pooling. Memory pooling allocates the required memory at startup and divides this block into small chunks. If there is no memory pooling, memory will get fragmented. Accessing fragmented memory is inefficient. When the existing memory is not enough for allocating, then it switches to use fragmented memory for allocating memory elements. In the considered situation, the existing memory (12 GB) is not sufficient for allocating the required amount (13 GB) which switches to use fragmented memory.

The total cache size of the machine is 8.25 MB, which is less than the minimum memory utilization of considered algorithms (16 MB to 8 GB) in relation to the sample size range from 2^{22} to 2^{31}. Only the sample size 2^{21} occupies 8 MB memory, which is less than the total cache memory. Except BRA_Split_4_1V structure, all algorithms reported constant CPE in relation to the entire sample size range. The best algorithms of each category, especially, reported very steady behaviour. This observation is in disagreement with the statement of Karp “that a machine with hierarchical memory does not perform well when array size is larger than the cache size” [10].

Comparison (Figure 6) of best reported version in the considered domain (four memory structure version) and the selected algorithms shows that the array version of EBR performs the best. The four-array version of new BRA reported 1% lower performance than the array version of EBR. However, the four-array version of new BRA reported 34% and 23% higher performances than array versions of LBRA and BRA-Ru. Also, the four-vector version of new BRA is reported to have the best performance among all the vector versions. It reported 16%, 10%, and 22% performances compared to vector versions of LBRA, EBR, and BRA-Ru, respectively.

4. Conclusion and Outlook

The main finding of this paper is the recursive pattern of BPR and the implementation method of it using multiple memory structures. With multiple memory structures, especially, the newly identified index mapping performs much better than multidimensional or single one-dimensional memory structure. Furthermore, findings of this paper show that the performance of primitive array is higher than vector type. The result is in disagreement with the statement of Karp “that a machine with hierarchical memory does not perform well when array size is larger than the cache size.” Almost all the sample sizes we used were higher than the total cache size

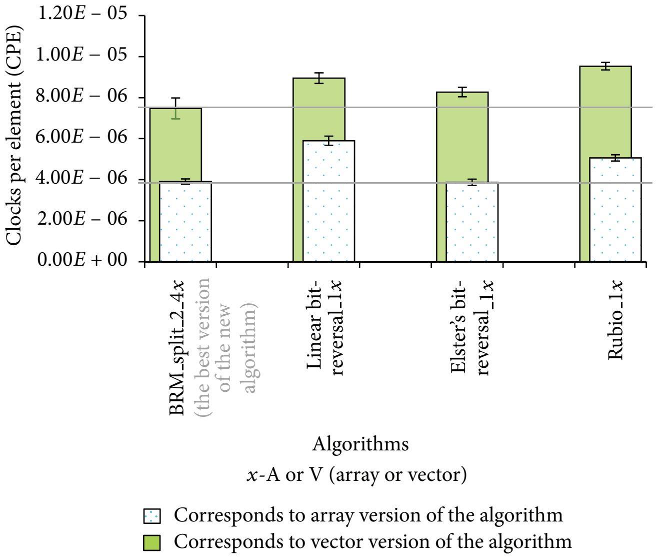


Figure 6: Clocks per element versus best version of new algorithm and selected algorithms, where blue dotted column corresponds to array version and green column corresponds to vector version of the algorithm.

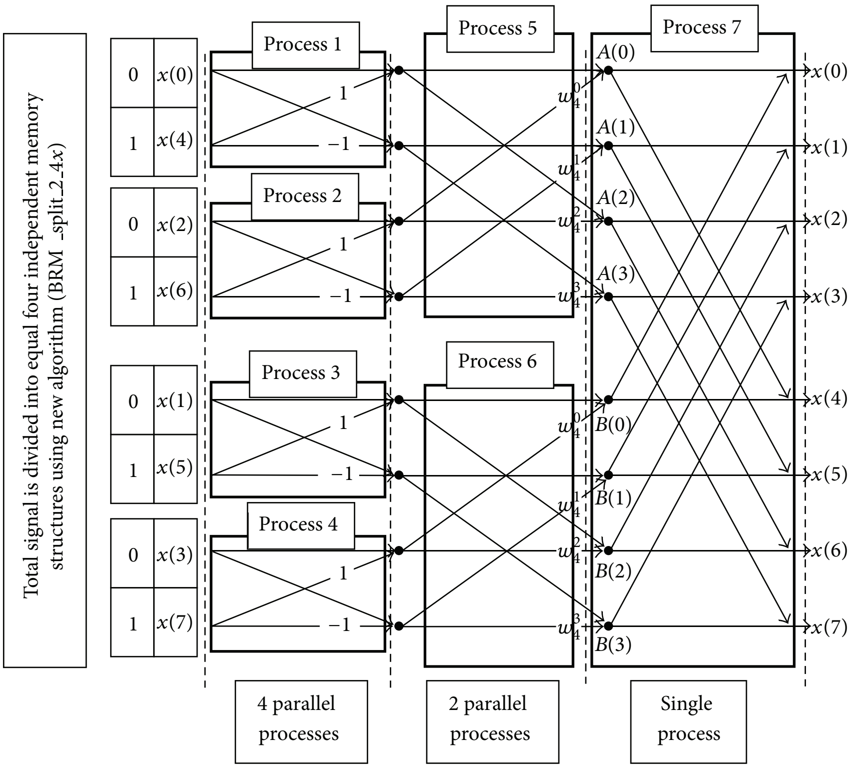


Figure 7: Four-memory-structure version of algorithm and the parallel processes for sample size = 16.

of the computer. However, multiple memory structure and the multidimensional memory structure versions showed reasonable steady performance with those samples. In general these results show the effects of data structures and memory allocation techniques and open a new window of creating efficient algorithms with multiple memory structures in many other fields where index mapping is involved.

The new bit reversal algorithm with 2^s independent memory structures splits the total signal into s independent portions and the total FFT process into $s + 1$ levels. Then, these s signal portions can be processed independently by means of s independent processes on the first level. On the next level the results from the previous level stored in independent memory structures can be processed with $s/2$

processes and so on, until the last level. Therefore, we suggest using the concept of multiple memory structures in total FFT process along with the new algorithm with multiple memory structures and suitable parallel processing technique. We expect that it is possible to achieve higher performance from FFT process with proper combination of parallel processing technique and new algorithm compared to using the new algorithm only to create bit reversal permutation. Figure 7 shows such approach with four (when $s = 2$) independent memory structures for sample size = 16.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This publication is financially supported by the University Grants Commission, Sri Lanka.

References

[1] C. S. Burrus, "Unscrambling for fast DFT algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1086–1087, 1988.

[2] A. C. Elster and J. C. Meyer, "A super-efficient adaptable bit-reversal algorithm for multithreaded architectures," in *Proceedings of the IEEE International Symposium on Parallel & Distributed Processing (IPDPS '09)*, vol. 1–5, pp. 1–8, Rome, Italy, May 2009.

[3] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[4] M. L. Massar, R. Bhagavatula, M. Fickus, and J. Kovačević, "Local histograms and image occlusion models," *Applied and Computational Harmonic Analysis*, vol. 34, no. 3, pp. 469–487, 2013.

[5] R. G. Lyons, *Understanding Digital Signal Processing*, Prentice Hall PTR, 2004.

[6] C. Deyun, L. Zhiqiang, G. Ming, W. Lili, and Y. Xiaoyang, "A superresolution image reconstruction algorithm based on landweber in electrical capacitance tomography," *Mathematical Problems in Engineering*, vol. 2013, Article ID 128172, 8 pages, 2013.

[7] Q. Yang and D. An, "EMD and wavelet transform based fault diagnosis for wind turbine gear box," *Advances in Mechanical Engineering*, vol. 2013, Article ID 212836, 9 pages, 2013.

[8] C. S. Burrus, *Fast Fourier Transforms*, Rice University, Houston, Tex, USA, 2008.

[9] C. van Loan, *Computational Frameworks for the Fast Fourier Transform*, vol. 10 of *Frontiers in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 1992.

[10] A. H. Karp, "Bit reversal on uniprocessors," *SIAM Review*, vol. 38, no. 1, pp. 1–26, 1996.

[11] A. C. Elster, "Fast bit-reversal algorithms," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 1099–1102, IEEE Press, Glasgow, UK, May 1989.

[12] M. Rubio, P. Gómez, and K. Drouiche, "A new superfast bit reversal algorithm," *International Journal of Adaptive Control and Signal Processing*, vol. 16, no. 10, pp. 703–707, 2002.

[13] B. Stroustrup, *Programming: Principles and Practice Using C++*, Addison-Wesley, 2009.

[14] B. Stroustrup, "Software development for infrastructure," *IEEE Computer Society*, vol. 45, no. 1, Article ID 6081841, pp. 47–58, 2012.

[15] B. Stroustrup, *The C++ Programming Language*, AT&T Labs, 3rd edition, 1997.

[16] S. Donovan, *C++ by Example, "UnderC" Learning Edition*, QUE Corporation, 2002.

[17] S. Mcconnell, *Code Complete*, Microsoft Press, 2nd edition, 2004.

[18] Microsoft, *STL Containers*, Microsoft, New York, NY, USA, 2012, http://msdn.microsoft.com/en-us/library/1fe2x6kt.aspx.

[19] Microsoft, *Arrays (C++ Component Extensions)*, Microsoft, New York, NY, USA, 2012, http://msdn.microsoft.com/en-us/library/vstudio/ts4c4dw6(v=vs.110).aspx.

[20] B. Stroustrup, "Evolving a language in and for the real world: C++ 1991–2006," in *Proceedings of the 3rd ACM SIGPLAN History of Programming Languages Conference (HOPL-III '07)*, June 2007.

[21] Microsoft, *Memory Limits for Windows Releases*, Microsoft, New York, NY, USA, 2012, http://msdn.microsoft.com/en-us/library/aa366778%28VS.85%29.aspx#memory_limits.

[22] A. Fog, "Optimizing software in C++," 2014, http://www.agner.org/optimize/.

[23] Code Project, "C++ Memory Pool," 2014, http://www.codeproject.com/Articles/15527/C-Memory-Pool.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 569580, 8 pages
http://dx.doi.org/10.1155/2014/569580

Research Article

A Multiobjective Optimization Algorithm Based on Discrete Bacterial Colony Chemotaxis

Zhigang Lu, Tao Feng, and Zhaozheng Liu

Key Lab of Power Electronics for Energy Conservation and Motor Drive of Hebei Province, Yanshan University, Qinhuangdao, Hebei 066004, China

Correspondence should be addressed to Zhigang Lu; zhglu@ysu.edu.cn

Received 18 February 2014; Accepted 26 June 2014; Published 14 July 2014

Academic Editor: Yang Xu

Bacterial colony chemotaxis algorithm was originally developed for optimal problem with continuous space. In this paper the discrete bacterial colony chemotaxis (DBCC) algorithm is developed to solve multiobjective optimization problems. The basic DBCC algorithm has the disadvantage of being trapped into the local minimum. Therefore, some improvements are adopted in the new algorithm, such as adding chaos transfer mechanism when the bacterium choose their next locations and the crowding distance operation to maintain the population diversity in the Pareto Front. The definition of chaos transfer mechanism is used to retain the elite solution produced during the operation, and the definition of crowding distance is used to guide the bacteria for determinate variation, thus enabling the algorithm obtain well-distributed solution in the Pareto optimal set. The convergence properties of the DBCC strategy are tested on some test functions. At last, some numerical results are given to demonstrate the effectiveness of the results obtained by the new algorithm.

1. Introduction

In the field of optimization, many researchers have been inspired by biological processes such as evolution [1, 2] or the food-searching behavior of ants [3] to develop new optimal methods such as evolutionary algorithms or ant codes. These techniques have been found to perform better than the classical heuristic or gradient-based optimization methods, especially when faced with the problem of optimizing multimodal, nondifferentiable, or discontinuous functions. Examples of applying these biologically inspired strategies in the field of engineering range from aerodynamic design [4] to job-shop scheduling problems [5]. Another biologically inspired optimization method is the chemotaxis algorithm, pioneered by Bremermann [6] and his coworkers [7, 8], that is proposed by analogy to the way bacteria react to chemoattractants in concentration gradients. This algorithm was analyzed for chemotaxis in a three-dimensional (3D) gradient by Bremermann [6] and employed for the training of neural networks [7, 8]. This strategy is evaluated on a number of test functions for local and global optimization, compared with other optimization techniques, and applied to the problem of inverse airfoil design [9]. A similar algorithm is the guided accelerated random search technique [10], which was applied to optimize parameters in flight control systems, [11] and to optimize perceptrons [12]. The BCC algorithm is a new colony intelligence optimization algorithm which was introduced in [13]. This novel algorithm considers not only the chemotactical strategy but also the communication between the colony members, and the performance has improved greatly.

Most real-world optimization problems require making decisions involving two or more goals. When these goals are the minimization or maximization of the objective functions they are typically referred to as multiobjective optimization (MO). From the 1950s, a variety of methods known as classical have been developed for the solution of multiobjective optimal problems (MOP). These methods are based on formal logic or mathematical programming [14]. Another interesting biological process that has been already implemented as an optimization technique is the BCC algorithm. This novel technique exposed the potential of implementing bacterial chemotaxis as a distributed optimization process, recognizing that in natural colonies, it is the interaction and communication between bacteria the mechanism that enables them to develop biologically advantageous patterns. Many real-world binary-discrete optimization problems require making decisions involving two or more goals that typically are in contradiction with each other. So in this paper the DBCC algorithm is developed to solve MOP. The basic DBCC algorithm has the disadvantage of being trapped into the local minimum and also cannot maintain the population diversity in the Pareto optimal set. Therefore, some improvements are adopted in the new algorithm, as adding arithmetic operator when the bacteria choose their next locations and taking into account the chaos transfer mechanism.

This paper is organized as follows. Section 2 describes the basic theorem of MOP. The model is shown in Section 3. The convergence properties of the developed DBCC strategy which is applied on test functions are given by comparing with other algorithms in Section 4. Finally, some conclusions are drawn in Section 5.

2. Preparative Theorem of MOP

A MOP [15] is defined as the problem of finding a vector of decision variables that satisfies some restrictions and optimizes a vector function, whose elements represent the values of the functions. A MOP may be formulated as follows [16]:

$$\begin{aligned} &\text{Min:} && f_m(x) \quad m = 1, 2, \ldots, M \\ &\text{Subject to:} && g_j(x) \geq 0 \quad j = 1, 2, \ldots, J \\ &&& h_k(x) = 0 \quad k = 1, 2, \ldots, K \\ &&& x_i \in \{0, 1\} \quad i = 1, 2, \ldots, n, \end{aligned} \tag{1}$$

where x is the vector of discrete decision variables $x = (x_1, x_2, \ldots, x_n)^T$ and $f_m(x)$ are the m objective functions. The J inequalities g_j and the K equalities h_k are known as constraint functions.

For MOP, instead of a single optimal, there is a set of optimal solutions known as Pareto optimal front (POF). Any solution of this set represents a balance between the objective functions; therefore, it is not possible to say that there is other solution in the search space which is superior to this one when all objectives are considered. In the minimization MOP, Pareto optimality can be mathematically defined as follows [17].

Definition 1. Pareto dominate: given two candidate solutions x and y from S, where S is the feasible region, the vector $f(x)$ is said to dominate vector $f(y)$ (denoted by $\prec y$) if and only if,

$$\begin{aligned} f_m(x) &\leq f_m(y), \quad \forall m \in \{1, 2, \ldots, M\}, \\ f_m(x) &< f_m(y), \quad \exists m \in \{1, 2, \ldots, M\}. \end{aligned} \tag{2}$$

Denoted by $x \neg \prec y$ if and only if,

$$\begin{aligned} f_m(x) &\leq f_m(y), \quad \exists m \in \{1, 2, \ldots, M\}, \\ f_j(x) &\geq f_j(y), \quad \exists j \in \{1, 2, \ldots, M\}. \end{aligned} \tag{3}$$

Definition 2. Pareto optimal: the candidate solution $x^* \in S$ is Pareto optimal if and only if,

$$x \prec x^*, \quad \neg \exists x \in S. \tag{4}$$

Definition 3. Pareto optimal set

$$F = \{f(x) \mid y \prec x, \neg \exists y \in S\}. \tag{5}$$

The set of solutions that satisfy all constraints is known as the Pareto optimal set and the fitness values corresponding to these solutions form the Pareto front or trade-off surface in objective space. In most cases it is not easy to find analytical expressions for the line or curve that contains the POF; thus, commonly optimal solutions points and the objective functions values in them are calculated. In order to find optimal solutions, there are two goals that any multiobjective optimization algorithm (MOA) seeks to achieve [18]:

(1) guide the search towards the global Pareto optimal region,

(2) maintain the population diversity in the Pareto optimal front.

3. Multiobjective Optimal Algorithm Based on DBCC (MOADBCC)

In this section, we present further improvements to the algorithm. MOADBCC can be summarized in the following steps:

The strategy parameters T_0, b, and τ_c are relevant to the calculation precision ε

$$\begin{aligned} T_0 &= \varepsilon^{0.30} \cdot 10^{-1.73}, \\ b &= T_0 \cdot \left(T_0^{-1.54} \cdot 10^{0.60}\right), \\ \tau_c &= \left(\frac{b}{T_0}\right)^{0.30} \cdot 10^{1.16}. \end{aligned} \tag{6}$$

Step 1. Initialize variables.

Generate N bacteria randomly by using binary-discrete code and make it turn to real code. Compute the velocity v. In this model, the velocity is assumed to be a scalar constant value $v = 1$. Evaluate the fitness of the bacteria and store the nondominated solutions in the Pareto optimal set.

Step 2. Individual bacterial optimization and find the new position $\vec{x}_1$.

(a) Compute the duration of the trajectory τ from the distribution of a random variable with an exponential probability density function

$$P(X = \tau) = \frac{1}{T} e^{-\tau/T}. \tag{7}$$

The time T is given by

$$T = \begin{cases} T_0 & \text{for } \vec{x}_{\text{pre}} \prec \vec{x}_{\text{cur}} \\ T_0 \left(1 + b * \min\left\{\left|\dfrac{f_{m\,\text{pr}}}{l_{\text{pr}}}\right|\right\}\right) & \text{for } \vec{x}_{\text{pre}} \neg \prec \vec{x}_{\text{cur}}, \end{cases} \tag{8}$$

where T_0 is minimal mean time, $\vec{x}_{\text{pre}}$ and $\vec{x}_{\text{cur}}$ are the position of a bacterium in the previous and current step, respectively, $f_{m\,\text{pr}}$ $m = 1, 2, \ldots, M$ is the difference between the actual and the previous function value, l_{pr} is vector connecting the previous and the actual position in the parameter space, and b is dimensionless parameter.

(b) The position and motion of a bacterium are defined by x, with a radius γ and $n-1$ angles $\Phi = \{\phi_1, \phi_2, \ldots, \phi_{n-1}\}$, $x_1 = \gamma \prod_{s=1}^{n-1} \cos(\phi_s)$, $x_i = \gamma \sin(\phi_{i-1}) \prod_{s=i}^{n-1} \cos(\phi_s)$, $i = 2, 3, \ldots, n$, $x_n = \gamma \sin(\cos(\phi_{n-1}))$.

The angle ϕ_i between the previous and the new direction obeys Gaussian distribution, for turning left or right, respectively,

$$P(X_i = \phi_i, \nu_i = \mu_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left[-\frac{(\phi_i - \nu_i)^2}{2\sigma_i^2}\right],$$
$$P(X_i = \phi_i, \nu_i = -\mu_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left[-\frac{(\phi_i - \nu_i)^2}{2\sigma_i^2}\right], \tag{9}$$

where $\phi_i \in [0^0, 180^0]$, the expectation value $\mu = E(X_i)$, and the variance $\sigma_i = \sqrt{\text{VAR}(X_i)}$

$$\mu_i = 62^\circ (1 - \cos(\theta)),$$
$$\sigma_i = 26^\circ (1 - \cos(\theta)) \tag{10}$$

with

$$\cos(\theta) = \begin{cases} 0 & \text{for } \vec{x}_{\text{pre}} \prec \vec{x}_{\text{cur}} \\ \exp\left(-\dfrac{\tau_c}{\tau_{\text{pr}}}\right) & \text{for } \vec{x}_{\text{pre}} \neg \prec \vec{x}_{\text{cur}}, \end{cases} \tag{11}$$

where τ_{pr} is the duration of the previous step.

(c) Compute the position $\vec{x}_1$ of the bacterium. The length of the path l is given by

$$l = v\tau. \tag{12}$$

The normalized new direction vector $\vec{n}_u$ with $|\vec{n}_u| = 1$ is multiplied by l to obtain the displacement vector $\vec{x}_1$

$$\vec{x} = \vec{n}_u l \tag{13}$$

such that the new location of the bacterium is

$$\vec{x}_1 = \vec{x}_{\text{pre}} + \vec{n}_u l. \tag{14}$$

Step 3. Optimize by the bacterial colony and find the new position $\vec{x}_2$.

The individual bacterial acquires information about its environment; compute the best neighbor center $\vec{x}_i$. We can get the new position by

$$\vec{x}_2 = \vec{x}_{\text{pre}} + \vec{l}, \tag{15}$$

where $\vec{l} = R(0,2) \cdot \text{dis}(\vec{x}_i, \text{Center}(\vec{x}_i))$, where $R(0,2)$ is the random number within the interval $(0,2)$ and $\text{dis}(x, y)$ is the distance connecting the vector x and y.

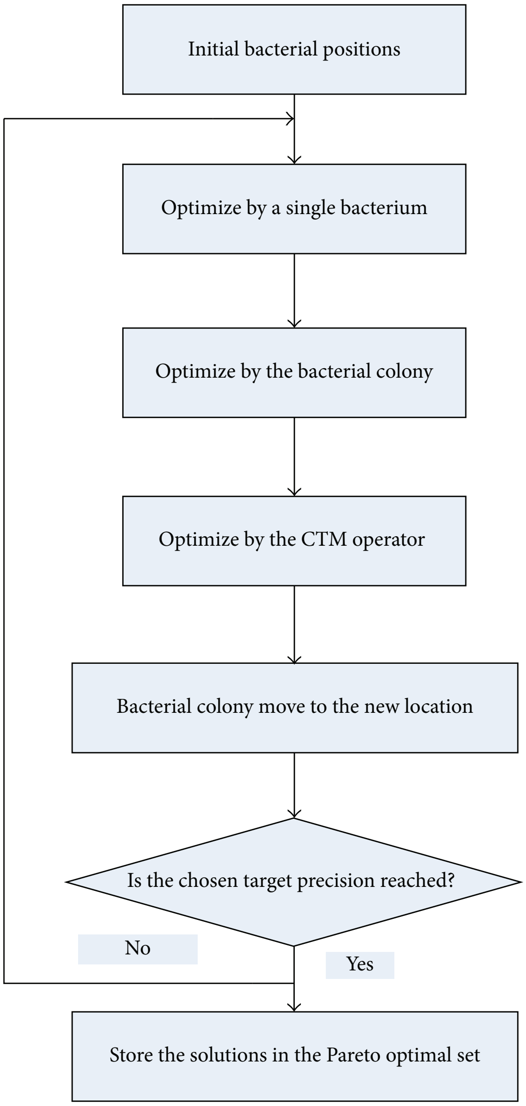

Initial bacterial positions
Optimize by a single bacterium
Optimize by the bacterial colony
Optimize by the CTM operator
Bacterial colony move to the new location
Is the chosen target precision reached?
No
Yes
Store the solutions in the Pareto optimal set

Figure 1: Flowchart of the MOADBCC algorithm.

Step 4. Optimize by the crowding distance operator and find the new position $\vec{x}_3$.

The new position of the individual bacterial with crowding distance operator is given by

$$\vec{x}_3 = \vec{x}_{\text{pre}} + F \cdot \vec{x}_{R\text{pr}}, \tag{16}$$

where $F \in [-1.2, 1.2]$ is assumed to be a real number and $\vec{x}_{R\text{pr}}$ is the biggest crowding distance position of bacterium in the previous step. The count of $\vec{x}_3$ is given no more than $N * 5\% + 2$.

Step 5. The reference colony $\vec{x}_{\text{ref}}$ is produced and bacteria move to the new location [19].

Step 6. If the chosen target precision ε_{end} is not reached, then go to Step 2; otherwise end.

Table 1: Test functions.

Problems	n	Constraint	Objective functions
SCH	1	$[-10^3, 10^3]$	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$
DEB	2	$[0, 1]$	$f_1(x) = x_1$ $f_2 = (1+10x_2) \times \left[1 - \left(\frac{x_1}{(1+10x_2)}\right)^2 - x_1 \times \frac{\sin(8\pi x_1)}{(1+10x_2)}\right]$
ZDT1	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \left(\frac{x_1}{g(x)}\right)^{0.5}\right]$ $g(x) = 1 + \frac{9(\sum_{i=2}^{n} x_i)}{(n-1)}$
ZDT2	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \left(\frac{x_1}{g(x)}\right)^{2}\right]$ $g(x) = 1 + \frac{9(\sum_{i=2}^{n} x_i)}{(n-1)}$
SRN	2	$[-20, 20]$	$f_1(x) = (x_1-2)^2 + (x_2-1)^2 + 2$ $f_2(x) = 9x_1 - (x_2-1)^2$ $g_1(x) = x_1^2 + x_2^2 - 225 \leq 0$ $g_2(x) = x_1 - 3x_2 + 10 \leq 0$

Table 2: Comparison table of CM.

Standard test function	SCH	ZDT1	ZDT2
MODBCC (with operators)			
Mean	0.003100	0.000909	0.000816
Stdev	0.000178	0.000458	0.005400
MODBCC (without operators)			
Mean	0.003300	0.003400	0.005600
Stdev	0.000258	0.005400	0.007100
NSGA-II			
Mean	0.003200	0.104700	0.187300
Stdev	0.000150	0.197700	0.229800

In practical optimization, the calculation precision ε can be adjusted adaptively with $\varepsilon_{\text{next}} = \varepsilon_{\text{begin}}/\alpha$, where $\alpha > 1$ is the precision update constant.

As the initial precision is reached, the parameters are adapted to another precision (defined by the number of parameter changes) and the search continues, until this new precision is reached. A given precision ε_{end} is reached if the difference between function values found by the bacterium is smaller than ε_{end}

$$\left|f_{m\,\text{pr}}\right| < \varepsilon_{\text{end}} \quad m = 1, 2, \ldots, M \tag{17}$$

for a given number n_{end}.

When the

$$\left|f_{m\,\text{pr}}\right| < \varepsilon_{i} \quad m = 1, 2, \ldots, M, \tag{18}$$

the difference between the actual and the previous function value is reached in a row n_i that has assumed before. The bacterial colony has to move to other positions and neglects information that the bacterial colony has held in previous step. This has been called bacterial migration. We modify the algorithm with chaos transfer mechanism (CTM).

Therefore, one has the following:

$$X_{n+1} = \mu X_n (1 - X_n). \tag{19}$$

So the algorithm can escape from the local optimal region.

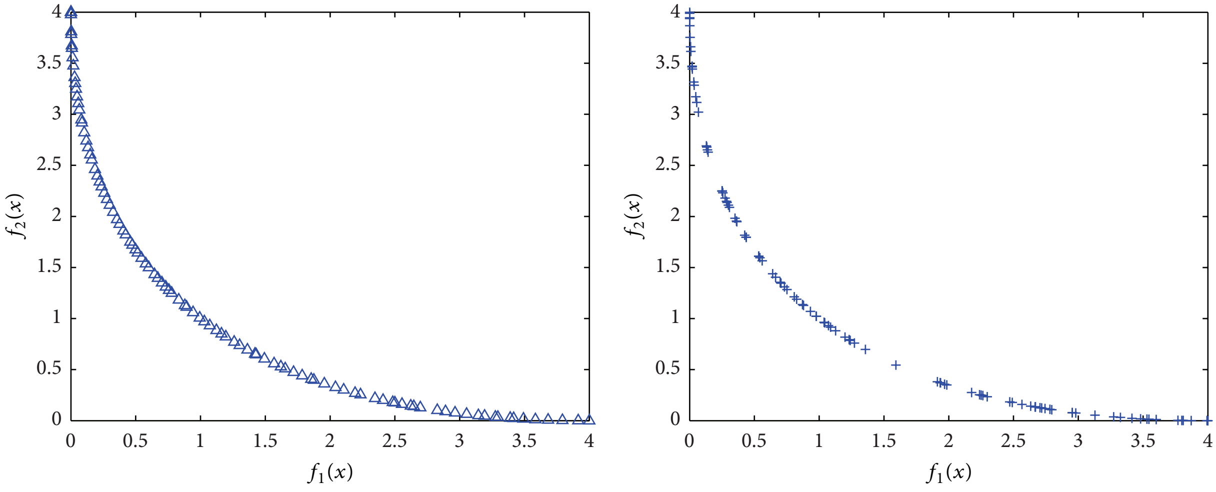
In the other side, sometimes the bacterium jumps out of the area of the global optimal and is never able to get back. To prevent the bacteria strategy from leaving the global optimal regions, the algorithm will search each of the bacterial migration positions, compare the function values with that in the POF, and then take the better one into the POF. Flowchart of the MOADBCC algorithm is given as shown in Figure 1.

4. Tests Results of the MOADBCC

MOADBCC with operators, MOADBCC without operators, and NSGA-II are, respectively, applied on simulation, among which NSGA-II applies binary coding to obtain Pareto

Table 3: Comparison table of RVI.

Standard test function	SCH	ZDT1	ZDT2
MODBCC (with operators)			
Mean	1.000000	0.970000	0.965000
Stdev	0.007200	0.012900	0.011100
MODBCC (without operators)			
Mean	0.990000	0.720000	0.665000
Stdev	0.029100	0.043600	0.056900
NSGA-II			
Mean	1.000000	1.000000	1.000000
Stdev	0.000000	0.000000	0.122100



(a)

(b)

Figure 2: The result of MOADBCC with and without operators on SCH.

optimal solution set. The parameter selection of MOADBCC algorithm can refer to [6]. The initial precision is set as 2.0 with a precision update constant $\alpha = 1.05$. The cross probability of NSGA-II is 0.9, and the variation probability is 0.01. The colony scales of all these three algorithms are 100 and the test function iterates 100 times except for the ZDT series function which iterates 200 times. Test condition was AMD Athlon(tm) IIX2 255 Processor, 2 GB Memory, Microsoft Windows XP operation system; the procedure was tested in Matlab 7.1.

This paper applies five test functions to examine the performance of the algorithm, including the research achievement SCH of Schaffer [20], DErB, and ZDT1 and ZDT2 [21–23] of Deb and SRN [24, 25] of test function Srinivas with constraint condition,. Binary coding and discretization treatment are applied by test function while the optimization is conducted in the discrete area. Among them, ZDT1 and ZDT2 and DEB apply 10 coding and SRN and SCH apply 20 coding. Table 1 shows test functions.

In order to evaluate the comprehensive performance of the algorithm, CM, and RVI performance evaluation indexes are applied in this paper to measure the advantages of the algorithm performance.

(1) Convergence index (CM) [26] is used to measure the intimacy level between the solutions obtained by the algorithm and real solution set. The smaller the CM index value is, the more intimate the obtained solutions are with real POF

$$\mathrm{CM} = \frac{\sqrt{\sum_{i=1}^{N} d_i^2}}{n}. \tag{20}$$

(2) Effective solution proportion index (RVI) refers to the proportion of nondominant solutions in all solutions, which is used to examine the convergence degree of final solutions. The bigger the RVI index value is, the better performance the algorithm is

$$\mathrm{RVI} = \frac{V}{N}, \tag{21}$$

where V means that the nondominant solutions minus the bacteria amount in same location.

The algorithm operates for 30 times independently for each index to obtain their mean value Mean and standard deviation Stdev. Tables 2 to 3 show the mean and Stdev of CM and RVI index for MOADBCC with operators, MOADBCC without operators, and NSGA-II respectively.

From the table of comparisons, the results show that all indexes of MOADBCC with operators are better than that of MOADBCC without operators. The performance of

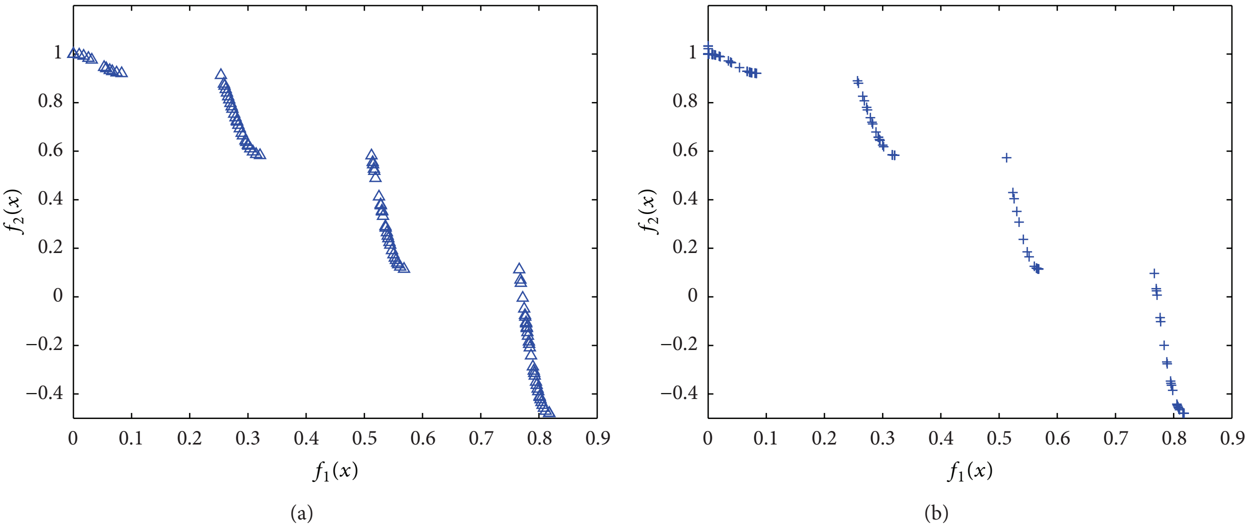


FIGURE 3: The result of MOADBCC with and without operators on DEB.

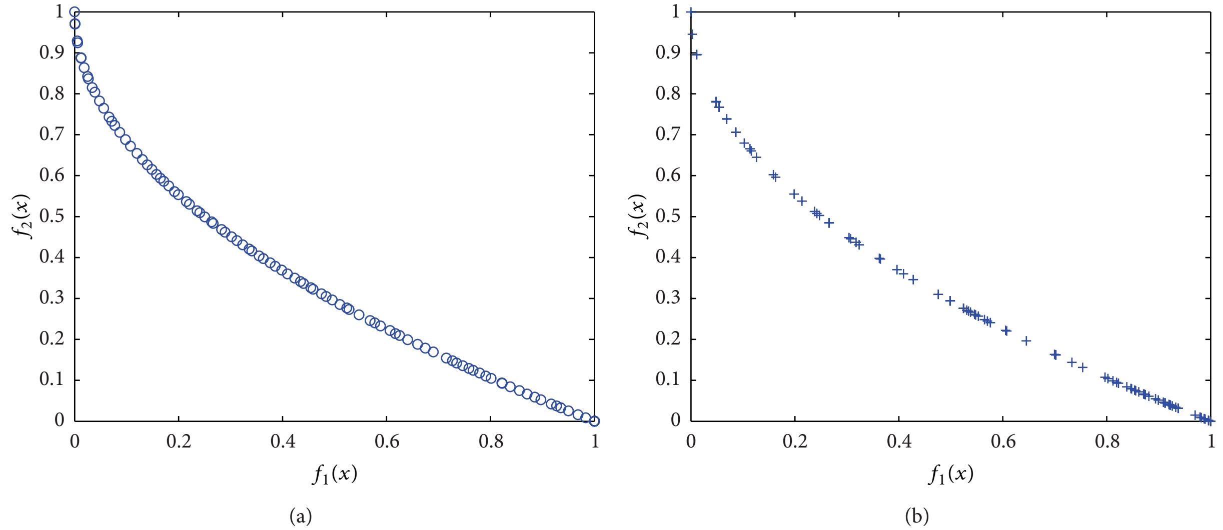


FIGURE 4: The result of MOADBCC with and without operators on ZDT1.

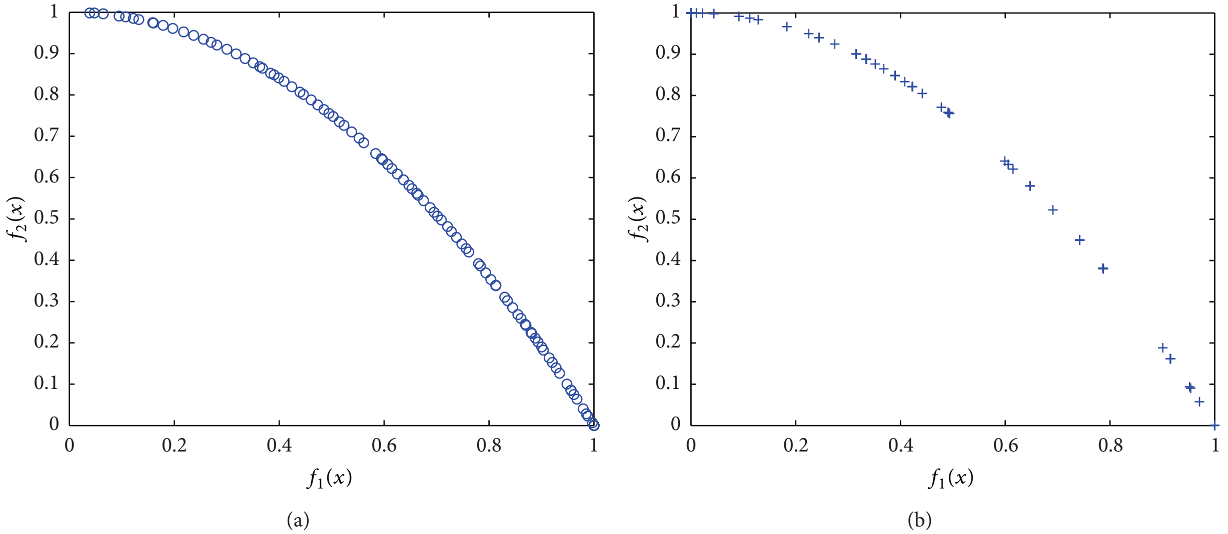


FIGURE 5: The result of MOADBCC with and without operators on ZDT2.

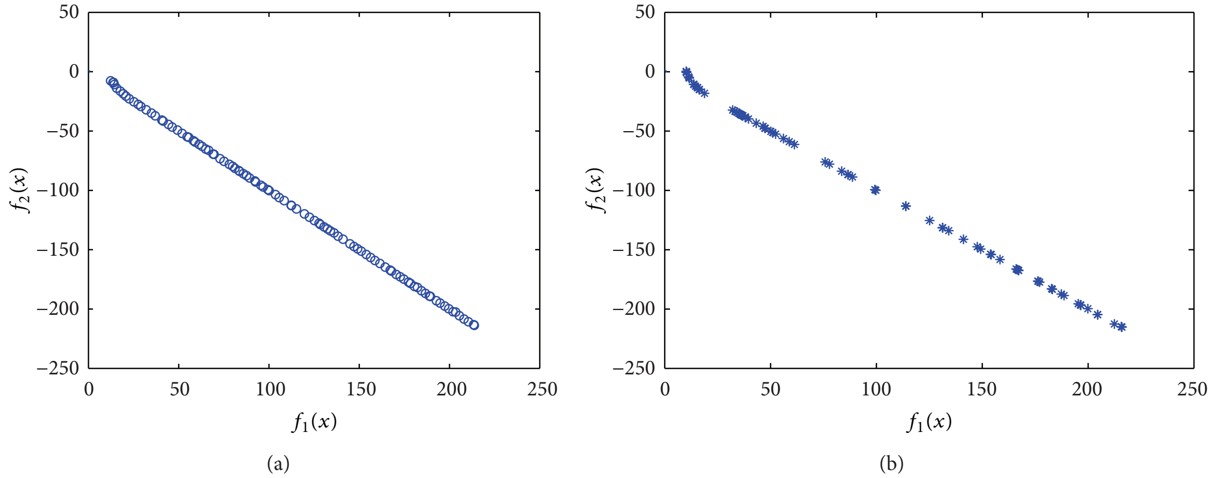


Figure 6: The result of MOADBCC with and without operators on SRN.

MODBCC and NSGA-II is equal to each other in terms of RVI index. However, the performance of MOADBCC is better than NSGA-II in terms of CM index, indicating it has better performance than NSGA-II algorithm. It is due to the fact that the operators proposed by this paper make it easier for bacteria to jump out of local convergence and accelerate the convergence speed of the algorithm, thus enabling it to obtain more effective solutions with same iterations and increasing the value of RVI.

In order to further compare the performances among these three algorithms, this paper listed the Figures of simulation results of three algorithms corresponding to SCH, DEB, and FON test functions, shown in Figures 2, 3, 4, 5, 6, and 7. From the simulation Figures, those results show that MOADBCC with operators is better than the other two algorithms in view of test problems in the Pareto Front, as it can converge to the Pareto Front more effectively and obtain more well-distributed solutions.

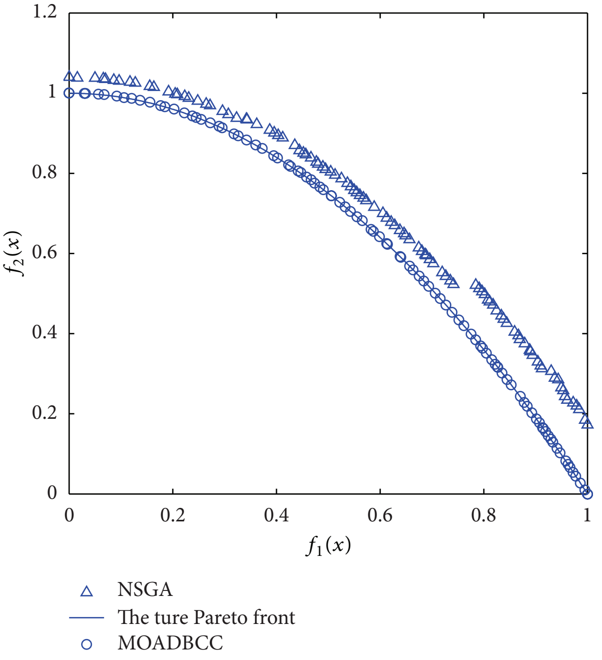


Figure 7: The result of MOADBCC and NSGA-II on ZDT2.

5. Conclusions

In this paper, a discrete bacterial colony chemotaxis (DBCC) algorithm is developed by chaos transfer mechanism and crowding distance strategy and used to solve multiobjective optimization problems. An additional term containing the chaos transfer mechanism draws the search away from local optima. Incorporation of crowding distance strategy also improves performance. The performance on tests is given to demonstrate the validity of the proposed algorithm. In fact, the results given in this paper not only show the effectiveness of the DBCC for the test functions, but also generally support the use of DBCC for other practical problems. Hence, there are future works including parameter settings and more applications of the proposed algorithmic framework to other optimization problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. Project 61374098 was supported by National Natural Science Foundation of China.

References

[1] H. P. Schwefel, *Evolution and Optimum Seeking*, Wiley, New York, NY, USA, 1995.

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[3] M. Dorigo, *Learning and natural algorithms [Ph.D. dissertation]*, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy, 1992.

[4] S. Obayashi, "Pareto genetic algorithm for aerodynamic design using the Navier-Stokes equations," in *Genetic Algorithms in Engineering and Computer Science*, Wiley, New York, NY, USA, 1997.

[5] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian, "Ant system for job-shop scheduling," *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 34, no. 1, pp. 39–53, 1994.

[6] H. J. Bremermann, "Chemotaxis and optimization," *Journal of the Franklin Institute*, vol. 297, no. 5, pp. 397–404, 1974.

[7] H. J. Bremermann and R. W. Anderson, "How the brain adjusts synapses-maybe," in *Automated Reasoning: Essays in Honor of Woody Bledsoe*, R. S. Boyer, Ed., pp. 119–147, Kluwer, Norwell, Mass, USA, 1991.

[8] R. W. Anderson, "Biased random-walk learning: a neurobiological correlate to trial-and-error," in *Neural Networks and Pattern Recognition*, O. M. Omidvar and J. Dayhoff, Eds., pp. 221–244, Academic Press, New York, NY, USA, 1998.

[9] S. Müller, S. Airaghi, and J. Marchetto, "Optimization based on bacterial chemotaxis," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 16–29, 2002.

[10] R. L. Barron, "Self-organizing and learning control systems," in *Proceedings of the Cybernetic Problems in Bionics. Bionics Symposium*, pp. 147–203, Gordon and Breach, New York, NY, USA, May 1966.

[11] "Neuromine nets as the basis for predictive component of robot brains," in *Proceedings of the 4th Annual Symposium American Society of Cybernetics—Cybernetics, Artificial Intelligence, and Ecology*, H. W. Robinson and D. E. Knight, Eds., pp. 159–193, Spartan, Washington, DC, USA, 1972.

[12] A. N. Mucciardi, "Adaptive flight control systems," in *Proceedings of the Principles and Practice of Bionics—NATO AGARD Bionics Symposium*, pp. 119–167, Brussels, Belgium, September 1968.

[13] W. W. Li, H. Wang, and Z. J. Zou, "Function optimization method based on bacterial colony chemotaxis," *Chinese Journal of Circuits and Systems*, vol. 10, no. 1, pp. 58–63, 2005 (Chinese).

[14] G. B. Dantzig and M. N. Thapa, *Linear Programming 1: Introduction*, Springer Series in Operations Research, Springer, New York, NY. USA, 1997.

[15] M. A. Guzmán, A. Delgado, and J. de Carvalho, "A novel multiobjective optimization algorithm based on bacterial chemotaxis," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 3, pp. 292–301, 2010.

[16] K. Deb, *Multi-objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester, UK, 2009.

[17] A. Chinchuluun and P. M. Pardalos, "A survey of recent developments in multiobjective optimization," *Annals of Operations Research*, vol. 154, pp. 29–50, 2007.

[18] K. Deb, "Evolutionary algorithms for multi-criterion optimization in engineering design," in *Proceedings of the Evolutionary Algorithms in Engineering and Computer Science (EUROGEN '99)*, K. Miettinen, M. Mäkelä, P. Neittaanmäki, and J. Périaux, Eds., pp. 135–161, Jyväskylä, Finland, 1999.

[19] C. Huilin, L. Zhigang, and S. Songqiang, "Multiobjective optimization using bacterial colony chemotaxis," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems*, 2011.

[20] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 93–100, 1985.

[21] K. Deb, "Multi-objective genetic algorithms: problem difficulties and construction of test problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.

[22] K. Deb, *Scalable Test Problems for Evolutionary Multi-Objective Optimization*, Springer, Berlin, Germany, 2005.

[23] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 253–274, 2003.

[24] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evolutionary Computation*, vol. 2, pp. 221–248, 1994.

[25] T. T. Binh and U. Korn, "MOBES: a multiobjective evolution strategy for constrained optimization problems," in *Proceedings of the IMACS World Congress on Scientific Computation*, pp. 357–362, Berlin, Germany, 1997.

[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 548957, 10 pages
http://dx.doi.org/10.1155/2014/548957

Research Article

Dealing with Nonregular Shapes Packing

Bonfim Amaro Júnior, Plácido Rogério Pinheiro,
Rommel Dias Saraiva, and Pedro Gabriel Calíope Dantas Pinheiro

Graduate Program in Applied Informatics, University of Fortaleza (UNIFOR), Avenue Washington Soares 1321,
Bl J Sl 30, 60811-905 Fortaleza, CE, Brazil

Correspondence should be addressed to Plácido Rogério Pinheiro; placidrp@uol.com.br

Received 11 April 2014; Accepted 7 June 2014; Published 8 July 2014

Academic Editor: Jer-Guang Hsieh

This paper addresses the irregular strip packing problem, a particular two-dimensional cutting and packing problem in which convex/nonconvex shapes (polygons) have to be packed onto a single rectangular object. We propose an approach that prescribes the integration of a metaheuristic engine (i.e., genetic algorithm) and a placement rule (i.e., greedy bottom-left). Moreover, a shrinking algorithm is encapsulated into the metaheuristic engine to improve good quality solutions. To accomplish this task, we propose a no-fit polygon based heuristic that shifts polygons closer to each other. Computational experiments performed on standard benchmark problems, as well as practical case studies developed in the ambit of a large textile industry, are also reported and discussed here in order to testify the potentialities of proposed approach.

1. Introduction

The constant competitiveness between the modern industries requires that a part of the investments has to be directed to the optimization of the production processes. In garment, glass, paper, sheet metal, textile, and wood industries, for instance, the main concern is to avoid the excessive expenditure of raw material required to meet a particular demand.

In this scenario, the two-dimensional irregular strip packing problem is included. Concisely, the irregular strip packing problem is a combinatorial optimization problem that consists of finding the most efficient design for packing irregular shaped items onto a single rectangular object with minimum waste material. More precisely, the problem can be defined as follows. Assume a rectangular object that has a constant width and infinite length. Consider also a collection of irregular items grouped in m types. For each piece type i, characterized by a set of points, there are an associated number of pieces b_i. The objective function of the problem aims to find an arrangement of items onto the rectangular object such that its length is minimized, and two geometric conditions hold. (1) No two pieces overlap with each other. (2) Each packed piece lies entirely onto the rectangular object.

A specialization of this problem is the placement of irregular figures with characteristics similar to regular cut, but dealing with irregular figures, the nesting problems [1]. They have been known as NP-hard due to their difficulty where few exact methods have been reported in the literature [2], where it is possible to find promising solutions by applying methodologies addressed in [3, 4].

The irregular strip packing problem is known to be NP-hard even without rotation [5], meaning that its globally optimal solution is unlikely to be found by polynomial-time algorithms. Solution techniques range from simple placement heuristics that convert a sequence of pieces into a feasible layout to local optimization techniques involving mathematical programming models. In this paper, we propose a novel approach based on the aggregation between a modified genetic algorithm and a greedy bottom-left procedure for tackling the target problem [6, 7]. A shrinking algorithm is also encapsulated into the metaheuristic engine in order to improve the arrangement of pieces.

To have better analysis, the remainder of this paper is organized as follows. In Section 2, we present state-of-the-art methodologies dedicated to the irregular strip packing problem. Section 3 conveys essential concepts related to the proposed methodology, which is introduced in Section 4. In Section 5, some control parameters are discussed. Computational results on benchmark problem instances and a case study on a textile industry are given in Section 6. Finally, in Section 7, we draw conclusions regarding the quality of the solutions provided by our algorithm and make some considerations concerning future development.

2. Literature Review

Problems involving irregular shapes comprise the most difficult class of packing problems. Whatever the constraints or secondary objectives, there are basically three approaches to find suitable layouts: (1) the polygons may be considered one at a time and packed onto the rectangular object according to the sorting criteria or (2) may be nested either singly or in groups into a set of enclosing polygons which are then packed onto the rectangular object, or (3) an initial allocation is improved iteratively.

Jakobs [9], for instance, presented a genetic algorithm in which each individual is represented by a list of pieces. The way they are disposed of in the chromosome defines the order they are placed by the Bottom-Left heuristic. A shrinking algorithm improves the partial solution by shifting the polygons closer to each other. On the other hand, a new method for implementing a bottom-left-fill packing algorithm which allows shapes that incorporate circular arcs and holes to be nested was presented by Burke and Kendall [10]. The placement rule is combined with hill climbing and tabu local search methods, which determine an efficient nesting sequence. Similarly, Oliveira et al. [11] developed the well-known constructive algorithm known as "*Técnicas de optimização para o posicionamento de figuras irregulares*" (TOPOS). The solution grows from a floating origin and both the next polygon to be packed and its position are defined by two heuristics called *local search* and *initial sort*. Different objective functions are proposed to evaluate and compare partial solutions.

In recent papers, mathematical programming techniques have been adopted for solving one of the following subproblems: *overlap minimization problem*, whose objective is to place all polygons onto a rectangular object with given width and length so that the total number of overlaps between polygons is made as small as possible; *compaction problem*, which requires a feasible design and relocates many polygons simultaneously so as to minimize the strip length; and *separation problem*, which takes an infeasible layout and performs a set of translations of the polygons that eliminates all overlaps and has a minimum total translation. Gomes and Oliveira [8], for instance, hybridized simulated annealing and linear programming. Firstly, the initial layout is obtained by a greedy bottom-left placement heuristic, being that each polygon is selected according to a *random weighted length* criterion. The simulated annealing algorithm guides the search over the solution space where each neighborhood structure handles linear programming models, which are a compaction algorithm (Figure 1) and a separation algorithm (Figure 2). Likewise, an extended local search algorithm based on nonlinear programming was conceived by Leung et al. [12]. The algorithm starts with a feasible layout, and its length is saved as the best length. Then, a new design is achieved by randomly swapping two polygons in the current solution. Within a time limit, the strip length is reduced, and local search method solves overlap minimization problems. If the new arrangement is feasible, the best solution is updated, and its length is further reduced to find even better solutions. Otherwise, the strip length is increased, and local search is invoked, which is conducted by a Tabu Search technique in order to escape from local minima. A compaction algorithm is used to improve results.

By other means, a successful approach that combines a local search method with a guided local search to deal with two- and three-dimensional irregular packing problems was proposed by Egeblad et al. [13]. An initial strip length is found by a fast placement heuristic. By reducing this value, overlap situations occur, which are removed by a local search that may apply one of the following four changes: horizontal translation; vertical translation; rotation; or flipping. The guided local search is selected to escape from local minima. Finally, in a recent paper [14], Bennell and Song modified the TOPOS placement heuristic and applied it to a beam search tree, which represents the placement order of polygons onto the rectangular object. Each node in the search tree corresponds to a partial solution, which means that partial solutions can be generated in parallel.

3. Related Concepts

To describe our proposed methodology, we first explain essential concepts related to its behavior, involving genetic algorithm, greedy bottom-left heuristic, and no-fit polygon.

3.1. Genetic Algorithm. Introduced by Holland [15] and perfected by Goldberg [16], *genetic algorithm* is a search and optimization technique inspired in the metaphor of biological evolution. Although a genetic algorithm is a heuristic method, which does not guarantee to find the optimal solutions, it can be applied in many optimization problems.

An implementation of a typical genetic algorithm begins with a population of (generally random) n chromosomes, which are evaluated and associated with a particular reproduction rate in such a way that those chromosomes that represent better solutions to the target problem are given more chances to "reproduce" than those which are poorer solutions. The quality of the solution is measured according to the current population. Then, the crossover and mutation mechanisms are implemented with some probability to the chosen chromosomes. Only the fittest individuals take part in the next generation. Once a new population is generated, the stopping condition of the metaheuristic is checked. If the genetic algorithm is not terminated, each candidate solution is again evaluated and the genetic search process renewed.

3.2. Greedy Bottom-Left Heuristic. A traditional constructive algorithm for solving any two-dimensional cutting or packing problem aims to order pieces and then place them in turn,

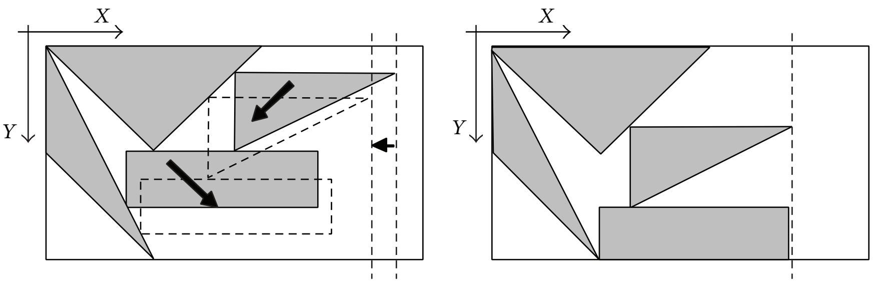


Figure 1: Compaction procedure (Gomes and Oliveira [8]).

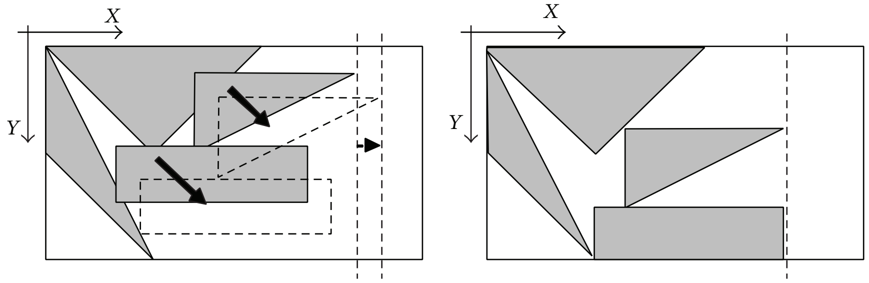


Figure 2: Separation procedure (Gomes and Oliveira [8]).

choosing the leftmost feasible position and breaking ties by selecting the lowest, as illustrated by Figure 3. This process, known as *greedy bottom-left heuristic*, was introduced by Baker et al. [17] for packing an arbitrary collection of rectangular pieces into a rectangular bin so as to minimize the layout height. The advantages of this type of approach are its speed and simplicity when compared with more sophisticated methods that may be able to produce solutions of higher quality.

Some papers have considered placement algorithms based on the greedy bottom-left rule in the field of two-dimensional cutting and in this work the aforementioned heuristic was chosen as placement policy.

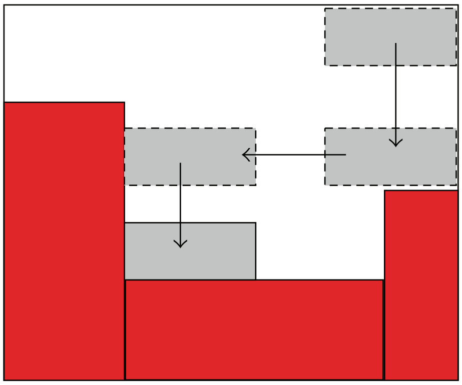


Figure 3: Greedy bottom-left procedure for an input piece.

3.3. No-Fit Polygon. The *no-fit polygon* is a powerful data structure used for fast and efficient handling of geometry in cutting and packing problems involving irregular shapes. The idea behind this trigonometric technique firstly described by Art [18] as *shape envelop* comes as follows. Given two polygons, A (the fixed piece) and B (the orbital piece), and a reference point on B called R_B, the no-fit polygon of A in relation to B, denoted by NFP_{AB}, is the set of points traced by R_B when B slides around the contour of A without overlapping, as displayed in Figure 4. Three situations may arise with respect to the interaction between both shapes. If polygon B is positioned so that its reference point is inside NFP_{AB}, then it overlaps with polygon A; if the reference point is on the boundary of NFP_{AB}, then polygon B touches polygon A; finally, if the reference point is outside of NFP_{AB}, then polygons A and B do not overlap or touch. So, the interior of the computed NFP_{AB} represents all intersecting positions of A and B, and the boundary represents all touching positions.

For our implementation, the construction of no-fit polygon was performed by using the Minkowski sum, whose concept involves two arbitrary point sets A and B. The *Minkowski sum* is obtained by adding each point in A to each point in B; that is, $A \oplus B = \{a + b : a \in A, b \in B\}$. Simple vector algebra can be used to show that $A \oplus -B$, defined as the *Minkowski difference* of A and B, is equivalent to no-fit polygon produced by both shapes. Since we follow the convention that polygons have counter-clockwise orientation, then $-B$ is simply B with clockwise orientation.

4. Proposed Methodology

The proposed methodology prescribes the integration of the distinct components described in Section 3. On what concerns the genetic algorithm, whose control parameters and calibration are set out in Section 5, each individual

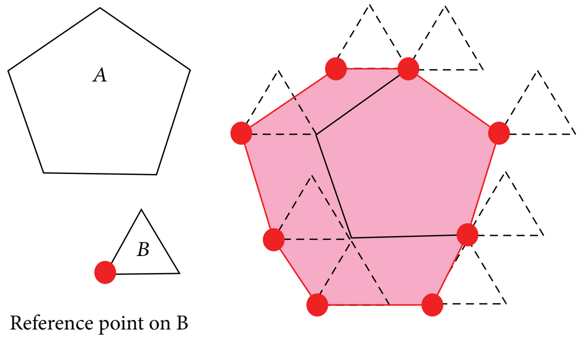


Figure 4: No-fit polygon generated by polygons A and B.

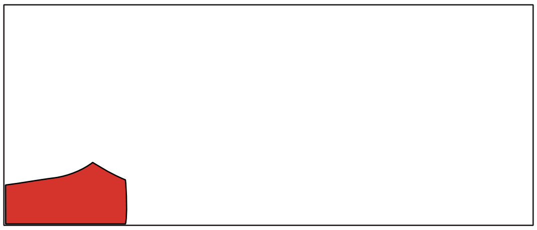


Figure 5: Allocation of the first piece.

is encoded by an integer chromosome that represents a placement vector $chrom(i)$, $(i = 1, \ldots, m)$, which determines the packing sequence of the m polygons onto the rectangular object. For each gene $chrom(i)$, characterized by a piece type t_i, there is an associated rotation variant r_i. The length required to pack all input polygons is assigned as fitness value of the corresponding individual. Having revealed these details, the steps executed by each individual of the current population are presented below.

Step 1. As depicted in Figure 5, polygon $chrom(i)$ is placed onto the rectangular object by the greedy bottom-left rule. Set $i \leftarrow i + 1$.

Step 2. According to the no-fit polygon technique, polygon $chrom(i)$ is positioned on the contour of polygon $chrom(i-1)$ such that the length of the rectangular object does not increase or is minimized, as presented in Figure 6. In the case of multiple positions, the leftmost one is considered.

Step 3. The greedy bottom-left rule is applied to polygon $chrom(i)$, as illustrated by Figure 7.

Step 4 (termination test). If $i \neq m$, set $i \leftarrow i + 1$ and go to Step 2;

Step 5 (final layout). A shrinking algorithm (Figure 8) is configured as a genetic algorithm operator, working in each generation on the two fittest individuals (i.e., layouts with short lengths) with an occurrence probability. Considering the no-fit polygon technique, this task is characterized by choosing polygons located in $P1$ (orbital piece) and $P2$ (stationary piece). After the $\text{NFP}_{P1,P2}$ is calculated, the aforementioned packing process is applied for both pieces.

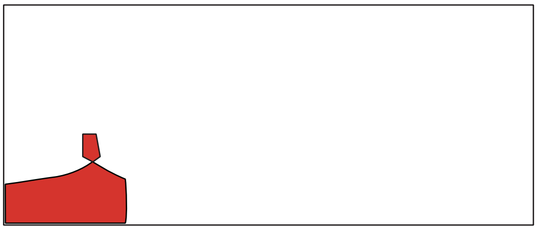


Figure 6: Best position computed by no-fit polygon.

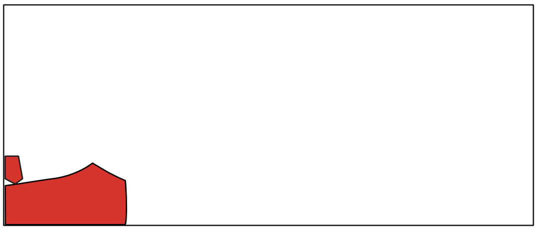


Figure 7: Final location of pieces.

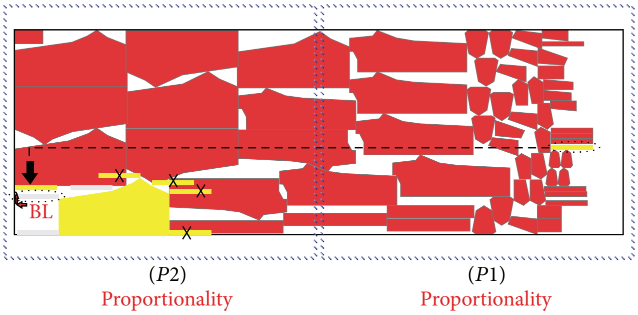


Figure 8: Shrinking heuristic.

The main objective of this method is to allocate polygons situated in rightmost spaces in leftmost free spaces in order to decrease the strip length. Furthermore, calculated NFP polygons between all the types of pieces reduce the number of operations during execution, considering that with Polygon processed NFP just a translation to the reference point of the stationary polygon is needed. The preprocessing of the NFP will only run once at the beginning of the algorithm and its complexity depends only on the quantity and types of possible angles.

The methodology flow chart is presented in Figure 9.

5. Control Parameters and Calibration

In this section, we discuss the parameters that control many aspects of the proposed methodology; some of these parameters represent tradeoffs between opposite goals, while others simply allow the fine-tuning of the effectiveness and/or efficiency of the methodology for particular problem instances. However, the set of control parameters discussed here allows the fine tuning of the methodology for a large range of optimization problems. We will see that several parameters were actually set to constant values during the

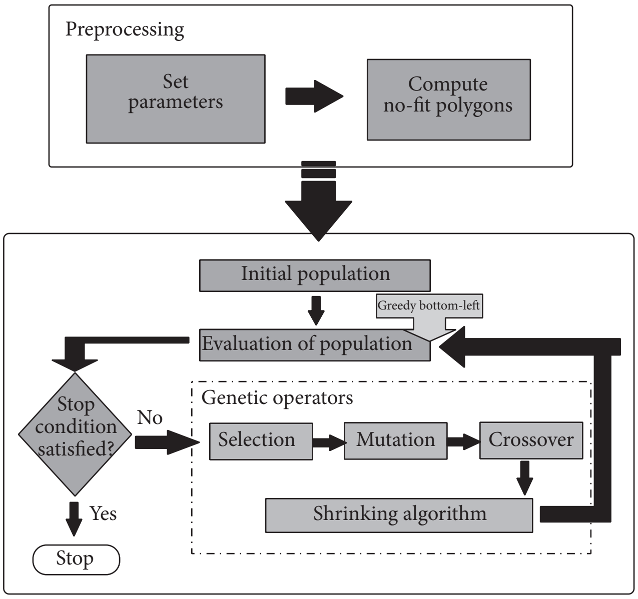


Figure 9: Methodology flow chart.

course of our computational experiments, suggesting that certain parameter-value pairs tend to work well on a wide variety of problem instances. On what concerns the metaheuristic engine, for the sake of computational performance, a simple genetic algorithm instance has been used, which is configured with uniform order-based crossover, swap mutation, fitness proportional parent selection, and age-based population replacement (whereby the parents are replaced by the entire set of offspring). Important control parameters related are described as follows.

(i) Elitism. This parameter indicates whether the best current individual is preserved from one generation to the next. Since elitism usually has a positive impact on the search process (as it allows better exploitation of the best individuals generated so far), in our experiments, we enabled this parameter, obtaining gains in convergence without any side effects.

(ii) Population Size. A large population size allows for more diversity and, thus, facilitates the exploration of several regions of the search space. On the other hand, a large population size implies slow evolutionary cycles. The best value obtained for this parameter was 400.

(iii) Crossover Rate. This parameter determines how often the crossover operator will be applied. If there is no crossover, offspring will be exact copies of parents. If there is a crossover, offspring is generated from parts randomly selected from their parents. To achieve the results reported in Section 6, we adopted the value of 0.90 for this parameter.

(iv) Mutation Rate. This parameter controls how often a gene of an individual has its value altered through mutation. While small values for this parameter tend to reduce the genetic algorithm exploration ability, larger values may cause the premature loss of good individuals from one generation to the next. In our experiments, we have selected the value of 0.2.

(v) Shrinking Probability. This parameter controls how often the shrinking method is applied. In our experiments, we have used the value of 0.6.

(vi) Total Time to Stop. This parameter (TTS, for short) stipulates the maximum amount of running time allowed for the genetic algorithm to execute. If its value is reached, the execution is interrupted and the best solution found so far is returned. The user according to his/her time availability should set this parameter.

(vii) Maximum Number of Generations. This is a secondary parameter used to cease the optimization process and is hereafter referred to as MNG. When the maximum number of generations is reached, the genetic algorithm process is interrupted and the best solution found so far is returned. If desired, this parameter can be set to an undetermined value in order to be disregarded.

6. Computational Experiments

A series of experiments were carried out on a desktop machine with a 3.60 GHz Intel i5 CPU and 4 GB of RAM. The genetic algorithm was implemented in Java and configured with those parameters described in Section 5. Concerning the MNG and TTS, they were set as 200 and 6 hours, respectively.

To evaluate the potentialities behind the proposed methodology, we conducted a series of experiments on benchmark problems available on the EURO Special Interest Group on Cutting and Packing (ESICUP) at http://www.fe.up.pt/esicup, which involved five data sets: DIGHE1 and DIGHE2 are jigsaw puzzles with known optimum conceived from Dighe and Jakiela [19]; JAKOBS1 and JAKOBS2 are artificial data sets proposed by Jakobs [9]; TROUSERS is an approximation of a real instance taken from the garment industry and it was firstly presented in Oliveira et al. [11]; SHAPES0 and SHAPES1 are artificially created data set, coordinates stated in Oliveira el al. [11]; ALBANO, MAO, and MARQUES are real instances from the textile industry and were presented in [20–22], respectively. Further descriptions of the instances are shown in Table 1.

Table 2 provides the best results achieved by 10 runs of our aggregation method (AM). In this table, for each data set, *Length*, *Utilization* and *Time* denote, respectively, the best length among those produced by all runs, the utilization percentage of the rectangular object of the best solution value, and the time elapsed from the beginning of the run until the best solution found.

In Table 3, we present a comparison of the best results achieved by some state-of-the-art methodologies to solve the irregular strip packing problem, where SAHA by Gomes and Oliveira [8], BLF by Burke and Kendall [10], 2DNest by Egeblad et al. [13], and BS by Bennell and Song [14] are included.

Table 1: Data set characteristics.

Data set	Number of different pieces	Total number of pieces	Orientations (degrees)
DIGHE1	16	16	0
DIGHE2	10	10	0
JAKOBS1	25	25	0
JAKOBS2	25	25	0
TROUSERS	17	64	0, 180
SHAPES0	4	43	0
SHAPES1	4	43	0, 180
MARQUES	8	24	90 incremental
MAO	9	20	90 incremental
ALBANO	8	24	0, 180
SWIM	10	48	0, 180

Table 2: Computational Results.

Data set	Length	Utilization (%)	Time (seconds)
DIGHE1	100.00	100.00	4113
DIGHE2	100.00	100.00	3751
JAKOBS1	12.22	84.46	13498
JAKOBS2	26.11	79.61	11985
TROUSERS	245.45	88.90	14775
SHAPES0	63.275	65.17	4303
SHAPES1	61.301	69.93	5032
MARQUES	81.89	87.80	6871
MAO	1840.37	82.69	6348
ALBANO	10247.28	88.40	10.230
SWIM	6099.00	72.63	15.204

We denote by Dif the relation, in percentage, between the solution found by AM and the best solution of the row:

$$\text{Dif} = \frac{\text{Solution found by AM } (\text{in percentage})}{\text{Best solution of the row } (\text{in percentage})} \times 100. \tag{1}$$

To sum up, regarding the utilization percentage of the rectangular object, it can be stated that AM has presented promising results. Taking as reference the scores achieved by BLF and SAHA, the proposed methodology presents better solutions in most of cases, as well as the average solutions. Regarding the 2DNest algorithm, we obtained better results in three problem instances, namely, ALBANO, DIGHE1, DIGHE2, and SWIM. Moreover, AM yielded better scores compared to the BS in two data sets, namely, SHAPES0 and ALBANO. Furthermore, even without applying the same rotation variants allowed by other studies (90 incremental), Table 3 shows that AM proved to be very competitive when compared with the reported approaches. We strongly believe that equivalent or better results can be found if we allow more rotations in a future extension of the work.

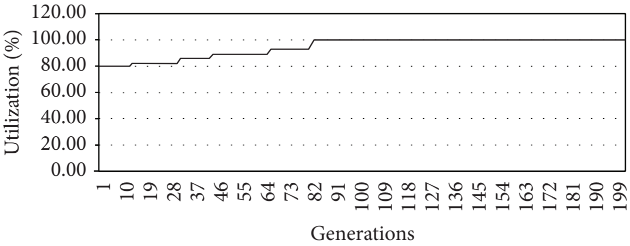


Figure 10: Dynamics of the genetic algorithm evolutionary process: a step-by-step improvement-DIGHE1.

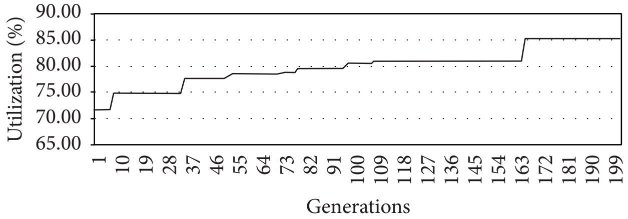


Figure 11: Dynamics of the genetic algorithm evolutionary process: a step-by-step improvement-ALBANO.

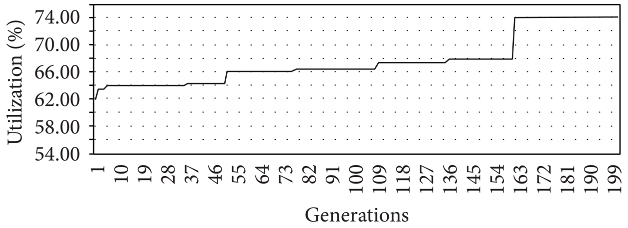


Figure 12: Dynamics of the genetic algorithm evolutionary process: a step-by-step improvement-JAKOBS2.

In Figures 10, 11, and 12, we show the convergence to the best solution found during the search of the genetic algorithm for DIGHE1, ALBANO, and JAKOBS2 data sets, while Figures 13, 14, and 15 display their best cutting configurations.

6.1. Applied Large Textile Industry. Brazil is the sixth largest textile producer country of the world and the main power, according to the Brasilian Textile Industry Association (ABIT). Among the Brazilian poles, we can highlight the state of Ceara. The success of cotton in this state, until the mid-1980s, stimulated the establishment of a solid textile and apparel park, which soon had to adapt to the new reality.

The aggregation method has also been applied in the ambit of a large textile industry. This particular industrial unit prints soccer team logos (Figure 16) on rectangular strips and then performs the cutting of these items for embroidering on caps, coats, shirts, shorts, and socks.

A significant difference in the production of shells is to define a feasible layout and has a prominent advantage. The acquisition of the points of the figures is performed by means of discretization, or some points are selected manually or automatically, in a manner that represent them.

The greater the number of points to describe the polygon is, the better the resolution is; however, all the processing routines consume high computational time. The width of a

Table 3: Comparative analysis.

Data set	SAHA	BLF	2DNest	BS	AM	Dif
DIGHE1	100.00	77.40	99.86	100.00	100.00	100
DIGHE2	100.00	79.40	99.95	100.00	100.00	100
JAKOBS1	78.89	82.60	89.07	85.96	84.46	95
JAKOBS2	77.28	74.80	80.41	80.40	79.61	99
TROUSERS	89.96	88.50	89.84	90.38	88.90	98
SHAPES0	66.50	60.50	67.09	64.35	65.17	97
SHAPES1	71.25	66.50	73.83	71.25	69.93	95
MARQUES	88.14	86.50	89.17	88.92	87.80	98
MAO	82.54	79.50	85.15	84.07	82.69	97
ALBANO	89.96	84.60	86.96	87.88	88.40	98
SWIM	74.36	71.60	71.53	75.04	72.63	96
Mean	83.53	77.45	84.81	84.39	83.60	98

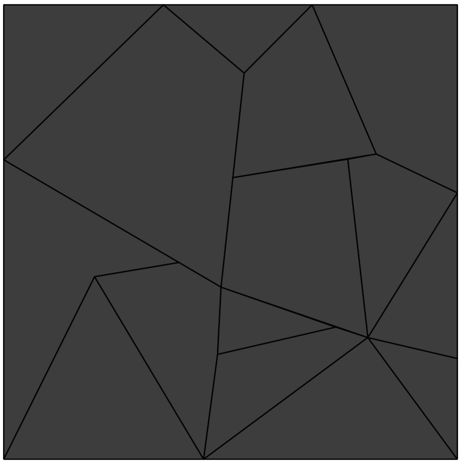


Figure 13: Best solution found for DIGHE1 instance.

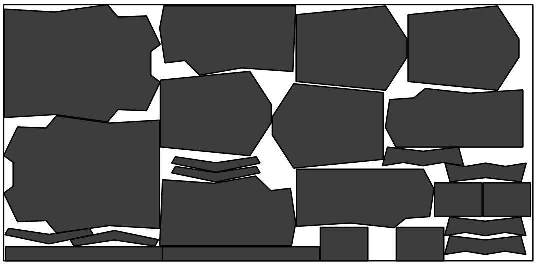


Figure 14: Best solution found for ALBANO instance.

rectangular strip is 210 mm and each logo is composed by a set of points (see input data for each instance at https://www.dropbox.com/sh/qzwnx03lwp8klbs/9ULx0YkFey).

Moreover, Table 4 contains the instance, the number of kinds, the number of polygons, and width to applications.

In Table 5, we present the results obtained by empirical methods (EM) in solving instances of the case study. We define "empirical" as the knowledge used by employees with extensive experience in textile cutting (without applying any combinatorial optimization technique).

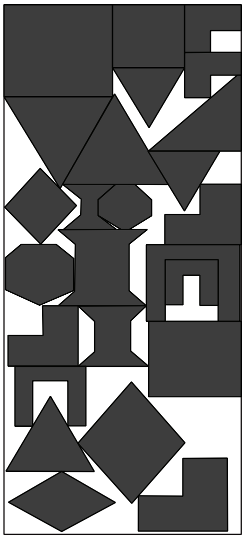


Figure 15: Best solution found for JAKOBS2 instance.

The layout produced by applying the empirical methods (EM) to Soccer Logos-I, Soccer Logos-II, and Soccer Logos-III instances is displayed in Figures 17, 18, and 19.

In addition, the layout produced by applying the aggregation method (AM) to Soccer Logos-I, Soccer Logos-II, and Soccer Logos-III instances is displayed in Figures 20, 21, and 22. Moreover, Table 6 shows the values obtained from the length, improvement, and execution time presented. Comparing Tables 5 and 6, there is a better use of the aggregation method (AM) over the empirical methods (EM).

The highlighted variation in a small increase in the number of generations from the compression method is applied to the methodology. Moreover, it is observed that the application of aggregation method for the experiments of the textile industry presented has made significant progress.

7. Conclusions and Future Work

The main objective is to minimize the length of the layouts, while the width remains fixed. For this, we developed a

Table 4: Features of the figures of the case study.

Data set	Number of different pieces	Total number of pieces	Orientations (degrees)	Length
Soccer Logos-I	2	40	45 incremental	210 mm
Soccer Logos-II	1	20	45 incremental	210 mm
Soccer Logos-III	1	30	45 incremental	210 mm

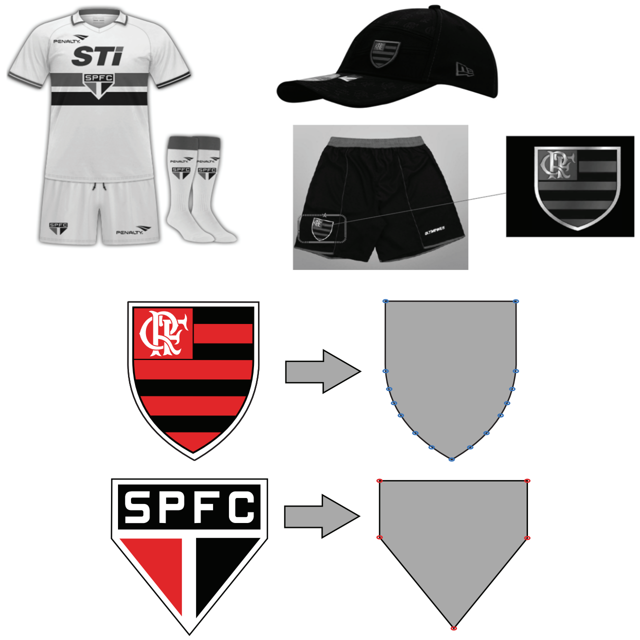


Figure 16: Soccer team logos.

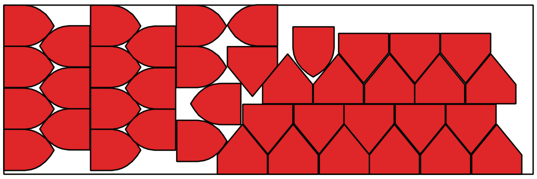


Figure 17: Best solution found for Soccer Logos-I instance.

Table 5: Results obtained from the practical case studied—empirical methods (EM).

Data set	Length	Improvement (%)	Time (seconds)
Soccer Logos-I	637.617	78.01	5100
Soccer Logos-II	307.474	79.21	1200
Soccer Logos-III	499.758	76.21	4800

strategy based on hybridization of genetic algorithms and a heuristic placement that applies the concepts of calculating the no-fit polygons methodology and bottom-left heuristic. In this initial study, we have introduced an aggregation methodology to cope with the irregular strip packing problem, which is based on a kind of hybridization between a genetic algorithm greedy bottom-left heuristic.

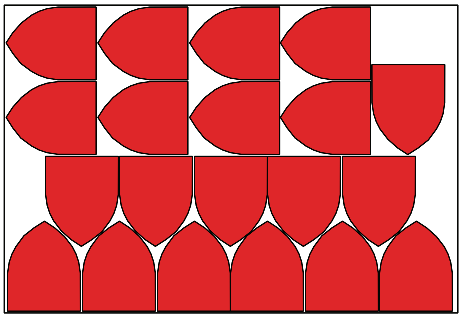


Figure 18: Best solution found for Soccer Logos-II instance.

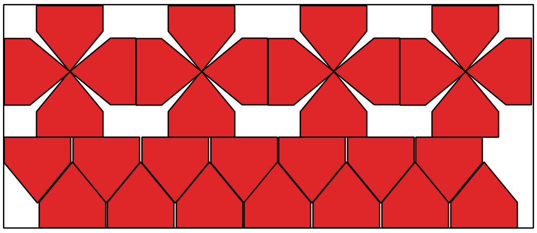


Figure 19: Best solution found for Soccer Logos-III instance.

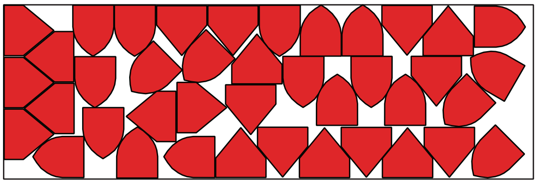


Figure 20: Best solution found for Soccer Logos-I instance: (AM).

Table 6: Results obtained from the practical case studied—aggregation method (AM).

Data set	Length	Improvement (%)	Time (seconds)
Soccer Logos-I	633.58	78.51	5220
Soccer Logos-II	300.95	80.94	3800
Soccer Logos-III	453.39	84.01	3690

For specific types of figures, certain approaches will produce best computational results. However, depending on the shape of polygons, the same may be a decrease in efficiency. The criteria for selection of various positioning method is an outstanding solution to the problem; however, the computational complexity involved in such methods is high. The computational results were good cheer, compared to other approaches, and especially according to the inherent difficulty of this problem.

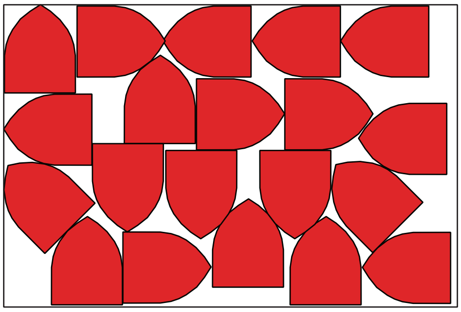


FIGURE 21: Best solution found for Soccer Logos-II instance: (AM).

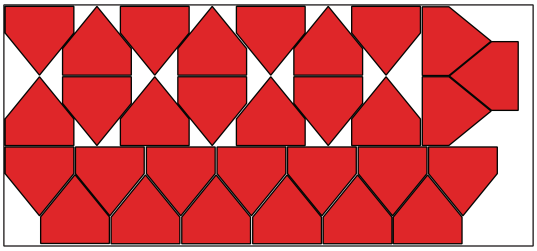


FIGURE 22: Best solution found for Soccer Logos-III instance: (AM).

Overall, the optimization performance achieved with the novel methodology has been promising, taking as reference the results achieved by other approaches and taking into account the inherent difficulties associated with this particular cutting and packing problem.

As future work, we plan to investigate the performance of the biased random-key genetic algorithm [23] with different criteria evaluations for the irregular strip packing problem, to apply processing in distributed genetic algorithm coupling observing criteria for each process, and to investigate other forms of representation of the polygon, so that irregular figures holes are characterized by applying the method of positioning. In addition, investigate other forms of representation of polygons, so that figures with irregular bores are characterized in the application of the method of placement. Another possibility is to investigate the application of Integer Linear Programming models for compacting layouts.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The first and third authors are thankful to Coordination for the Improvement of Higher Level or Education Personnel (CAPES) and the second and fourth authors are thankful to National Counsel of Technological and Scientific Development (CNPq) via Grants no. 475239/2012-1.

References

[1] G. Wäscher, H. Haußner, and H. Schumann, "An improved typology of cutting and packing problems," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1109–1130, 2007.

[2] F. M. B. Toledo, M. A. Carravilla, C. Ribeiro, J. F. Oliveira, and A. M. Gomes, "The dotted-board model: a new MIP model for nesting irregular shapes," *International Journal of Production Economics*, vol. 145, no. 2, pp. 478–487, 2013.

[3] P. R. Pinheiro and P. R. Oliveira, "A hybrid approach of bundle and Benders applied large mixed linear integer problem," *Journal of Applied Mathematics*, vol. 2013, Article ID 678783, 11 pages, 2013.

[4] P. R. Pinheiro, A. L. V. Coelho, A. B. de Aguiar, and T. O. Bonates, "On the concept of density control and its application to a hybrid optimization framework: investigation into cutting problems," *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 463–472, 2011.

[5] B. K. Nielsen and A. Odgaard, "Fast neighborhood search for the nesting problem," Tech. Rep. 03/03, Department of Computer Science, University of Copenhagen, 2003.

[6] B. A. Junior, P. R. Pinheiro, and R. D. Saraiva, "Tackling the irregular strip packing problem by hybridizing genetic algorithm and bottom-left heuristic," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, pp. 3012–3018, Cancun, Mexico, June 2013.

[7] B. Amaro Jr., P. R. Pinheiro, and R. D. Saraiva, "A hybrid methodology for tackling the irregular strip packing problem," in *Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS '13)*, 11, pp. 396–401, May 2013.

[8] A. M. Gomes and J. F. Oliveira, "Solving irregular strip packing problems by hybridising simulated annealing and linear programming," *European Journal of Operational Research*, vol. 171, no. 3, pp. 811–829, 2006.

[9] S. Jakobs, "On genetic algorithms for the packing of polygons," *European Journal of Operational Research*, vol. 88, no. 1, pp. 165–181, 1996.

[10] E. Burke and G. Kendall, "Applying evolutionary algorithms and the no fit polygon to the nesting problem," in *Proceedings of the International Conference on Artificial Intelligence*, vol. 1, pp. 51–57, 1999.

[11] J. F. Oliveira, A. M. Gomes, and J. S. Ferreira, "TOPOS: a new constructive algorithm for nesting problems," *OR Spektrum: Quantitative Approaches in Management*, vol. 22, no. 2, pp. 263–284, 2000.

[12] S. C. H. Leung, Y. Lin, and D. Zhang, "Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem," *Computers and Operations Research*, vol. 39, no. 3, pp. 678–686, 2012.

[13] J. Egeblad, B. K. Nielsen, and A. Odgaard, "Fast neighborhood search for two- and three-dimensional nesting problems," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1249–1266, 2007.

[14] J. A. Bennell and X. Song, "A beam search implementation for the irregular shape packing problem," *Journal of Heuristics*, vol. 16, no. 2, pp. 167–188, 2010.

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Oxford, UK, 1975.

[16] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[17] B. S. Baker, E. G. Coffman, and R. L. Rivest, "Orthogonal packings in two dimensions," *SIAM Journal on Computing*, vol. 9, no. 4, pp. 846–855, 1980.

[18] R. C. Art, "An approach to the two-dimensional irregular cutting stock problem," Tech. Rep. 36.008, IBM Cambridge Centre, 1966.

[19] R. Dighe and M. J. Jakiela, "Solving pattern nesting problems with genetic algorithms employing task decomposition and contact detection," *Evolutionary Computation*, vol. 3, no. 3, pp. 239–266, 1996.

[20] A. Albano and G. Sapuppo, "Optimal allocation of two-dimensional irregular shapes using heuristic search methods," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 10, no. 5, pp. 242–248, 1980.

[21] C. Bounsaythip and S. Maouche, "Irregular shape nesting and placing with evolutionary approach," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, pp. 3425–3430, October 1997.

[22] V. M. M. Marques, C. F. G. Bispo, and J. J. S. Sentieiro, "A system for the compactation of two-dimensional irregular shapes based on simulated annealing," in *Proceedings of the International Conference on Industrial Electronics, Control and Instrumentation (IECON '91)*, pp. 1911–1916, Kobe, Japan, November 1991.

[23] J. F. Gonçalves and M. G. C. Resende, "Biased random-key genetic algorithms for combinatorial optimization," *Journal of Heuristics*, vol. 17, no. 5, pp. 487–525, 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 507401, 8 pages
http://dx.doi.org/10.1155/2014/507401

Research Article

The Dual Triple I Methods of FMT and IFMT

Liu Yan[1] and Zheng Mucong[2]

[1] *College of Science, Xi'an University of Science and Technology, Xi'an 710054, China*
[2] *College of Mathematics and Information Science, Shaanxi Normal University, Xi'an 710062, China*

Correspondence should be addressed to Zheng Mucong; zhengmucong@gmail.com

Received 11 April 2014; Revised 5 June 2014; Accepted 11 June 2014; Published 7 July 2014

Academic Editor: Ker-Wei Yu

The Triple I method for the model of intuitionistic fuzzy modus tollens (IFMT) satisfies the local reductivity instead of the reductivity. In order to improve the quality of the Triple I method for lack of reductivity, the paper is intended to present a new approximate reasoning method for IFMT problem. First, the concept of intuitionistic fuzzy difference operator is proposed and its properties on the lattice structure of intuitionistic fuzzy sets are studied. Then, the dual Triple I method for FMT based on residual fuzzy difference operator is presented and the dual Triple I method is generated for IFMT. Moreover, a decomposition method of IFMT is provided. Furthermore, the reductivity of methods is investigated. Finally, α-dual Triple I method of IFMT is proposed.

1. Introduction

The real world is too complicated to be described precisely and it is full of uncertainty. It seems that humans have a remarkable capability to deal with the uncertain information. We need a theory to formulate human knowledge representation. The theory of fuzzy sets introduced by Zadeh [1] has been found to be useful to deal with uncertainty, imprecision, and vagueness of information. It is well known that the fuzzy logic and approximate reasoning are significant parts of the theory of fuzzy sets. To provide foundations for approximate reasoning with fuzzy propositions, the basic models of deductive processes with fuzzy sets, which were called the fuzzy modus pones (FMP) and the fuzzy modus tollens (FMT), were proposed in the seminal paper of Zadeh [2]. The basic reasoning principle was the composition rule of inference (CRI). Based on the CRI method, fuzzy reasoning has been successfully applied to a wide variety of fields [3–6].

In [7], Wang pointed out that the composition seems to lack logic foundation. For trying to provide a logic foundation for fuzzy reasoning, Wang proposed the full implication Triple I method (Triple I method for short). Triple I method has attracted the attention of many scholars and many results have been reported [7–14]. Pei [10] comprehensively investigated the method based on a class residual fuzzy implication derived from left-continuous t-norms. Liu and Wang [8] obtained a class restriction Triple I solution for FMP and FMT. For the formalization of the Triple I fuzzy reasoning, Wang [13] and Pei [11] considered the propositional logic system and the first order logic system, respectively.

From a knowledge representation point of view, however, the role of fuzziness is not always to capture uncertainty [15, 16]. The intuitionistic fuzzy sets introduced by Atanassov [17] is a pair of fuzzy sets, namely, a membership and a nonmembership function, which represent positive and negative aspects of the given information. It constitutes an appropriate knowledge representation framework.

The intuitionistic fuzzy set theory has been widely applied in many fields such as pattern recognition, machine learning, decision making, market prediction, and image processing [18–23]. In order to construct the theoretical foundation of intuitionistic fuzzy reasoning, Cornelis et al. [24–26] have made fruitful pioneering work. They presented the intuitionistic fuzzy t-norms and t-conorms [25]. In [24, 26], they concentrated on the intuitionistic fuzzy implication operator theory, and the CRI method of the intuitionistic fuzzy reasoning is discussed in [24]. However, because restrictions on the intuitionistic fuzzy implications are much more complicated than that of fuzzy implication operator, the Triple I method of the intuitionistic fuzzy reasoning has not been paid enough attention to. Zheng et al. [27] presented the Triple I method of intuitionistic fuzzy reasoning, proved the reductivity of the Triple I method for IFMP, and showed that the Triple I method of IFMT satisfied the local reductivity instead of the reductivity. In order to improve the quality of the Triple I method for lack of reductivity, we intend to propose the dual Triple I method of approximate reasoning for IFMT. First of all, we introduce the concepts of the intuitionistic fuzzy difference operators and coadjoint pair and provide the unified form of the residual intuitionistic fuzzy difference operators adjoint to intuitionistic t-conorms derived from the left-continuous t-norms. Secondly, we present the dual Triple I method of fuzzy reasoning for FMT and give the equivalence conditions between dual Triple I method and Triple I method. Lastly, we propose the dual Triple I method and the decomposition method of approximate reasoning for IFMT and discuss the reductivity of methods. Furthermore, α-Triple I method of IFMT is proposed.

Throughout this paper, we denote $L = [0,1]$, $a \vee b = \max\{a,b\}$, $a \wedge b = \min\{a,b\}$, $a^c = 1 - a$, $a, b \in [0,1]$, and $L^* = \{(u,v) \in [0,1]^2 \mid u + v \leq 1\}$. I stands for any index set. We denote by FS$s(X)$, IFS$s(X)$ the set of all fuzzy sets in X and the set of all intuitionistic fuzzy sets, respectively.

2. Preliminaries

In this section, we recall some basic concepts and results, which we will need in the subsequent sections.

Definition 1 (see [28]). A triangular norm (t-norm for short) is a binary operation $\otimes$ on L satisfying commutativity, associativity, monotonicity, and boundary condition $a\otimes 1 = a$, $\forall a \in L$. A triangular conorm (t-conorm for short) is a binary operation $\oplus$ on L satisfying commutativity, associativity, monotonicity, and boundary condition $a \oplus 0 = a$, $\forall a \in L$.

The t-conorm $\oplus$ is called the dual t-conorm of the norm $\otimes$ if $a \oplus b = 1 - (1 - a) \otimes (1 - b)$, $\forall a, b \in L$ and, analogously, the t-norm $\otimes$ is called the dual t-norm of the conorm $\oplus$ if $a \otimes b = 1 - (1 - a) \oplus (1 - b)$, $\forall a, b \in L$.

Definition 2 (see [28]). A t-norm $\otimes$ is a left-continuous t-norm if for all $a_i, b \in L$ and $\otimes$ satisfies $(\vee_{i\in I} a_i)\otimes b = \vee_{i\in I}(a_i \otimes b)$. A t-conorm $\oplus$ is a right-continuous t-conorm if for all $a_i, b \in L$, $\oplus$ satisfies $(\wedge_{i\in I} a_i) \oplus b = \wedge_{i\in I}(a_i \oplus b)$.

Proposition 3 (see [28]). *A t-norm is left-continuous if and only if the dual i-conorm of the t-norm is right-continuous.*

Proposition 4 (see [29]). *If $\otimes$ is a left-continuous t-norm, then there exists a binary operation $\rightarrow$ (the $\otimes$-residuum) on L such that $(\otimes, \rightarrow)$ satisfy the residual principle; that is, $a \otimes b \leq c$ if and only if $a \leq b \rightarrow c$, where $\rightarrow$ is given by*

$$a \longrightarrow b = \vee \{c \in L \mid c \otimes a \leq b\} \tag{1}$$

and is called residual implication derived from $\otimes$.

Proposition 5 (see [30]). *If $\oplus$ is a right-continuous t-conorm then there exists a binary operation $\ominus$(the $\oplus$ -coresiduum) on L such that $(\oplus, \ominus)$ forms a coadjoint pair; that is, $a \leq b \oplus c$ if and only if $a \ominus c \leq b$, and $\ominus$ is given by*

$$a \ominus b = \wedge \{c \in L \mid a \leq c \oplus b\}. \tag{2}$$

The $\oplus$-coresiduum $\ominus$ is called fuzzy difference operator derived from $\oplus$.

Definition 6 (see [27]). $\rightarrow$, $\oplus$, and $\ominus$ are called associated operators of $\otimes$ if $(\otimes, \rightarrow)$ is a adjoint pair, $(\oplus, \ominus)$ is a coadjoint pair, and $\oplus$ is the dual t-conorm of the norm $\otimes$.

Proposition 7 (see [27]). *If $\rightarrow$, $\oplus$, and $\ominus$ are associated operators of $\otimes$, then $a \ominus b = 1 - (1 - b) \rightarrow (1 - a)$.*

Example 8. The following are four important t-norms. The first three are all continuous t-norms, but the last one is left-continuous:

(1) Gödel t-norm $a\otimes_{\mathrm{G}} b = a \wedge b$,

(2) Lukasiewicz t-norm $a\otimes_{\mathrm{Lu}} b = (a + b - 1) \vee 0$,

(3) product t-norm $a\otimes_{\pi} b = ab$,

(4) R_0 t-norm.

Consider

$$a\otimes_0 b = \begin{cases} 0, & a + b \leq 1; \\ a \wedge b, & a + b > 1. \end{cases} \tag{3}$$

The associated operators of the above four t-norms are as follows, respectively:

(1′)

$$\begin{gathered} a \longrightarrow_{\mathrm{G}} b = \begin{cases} 1, & a \leq b; \\ b, & a > b, \end{cases} \\ a\oplus_{\mathrm{G}} b = a \vee b, \\ b\ominus_{\mathrm{G}} a = \begin{cases} 0, & b \leq a; \\ b, & b > a, \end{cases} \end{gathered} \tag{4}$$

(2′)

$$\begin{gathered} a \longrightarrow_{\mathrm{Lu}} b = (1 - a + b) \wedge 1, \\ a\oplus_{\mathrm{Lu}} b = (a + b) \wedge 1, \\ b\ominus_{\mathrm{Lu}} a = (b - a) \vee 0, \end{gathered} \tag{5}$$

(3′)

$$\begin{gathered} a \longrightarrow_{\pi} b = \begin{cases} 1, & a \leq b; \\ \dfrac{b}{a}, & a > b, \end{cases} \\ a\oplus_{\pi} b = a + b - ab, \\ b\ominus_{\pi} a = \begin{cases} 0, & b \leq a; \\ \dfrac{b - a}{1 - a}, & b > a, \end{cases} \end{gathered} \tag{6}$$

(4′)

$$a \longrightarrow_0 b = \begin{cases} 1, & a \le b; \\ (1-a) \vee b, & a > b, \end{cases}$$

$$a \oplus_0 b = \begin{cases} 1, & (a+b) \ge 1; \\ a \vee b, & a+b < 1, \end{cases} \quad (7)$$

$$b \ominus_0 a = \begin{cases} 0, & b \le a; \\ b \wedge (1-a), & b > a. \end{cases}$$

Lemma 9. *If $a \le b$, $a, b \in L$, then $b \ominus_{Lu} (b \ominus_{Lu} a) = a$.*

Lemma 10. *If $1/2 \le a \le b$, $a, b \in L$, then $b \ominus_0 (b \ominus_0 a) = a$.*

Definition 11 (see [17]). An intuitionistic fuzzy set on the nonempty universe of discourse X is given by $A = \{\langle x, A_t(x), A_f(x)\rangle \mid x \in X\}$, where

$$A_t : X \longrightarrow [0,1], \qquad A_f : X \longrightarrow [0,1] \quad (8)$$

with the condition

$$0 \le A_t(x) + A_f(x) \le 1, \quad \forall x \in X. \quad (9)$$

$A_t(x)$ and $A_f(x)$ denote a membership function and a nonmembership function of x to A, respectively. It is clear that the intuitionistic fuzzy set A in X can be written as

$$A(x) = (u, v), \quad 0 \le u + v \le 1, \qquad u, v \in [0,1], \quad \forall x \in X. \quad (10)$$

As a generalization of fuzzy sets, intuitionistic fuzzy sets extend the character value from $[0,1]$ to the triangle domain $L^* = \{(u,v) \in [0,1]^2 \mid u + v \le 1\}$.

If $A_t(x) = 1 - A_f(x)$, $\forall x \in X$, then intuitionistic fuzzy sets degenerate into fuzzy sets. We denote by IFSs(X) the set of all intuitionistic fuzzy sets in X.

We can define a partial order on L^* as follows:

$$\alpha, \beta \in L^*, \quad \alpha = (a_1, a_2), \quad \beta = (b_1, b_2), \qquad \alpha \le \beta \quad \text{iff } a_1 \le b_1, \quad a_2 \ge b_2. \quad (11)$$

Obviously, $\alpha \wedge \beta = (a_1 \wedge b_1, a_2 \vee b_2)$, $\alpha \vee \beta = (a_1 \vee b_1, a_2 \wedge b_2)$, $0^* = (0,1)$ and $1^* = (1,0)$, are the smallest element and the greatest element of L^*, respectively. It is easy to verify the fact that $(L^*, \le)$ is a complete lattice.

Definition 12 (see [27]). $\otimes_{L^*}$ is called an intuitionistic t-norm derived from t-norm $\otimes$ if

$$\alpha \otimes_{L^*} \beta = (a_1 \otimes b_1, a_2 \oplus b_2) \quad (12)$$

and $\oplus_{L^*}$ is called an intuitionistic t-conorm derived from t-norm if

$$\alpha \oplus_{L^*} \beta = (a_1 \oplus b_1, a_2 \otimes b_2), \quad (13)$$

where $\oplus$ is the dual t-conorm of the t-norm $\otimes$.

Proposition 13 (see [27]). *$(L^*, \otimes_{L^*}, 1^*)$ is a commutative monoid and $\otimes_{L^*}$ is isotone; $(L^*, \oplus_{L^*}, 0^*)$ is a commutative monoid and $\oplus_{L^*}$ is isotone.*

Proposition 14 (see [27]). *Let $\otimes$ be a left-continuous t-norm; then*

(1) *$\otimes_{L^*}$ derived from $\otimes$ is a left-continuous intuitionistic t-norm on L^*; that is, $(\vee_{i \in I} \alpha_i) \otimes_{L^*} \gamma = \vee_{i \in I} (\alpha_i \otimes_{L^*} \gamma)$;*

(2) *$\oplus_{L^*}$ derived from $\otimes$ is a right-continuous intuitionistic t-conorm on L^*; that is, $(\wedge_{i \in I} \alpha_i) \oplus_{L^*} \gamma = \wedge_{i \in I} (\alpha_i \oplus_{L^*} \gamma)$.*

Theorem 15 (see [27]). *Let $\otimes_{L^*}$ be an intuitionistic t-norm derived from a left-continuous t-norm $\otimes$; then there exists a binary operation $\rightarrow_{L^*}$ on L^* such that*

$$\gamma \otimes_{L^*} \alpha \le \beta \quad iff \ \gamma \le \alpha \longrightarrow_{L^*} \beta \quad (14)$$

and $\rightarrow_{L^*}$ is given by

$$\alpha \rightarrow_{L^*} \beta = \vee \{\eta \in L^* \mid \eta \otimes_{L^*} \alpha \le \beta\}. \quad (15)$$

Definition 16 (see [27]). $(\otimes_{L^*}, \rightarrow_{L^*})$ is called an intuitionistic adjoint pair if $(\otimes_{L^*}, \rightarrow_{L^*})$ satisfy the residual principle (14), and $\rightarrow_{L^*}$ is called a residual intuitionistic implication derived from a left-continuous $\otimes$ if $\otimes_{L^*}$ is an intuitionistic t-norm derived from a left-continuous $\otimes$.

Theorem 17 (see [27]). *Let $\alpha, \beta \in L^*$, $\alpha = (a_1, a_2)$, $\beta = (b_1, b_2)$, and $\rightarrow_{L^*}$ be a residual intuitionistic implication derived from a left-continuous t-norm $\otimes$; then*

$$\alpha \longrightarrow_{L^*} \beta = ((a_1 \rightarrow b_1) \wedge (1 - (b_2 \ominus a_2)), b_2 \ominus a_2). \quad (16)$$

3. The Triple I Method of IFMT

As one of the basic inference models of fuzzy reasoning, FMT has the following form:

$$\begin{array}{llll} \text{Suppose that} & A(x) \longrightarrow & B(y) & \cdots \text{ major premise} \\ \text{and given} & & B^*(y) & \cdots \text{ minor premise} \\ \hline \text{caculate} & A^*(x) & & \cdots \quad \text{conclution} \end{array}, \quad (17)$$

where $A(x)$, $A^*(x)$ are the fuzzy sets on the nonempty universe of discourse X and $B(y)$, $B^*(y)$ are the fuzzy sets on the nonempty universe of discourse Y.

The Triple I principle is as follows:

$A^*(x)$ should be the biggest fuzzy set on X satisfying

$$(A(x) \longrightarrow B(y)) \longrightarrow (A^*(x) \longrightarrow B^*(y)) = 1. \quad (18)$$

Theorem 18 (see [10]). *Let $\rightarrow$ be a residual implication derived from a left-continuous t-norm; the expression of the Triple I solution A^* for FMT problem* (17) *is as follows:*

$$A^*(x) = \wedge_{y \in Y} \{(A(x) \longrightarrow B(y)) \longrightarrow B^*(y)\}. \quad (19)$$

Theorem 19 (see [10, 14]). *Let $\rightarrow$ be a residual implication derived from a left-continuous t-norm and satisfy contrapositive symmetry; that is, $a \rightarrow b = b^c \rightarrow a^c$; then the expression of the Triple I solution A^* for FMT problem* (17) *becomes*

$$A^*(x) = \wedge_{y \in Y} \{B^*(y) \oplus (A(x) \longrightarrow B(y))^c\}. \quad (20)$$

The Triple I method of approximate reasoning was extended from FMT to IFMT in [27]. IFMT has the same form as FMT as follows:

$$\begin{array}{llll} \text{Suppose that} & A(x) \longrightarrow_{L^*} & B(y) & \cdots \text{ major premise} \\ \text{and given} & & B^*(y) & \cdots \text{ minor premise} \\ \hline \text{caculate} & A^*(x) & & \cdots \text{ conclution} \end{array}, \tag{21}$$

where $A(x)$, $A^*(x)$ are the intuitionistic fuzzy sets on the nonempty universe of discourse X; $B(y)$, $B^*(y)$ are the intuitionistic fuzzy sets on the nonempty universe of discourse Y; and $\rightarrow_{L^*}$ is a residual intuitionistic fuzzy implication on L^*. We denote $A(x) = (A_t(x), A_f(x))$, $B(y) = (B_t(y), B_f(y))$, $B^*(y) = (B^*_t(y), B^*_f(y))$, $A_{-f}(x) = 1 - A_f(x)$, $B_{-f}(y) = 1 - B_f(y)$, and $B^*_{-f}(y) = 1 - B^*_f(y)$. Clearly, A_t, A_f, and A_{-f} are the fuzzy sets on X, respectively, and B_t, B_f, B^*_t, B^*_f, and B_{-f} are the fuzzy sets on Y, respectively.

Because $\rightarrow_{L^*}$ is the residual intuitionistic fuzzy implication on L^*, the extension of the Triple I principle is as follows:

$A^*(x)$ should be the biggest intuitionistic fuzzy set on X satisfying

$$(A(x) \longrightarrow B(y)) \longrightarrow (A^*(x) \longrightarrow B^*(y)) = 1^* \tag{22}$$

under the order of L^*.

Theorem 20 (see [27]). *Let the implication $\rightarrow_{L^*}$ in IFMT be the residual implication derived from a left-continuous t-norm $\otimes$; then the expression of the Triple I solution A^* for IFMT problem* (21) *is as follows:*

$$A^*(x) = \wedge_{y\in Y}\left\{(A(x) \longrightarrow_{L^*} B(y)) \longrightarrow_{L^*} B^*(y)\right\}, \quad x \in X. \tag{23}$$

We know that the Triple I method of FMT possesses virtue of reductivity if $B(y)$ satisfies the condition $\exists y_0 \in Y$ such that $B(y_0) = 0$ (see [10]). Unfortunately, the Triple I method of IFMT only possess the local reductivity instead of the reductivity (see [27]).

Theorem 21 (see [27]). *Let the implication $\rightarrow_{L^*}$ in IFMT be the residual implication derived from a left-continuous t-norm $\otimes$ satisfying $1 \ominus (1 \ominus a) = a$; then the Triple I method is local reductive; that is, $A^*_f = A_f$ whenever $B^* = B$ satisfying $\exists y_0 \in Y$ such that $B(y_0) = 0^*$.*

Corollary 22 (see [27]). *Let the implication $\rightarrow_{L^*}$ in IFMT be the residual implication derived from Lukasiewicz t-norm or R_0 t-norm; then the Triple I method is local reductive; that is, $A^*_f = A_f$ whenever $B^* = B$ satisfying $\exists y_0 \in Y$, $B(y_0) = 0^*$.*

Theorem 23 (see [27]). *Let the implication $\rightarrow_{L^*}$ in IFMT be the residual implication derived from Lukasiewicz t-norm; then the Triple I method is local reductive, that is, $A^*_f = A_f$ whenever $B^*_f = B_f$ satisfying $\forall x \in X$, $\exists y_0 \in Y$ such that $A_f(x) \le B_f(y_0)$.*

Theorem 24 (see [27]). *Let the implication $\rightarrow_{L^*}$ in IFMT be the residual implication derived from R_0 t-norm; then the Triple I method is local reductive; that is, $A^*_f = A_f$ whenever $B^*_f = B_f$ satisfying $\forall x \in X$, $\exists y_0 \in Y$ such that $1/2 \le A_f(x) \le B_f(y_0)$.*

4. Intuitionistic Fuzzy Difference Operator

In this section, we give the unified form of the adjoint operator for the intuitionistic t-conorm derived from a left-continuous t-norm.

Theorem 25. *Let $\oplus_{L^*}$ be an intuitionistic t-conorm derived from a left-continuous t-norm $\otimes$; then there exists a binary operation $\ominus_{L^*}$ on L^* such that*

$$\alpha \le \gamma \oplus_{L^*} \beta \quad iff \;\; \alpha \ominus_{L^*} \beta \le \gamma, \tag{24}$$

and $\ominus_{L^*}$ is given by

$$\alpha \ominus_{L^*} \beta = \wedge\left\{\eta \in L^* \mid \alpha \le \eta \oplus_{L^*} \beta\right\}. \tag{25}$$

Proof. By (25), if $\alpha \le \gamma\oplus_{L^*}\beta$, then $\alpha\ominus_{L^*}\beta \le \gamma$. Conversely, if $\alpha\ominus_{L^*}\beta \le \gamma$, then $\wedge\{\eta \mid \alpha \le \eta\oplus_{L^*}\beta\} \le \gamma$. From the monotonicity of $\oplus_{L^*}$, $(\wedge\{\eta \mid \alpha \le \eta\oplus_{L^*}\beta\})\oplus_{L^*}\beta \le \gamma\oplus_{L^*}\beta$. According to the left-continuity of $\oplus_{L^*}$, $\wedge\{\eta\oplus_{L^*}\beta \mid \alpha \le \eta\oplus_{L^*}\beta\} \le \gamma\oplus_{L^*}\beta$. Thus $\alpha \le \gamma\oplus_{L^*}\beta$. □

Definition 26. $(\oplus_{L^*}, \ominus_{L^*})$ is called an intuitionistic coadjoint pair if $(\oplus_{L^*}, \ominus_{L^*})$ satisfy the residual principle (24), and $\ominus_{L^*}$ is called a residual intuitionistic fuzzy difference operator derived from a left-continuous t-norm $\otimes$ if $\oplus_{L^*}$ is an intuitionistic t-conorm derived from a left-continuous t-norm $\otimes$.

Proposition 27. *Suppose that $\ominus_{L^*}$ is a residual intuitionistic fuzzy difference operator derived from a left-continuous t-norm and $(\oplus_{L^*}, \ominus_{L^*})$ is an intuitionistic coadjoint pair; then*

(1) $\alpha = \alpha\ominus_{L^*}0^*$;

(2) $\alpha\ominus_{L^*}\beta = 0^*$*iff* $\alpha \le \beta$;

(3) $\alpha\ominus_{L^*}\beta \le \gamma$*iff* $\alpha\ominus_{L^*}\gamma \le \beta$;

(4) $\alpha\ominus_{L^*}(\beta\oplus_{L^*}\gamma) = (\alpha\ominus_{L^*}\beta)\ominus_{L^*}\gamma = (\alpha\ominus_{L^*}\gamma)\ominus_{L^*}\beta$;

(5) $(\alpha\oplus_{L^*}\beta)\ominus_{L^*}\beta \le \alpha \le (\alpha\ominus_{L^*}\beta)\oplus_{L^*}\beta$;

(6) $((\alpha\oplus_{L^*}\beta)\ominus_{L^*}\beta)\ominus_{L^*}\alpha = ((\alpha\oplus_{L^*}\beta)\ominus_{L^*}\alpha)\ominus_{L^*}\beta = 0$;

(7) $(\alpha\oplus_{L^*}\gamma)\ominus_{L^*}(\gamma\oplus_{L^*}\beta) \le \alpha\ominus_{L^*}\beta \le (\alpha\ominus_{L^*}\gamma)\oplus_{L^*}(\gamma\ominus_{L^*}\beta)$;

(8) $\alpha\ominus_{L^*}(\wedge_{i\in I}\beta_i) = \vee_{i\in I}(\alpha\ominus_{L^*}\beta_i)$;

(9) $(\vee_{i\in I}\alpha_i)\ominus_{L^*}\beta = \vee_{i\in I}(\alpha_i\ominus_{L^*}\beta)$;

(10) *$\alpha\ominus_{L^*}\beta$ is isotone in the first variable and antitone in the second variable.*

Theorem 28. *Suppose that $\alpha, \beta \in L^*$, $\alpha = (a_1, a_2)$, $\beta = (b_1, b_2)$, and $\ominus_{L^*}$ is a residual intuitionistic fuzzy difference operator derived from a left-continuous t-norm $\otimes$; then*

$$\alpha\ominus_{L^*}\beta = \left(a_1 \ominus b_1, \left(b_2 \longrightarrow a_2\right) \wedge \left(1 - a_1 \ominus b_1\right)\right). \tag{26}$$

Proof. Let $\gamma = (u, v) = \alpha\ominus_{L^*}\beta$ and $\gamma_i = (u_i, v_i) \in L^*$.

By Theorem 25,

$$\begin{aligned}\gamma = (u, v) &= \alpha \ominus_{L^*} \beta = \wedge \{\gamma_i \in L^* \mid \alpha \leq \gamma_i \oplus_{L^*} \beta\} \\ &= \wedge \{(u_i, v_i) \mid (a_1, a_2) \leq (u_i, v_i) \oplus_{L^*} (b_1, b_2)\} \\ &= \wedge \{(u_i, v_i) \mid a_1 \leq u_i \oplus b_1, a_2 \leq v_i \otimes b_2, u_i + v_i \leq 1\} \\ &= (\wedge u_i, \vee v_i).\end{aligned} \tag{27}$$

For the first argument u, it follows from Proposition 13 that $u = \wedge\{u_i \mid a_1 \leq u_i \oplus b_1\} = a_1 \ominus b_1$.

For the second argument v,

$$\begin{aligned}v &= \vee \{v_i \mid v_i \otimes b_2 \leq a_2, v_i \leq 1 - u_i, a_1 \leq v_i \oplus b_1\} \\ &\leq (\vee \{v_i \mid v_i \otimes b_2 \leq a_2\}) \\ &\quad \wedge (\vee \{v_i \mid v_i \leq 1 - u_i, a_1 \leq u_i \oplus b_1\}) \\ &\leq (\vee \{v_i \mid v_i \otimes b_2 \leq a_2\}) \wedge (1 - \wedge \{u_i \mid a_1 \leq u_i \oplus b_1\}) \\ &= (b_2 \longrightarrow a_2) \wedge (1 - a_1 \ominus b_1),\end{aligned} \tag{28}$$

so $\eta = (u, v) \leq ((a_1 \rightarrow b_1) \wedge (1 - b_2 \ominus a_2), b_2 \ominus a_2)$. Moreover,

$$\begin{aligned}&(a_1 \ominus b_1, (b_2 \longrightarrow a_2) \wedge (1 - a_1 \ominus b_1)) \oplus_{L^*} \beta \\ &\quad = (a_1 \ominus b_1, (b_2 \longrightarrow a_2) \wedge (1 - a_1 \ominus b_1)) \oplus_{L^*} (b_1, b_2) \\ &\quad \geq ((a_1 \ominus b_1) \oplus b_1, (b_2 \longrightarrow a_2) \otimes b_2) \geq (a_1, a_2) = \alpha.\end{aligned} \tag{29}$$

According to Theorem 25

$$\gamma = (u, v) = \wedge \{(u_i, v_i) \mid \alpha \leq (u_i, v_i) \oplus_{L^*} \beta\}. \tag{30}$$

Therefore, $\gamma = (u, v) = (a_1 \ominus b_1, (b_2 \rightarrow a_2) \wedge (1 - a_1 \ominus b_1))$. □

Example 29. The residual intuitionistic fuzzy difference operators derived from Gödel t-norm, Lukasiewicz t-norm, product t-norm, and R_0 t-norm are as follows, respectively:

(1)

$$\alpha \ominus_{L^*} \beta = \begin{cases} (0, 1), & a_1 \leq b_1, \quad b_2 \leq a_2; \\ (0, a_2), & a_1 \leq b_1, \quad b_2 > a_2; \\ (a_1, 1 - a_1), & a_1 > b_1, \quad b_2 \leq a_2; \\ (a_1, a_2), & a_1 > b_1, \quad b_2 > a_2, \end{cases} \tag{31}$$

(2)

$$\alpha \ominus_{L^*} \beta = ((a_1 - b_1) \vee 0, (1 - b_2 + a_2) \wedge (1 - a_1 + b_1) \wedge 1), \tag{32}$$

(3)

$$\alpha \ominus_{L^*} \beta = \begin{cases} (0, 1), & a_1 \leq b_1, \quad b_2 \leq a_2; \\ \left(0, \dfrac{a_2}{b_2}\right), & a_1 \leq b_1, \quad b_2 > a_2; \\ \left(\dfrac{a_1 - b_1}{1 - b_1}, \dfrac{1 - a_1}{1 - b_1}\right), & a_1 > b_1, \quad b_2 \leq a_2; \\ \left(\dfrac{a_1 - b_1}{1 - b_1}, \dfrac{1 - a_1}{1 - b_1} \wedge \dfrac{a_2}{b_2}\right), & a_1 > b_1, \quad b_2 > a_2, \end{cases} \tag{33}$$

(4)

$$\alpha \ominus_{L^*} \beta = \begin{cases} (0, 1), \\ \quad a_1 \leq b_1, \quad b_2 \leq a_2; \\ (0, (1 - b_2) \vee a_2), \\ \quad a_1 \leq b_1, \quad b_2 > a_2; \\ (a_1 \wedge (1 - b_1), (1 - a_1) \vee b_1), \\ \quad a_1 > b_1, \quad b_2 \leq a_2; \\ (a_1 \wedge (1 - b_1), \\ ((1 - a_1) \vee b_1) \wedge ((1 - b_2) \vee a_2)), \\ \quad a_1 > b_1, \quad b_2 > a_2. \end{cases} \tag{34}$$

5. The Dual Triple I Method of FMT and IFMT

If we take the fuzzy difference operator instead of the fuzzy implication, then the model FMT has the following form:

$$\begin{array}{lll} \text{Suppose that } A(x) \ominus B(y) & \cdots & \text{major premise} \\ \text{and given} \qquad\qquad B^*(y) & \cdots & \text{minor premise} \\ \hline \text{caculate} \qquad A^*(x) & \cdots & \text{conclution} \end{array}, \tag{35}$$

where $A(x)$, $A^*(x)$ are the fuzzy sets on the nonempty universe of discourse X; $B(y)$, $B^*(y)$ are the fuzzy sets on the nonempty universe of discourse Y; and $\ominus$ is a fuzzy difference operator.

The dual Triple I principle is as follows:

$A^*(x)$ should be the biggest fuzzy set on X satisfying

$$(A^*(x) \ominus B^*(y)) \ominus (A(x) \ominus B(y)) = 0. \tag{36}$$

Remark 30. Because there are three fuzzy difference operators in formula (36), the dual Triple I method could be called Triple D method.

Theorem 31. *Let $\ominus$ be the fuzzy difference operator derived from a right-continuous t-conorm $\oplus$; the expression of the Triple D solution A^* of FMT problem* (35) *is as follows:*

$$A^*(x) = \wedge_{y \in Y} \{(A(x) \ominus B(y)) \oplus B^*(y)\}. \tag{37}$$

Proof. It follows from formula (37) that

$$\forall y \in Y, \quad A^*(x) \leq (A(x) \ominus B(y)) \oplus B^*(y), \quad x \in X. \tag{38}$$

Since $\ominus$ is the residual fuzzy difference operator, then

$$\forall y \in Y, \quad A^*(x) \ominus B^*(y) \leq A(x) \ominus B(y), \quad x \in X. \tag{39}$$

That is,

$$\forall y \in Y, \quad (A^*(x) \ominus B^*(y)) \ominus (A(x) \ominus B(y)) = 0, \quad x \in X. \tag{40}$$

Suppose that $C(x)$ is a fuzzy set on X such that

$$(C(x) \ominus B^*(y)) \ominus (A(x) \ominus B(y)) = 0, \quad x \in X, \; y \in Y. \tag{41}$$

Since $\ominus$ is the residual fuzzy difference operator, then

$$\forall y \in Y, \quad C(x) \ominus B^*(y) \leq A(x) \ominus B(y). \tag{42}$$

That is,

$$\forall y \in Y, \quad C(x) \leq (A(x) \ominus B(y)) \oplus B^*(y). \tag{43}$$

It follows from formula (37) that $C(x) \leq A^*(x)$. Therefore, A^* given by the formula (37) is the Triple D solution of IFMT. □

Theorem 32. *Let $\ominus$ be the fuzzy difference operator derived from a right-continuous t-conorm $\oplus$; then the Triple D solution A^* of FMT given by* (37) *is reductive; that is, $A^*(x) = A(x)$ whenever $B^*(y) = B(y)$ satisfying $\exists y_0 \in Y$ such that $B(y_0) = 0$.*

Proof. It follows from (37) that if $B^*(y) = B(y)$ then $A^*(x) = \wedge_{y\in Y}\{(A(x) \ominus B(y)) \oplus B(y)\}$. Because $A(x) \ominus B(y) \leq A(x) \ominus B(y)$ and $(\oplus, \ominus)$ is a coadjoint pair, then $A(x) \leq (A(x) \ominus B(y)) \oplus B(y)$. Thus it follows from (37) that $A(x) \leq A^*(x)$, $x \in X$. Moreover, if there exists $y_0 \in Y$ such that $B(y_0) = 0$, then it follows from formula (37) that $A^*(x) \leq (A(x) \ominus 0) \oplus 0 = A(x)$. Therefore, $A^* = A$. □

Theorem 33. *Suppose that $\rightarrow$, $\oplus$, and $\ominus$ are associated operators of $\otimes$; if $\rightarrow$ satisfies contrapositive symmetry, then the Triple I solution A^* is equivalent to the Triple D solution A^*.*

Proof. Since $\rightarrow$ satisfies contrapositive symmetry, then by Theorem 19 and Proposition 7

$$\begin{aligned} A^*(x) &= \wedge_{y\in Y}\left\{B^*(y) \oplus (A(x) \longrightarrow B(y))^c\right\} \\ &= \wedge_{y\in Y}\left\{B^*(y) \oplus \left((B(y))^c \longrightarrow (A(x))^c\right)^c\right\} \\ &= \wedge_{y\in Y}\left\{B^*(y) \oplus (A(x) \ominus B(y))\right\} \\ &= \wedge_{y\in Y}\left\{(A(x) \ominus B(y)) \oplus B^*(y)\right\}. \end{aligned} \tag{44}$$

Therefore, the Triple I solution A^* is equivalent to the Triple D solution A^*. □

It is natural that the model IFMT can be transformed to the following form:

$$\begin{array}{llll} \text{Suppose that} & A(x) \ominus_{L^*} \; B(y) & \cdots & \text{major premise} \\ \text{and given} & \qquad\qquad B^*(y) & \cdots & \text{minor premise} \\ \hline \text{caculate} & A^*(x) & \cdots & \text{conclution} \end{array}, \tag{45}$$

where $A(x)$, $A^*(x)$ are the intuitionistic fuzzy sets on the nonempty universe of discourse X; $B(y), B^*(y)$ are the intuitionistic fuzzy sets on the nonempty universe of discourse Y; and $\ominus_{L^*}$ is a residual intuitionistic fuzzy difference operator on L^*. We denote $A(x) = (A_t(x), A_f(x))$, $B(y) = (B_t(y), B_f(y))$, $B^*(y) = (B_t^*(y), B_f^*(y))$, $A_{-f}(x) = 1 - A_f(x)$, $B_{-f}(y) = 1 - B_f(y)$, and $B_{-f}^*(y) = 1 - B_f^*(y)$. Clearly, A_t, A_f, and A_{-f} are the fuzzy sets on X, respectively, and B_t, B_f, B_t^*, B_f^*, B_{-f}, and B_{-f}^* are the fuzzy sets on Y, respectively.

The Triple D principle of IFMT is as follows:

$A^*(x)$ should be the biggest intuitionistic fuzzy set on X satisfying

$$(A^*(x) \ominus_{L^*} B^*(y)) \ominus_{L^*} (A(x) \ominus_{L^*} B(y)) = 0^* \tag{46}$$

under the order of L^*.

Theorem 34. *Let $\ominus_{L^*}$ be the residual intuitionistic fuzzy difference operator derived from a left-continuous t-norm; the expression of the Triple D solution A^* for IFMT problem* (45) *is as follows:*

$$A^*(x) = \wedge_{y\in Y}\left\{(A(x) \ominus_{L^*} B(y)) \oplus_{L^*} B^*(y)\right\}. \tag{47}$$

Proof. It is similar to the proof of Theorem 31. □

Corollary 35. *Suppose that $\ominus_{L^*}$ is the intuitionistic fuzzy difference operator derived from a left-continuous t-norm; then the Triple D solution A^* of IFMT is given by the following formula:*

$$A^*(x) = \left(A_t^*(x), A_f^*(x)\right), \tag{48}$$

where

$$\begin{aligned} A_t^*(x) &= \wedge_{y\in Y}\left\{(A_t(x) \ominus B_t(y)) \oplus B_t^*(y)\right\} \\ A_f^*(x) &= \vee_{y\in Y}\left\{\left(\left(B_f(y) \longrightarrow A_f(x)\right)\right.\right. \\ &\qquad \left.\left. \wedge \left(1 - A_t(x) \ominus B_t(y)\right)\right) \otimes B_f^*(y)\right\}. \end{aligned} \tag{49}$$

Proof. The proof is trivial by Theorems 28 and 34. □

According to Corollary 35, we consider the Triple D solutions of the following two FMT problems and the Triple I solution of the following FMP problem:

$$\begin{array}{llll} \text{Suppose that} & A(x) \ominus \; B_t(y) & \cdots & \text{major premise} \\ \text{and given} & \qquad\quad B_t^*(y) & \cdots & \text{minor premise} \\ \hline \text{caculate} & \widetilde{A}_t^*(x) & \cdots & \text{conclution} \end{array}, \tag{50}$$

$$\begin{array}{llll} \text{Suppose that} & A_t(x) \ominus \; B_t(y) & \cdots & \text{major premise} \\ \text{and given} & \qquad\quad B_{-f}^*(y) & \cdots & \text{minor premise} \\ \hline \text{caculate} & \overline{A}_t^*(x) & \cdots & \text{conclution} \end{array}, \tag{51}$$

$$\begin{array}{llll} \text{Suppose that} & B_f(y) \longrightarrow \; A_f(x) & \cdots & \text{major premise} \\ \text{and given} & B_f^*(y) & \cdots & \text{minor premise} \\ \hline \text{caculate} & \qquad\quad \widetilde{A}_f^*(x) & \cdots & \text{conclution} \end{array}. \tag{52}$$

Theorem 36. *Suppose that $\widetilde{A}_t^*(x)$ and $\overline{A}_t^*(x)$ are the Triple D solutions of FMT problem* (50) *and* (51), *respectively, and $\widetilde{A}_f^*(x)$ is the Triple I solution of FMP problem* (52); *then*

$$\begin{gathered} A_t^*(x) = \widetilde{A}_t^*(x), \\ A_f^*(x) \leq \left(1 - \overline{A}_t^*(x)\right) \wedge \widetilde{A}_f^*(x) \leq 1 - A_t^*(x), \end{gathered} \tag{53}$$

where $A^*(x) = (A_t^*(x), A_f^*(x))$ is the Triple D solution of IFMT problem (45) *given by* (49).

Proof. Obviously, $A_t^*(x) = \widetilde{A}_t^*(x)$ from Theorem 31. It follows from formula (49) that

$$\begin{aligned} A_f^*(x) &= \vee_{y\in Y}\left\{\left(\left(B_f(y) \longrightarrow A_f(x)\right)\right.\right.\\ &\qquad \left.\wedge\left(1 - A_t(x)\ominus B_t(y)\right))\otimes B_f^*(y)\right\}\\ &\le \vee_{y\in Y}\left\{\left(\left(B_f(y) \longrightarrow A_f(x)\right)\otimes B_f^*(y)\right.\right.\\ &\qquad \left.\wedge\left(\left(1 - A_t(x)\ominus B_t(y)\right)\otimes B_f^*(y)\right)\right\}\\ &\le \left(\vee_{y\in Y}\left\{\left(B_f(y)\longrightarrow A_f(x)\right)\otimes B_f^*(y)\right\}\right)\\ &\quad \wedge\left(\vee_{y\in Y}\left\{\left(1 - A_t(x)\ominus B_t(y)\right)\otimes\left(1 - B_{-f}^*(y)\right)\right\}\right)\\ &= \left(\vee_{y\in Y}\left\{\left(B_f(y)\longrightarrow A_f(x)\right)\otimes B_f^*(y)\right\}\right)\\ &\quad \wedge\left(1 - \wedge_{y\in Y}\left\{\left(A_t(x)\ominus B_t(y)\right)\oplus B_{-f}^*(y)\right\}\right)\\ &= \widetilde{A}_f^*(x)\wedge\left(1 - \overline{A}_t^*(x)\right). \end{aligned} \tag{54}$$

Since $\forall y \in Y, B_t^*(y) + B_f^*(y) \le 1$, that is, $\forall y \in Y, B_t^*(y) \le 1 - B_f^*(y) = B_{-f}^*(y)$, then

$$\begin{aligned} &\wedge_{y\in Y}\left\{\left(A_t(x)\ominus B_t(y)\right)\oplus B_t^*(y)\right\}\\ &\qquad \le \wedge_{y\in Y}\left\{\left(A_t(x)\ominus B_t(y)\right)\oplus B_{-f}^*(y)\right\}. \end{aligned} \tag{55}$$

That is, $\widetilde{A}_t^*(x) \le \overline{A}_t^*(x)$. Thus $(1 - \overline{A}_t^*(x)) \wedge \widetilde{A}_f^*(x) \le 1 - \widetilde{A}_t^*(x) = 1 - A_t^*(x)$. The proof is completed. □

Definition 37. $\widetilde{A}^*(x)$ is called the decomposition method solution of IFMT problem (45) if

$$\widetilde{A}^*(x) = \left(\widetilde{A}_t^*(x), \left(1 - \overline{A}_t^*(x)\right)\wedge\widetilde{A}_f^*(x)\right). \tag{56}$$

Do the Triple D solution $A^*(x)$ and the decomposition methods solution $\widetilde{A}^*(x)$ degenerate into the fuzzy sets if the intuitionistic fuzzy sets $A(x)$, $A^*(x)$, and $B(y)$ degenerate into the fuzzy sets? The following theorem answers this question.

Theorem 38. *If the intuitionistic fuzzy sets $A(x)$, $B(y)$, and $B^*(y)$ in IFMT problem (45) degenerate into the fuzzy sets, then the Triple D solution $A^*(x)$ and the decomposition methods solution $\widetilde{A}^*(x)$ accordingly degenerate into the fuzzy sets and coincide with the solution $A^*(x)$ given by Theorem 31.*

Proof. According to Theorem 38, we should only prove that $A_f^*(x) = 1 - A_t^*(x)$. Since $A(x)$, $B(y)$, and $B^*(y)$ degenerate into the fuzzy sets, then $A_f(x) = 1 - A_t(x)$, $B_f(x) = 1 - B_t(x)$, and $B_f^*(x) = 1 - B_t^*(x)$. It follows from Proposition 27 and Corollary 35 that

$$\begin{aligned} A_f^*(x) &= \vee_{y\in Y}\left\{\left(B_f(y)\longrightarrow A_f(x)\right)\otimes B_f^*(y)\right\}\\ &= \vee_{y\in Y}\left\{\left(1 - A_t(x)\ominus B_t(y)\right)\otimes B_f^*(y)\right\}\\ &= \vee_{y\in Y}\left\{\left(1 - A_t(x)\ominus B_t(y)\right)\otimes\left(1 - B_t^*(y)\right)\right\}\\ &= 1 - \wedge_{y\in Y}\left\{\left(A_t(x)\ominus B_t(y)\right)\oplus B_t^*(y)\right\}\\ &= 1 - A_t^*(x). \end{aligned} \tag{57}$$

□

Theorem 39. *Let $\ominus_{L^*}$ be the residual intuitionistic fuzzy difference operator derived from a left-continuous t-norm, then the Triple D solution A^* of IFMT given by (47) is reductive; that is, $A^*(x) = A(x)$ whenever $B^*(y) = B(y)$ satisfying $\exists y_0 \in Y$ such that $B(y_0) = 0^*$.*

Proof. If $B^*(y) = B(y)$, then

$$A^*(x) = \wedge_{y\in Y}\left\{\left(A(x)\ominus_{L^*}B(y)\right)\oplus_{L^*}B(y)\right\}. \tag{58}$$

It follows from Proposition 27 (13) that $(A(x) \le (A(x)\ominus_{L^*}B(y))\oplus_{L^*}B(y)$, so $A(x) \le A^*(x)$. On the other hand, $A^*(x) \le (A(x)\ominus_{L^*}B(y_0))\oplus_{L^*}B(y_0) = (A(x)\ominus_{L^*}0^*)\oplus_{L^*}0^* = A(x)$. Thus $A^*(x) = A(x)$. □

It indicates that the Triple D method is more meaningful than the Triple I method in point of reductivity for IFMT.

We know that the Triple D method of FMT is reductive; it is easy to prove that the decomposition method of IFMT is reductive.

Theorem 40. *Let $\ominus_{L^*}$ be the residual intuitionistic fuzzy difference operator derived from a left-continuous t-norm; then the decomposition method solution $\widetilde{A}^*(x)$ of IFMT given by (56) is reductive; that is, $A^*(x) = A(x)$ whenever $B^*(y) = B(y)$ satisfying $\exists y_0 \in Y$ such that $B(y_0) = 0^*$.*

Taking into account 0^* being the smallest element of L^* in the Triple D Principle of IFMT, we propose the α-Triple D principle as follows:

$A^*(x)$ should be the biggest intuitionistic fuzzy set on X satisfying

$$\left(A^*(x)\ominus_{L^*}B^*(y)\right)\ominus_{L^*}\left(A(x)\ominus_{L^*}B(y)\right) \le \alpha \tag{59}$$

under the order of L^* where $\alpha \in L^*$.

Theorem 41. *Let $\ominus_{L^*}$ be the residual intuitionistic fuzzy difference operator derived from a left-continuous t-norm; the expression of the α-Triple D solution A^* for IFMT is as follows:*

$$A^*(x) = \wedge_{y\in Y}\left\{\alpha\oplus_{L^*}\left(\left(A(x)\ominus_{L^*}B(y)\right)\oplus_{L^*}B^*(y)\right)\right\}. \tag{60}$$

6. Conclusion

In [27], the Triple I method and the decomposition method of IFMP were first presented and the reductivity of methods were verified; however, it was confirmed that the Triple I method of IFMT satisfied the local reductivity instead of reductivity. In order to achieve the improvement of reductivity of Triple I method for IFMT, the Triple D method and the decomposition method of IFMT are presented and the reductivity of methods is proved. Moreover, the concepts of the intuitionistic fuzzy difference operators and coadjoint pair are proposed, and the unified form of the residual intuitionistic fuzzy difference operators adjoint to intuitionistic t-conorms derived from left-continuous t-norms is provided.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors acknowledge the support from the Natural Science Foundation of China (nos. 11101253, 11301321, and 11301319) and the Fundamental Research Funds for the Central Universities (no. GK201403001).

References

[1] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[2] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 1, pp. 28–44, 1973.

[3] D. Dubois, "The role of fuzzy sets in decision sciences: old techniques and new directions," *Fuzzy Sets and Systems*, vol. 184, no. 1, pp. 3–28, 2011.

[4] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 2, pp. 405–435, 1990.

[5] E. H. Mamdani and B. R. Gaines, *Fuzzy Reasoning and Its Applications*, Academic Press, London, UK, 1981.

[6] L. X. Wang, *A Course in Fuzzy Systems and Control*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1997.

[7] G. J. Wang, "Formalized theory of general fuzzy reasoning," *Information Sciences*, vol. 160, pp. 251–266, 2004.

[8] H. W. Liu and G. J. Wang, "Unified forms of fully implicational restriction methods for fuzzy reasoning," *Information Sciences*, vol. 177, no. 3, pp. 956–966, 2007.

[9] D. W. Pei, "On the strict logic foundation of fuzzy reasoning," *Soft Computing*, vol. 8, no. 8, pp. 539–545, 2004.

[10] D. W. Pei, "Unified full implication algorithms of fuzzy reasoning," *Information Sciences*, vol. 178, no. 2, pp. 520–530, 2008.

[11] D. W. Pei, "Formalization of implication based fuzzy reasoning method," *International Journal of Approximate Reasoning*, vol. 53, no. 5, pp. 837–846, 2012.

[12] S. Song, C. Feng, and E. S. Lee, "Triple I method of fuzzy reasoning," *Computers and Mathematics with Applications*, vol. 44, no. 12, pp. 1567–1579, 2002.

[13] G. J. Wang, "Formalized theory of general fuzzy reasoning," *Information Sciences*, vol. 160, pp. 251–266, 2004.

[14] G. J. Wang and L. Fu, "Unified forms of Triple I method," *Computers and Mathematics with Applications*, vol. 49, no. 5-6, pp. 923–932, 2005.

[15] D. Dubois and H. Prade, "Gradualness, uncertainty and bipolarity: making sense of fuzzy sets," *Fuzzy Sets and Systems*, vol. 192, pp. 3–24, 2012.

[16] C. Franco, J. Montero, and J. T. Rodríguez, "A fuzzy and bipolar approach to preference modeling with application to need and desire," *Fuzzy Sets and Systems*, vol. 214, pp. 20–34, 2013.

[17] K. T. Atanassov, "Intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 87–96, 1986.

[18] K. Atanassov, *Intuitionistic Fuzzy Sets: Theory and Applications*, Physica, Heidelberg, Germany, 1999.

[19] I. Bloch, "Mathematical morphology on bipolar fuzzy sets: general algebraic framework," *International Journal of Approximate Reasoning*, vol. 53, no. 7, pp. 1031–1060, 2012.

[20] D. F. Li and C. T. Cheng, "New similarity measures of intuitionistic fuzzy sets and application to pattern recognitions," *Pattern Recognition Letters*, vol. 23, pp. 221–225, 2002.

[21] H. W. Liu, "New similarity measures between intuitionistic fuzzy sets and between elements," *Mathematical and Computer Modelling*, vol. 42, no. 1-2, pp. 61–70, 2005.

[22] H. W. Liu and G. J. Wang, "Multi-criteria decision-making methods based on intuitionistic fuzzy sets," *European Journal of Operational Research*, vol. 179, no. 1, pp. 220–233, 2007.

[23] Z. S. Xu, *Intuitionistic Fuzzy Information Aggregating Theory and Its Application*, Science Press, Beijing, China, 2008, (Chinese).

[24] C. Cornelis, G. Deschrijver, and E. E. Kerre, "Implication in intuitionistic fuzzy and interval-valued fuzzy set theory: construction, classification, application," *International Journal of Approximate Reasoning*, vol. 35, no. 1, pp. 55–95, 2004.

[25] G. Deschrijver, C. Cornelis, and E. E. Kerre, "On the representation of intuitionistic fuzzy t-norms and t-conorms," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 1, pp. 45–61, 2004.

[26] G. Deschrijver and E. E. Kerre, "Classes of intuitionistic fuzzy t-norms satisfying the residuation principle," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 11, no. 6, pp. 691–709, 2003.

[27] M. C. Zheng, Z. K. Shi, and Y. Liu, "Triple I method of approximate reasoning on Atanassov's intuitionistic fuzzy sets," *International Journal of Approximate Reasoning*, vol. 55, no. 6, pp. 1369–1382, 2014.

[28] E. P. Klement, R. Mesiar, and E. Pap, *Triangular Norms*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[29] G. J. Wang, *Introduction to Mathematical Logic and Resolution Principle*, Science Press, Beijing, China, 2nd edition, 2006, (Chinese).

[30] M. C. Zheng and G. J. Wang, "Co-residuated lattice with application," *Fuzzy Systems and Mathematics*, vol. 19, no. 4, pp. 1–6, 2005 (Chinese).

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 138927, 11 pages
http://dx.doi.org/10.1155/2014/138927

Research Article

An Innovative SIFT-Based Method for Rigid Video Object Recognition

Jie Yu,[1] Fengli Zhang,[1] and Jian Xiong[2]

[1] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, No. 2006, Xiyuan Avenue, West Hi-Tech Zone, Chendu, Sichuan 611731, China*
[2] *Mathematics and Computer Science Department, Shangrao Normal University, No. 85, Zhiming Avenue, Shangrao, Jiangxi 334001, China*

Correspondence should be addressed to Jie Yu; yj19731201@126.com

Received 11 April 2014; Revised 5 June 2014; Accepted 9 June 2014; Published 6 July 2014

Academic Editor: Jer-Guang Hsieh

This paper presents an innovative SIFT-based method for rigid video object recognition (hereafter called RVO-SIFT). Just like what happens in the vision system of human being, this method makes the object recognition and feature updating process organically unify together, using both trajectory and feature matching, and thereby it can learn new features not only in the training stage but also in the recognition stage, which can improve greatly the completeness of the video object's features automatically and, in turn, increases the ratio of correct recognition drastically. The experimental results on real video sequences demonstrate its surprising robustness and efficiency.

1. Introduction

In recent years, security surveillance systems being called "sky-eye" and hand-held video cameras have increasingly grown in popularity, and the need of applications such as video-based object recognition and tracking or retrieval [1, 2] goes up rapidly. However, to identify a (identical) 3D object in videos or image sequences is still a challenging problem mainly because a 3D object's visual appearance may be different due to viewpoint or lighting changes.

For example, in Figure 1, there is a series of frame images captured from a video clip in which the vehicle is turning. If only (e) is used as the training image, then the vehicles in (h) or (t) would not be correctly recognized due to the changing of the viewpoints. Even for a human being with high sense of responsibility, just only providing him with (d) as the training image, he cannot confirm also that the cars in (h) or (t) are identical to the one in (d). However, when browsing the video clip, the source of Figure 1, everyone having normal cognitive ability can tell that the cars in (a) ~ (t) are all identical easily. Why?

With the help of selected regions in Figure 1, let us describe briefly what happens in the video browsing process of human being to serve as a modest spur to introduce the novel method proposed in this paper to come.

Before browsing the video clip, someone else, who has known the target object, should tell us which object is the target, for example, the selected region in frame image (a). Then we focus on the target object and try to dig its typical features out for saving—this part corresponds to the feature initialization of the target object. After that we keep going on to the next frame. What will we do with it? Firstly, we try to judge whether the target object is in it or not. Secondly, if the target objects are recognized in the frame, then we judge whether any new features of the target arise or not. However, how do we do that? Do we just only use incomplete target object features extracted just before? The answer is no. For our human beings, we do it by using both features and moving trajectories matching together. For example, comparing to the selected region in frame (a), we can read that although frames (b), (c), and (d) are not continuous, the cars in them not only share many features, but also keep going on the same

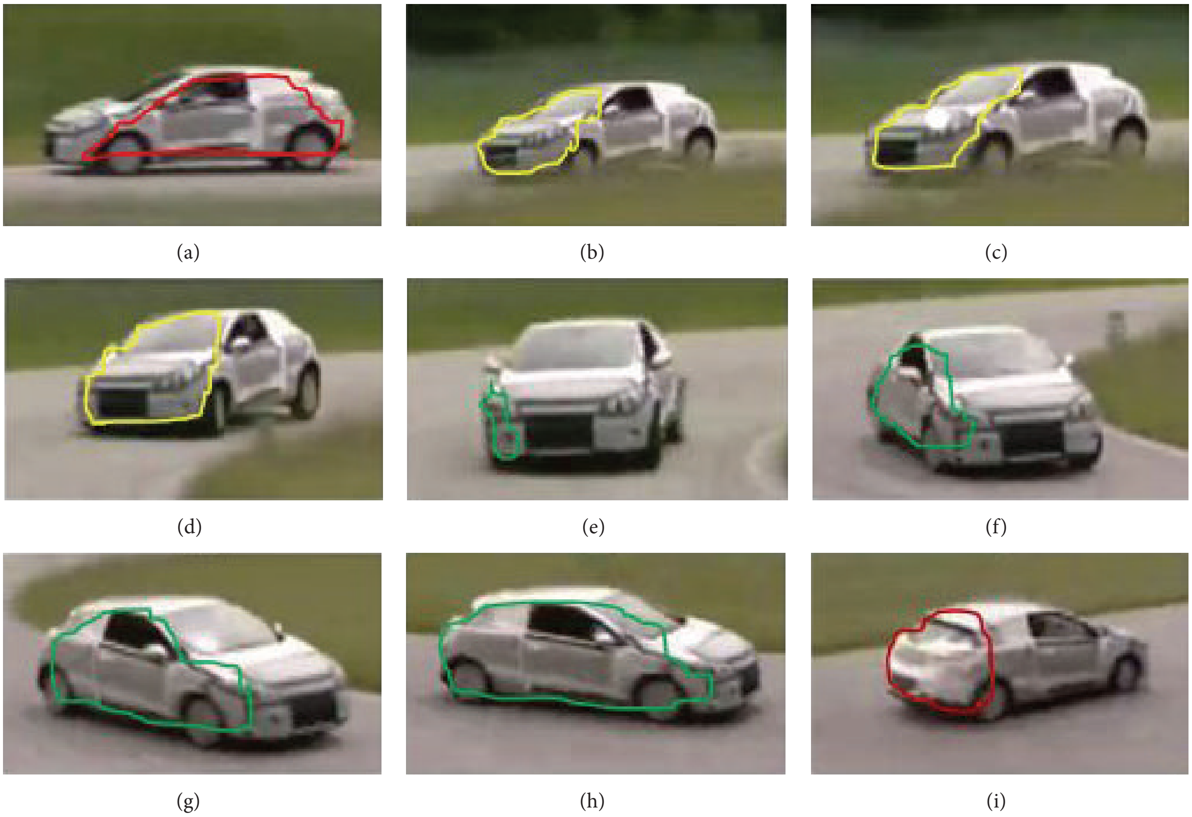


Figure 1: Frames captured from a surveillance video in which the vehicle is turning.

turning trajectory, so a conclusion can be drawn that cars in them are all identical with high probability. Meanwhile, we can find that regions in them, being surrounded by yellow line, seem to be similar and moreover keep moving on a similar trajectory, so we can conclude that they are part of the target car with high probability also. Then, new features can be extracted from the regions and be used for further recognition.

Processes of object recognizing and feature updating are being executed alternately and iteratively, and eventually all vehicles would be recognized and most of the new distinctive features would be extracted and saved.

Inspired by this physiological process of human being, a novel method for rigid video object recognition is proposed in this paper; its main contribution lies in the following.

(1) Modeled on the human recognition system, it makes the object recognition and feature updating processes organically unify together, which means that feature extraction and updating can be done not only in training stage, but also in recognition stage, which can improve greatly the completeness of features of the target video object and can in turn increase the ratio of correct recognition dramatically.

(2) Its object recognition is based on models of both feature and trajectory matching, which improve greatly the accuracy of the identification.

(3) Even if provided with only a single training image, it can create a relatively complete model of the target 3D object, using multiple 2D views automatically.

This paper is organized as follows. In Section 2, related researches are reviewed. In Section 3, the initialization of the target video object's feature database is given. In Section 4, feature point's trajectory is discussed and the iterative object is recognized, and then feature updating process is described. The experimental setup is presented and the analysis of the results is given in Section 5. Finally, in Section 6, conclusions are drawn.

2. Related Researches

There are lots of researches for 3D object recognition in which most of them are to model 3D objects using multiple 2D views. For example, in [3, 4], a method is proposed for combining multiple images of a 3D object into a single model representation; however, this approach requires that the single target object should occupy the majority of each of the training view images, which makes it meaningless in practice; the other most primary approaches are to get and describe the object's stable surface features from 2D view images, such as color feature [5], texture feature, shape feature [6], and contour feature [7].

Another direction of research is to use automatic background segmentation [2, 8], which digs the moving objects

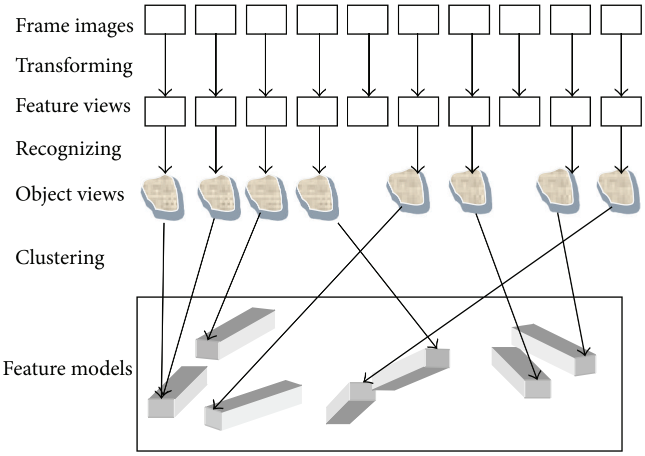


Figure 2: The Relationship between frames and feature models. Each frame is transformed into feature view; similar object views contained in different frames are clustered into one or more feature models.

out from the scenes first and then does recognition. However this can only work well with videos in which there is no dramatic background changing.

Motivated by biological vision systems, as this paper is, some simulating approaches are proposed in the long history of research. In [9], a method is proposed in which a scene is analyzed to produce multiple feature maps which are combined to form a saliency map which is used to bias attention to regions of the highest activity. And then some adjustments and improvements have been suggested to [9] by Itti et al. [10, 11].

Our work seemingly shares some themes with [12, 13] in the fact that features are learned from image sequences and/or video frames automatically. However, there are major differences between them; that is, in [12], its goal is to optimize the parameters of the known features in the tracked patches, not to learn new features of the targets; for example, for car detection, it can only extract and track features in the fixed manually labeled car's image area and optimize parameters of the known features in corresponding different frame, but it cannot learn new features of the target car from the correlations between the video frames, and in [13], its goal is to utilize offline feature tracking to observe feature transformations under large camera motions and then one can construct the database accordingly, keeping fewer images of each scene than would otherwise be needed; that is, its essence lies in using feature transformation to reduce storage costs, not feature learning either.

3. Feature Database Initialization

In this paper, the target object's feature database is organized as a set of feature models which represent different views of the target object, and each model consisted of all stable features extracted from corresponding object views which is contained in corresponding frame image. The relationship between frame image, object view, and feature model is shown in Figure 2. Furthermore, feature models are linked with each other by their sharing features. For example, in Figure 1, the vehicle views in frame images (a) ~ (d) seem to be similar and share many features, so four feature models may be built in feature database accordingly and may be linked with each other by their sharing of features. Of course, (b) and (c) seem to be identical, and maybe only one feature model needs to be built according to them. How do we estimate the degree of the similarity between object views and then decide how many feature models should be created accordingly? Refer to Section 4 please.

To initialize the feature database, it is needed to specify the target object firstly, that is, to specify the representative region of the target object firstly and then to initialize the object's feature database with it.

3.1. To Specify the Target Object. However, as shown in Figure 3, there are six selection modes; then which type is better or the best?

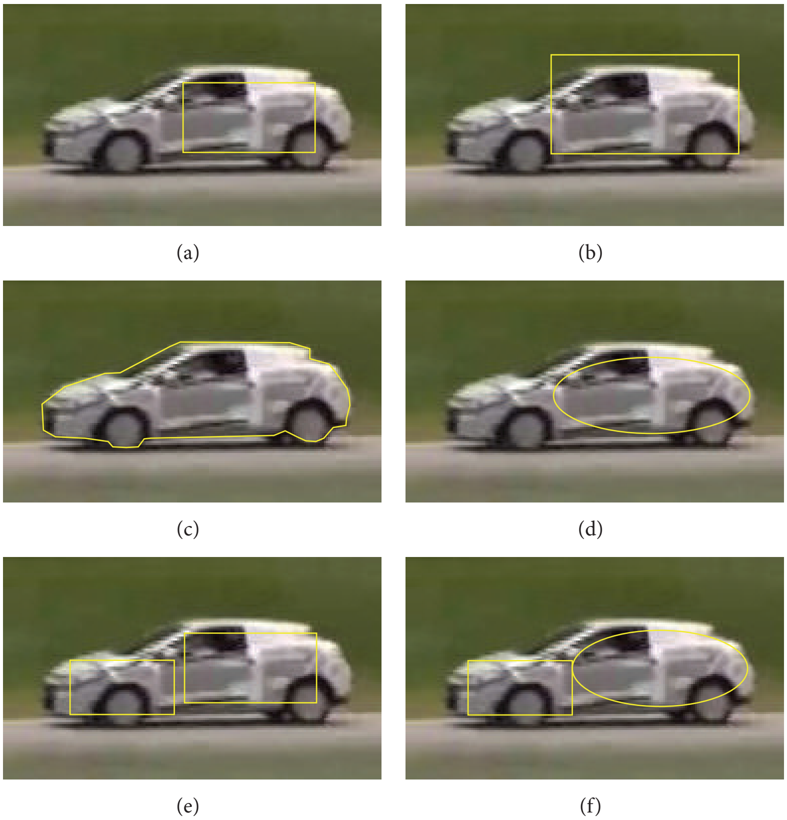
Before drawing a conclusion, we should figure out the meaning of the selected region to the novel recognition method thoroughly. Firstly, the region is the representative of the target object and the feature points in it are feature seeds for the next stage, so all pixels selected should belong to the target object; meanwhile, the more pixels to be selected and the more distinctive they are, which means more potential feature points, the better. Secondly, all feature points in the selected region are used to recognize the object views in the next frames, so the more the duration of the region keeping in the next frames, the better.

Based on the consideration above, the selection modes are all desirable except (b), meanwhile, of all the option, mode (e) and (f) seems to be the better, mode (c) seems to be the best. However, the mode (c) seems to be too complicated to operate, and the extraction of the point features in (c), (d), and (f) involves the operation of padding with zero pixels, so taking the convenience of the select operation and the computation complexity into account additionally, we prefer the mode (e).

3.2. To Initialize Feature Database. After specifying the representative region of the target object, to allow for efficient matching between the region and frames in video clips (namely, the training or being searched video clips), the selected region and the video frames should be represented as a set of stable features, using some kind of feature descriptor.

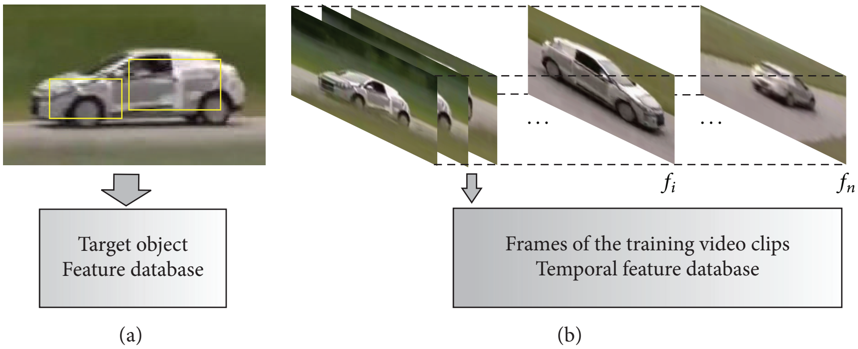
In this paper, RGB-SIFT [15] is adopted to compute features. For the RGB-SIFT descriptor, SIFT descriptor is computed for each of the three color channels (R/G/B), respectively. To see the details of the transformation process, refer to [15] please.

As Figure 4 showed, after the processing with the above steps of RGB-SIFT, the selected region is transformed into stable features and saved into a feature model in the target object's feature database directly. Meanwhile, each of the frames in videos is transformed into feature view and saved into the temporal feature database, respectively.



(a) (b) (c) (d) (e) (f)

Figure 3: The region selection modes of the target object.



(a) (b)

Figure 4: Features database initialization using some kind of feature descriptor.

Each SIFT feature, along with a record of the location, orientation, and scale, represents a vector of local image measurements in a manner that is invariant to scaling, translation, changes in illumination of the scene, and limited rotation of the camera's viewpoint. The size of frame image region that is sampled for each feature can be varied, but the experiments described in this paper all use a vector of 3 × 128 samples for each feature to sample 8 gradient orientations over a 4 × 4 sampling region in each color channel image. A typical frame image may produce several thousand overlapping features at a wide range of scales that form a redundant representation between the adjacent frame images.

4. Video Object Recognition Accompanied by Feature Database Updating

After the initialization stage described in the above section, in the object's feature database, there is only one feature model which consisted of all feature vectors extracted from the

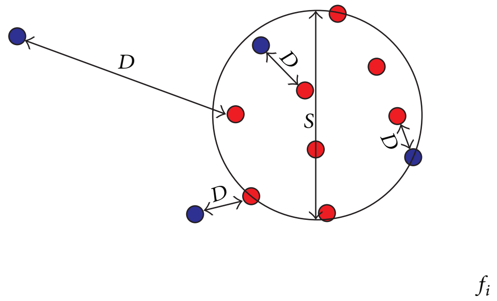


Figure 5: The diameter $S(fv_i)$ of the minimum circle covering all the matched feature keypoints in $\mathrm{Fsq}(fv_i)$ represents approximately the dimension of the target object in f_i, and D denotes the Euclidian distance from the candidate feature point to the nearest feature keypoint in the circle.

selected region. Obviously, one feature model is not enough for video object recognizing, and something should be done to enrich the feature database. Just like human being does, the process of video object recognition proposed in this paper is accompanied by object feature updating, and its main work includes the following.

(1) Video object recognition: to recognize the target object views in video frames, using both feature and trajectory matching.

(2) Feature database updating: to enrich feature models and features in it with different views of the target object contained in corresponding frames, using both feature and trajectory matching also.

The initial feature model and features in it can be used as seeds to do these works.

4.1. Feature Point Trajectory Matching. Thankfully, according to affine camera model [16, 17], each of the feature points of the target object may be a trajectory point; that is, the motion trajectory of the target object is shown as its feature point trajectories.

That is, to suppose that M feature points have been tracked between frame views f_i and f_j, then Ψ^{ij}, the corresponding affine transform matrix, can be worked out and $\bar{e}_{f_i,f_j}$, the average error of trajectories matching between them, can be estimated.

The point is that $\bar{e}$ can be used to evaluate the similarity between object views and models, which is the foundation for object recognition and feature database updating with corresponding feature views in this paper.

4.2. Feature Keypoint Matching. A set of matching feature keypoint pairs can be gained by efficient feature matching between feature models and feature views. This is the base for recognizing candidate video object views in corresponding frame images. According to [18], the best candidate match for each feature keypoint is found by identifying its nearest neighbor which is defined as the keypoint with minimum Euclidean distance for the invariant feature vector.

No algorithm is known for being any more efficient than exhaustive search in identifying the exact nearest neighbors of points in high-dimensional spaces. Our RGB-SIFT keypoint descriptor has a 3 × 128-dimensional feature vector; therefore, we have used an approximate algorithm, called the Best-Bin-First (BBF) algorithm [19]. This is approximate in the sense that it returns the closest neighbor with high probability. To see the details of keypoint matching, refer to [19] please.

This feature matching process between feature models and views is expressed as function feature point match (feature models, feature view, etc.) in this paper.

4.3. Notations Defined Specifications. For the sake of describing the procedure of video object recognition and feature updating conveniently, we adopt the following notations:

(1) f_i: the ith frame image in video;

(2) Q: the feature database of the target object in which all the known feature models are saved;

(3) TQ: the temporal feature database in which all the feature views are saved;

(4) fv_i: the ith feature view in TQ;

(5) Model(i): the ith feature model in Q;

(6) SSW: the scalable sliding window in which feature views being recognized with an identical feature model are saved temporally;

(7) SQ, Fsq(): feature sets in which the matching features in the matched feature model and feature view are saved, respectively;

(8) $S(fv_i)$: the dimension of the area occupied in the frame view fv_i by the known feature points in $\mathrm{Fsq}(fv_i)$. The value of it is approximated by the diameter of the minimum circle which covers all the feature points in $\mathrm{Fsq}(fv_i)$. The circle can be gained by using the Hough transform method [20] (see Figure 5);

(9) $D_{f_i}(p)$: the distance of one candidate feature point p to the known region of the target object in frame f_i. Its value is approximated by its Euclidean distance to the nearest keypoint in $\mathrm{Fsq}(fv_i)$ (see Figure 5).

4.4. Video Object Recognition in One Frame. Feature models in the object's feature database represent corresponding views of the target video object. According to [3], they can be used to recognize similar views of the object over a range of rotations in depth of at least 20 degrees in any direction.

Now, to suppose that a feature view fv_i has matched with a feature model model(j), using the feature matching procedure described in Section 4.2, and the matching features (≥3) between them are already saved, respectively, in $\mathrm{Fsq}(fv_i)$, SQ, then the procedure to recognize the target video object in fv_i can be described in Algorithm 1.

4.5. Feature Database Updating. Once object views are recognized in corresponding frame images, then the process of

```
(1) If number(Fsq($fv_i$)) ≥ 3 then
(2) {
(3) Set $SQ \leftarrow SQ \cap$ Fsq($fv_i$)//to eliminate feature points without matching with any feature
      points in Fsq($fv_i$) to ensure the similarity property between the recognized object views next
(4) Set $n \leftarrow 0$
(5) Set recognized ← false
(6) Set $T1 \leftarrow$ a threshold value //for example, 0.8
(7) Set $T2 \leftarrow$ a threshold value //for example, 0.8
(8) Calculate $S(Fv_i)$ with feature points Fsq($fv_i$) //to estimate the dimension of the target
      object in $f_i$, using the Hough Transform method with feature points in Fsq($fv_i$)
(9) $k \leftarrow$ count the number of feature models linked by features in $SQ$
(10) For each of feature models linked with features in $SQ$ do
(11) {
(12)   Calculate $S(SQ)$ with feature points in $SQ$ //to estimate the dimension of the target
         object in the feature model.
(13)   $M \leftarrow$ Count the number of feature keypoints in the minimum circle determined by $S(SQ)$
(14)   Calculate $\bar{e}_{SQ,f_i}$//to estimate the residual error which shows the degree of the similarity
         between views that SQ and $f_i$ implying, denoted by model($SQ$) and model($f_i$) respectively
(15)   If $\bar{e}_{SQ,f_i} \leq 0.25 \times \max(S(SQ), S(fv_i))$ and $(\text{num}(\text{Fsq}(fv_i)))/M \geq T1$ then
(16)     {
(17)       $n = n + 1$
(18)     }
(19) }
(20) If $(n/k) \geq T2$ then recognized ← true
(21) Ouput $f_i$
(22) }
(23) Return $S(SQ)$, $S(Fv_i)$
```

Algorithm 1: Recognizinginframe(output: recognized or not, $S(SQ)$, $S(Fv_i)$; input: SQ, Fsq(fv_i)) *//to judge whether the target video object is in the frame f_i or not.*

feature database updating with them can be started immediately. The video object feature database consisted of feature models in which features are extracted from corresponding object views. Then the procedure of feature database updating includes two different level aspects.

(1) Feature models updating: to enrich feature models with different views of the target object contained in corresponding feature views.

(2) Model features updating: to enrich features in feature models with new features found in the corresponding similar feature views.

4.5.1. Feature Model Updating. To update feature models, it is required that at least one similar view of the target object is recognized in corresponding feature view with a specific feature model.

Now, the feature view fv_i is recognized with a feature model, and the matching feature points between them are already saved, respectively, in Fsq(f_i), SQ, and then the procedure to enrich the feature model with fv_i can be described in Algorithm 2.

4.5.2. Model Features Updating. To update features in corresponding feature models, it is required that at least two similar views of the target video object are recognized.

Also, to suppose that feature views fv_i and fv_j are recognized with a feature model in Q, and the matching features between them are already saved, respectively, in Fsq(fv_i), Fsq(fv_j), SQ, then the procedures to enrich features in corresponding models can be described in Algorithm 3.

Why is the constant multiplier in line 12 of Algorithm 1, line 2 of Algorithm 2, and lines 3 and 10 in Algorithm 3 0.25, 0.05, 0.25, and 0.85, respectively? According to [21], if we imagine placing a sphere around an object, then rotation of the sphere by 30 degrees will move no point within the sphere by more than 0.25 times the projected diameter of the sphere, and for the examples of typical 3D objects used in [16, 21], an affine solution works well with allowing residual errors up to 0.25 times the maximum projected dimension of the object. In addition, $S(fv_i)$ is less than the actual projected diameter of the target object in f_i generally, so the constant multipliers adopted in this paper would work well too. Thankfully, the experimental results support them.

4.6. General Procedure of the Video Object Recognition. So far, methods for recognizing the target video object in one frame and updating the feature database with one or two frames have been described in the above paragraphs.

So, we can write out the integrated cyclic procedure of recognizing the video object accompanied with updating the feature database briefly in Algorithm 4.

```
Calculate $\bar{e}_{SQ,f_i}$ //to estimate the residual error which shows the degree of the similarity between the object feature models that SQ and $f_i$ implying, denoted by model(SQ) and model($f_i$) respectively
set $T = 0.05 * S(fv_i)$ //to set a lower limit to the degree of their similarity between model(SQ) and model($f_i$)
if $\bar{e}_{SQ,f_i} > T$ then
  {
    New(model($f_i$), Q) //to create a new feature model in Q and save features in Fsq($fv_i$) into it
    Link(model($f_i$), model(SQ)) //to link model($f_i$) to model(SQ) withall matching features between them
  }
Else
  {
    Combine(model($f_i$), model(SQ)) //to combine model($f_i$) with model(SQ), which means the new features from $f_i$ should be added to the existing model model(SQ)
  }
Endif
return Q
```

Algorithm 2: FModelUpdating(Output: Q; Input: Q, SQ, $S(fv_i)$, Fsq(fv_i)) *//to update feature models in Q with recognized frame image f_i.*

```
Calculate $X^{i,j}$ with Fsq($fv_i$), Fsq($fv_j$) //to calculate the column vector determined by the transform matrix with matching feature point pairs in Fsq($fv_i$) ∩ Fsq($fv_j$)
Calculate $\bar{e}_{f_i,f_j}$// to estimate the residual error between model($f_i$) and model($f_j$), subjecting $X^{i,j}$ and feature point pairs in Fsq($fv_i$) ∩ Fsq($fv_j$) to equation $\bar{e}$
If $\bar{e}_{f_i,f_j} \geq 0.05 * \min\left(S(fv_i), S(fv_j)\right)$ then //To determine whether there is a perceptible change between $f_i$ and $f_j$ or not
{
  FeaturepointMatch(Output: output1, output2; Input: $fv_i$, $fv_j$) //to match features between $fv_i$ and $fv_j$ and the matching feature keypoints are all saved temporally in Output1and Output2 respectively except features in Fsq($fv_i$) and Fsq($fv_j$)
  For $k \leftarrow 1$ to Num(FeaturepointMatch.Output) do //for each matching feature keypoint pair $\left(p_k^{f_i}, p_k^{f_j}\right)$ in the Outputs
    {
      Calculate $D_{f_i}$, $D_{f_j}$//to estimate the corresponding distance of this candidate feature points to the known area of the target object in $f_i$ and $f_j$, respectively
      Calculate $e_{f_i,f_j} = \left\| \begin{pmatrix} u^i & v^i & 0 & 0 & 1 & 0 \\ 0 & 0 & u^i & v^i & 0 & 1 \end{pmatrix} X^{i,j} - \begin{pmatrix} u^j \\ v^j \end{pmatrix} \right\|$//to evaluate the degree of agreement between this feature point pair. $(u^i \quad v^i)$, $(u^j \quad v^j)$ represent locations of the candidate matching feature point in $fv_i$, $fv_j$ respectively
      If $D_{f_i} \leq 0.25 * S(f_i)$ and $D_{f_j} \leq 0.25 * S(f_j)$ and $e_{f_i,f_j} \leq 0.85 * \bar{e}_{f_i,f_j}$ then
        {
          AddFeaturetoModel(($p_k^{f_i}$, model($f_i$); $p_k^{f_j}$, model($f_j$)) //if its distance is less than 0.25 times the projected diameter of the known area of the target object in corresponding frame view and the residual error of the projection is less than 0.85 times the average residual error, then add them into corresponding feature models, however, they will be discarded if already existed. By the way, the two models may point to a same feature model
        }
    }
}
Return Q
```

Algorithm 3: FFeatureUpdating(Output: Q; Input: SQ, Fsq(fv_i), Fsq(fv_j)) *//to search new features belonging to the target video object using features in Fsq(fv_i) and Fsq(fv_j), and then to add them into corresponding feature models in Q or discard.*

```
$T \leftarrow$ True //to initiate the loop variable
While $T$ Do //to begin the updating loop
{
  Set $SQ \leftarrow Q, n \leftarrow 1$
  For $i \leftarrow 1$ to Num($TQ$) do //$Num(TQ)$ represents the number of frame views in $Tq$.
  {
    Featurepointmatch(Output: $SQ$, Fsq($f_i$); Input: $SQ$, $fv_i$) //to gain matching feature keypoint pairs between the feature view $fv_i$ and feature models
    Recognizinginfrmae(output: recognized or not, $S(SQ)$, $S(Fv_i)$; input: $SQ$, Fsq($f_i$)) //to confirm whether the frame image $f_i$ contain the target object or not.
    If recognized then
    {
      $SSW(n) \leftarrow fv_i, n \leftarrow n + 1$ //to save the corresponding feature views of the frame image into the scalable sliding window, as Figure 6 shown.
      FModelUpdating(Output: Q; Input: Q, $SQ$, $S(fv_i)$, Fsq($fv_i$)) //to update feature models in Q with recognized feature view $fv_i$
    }
  }
  If Empty($SSW$) then //to judge the scalable sliding window $SSW$ is empty or not
  {
    Break //to jump out the loop
  }
  Else
  {
    For $i \leftarrow 1$ to $n - 1$ do
    {
      For $j \leftarrow i + 1$ to $n$ do
      {
        FFeatureUpdating(Output: $Q$; Input: $SQ$, Fsq($fv_i$), Fsq($fv_j$))
      }
      Delete($TQ$, $fv_i$) //to delete feature view $fv_i$ from the temporal feature database $TQ$
    }
  }
  Dump($SSW$) //to empty $SSW$
}
return $Q$
```

Algorithm 4: RecognizingthenUpdating(Output:recognized frames, Q; Input: Q, TQ) //*to find frames that contain the target object and update feature database with them iteratively*.

After this cyclic procedure of recognizing and then updating, most of the frames containing the target object can be recognized.

5. Experimental Results

The RVO-SIFT method with recognizing and then updating mechanism has shown its better abilities in experiments. The following experimental results are obtained on a computer with AMD Athlon 64 X2 2.6 GHz processor and 4G memory.

In order to fully demonstrate the ability of RVO-SIFT to acquire new features of the target video object, which is the key contribution of this paper, we use an about 2-minute-long surveillance video clip as the training video in which a Renault Megane comprehensive performance is testing and another about 20-minute video clip in which the testing vehicle is running on the highway as the target video to be recognized in, and due to that almost every perspective of the running vehicle exists in the video.

Figure 7 shows that the number of features in the feature database varies as a function of the number of training frame images containing the target video vehicle in the surveillance video with RVO-SIFT and classic RGB-SIFT, respectively. With RVO-SIFT, it can be seen that the number of feature keypoints does increase with the increasing of the number of training frame images. However, after the completeness of the feature database having increased to a considerable degree, the contribution of frame images reduces relatively. Meanwhile, with classic RGB-SIFT, it only extracts features in the specified region which is shown in Figure 1(a), and the size of the feature database keeps invariable.

In order to show how much the feature number affects the outcome of the recognition, the process to recognize the target vehicle in the target video is performed. Figure 8 shows the ratio of the correctly recognized vehicles in frames as a function of the number of feature keypoints in feature databases. We can read from the graph that the ratio of correctly recognized objects increases obviously with the

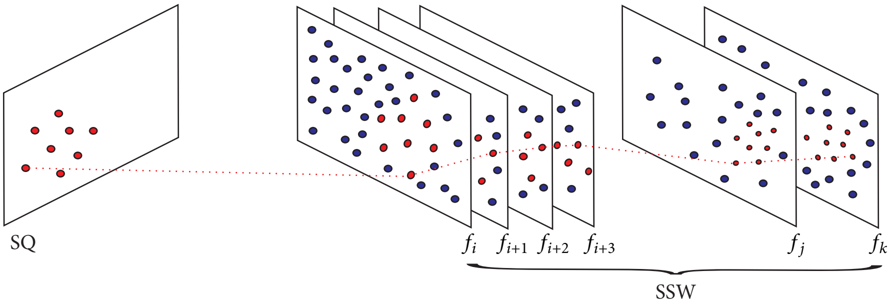


Figure 6: Red dots represent sharing feature keypoints that have matched between an existing model of the target object and similar frame image views in SSW, and blue dots indicate other candidate feature keypoints in each frame image view. The dotted line shows the virtual trajectory of one of the matched feature keypoints.

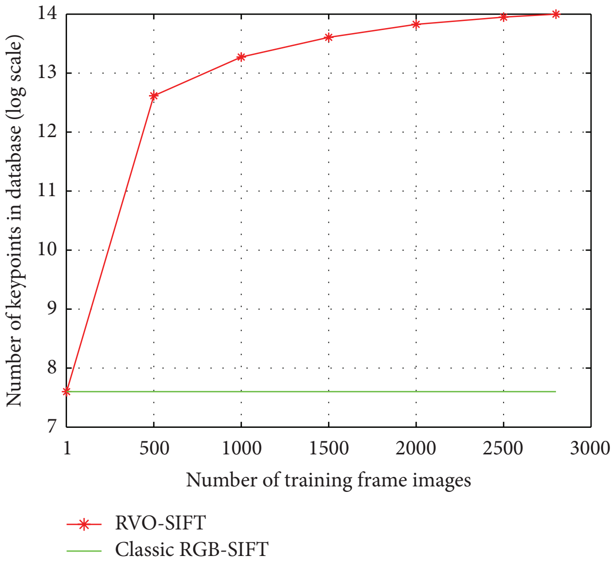


Figure 7: The red line shows the number of keypoints in the feature database (log scale) as a function of the number of training frame images containing the running vehicle with RVO-SIFT and classic RGB-SIFT, respectively.

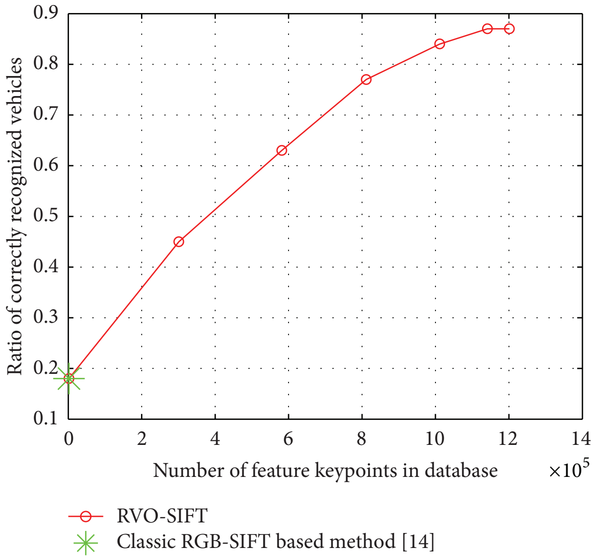


Figure 8: The red line and the green star show the percent of correctly recognized vehicles as a function of database size (log scale) with RVO-SIFT and classic RGB-SIFT based method [14].

increasing number of feature keypoints in database and, meanwhile, the growth rate slows down when the completeness of feature database reaches a certain degree. Some view images recognized correctly are shown in the top line of Figure 10.

Of course, accompanied by the feature database updating process, the recognition process of the RVO-SIFT consumes much more computation time. However, its average delay time is affordable. The experimental results are shown in Figure 9.

In order to show the generality of the RVO-SIFT, the recognition is performed additionally in the micromovie “The New Year, The Same Days” in which the face of the wife character is recognized and a trailer video for a blue and white porcelain, with which a beautiful chinaware is to be recognized in; the results are shown in the two lines below in Figure 10.

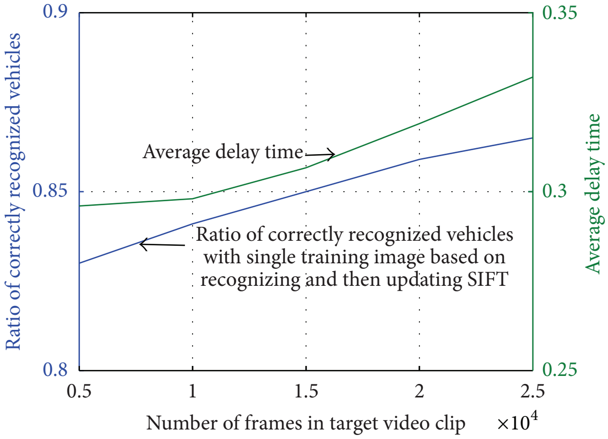


Figure 9: The top line in the graph shows the average delay time as a function of the number of frames in the video clip to be recognized. The bottom line shows the percent of correctly recognized vehicles in frames.

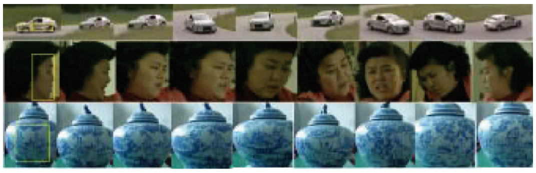


Figure 10: The experimental results show that using the selected areas in bounding boxes in the left images as the training image, the other objects on the right are recognized correctly in corresponding video by RVO-SIFT eventually.

As shown in Figure 10, the RVO-SIFT performs well with real-world videos. Even human faces without exaggerated facial expression changes can be recognized correctly with relatively high rate, just as shown in the middle line. Furthermore, experimental results also show that the RVO-SIFT even can tolerate local camouflages, which is a basic but wonderful ability of human beings, due to the fact that the features of the camouflaged region would be added into the feature database by recognizing and then updating procedure gradually and then they play their roles in recognition subsequently.

6. Conclusion and Future Work

The RVO-SIFT, in which the novel recognizing and then updating mechanism is adopted, is particularly not only a wonderful rigid video object recognizer but also a wonderful feature automatic extractor for rigid video objects. It mixes processes of the object recognizing and feature studying together, just like what human being does in recognition process. It can improve greatly the completeness of the feature database of the target video object automatically and in turn increases drastically the ratio of correctly recognized objects consequently, at the expense of the more affordable millisecond level computation time. In addition to rigid video object recognition, its other potential applications include rigid video motion tracking and segmentation and any others that require feature extraction of the rigid targets in videos or image sequences.

However, RVO-SIFT is based on rigid video object theoretically and experimentally in this paper, so one of the directions for further research is to try to apply it to semirigid video objects, such as video face recognition with exaggerated facial expression changes.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported by the National Nature Science Foundation of China under Grant no. 61163066.

References

[1] S. Gould, J. Arfvidsson, A. Kaehler et al., "Peripheral-foveal vision for real-time object recognition and tracking in video," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI '07)*, vol. 7, pp. 2115–2121, January 2007.

[2] P. Chaturvedi, A. S. Rajput, and A. Jain, "Video object tracking based on automatic background segmentation and updating using RBF neural network," *International Journal of Advanced Computer Research*, vol. 3, no. 2, p. 866, 2013.

[3] D. G. Lowe, "Local feature view clustering for 3D object recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. I-682–I-688, December 2001.

[4] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence*, vol. 31, no. 3, pp. 355–395, 1987.

[5] T. Gevers and A. W. M. Smeulders, "Color-based object recognition," *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, 1999.

[6] Y. Zhong and A. K. Jain, "Object localization using color, texture and shape," *Pattern Recognition*, vol. 33, no. 4, pp. 671–684, 2000.

[7] C. Y. Chung and H. H. Chen, "Video object extraction via MRF-based contour tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 149–155, 2010.

[8] J. Yu and F. L. Zhang, "Distinguishing moving objects from traffic video by the dynamic background skeleton based model," in *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS '13)*, vol. 1, pp. 271–275, IEEE, 2013.

[9] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," in *Matters of Intelligence*, pp. 115–141, Springer, Amsterdam, The Netherlands, 1987.

[10] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[11] Y. Hu, X. Xie, W. Y. Ma et al., "Salient region detection using weighted feature maps based on the human visual attention model," in *Advances in Multimedia Information Processing—PCM 2004*, vol. 3332 of *Lecture Notes in Computer Science*, pp. 993–1000, Springer, Berlin, Germany, 2004.

[12] D. Stavens and S. Thrun, "Unsupervised learning of invariant features using video," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1649–1656, IEEE, San Francisco, Calif, USA, June 2010.

[13] A. Makadia, "Feature tracking for wide-baseline image retrieval," in *Computer Vision—ECCV 2010*, vol. 6315, pp. 310–323, Springer, Berlin, Germany, 2010.

[14] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 2, pp. 1150–1157, September 1999.

[15] K. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[16] C. Kolb, D. Mitchell, and P. Hanrahan, "Realistic camera model for computer graphics," in *Proceedings of the 22nd Annual ACM Conference on Computer Graphics and Interactive Techniques*, pp. 317–324, August 1995.

[17] L. Quan, "Self-calibration of an affine camera from multiple views," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 93–105, 1996.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[19] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1000–1006, June 1997.

[20] J. Illingworth and J. Kittler, "A survey of the hough transform," *Computer Vision, Graphics and Image Processing*, vol. 44, no. 1, pp. 87–116, 1988.

[21] G. L. Foresti, "Object recognition and tracking for remote video surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1045–1062, 1999.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 782351, 10 pages
http://dx.doi.org/10.1155/2014/782351

Research Article

Comparing the Selected Transfer Functions and Local Optimization Methods for Neural Network Flood Runoff Forecast

Petr Maca, Pavel Pech, and Jiri Pavlasek

Department of Water Resources and Environmental Modeling, Faculty of Environmental Sciences, Czech University of Life Sciences Prague, Kamycka 1176, 165 21 Praha 6, Suchdol, Czech Republic

Correspondence should be addressed to Petr Maca; maca@fzp.czu.cz

Received 11 April 2014; Accepted 18 June 2014; Published 2 July 2014

Academic Editor: Jer-Guang Hsieh

The presented paper aims to analyze the influence of the selection of transfer function and training algorithms on neural network flood runoff forecast. Nine of the most significant flood events, caused by the extreme rainfall, were selected from 10 years of measurement on small headwater catchment in the Czech Republic, and flood runoff forecast was investigated using the extensive set of multilayer perceptrons with one hidden layer of neurons. The analyzed artificial neural network models with 11 different activation functions in hidden layer were trained using 7 local optimization algorithms. The results show that the Levenberg-Marquardt algorithm was superior compared to the remaining tested local optimization methods. When comparing the 11 nonlinear transfer functions, used in hidden layer neurons, the RootSig function was superior compared to the rest of analyzed activation functions.

1. Introduction

In recent three decades, the implementations of various models based on artificial neural networks (ANN) were intensively explored in hydrological engineering. The general reviews of ANNs modeling strategies and applications with the emphases on modeling of hydrological processes are presented in [1–3]. They confirm that the class of multilayer perceptron (MLP) [4, 5] belongs to the most frequently studied ANN's models in hydrological modeling [6–9].

The MLP forms the nonlinear data driven model. According to its architecture, it is a fully connected feed-forward network, which organizes the processing units (neurons) into the layers and allows the interconnections only between neurons in two following layers. As it was proved by [10], the MLP is the universal function approximator. This important property has been widely confirmed by many hydrological studies [11–14].

Despite the positive research results of a large number of studies on MLP runoff forecasting, there is a need for clear methodological recommendations of MLP transfer function selection [15, 22–24] combined together with the training method assessment and the implementation of new training method [8, 18, 19, 25].

Main aims of presented paper are (1) to analyze the hourly flood runoff forecast on small headwater catchment with MLP-ANN models, which are based on 12 different MLP's transfer functions following the work of [15, 24], (2) to compare the 7 local optimization algorithms [5, 17, 19], and finally (3) to evaluate the MLP performance with 4 selected model evaluation measures [26, 27].

2. Material and Methods

The tested runoff prediction using the MLP-ANN models uses the set of rainfall runoff data. The MLP-ANN implementation for runoff forecast generally consists of data preprocessing, model architecture selection, MLP training, and model validation. In this section, we give a very brief description of the MLP-ANN model architecture and tested optimization schemes and datasets.

Table 1: The implemented transfer functions [15]; a is the neuron's activation, y is the neuron output.

Function name	Transfer function	Derivatives of transfer function
Logistic sigmoid (LSi)	$y(a) = \frac{1}{1+\exp(-a)}$	$y'(a) = y(a)(1-y(a))$
Hyperbolic tangent (HT)	$y(a) = \tan h(a)$	$y'(a) = 1 - y(a)^2$
Linear function (LF)	$y(a) = a$	$y'(a) = 1$
Gaussian function (GF)	$y(a) = \exp(-a^2)$	$y'(a) = -0.5y(a)a$
Inverse abs (IA)	$y(a) = \frac{a}{1+\lvert a\rvert}$	$y'(a) = \frac{1}{(1+\lvert a\rvert)^2}$
LogLog (LL)	$y(a) = \exp(-\exp(-a))$	$y'(a) = \exp(-a)y(a)$
ClogLog (CL)	$y(a) = 1 - \exp(-\exp(a))$	$y'(a) = (1-y(a))\exp(a)$
ClogLogm (CLm)	$y(a) = 1 - 2\exp(-0.7\exp(a))$	$y'(a) = 1.4\exp(a)\exp(-0.7\exp(a))$
RootSig (RS)	$y(a) = \frac{a}{1+\sqrt{1+a^2}}$	$y'(a) = \frac{1}{\left(1+\sqrt{1+a^2}\right)\sqrt{1+a^2}}$
LogSig (LS)	$y(a) = \left(\frac{1}{1+\exp(-a)}\right)^2$	$y'(a) = \frac{2\exp(-a)}{(1+\exp(-a))^3}$
Sech (SF)	$y(a) = \frac{2}{\exp(a)+\exp(-a)}$	$y'(a) = -y(a)\tan h(a)$
Wave (WF)	$y(a) = (1-a^2)\exp(-a^2)$	$y'(a) = 2a\exp(-a^2)(-2+a^2)$

Table 2: The implemented local training gradient based methods.

Training method	References
Batch propagation (BP)	[4, 5, 16]
Batch backpropagation with regularization (BP_regul)	[4, 5, 16]
Levenberg-Marquardt (LM)	[4, 17, 18]
Scaled conjugate gradient, Perry (PER)	[19–21]
Scaled conjugate gradient, Polak (POL)	[19–21]
Scaled conjugate gradient, Hestenes (HEST)	[19–21]
Scaled conjugate gradient, Fletcher (FLET)	[19–21]

2.1. MLP-ANN Model. We analyzed the MLP model with one hidden layer. The similar ANN architecture was used in a large number of hydrologically oriented studies [18, 28–31]. The studied MLP models had in total three layers of neurons, the input layer, the hidden layer, and the output layer. As proved by Hornik et al. [10], this type of artificial neural network with sufficiently a large number of neurons in the second layer can approximate with desired precision any measurable functional relationship.

The implemented MLP-ANN models had a general form

$$R_f = v_0 + \sum_{j=1}^{N_{\text{hd}}} v_j f\left(w_{0j} + \sum_{i=1}^{N_{\text{in}}} w_{ji} x_i\right), \tag{1}$$

where the R_f is a network output, that is, flood runoff forecast for given time interval, x_i is network input for input layer neuron i, N_{in} is the number of MLP inputs, the w_{ji} is the weight of i input to j hidden layer neuron, $f()$ is the activation function constant for all hidden layer neurons, N_{hd} is the number of hidden neurons, v_j is the weight for output from hidden neuron j, and w_{0j}, v_0 are neuron biases [2–4, 18, 25, 31].

2.1.1. MLP-ANN Transfer Functions. The type of activation function together with network architecture influences the generalization of neural network. Imrie et al. [32] empirically confirmed that the transfer function bounding influences the ANN generalization and hydrological extreme simulations during runoff forecast. Following the work of [15], we implemented the 12 different types of transfer functions, and 11 of them were tested in hidden neuron layer of analyzed MLP-ANN models. Table 1 provides their list.

The activation functions type combined with specific type of training methods influences the average performance of leaning algorithm and computing time [15, 24]. For example, the Bishop [4] pointed out that the implementation of hyperbolic function speeds up the training process compared to the use of logistic sigmoid.

2.1.2. MLP-ANN Local Optimization Methods. We selected 7 gradient based local optimization methods. Table 2 shows their list together with their references. All MLP-ANN optimization was performed using the batch learning mode [4].

All tested gradient local search methods (except BP_regul) minimized the error function represented as a the sum of square of residuals

$$E_r = \frac{1}{2}\sum_{i=1}^{N} r_i^2, \tag{2}$$

and the residuals $r_i = R_o[i] - R_f[i]$ were defined as differences between observed R_o and computed $R_f[i]$ flood runoff.

The two first order local training methods are represented by the standard backpropagation and backpropagation with regularization term. Both backpropagation methods implement the following modification: constant learning rate and momentum parameter. The BP_regul used the regularization term, which penalizes the size of estimated weights, and the error function is defined as

$$E = \beta E_r + \alpha \frac{1}{2} \sum_{l=1}^{Nw} w_l^2, \quad (3)$$

where the Nw is a total number of MLP-ANN weights w_l. The hyperparameters α and β were constant within the standard backpropagation with the regularization term [4, 16].

The scaled conjugate gradient methods are built together with safe line search based on golden section search combined with bracketing the minima [33, 34]. The implementation enables the restarting during the iteration search based on the recommendations of [21, 35]. The restarting controls the prescribed number of iterations or gradient norm. The implementation of scaled conjugate gradient uses four different updating schemes in detail described by [19, 36].

All gradient based methods apply the standard backpropagation algorithm for the estimation derivatives of the objective function with respect to weights [37]. The Levenberg-Marquardt methods approximate the Hessian matrix using first order derivatives neglecting the terms with the second order derivatives [4, 17].

2.1.3. *The MLP-ANN Performance.*

We based the evaluation of MLP-ANN model simulations of training, testing, and validation datasets on the following statistics [26, 27, 38]:

mean absolute error (MAE)

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| R_o[i] - R_f[i] \right|, \quad (4)$$

Nash Sutcliffe efficiency (NS)

$$\mathrm{NS} = 1 - \frac{\sum_{i=1}^{n} \left(R_o[i] - R_f[i] \right)^2}{\sum_{i=1}^{n} \left(R_o[i] - \overline{R_o} \right)^2}, \quad (5)$$

fourth root mean quadrupled error (R4MS4E)

$$\mathrm{R4MS4E} = \sqrt[4]{\frac{1}{n} \sum_{i=1}^{n} \left(R_o[i] - R_f[i] \right)^4}, \quad (6)$$

persistency index (PI)

$$\mathrm{PI} = 1 - \frac{\sum_{i=1}^{n} \left(R_o[i] - R_f[i] \right)^2}{\sum_{i=1}^{n} \left(R_o[i] - R_o[i - \mathrm{LAG}] \right)^2}, \quad (7)$$

where the n represents the total number of time intervals to be predicted, the $\overline{R_o}$ is the average of observed flood runoff R_o, and LAG is the time shift describing last observed flood runoff $R_o[i - \mathrm{LAG}]$.

2.1.4. *The PONS2train.*

The tested MLP-ANN models were implemented using the PONS2train software application. The PONS2train is software written in C++ programing language, whose main goal is to test MLP models with different architectures. The software application uses the LAPACK, BLAS, and ARMADILLO C++ linear algebra libraries [39–41]. The application is freely distributed upon a request to authors.

The PONS2train has additional features: the weight initialization can be performed using two methods. The first one follows the work of Nguyen and Widrow [42], while the second one uses random initialization coming from the uniform distribution.

Giustolisi and Laucelli [25] extensively studied the eight methods for improving the MLP performance and generalization. One of them the early stopping is incorporated in designed application. Following the recommendations of Stäger and Agarwal [43], the PONS2train also controls the avoiding of the neuron's saturation.

The important PONS2train implementation feature is the multirun and ensemble simulation. Its software design also enables further multimodel or hybrid MLP extensions [29, 44].

The software design also allows the comparative analysis of MLP's architectures with or without bias neurons in layers. The PONS2train also enables the comparison of MLP trained on shuffled and unshuffled dataset. The shuffling of data patterns follows the random permutation algorithm of Durstenfeld [45].

The MLP datasets are scaled using two methods. Both methods scale the analyses datasets into the interval $(0, k)$ with arbitrary chosen upper bound $k \leq 1$. The nonlinear scaling provides the transformed data D_{trans} obtained from original data D_{orig} using exponential transformation

$$D_{\mathrm{trans}} = \left(1 - \exp\left(-\gamma D_{\mathrm{orig}} \right) \right), \quad (8)$$

where the γ is a control parameter. The second scaling methods is a linear one.

2.2. *The Dataset Description.*

We explored the MLP-ANN models using the rainfall and runoff time series data obtained from 10-year monitoring in the Modrava catchment 0.17 km^2. The experimental watershed was established in 1998 in upper parts of Bohemian Forest National Park. The basin belongs to the set of testbeds designed to monitor the hydrological behavior of headwater forested catchments. The watershed description shows that of Pavlasek et al. [46].

The forest cover is a clearing with young artificially planted forest combined with an undergrowth of herbs (mainly *Calamagrostis villosa*, *Avenella flexuosa*, *Scirpus sylvaticus,* and *Vaccinium myrtillus*) and bryophyte (*Polytrichastrum formosum*, *Dicranum scoparium,* and *Sphagnum girgensohnii*). A small part of the catchment (less than 10%) is covered by 40-year-old forest. The bark beetle calamity removed the original forest cover. Catchment bedrock is formed by granite, migmatite, and paragneiss covered by Haplic Podzols with depths of up to 0.9 m. The mean runoff coefficient is 0.2, mean daily runoff 1.2 mm.

Table 3: The rainfall runoff events characteristics: T_o outflow duration, Q_p flood peak, H_o runoff depth, T_r rainfall duration, I_p peak rainfall intensity, H_r rainfall depth, RC runoff coefficient, $H_{r\text{-}5D}$ antecedent rainfall depth 5 days before event, RET_e basin retention during R-R event, and RET basin retention of the whole catchment before event.

R-R event	T_o [hour]	Q_p [m^3·s^{-1}·km^{-2}]	H_o [mm]	T_r [hour]	I_p [mm·hour^{-1}]	H_r [mm]	RC [—]	$H_{r\text{-}5D}$ [mm]	RET_e [mm]	RET [mm]
M2_19980915-11	125	0.559	43.8	102	8.2	164	0.27	31	120.2	151.2
M2_19981027-23	100	0.902	51.8	52	9.0	128	0.4	29	76.6	105.6
M2_20010908-15	158	0.322	29.9	123	4.6	105.2	0.28	19.6	75.3	94.9
M2_20011108-11	90	0.405	22.1	45	5.8	73.6	0.3	3.2	51.5	54.7
M2_20040923-22	68	0.448	14.8	55	7.4	110.8	0.13	7.0	96.0	103.0
M2_20060527-03	77	1.093	67.2	47	12.4	156.0	0.43	21.8	88.8	110.6
M2_20070119-03	99	0.788	35.0	73	8.8	73.4	0.48	14.4	38.4	52.8
M2_20070906-18	52	0.369	14.1	52	5.6	68.6	0.21	39.6	54.5	94.1
M2_20080808-01	18	1.14	11.7	2	73.6	85.6	0.14	8.2	73.9	82.1
Mean	87.4	0.667	32.3	61.2	15.04	107.24	0.29	19.3	75.0	94.3
St. dev.	40.7	0.318	19.1	35.0	22.08	35.86	0.12	12.3	25.0	29.9

Table 4: The selected quantiles of empirical distribution functions on runoff and rainfall depths for training, testing, and validation data sets.

	Minimum	1st Quartile	Median	Mean	2nd Quartile	Maximum	St. dev.
Runoff depth [mm·hour^{-1}]							
Training runoff	0.000	0.0154	0.0194	0.1031	0.0592	3.828	0.326
Testing runoff	0.00380	0.00760	0.0231	0.09395	0.0622	4.107	0.249
Validation runoff	0.000	0.0040	0.0270	0.1008	0.0591	3.250	0.307
Rainfall depth [mm·hour^{-1}]							
Training rainfall	0.2	0.4	0.8	1.365	1.700	12.400	1.658
Testing rainfall	0.2	0.2	0.6	1.439	1.400	73.9	4.485
Validation rainfall	0.2	0.2	0.8	1.575	2.200	15.800	1.900

Table 5: The values of statistical measures for the benchmark SLMB model.

	Calibration dataset	Testing dataset	Validation dataset
PI [—]	0.36	0.20	0.00
NS [—]	0.96	0.82	0.96
MAE [mm·hour^{-1}]	0.03	0.03	0.02
R4MS4E [mm·hour^{-1}]	0.17	0.41	0.16

The most significant nine rainfall runoff events observed in hourly time step were selected from 10-year measurement. The flood runoff prediction was analyzed via proposed MLP-ANN models. The characteristics of flood events are described in Table 3. All floods events were complemented with the periods of 5 preceding days. The rainfall runoff events were divided into the nonoverlapping training, testing, and validation dataset.

The division of flood events into the datasets was made with respect to the similarity of empirical distribution functions of training, testing, and validation datasets and to their independence. The empirical distribution functions were estimated using the quantile estimation method, which was specifically developed for the description of hydrological time series (for detailed information see [47]). The selected quantiles of all datasets are shown in Table 4. The quantiles show that the distinctions of the information in training, testing, and validation datasets are not significant.

3. Results and Discussion

We tested MLP-ANN models with 4 MLP architectures; they are different according to the number of hidden layer neurons $N_{\mathrm{hd}} = 3, 4, 5, 6$. For each MLP architecture, we prepared 11 types of MLP-ANN models according to the type of hidden layer activation function (AF) (see Table 1). Each of them was trained with 7 training algorithms (TA) (see Table 2).

All MLP-ANN datasets consisted of all available pairs of four inputs and one output. The inputs x_i were one runoff interval $Q[t-1]$ and three rainfall intervals $P[t-1]$, $P[t-2]$, and $P[t-3]$ and output R_f was formed from one runoff output $Q[t]$ for all available time intervals $[t]$. The total number of training pairs was 1270, the testing input-output datasets were 1221, and validation datasets were 1423.

Although there are suitable methodologies for selection of the proper input vector for MLP model, that is, [48–50], we based our flood forecast on small number of previous rainfall intervals and one previous runoff mainly due to fast hydrological response of analyzed watershed. The datasets were transformed using the nonlinear exponential transformation.

Table 6: The results of persistency index on analyzed optimization algorithms, the best models are marked with bold fonts. The MLP architecture with $N_{\text{hd}} = K$ labeled as 4-K-1, the best models are marked with bold font.

	ntrain	ntest	nval	PI_train	PI_test	PI_val	mPI_train	mPI_test	mPI_val
4-3-1									
FLET	298	225	111	0.67	0.52	0.14	0.33	0.24	0.07
HEST	114	91	26	0.64	0.57	0.14	0.26	0.23	0.07
PER	**1045**	**716**	**408**	0.72	0.56	**0.32**	0.40	0.26	0.08
POL	89	63	16	0.52	0.49	0.26	0.24	0.20	0.08
LM	816	511	130	**0.84**	**0.61**	0.25	**0.48**	**0.27**	**0.09**
BP	505	371	171	0.63	0.53	0.23	0.34	0.25	0.07
BP_regul	486	229	99	0.63	0.54	0.26	0.26	0.24	0.09
4-4-1									
FLET	395	292	151	0.63	0.54	0.28	0.33	0.25	0.08
HEST	186	130	21	0.64	0.49	0.16	0.25	0.21	0.06
PER	**1107**	**755**	**416**	0.75	0.60	0.22	0.43	0.27	0.09
POL	112	92	22	0.66	0.52	0.21	0.27	0.20	0.08
LM	819	550	119	**0.88**	**0.61**	**0.25**	**0.54**	**0.29**	**0.09**
BP	579	417	216	0.66	0.57	0.21	0.38	0.27	0.08
BP_regul	578	251	99	0.72	0.55	0.25	0.30	0.24	0.08
4-5-1									
FLET	413	288	157	0.77	0.52	0.18	0.33	0.25	0.08
HEST	217	165	39	0.62	0.47	0.16	0.28	0.22	0.07
PER	**1168**	**787**	**453**	0.77	0.56	0.31	0.43	0.27	0.09
POL	117	86	15	0.63	0.47	0.12	0.25	0.24	0.07
LM	859	570	91	**0.89**	**0.61**	**0.32**	**0.55**	**0.28**	**0.09**
BP	606	451	225	0.68	0.55	0.21	0.37	0.25	0.08
BP_regul	643	291	143	0.71	0.61	0.29	0.31	0.24	0.10
4-6-1									
FLET	451	342	180	0.68	0.56	0.21	0.33	0.25	0.07
HEST	218	178	42	0.66	0.51	0.14	0.26	0.21	0.06
PER	**1181**	**838**	**468**	0.82	0.59	0.23	0.43	0.28	0.08
POL	153	114	31	0.68	0.53	0.19	0.28	0.20	0.09
LM	839	579	86	**0.89**	**0.61**	0.19	**0.55**	**0.29**	0.06
BP	621	484	256	0.67	0.59	0.23	0.39	0.26	0.07
BP_regul	679	306	126	0.66	0.59	**0.24**	0.32	0.25	**0.09**

Each training algorithm was repeated 150 times. The random initialization of network weights was performed by the method of [42]. Each optimization multirun used the same values of 150 mutually different initial random vectors of weights, in order to ensure that the comparison of performances of optimization algorithms was based on similar random weights initializations.

3.1. The Benchmark Model. The flood forecast was simulated using the benchmark model based on simple linear model—SLMB. The SLMB parameters were calculated using the ordinary least squares. Table 5 shows results obtained from the simulation of SLMB benchmark model.

Since the benchmark model provides the single simulation and one value for all tested model comparison measures, we compared the results of SLMB with results of the best selected single MLP-ANN models. In model ensemble, we found MLP-ANN models, which were superior compared SLMB.

For example, the model performance based on the PI index shows all MLP-ANN provided models, which were superior compared to SLMB (see the results of Table 6). The highest differences between the best PI values of ANN and PI of SLMB were obtained on MLP-ANN trained using LM algorithm on training dataset ($\text{PI}_{\text{ANN}} - \text{PI}_{\text{SLMB}} = 0.53$). The LM and PER training algorithms provided models with the highest values of PI on testing and validation datasets ($\text{PI}_{\text{ANN}} - \text{PI}_{\text{SLMB}} = 0.41$, resp., $\text{PI}_{\text{ANN}} - \text{PI}_{\text{SLMB}} = 0.32$).

These conclusions are in agreement with the values of remaining model performance measures—MAE, NS, and R4MS4E (see Table 7). The LM and BP_regul were superior in terms of differences with SLBM according to the MAE and R4MS4E. The LM and PER were superior compared to SLMB for NS values on training, testing, and validation datasets.

Table 7: The MAE, NS, and R4M4E—trainings algorithms.

	$\mathrm{MAE}_{\mathrm{train}}$	$\mathrm{MAE}_{\mathrm{test}}$	$\mathrm{MAE}_{\mathrm{val}}$	$\mathrm{NS}_{\mathrm{train}}$	$\mathrm{NS}_{\mathrm{test}}$	$\mathrm{NS}_{\mathrm{val}}$	$\mathrm{R4MS4E}_{\mathrm{train}}$	$\mathrm{R4MS4E}_{\mathrm{test}}$	$\mathrm{R4MS4E}_{\mathrm{val}}$
4-3-1									
FLET	0.017	0.018	0.016	0.98	0.89	0.97	0.131	0.35	0.14
HEST	0.015	0.020	0.017	0.98	0.90	0.97	0.142	0.34	0.14
PER	0.012	0.017	0.015	0.98	0.90	**0.98**	0.124	0.34	0.12
POL	0.016	0.017	0.015	0.97	0.88	0.97	0.149	0.35	0.13
LM	**0.010**	0.015	**0.015**	**0.99**	**0.91**	0.97	**0.093**	**0.33**	0.12
BP	0.014	0.017	0.016	0.98	0.89	0.97	0.133	0.35	0.13
BP_regul	0.015	**0.014**	0.015	0.98	0.89	0.97	0.129	0.35	**0.11**
4-4-1									
FLET	0.0172	0.018	0.016	0.98	0.89	0.97	0.14	0.35	0.13
HEST	0.0154	0.020	0.016	0.98	0.88	0.97	0.15	0.35	0.14
PER	0.0125	0.017	0.015	0.98	0.91	0.97	0.13	0.34	0.13
POL	0.0138	0.019	0.015	0.98	0.89	0.97	0.15	0.35	0.13
LM	**0.0077**	**0.014**	**0.014**	**0.99**	**0.91**	0.97	**0.09**	**0.33**	**0.12**
BP	0.0140	0.018	0.016	0.98	0.90	0.97	0.13	0.35	0.13
BP_regul	0.0145	0.016	0.015	0.98	0.90	0.97	0.12	0.34	0.12
4-5-1									
FLET	0.0121	0.016	0.016	0.99	0.89	0.97	0.117	0.35	0.13
HEST	0.0159	0.020	0.017	0.98	0.88	0.97	0.137	0.35	0.12
PER	0.0129	0.015	0.015	0.99	0.90	0.98	0.115	0.34	0.12
POL	0.0159	0.022	0.018	0.98	0.88	0.97	0.143	0.35	0.14
LM	**0.0072**	0.015	**0.013**	**0.99**	**0.91**	**0.98**	**0.086**	**0.33**	**0.12**
BP	0.0139	0.016	0.016	0.98	0.90	0.97	0.122	0.35	0.13
BP_regul	0.0132	**0.014**	0.014	0.98	0.91	0.97	0.132	0.34	0.12
4-6-1									
FLET	0.0150	0.016	0.016	0.98	0.90	0.97	0.132	0.35	0.13
HEST	0.0139	0.018	0.017	0.98	0.89	0.97	0.134	0.35	0.14
PER	0.0117	0.016	0.014	0.99	0.91	0.97	0.104	0.34	0.13
POL	0.0162	0.018	0.018	0.98	0.89	0.97	0.136	0.35	0.14
LM	**0.0076**	**0.015**	**0.014**	**0.99**	**0.91**	0.97	**0.087**	**0.32**	0.14
BP	0.0139	0.017	0.015	0.98	0.90	0.97	0.131	0.35	0.14
BP_regul	0.0130	0.015	0.014	0.98	0.90	0.97	0.130	0.34	**0.12**

The similar results can be found, when comparing the results of SLMB with the best MLP-ANN models organized in terms of different transfer functions. The highest differences of PI values were on training dataset for MLP-ANN with LL transfer function ($\mathrm{PI}_{\mathrm{ANN}} - \mathrm{PI}_{\mathrm{SLMB}} = 0.48$), for testing dataset on RS transfer function ($\mathrm{PI}_{\mathrm{ANN}} - \mathrm{PI}_{\mathrm{SLMB}} = 0.41$) and for validation dataset on LL transfer function ($\mathrm{PI}_{\mathrm{ANN}} - \mathrm{PI}_{\mathrm{SLMB}} = 0.31$). These were calculated for MLP-ANN with transfer functions, which were successful in more than 10% of simulations on validation dataset.

Those results were confirmed by the values of MAE, NS, and R4MS4E obtained for the best model of a simulation ensemble. The RS transfer function provided the best results in terms of differences between$\mathrm{MAE}_{\mathrm{ANN}} - \mathrm{MAE}_{\mathrm{SLMB}} = (-0.019, -0.015, -0.005)$, $\mathrm{NS}_{\mathrm{ANN}} - \mathrm{NS}_{\mathrm{SLMB}} = (0.02, 0.09, 0.01)$, and $\mathrm{R4MS4E}_{\mathrm{ANN}} - \mathrm{R4MS4E}_{\mathrm{SLMB}} = (-0.111, -0.06, -0.03)$ on training, testing, and validation datasets.

3.2. The Optimization Algorithms. The results of MLP-ANN models were explained through the values of model performance measures, which are shown in Tables 6 and 7. All training computations controlled the neuron's saturation using the method of Stäger and Agarwal [43]. The parameters of TA (i.e., number of epochs, learning rate, etc.) were selected in such a way that the number of MLP-ANN evaluations was similar in all tested TA.

Table 6 shows the results of persistency index, which was used as a main reference index, since the PI compares the model with last observed information [38]. The best TA according to the number of successful models with $\mathrm{PI} > 0$ was the PER (the scaled conjugate gradient method with Perry updating formula). The highest number of successfully trained models was found on MLP with $N_{\mathrm{hd}} = 6$ (see the ntrained = 1181, ntest = 838, and nval = 468 in Table 6).

When comparing the performance of TA according to the best single value of PI (see columns PI_train, PI_test,

Table 8: The results of persistency index on tested transfer functions, the best models are marked with bold font.

	ntrain	ntest	nval	PI_train	PI_test	PI_val	mPI_train	mPI_test	mPI_val
4-3-1									
CL	275	225	110	0.80	0.46	0.24	0.35	0.24	**0.08**
CLm	555	381	**207**	0.70	**0.56**	0.26	0.37	0.25	0.08
HT	489	283	148	0.72	0.56	**0.27**	**0.42**	0.26	0.08
LL	355	273	139	**0.84**	0.55	0.21	0.35	0.25	0.08
RS	**566**	**417**	198	0.73	0.56	0.22	0.40	**0.27**	0.08
4-4-1									
CL	290	241	129	0.74	0.52	0.17	0.35	0.27	0.08
CLm	**615**	402	**237**	0.75	0.56	0.24	0.38	0.27	0.08
HT	578	351	152	0.75	0.56	0.25	**0.44**	0.27	0.08
LL	384	300	157	**0.83**	0.57	**0.28**	0.35	0.26	**0.09**
RS	608	**475**	209	0.75	**0.61**	0.25	0.42	**0.28**	0.08
4-5-1									
CL	311	256	144	0.77	0.53	0.24	0.35	0.26	0.09
CLm	632	437	**245**	0.70	0.58	0.21	0.39	0.26	0.08
HT	574	321	147	0.75	0.61	0.31	**0.45**	0.27	**0.09**
LL	432	337	187	**0.79**	0.56	**0.31**	0.35	0.25	0.08
LS	377	291	115	0.74	0.57	0.21	0.35	0.25	0.08
RS	**659**	**517**	242	0.74	**0.61**	0.29	0.42	**0.27**	0.09
4-6-1									
CL	319	269	145	0.77	0.56	0.21	0.34	0.27	**0.08**
CLm	654	471	259	0.69	0.59	**0.24**	0.39	0.27	0.07
HT	601	361	162	0.71	0.59	0.20	**0.45**	0.27	0.08
LL	437	358	185	**0.75**	0.53	0.17	0.35	0.27	0.08
LS	391	324	114	0.71	0.55	0.20	0.35	0.22	0.07
RS	**651**	**539**	**263**	0.72	**0.61**	0.22	0.42	**0.28**	0.08

and PI_val in Table 6) and the average performance of best MLP-ANN models on PI (see columns mPI_train, mPI_test, and mPI_val in Table 6), the Levenberg-Marquardt algorithm was mostly superior compared to all remaining TA, except for three cases, when the PER and BP_regul were better on validation datasets for MLP with $N_{\mathrm{hd}} = 3, 6$ on best single value of PI and for average of mPI_val for $N_{\mathrm{hd}} = 6$.

Table 7 displays the results of best models for remaining statistical measures of MLP-ANN models trained on tested TA. Only three algorithms were superior at least for one architecture of MLP and on one dataset. They are LM, PER, and BP_regul. Again, the LM was mostly superior compared to the other tested TA. The differences between results of LM and PER and BP_regul were very small.

The best values of NS were in agreement with values of PI (see, e.g., the PER on MLP with $N_{\mathrm{hd}} = 3$). The BP_regul was better in terms of the length of residuals for MAE_test on MLP ANN models with $N_{\mathrm{hd}} = 3, 5$. Also when comparing the simulation of peak flow in terms of R4MS4E, the BP_regul was better on MLP with $N_{\mathrm{hd}} = 3, 6$ for validation dataset.

Our finding are in agreement with results on runoff forecast of Piotrowski and Napiorkowski [18], who compared the Levenberg-Marquardt approach even with more robust global optimization schemes, and found that the LM provides comparable results with MLP trained using the selected evolutionary computation methods.

3.3. The Transfer Functions. The results of PI, MAE, NS, and R4MS4E are shown in Tables 8 and 9. The PI has again served as a reference. We trained the MLP with all AF listed in Table 1. Tables 8 and 9 show the results of AF for MLP-ANN models, which were successful in more than 10% of simulations on validation dataset.

When comparing the absolute values of number of MLP-ANN models with PI $>$ 0, the models with two AF (RS and CLm) were superior compared to MLP models with remaining 9 AFs. The MLP with RS provided the larger number of better models in terms of PI value on 8 datasets, while the MLP with CLm transfer function was successful on 4 datasets.

RS was also the most successful TA on training dataset at MLPs with $N_{\mathrm{hd}} = 4, 5, 6$ (note that for $N_{\mathrm{hd}} = 3$ the differences in PI between RS and CLm are almost insignificant). The LL also provided good results on training dataset (for all tested values of N_{hd}) and on validation data for $N_{\mathrm{hd}} = 4, 5$.

The mean performances based on arithmetical means of PI values of best models showed that three AFs were superior compared to remaining 8 AFs (see mPI_train, mPI_test, and

Table 9: The MAE, NS, and R4MS4E on activation functions, the best models are marked with bold font.

	$\text{MAE}_{\text{train}}$	MAE_{test}	MAE_{val}	NS_{train}	NS_{test}	NS_{val}	$\text{R4MS4E}_{\text{train}}$	$\text{R4MS4E}_{\text{test}}$	$\text{R4MS4E}_{\text{val}}$
4-3-1									
CL	0.10	0.10	0.10	0.72	0.52	0.68	0.40	0.63	0.42
CLm	0.06	0.06	0.06	0.82	**0.65**	**0.81**	0.34	0.71	0.36
HT	0.06	0.06	0.06	0.81	0.60	0.79	0.34	0.74	0.37
LL	0.08	0.08	0.08	0.76	0.57	0.73	0.34	0.64	0.37
RS	**0.06**	**0.06**	**0.05**	**0.82**	0.65	0.80	**0.31**	**0.61**	**0.34**
4-4-1									
CL	0.10	0.10	0.10	0.71	0.53	0.68	0.40	0.62	0.41
CLm	0.06	0.06	0.05	**0.83**	0.66	**0.82**	0.30	0.67	0.33
HT	0.06	0.06	0.06	0.83	0.63	0.81	0.32	0.71	0.36
LL	0.08	0.08	0.07	0.79	0.59	0.75	0.31	0.62	0.35
RS	**0.05**	**0.05**	**0.05**	0.82	**0.67**	0.81	**0.29**	**0.59**	**0.32**
4-5-1									
CL	0.10	0.10	0.10	0.72	0.56	0.68	0.41	0.62	0.42
CLm	0.05	0.05	0.05	0.84	0.68	**0.84**	0.29	0.69	0.32
HT	0.05	0.05	0.05	**0.84**	0.64	0.83	0.29	0.70	0.33
LL	0.07	0.07	0.07	0.82	0.61	0.78	0.30	0.59	0.33
LS	0.07	0.07	0.06	0.82	0.60	0.79	0.28	**0.56**	0.31
RS	**0.05**	**0.05**	**0.05**	0.84	**0.69**	0.83	**0.28**	0.58	**0.30**
4-6-1									
CL	0.10	0.10	0.10	0.75	0.59	0.69	0.42	0.60	0.42
CLm	**0.05**	**0.05**	**0.05**	0.84	0.69	0.83	0.28	0.65	0.31
HT	0.05	0.05	0.05	**0.85**	0.65	**0.83**	0.28	0.65	0.32
LL	0.07	0.07	0.07	0.81	0.62	0.78	0.29	0.57	0.32
LS	0.07	0.07	0.06	0.82	0.60	0.78	**0.28**	**0.55**	0.31
RS	0.05	0.05	0.05	0.84	**0.70**	0.82	0.28	0.55	**0.30**

mPI_val in Table 8). They were CL, HT, and RS MLP ANN models. Their differences of PI were again very small.

Table 9 shows the averages of MAE, NS, and R4MS4E on set of tested models. The results point out that the RS transfer function provided in summary superior values compared to rest of tested AF. The CLm, HT, and LS activation functions were on some datasets better in terms of mean values of tested statistical measures but the differences between the RS MLP ANN models were again negligible.

When reflecting the results of da S. Gomes et al. [15], who recommended the CL, CLm, and LL functions on MLP ANN models, we point out the ability of the MLP models with RS to improve the flood runoff forecast.

Our findings on the selection of suitable AF on MLP ANN models recommend that different AF should be tested during the implementation of MLP models for flood runoff forecast.

4. Conclusions

During the extensive computational test, we trained in total the 46200 models of multilayer perceptron with one hidden layer. The main aim of computational exercise was the evaluation of the impacts of the transfer function selection and the test of selected local optimization schemes on flood runoff forecast.

Using the rainfall runoff data of nine of the most significant flood events, we analyzed the short term runoff forecast on small watershed with fast hydrological response. The developed MLP ANN models were able to predict flood runoff using the records of past rainfall and runoff from the basin.

When comparing the tested MLP ANN models with benchmark simple linear model, the developed MLP models were superior in terms of values of model performance measures compared to the SLMB.

The PONS2Train software application was developed for the purposes of the evaluation of MLP-ANN models with different architectures and for providing the simulations of neural network flood forecast.

When analyzing the 7 different gradient oriented optimization schemes we found that the Levenberg-Marquardt algorithm was superior compared to the tested set of scaled conjugate gradient methods and two first order local optimization schemes.

When analyzing the 11 different transfer functions used in hidden neurons we found that the RootSig function was according to the values of four model performance measures most promising activation function in terms of flood runoff forecast.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] H. R. Maier and G. C. Dandy, "Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications," *Environmental Modelling and Software*, vol. 15, no. 1, pp. 101–124, 2000.

[2] C. W. Dawson and R. L. Wilby, "Hydrological modelling using artificial neural networks," *Progress in Physical Geography*, vol. 25, no. 1, pp. 80–108, 2001.

[3] H. R. Maier, A. Jain, G. C. Dandy, and K. P. Sudheer, "Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions," *Environmental Modelling and Software*, vol. 25, no. 8, pp. 891–909, 2010.

[4] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, New York, NY, USA, 1995.

[5] R. D. Reed and R. J. Marks, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*, MIT Press, Cambridge, Mass, USA, 1998.

[6] A. W. Minns and M. J. Hall, "Artificial neural networks as rainfall-runoff models," *Hydrological Sciences Journal*, vol. 41, no. 3, pp. 399–417, 1996.

[7] H. K. Cigizoglu, "Estimation, forecasting and extrapolation of river flows by artificial neural networks," *Hydrological Sciences Journal*, vol. 48, no. 3, pp. 349–361, 2003.

[8] N. J. de Vos and T. H. M. Rientjes, "Constraints of artificial neural networks for rainfall-runoff modelling: trade-offs in hydrological state representation and model evaluation," *Hydrology and Earth System Sciences*, vol. 9, no. 1-2, pp. 111–126, 2005.

[9] G. Napolitano, F. Serinaldi, and L. See, "Impact of EMD decomposition and random initialisation of weights in ANN hindcasting of daily stream flow series: an empirical examination," *Journal of Hydrology*, vol. 406, no. 3-4, pp. 199–214, 2011.

[10] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[11] N. J. de Vos and T. H. M. Rientjes, "Multiobjective training of artificial neural networks for rainfall-runoff modeling," *Water Resources Research*, vol. 44, no. 8, 2008.

[12] E. Toth and A. Brath, "Multistep ahead streamflow forecasting: role of calibration data in conceptual and neural network modeling," *Water Resources Research*, vol. 43, no. 11, 2007.

[13] M. P. Rajurkar, U. C. Kothyari, and U. C. Chaube, "Modeling of the daily rainfall-runoff relationship with artificial neural network," *Journal of Hydrology*, vol. 285, no. 1–4, pp. 96–113, 2004.

[14] C. M. Zealand, D. H. Burn, and S. P. Simonovic, "Short term streamflow forecasting using artificial neural networks," *Journal of Hydrology*, vol. 214, no. 1–4, pp. 32–48, 1999.

[15] G. S. da S.Gomes, T. B. Ludermir, and L. M. M. R. Lima, "Comparison of new activation functions in neural network for forecasting financial time series," *Neural Computing & Applications*, vol. 20, no. 3, pp. 417–439, 2011.

[16] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, New York, NY, USA, 2003.

[17] M. Hagan and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[18] A. P. Piotrowski and J. J. Napiorkowski, "Optimizing neural networks for river flow forecasting—evolutionary computation methods versus the levenberg-marquardt approach," *Journal of Hydrology*, vol. 407, no. 1–4, pp. 12–27, 2011.

[19] A. E. Kostopoulos and T. N. Grapsa, "Self-scaled conjugate gradient training algorithms," *Neurocomputing*, vol. 72, no. 13–15, pp. 3000–3019, 2009.

[20] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[21] M. J. D. Powell, "Restart procedures for the conjugate gradient method," *Mathematical Programming*, vol. 12, no. 2, pp. 241–254, 1977.

[22] A. Shamseldin, A. Nasr, and K. O'Connor, "Comparsion of different forms of the multi-layer feed-forward neural network method used for river flow forecasting," *Hydrology and Earth System Sciences*, vol. 6, no. 4, pp. 671–684, 2002.

[23] R. R. Shrestha, S. Theobald, and F. Nestmann, "Simulation of flood flow in a river system using artificial neural networks," *Hydrology and Earth System Sciences*, vol. 9, no. 4, pp. 313–321, 2005.

[24] H. Yonaba, F. Anctil, and V. Fortin, "Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting," *Journal of Hydrologic Engineering*, vol. 15, no. 4, pp. 275–283, 2010.

[25] O. Giustolisi and D. Laucelli, "Improving generalization of artificial neural networks in rainfall-runoff modelling," *Hydrological Sciences Journal*, vol. 50, no. 3, pp. 439–457, 2005.

[26] C. W. Dawson, R. J. Abrahart, and L. M. See, "HydroTest: further development of a web resource for the standardised assessment of hydrological models," *Environmental Modelling & Software*, vol. 25, no. 11, pp. 1481–1482, 2010.

[27] C. W. Dawson, R. J. Abrahart, and L. M. See, "HydroTest: a web-based toolbox of evaluation metrics for the standardised assessment of hydrological forecasts," *Environmental Modelling and Software*, vol. 22, no. 7, pp. 1034–1052, 2007.

[28] A. Y. Shamseldin, "Application of a neural network technique to rainfall-runoff modelling," *Journal of Hydrology*, vol. 199, no. 3-4, pp. 272–294, 1997.

[29] W. Wang, P. H. A. J. M. V. Gelder, J. K. Vrijling, and J. Ma, "Forecasting daily streamflow using hybrid ANN models," *Journal of Hydrology*, vol. 324, no. 1–4, pp. 383–399, 2006.

[30] B. Pang, S. Guo, L. Xiong, and C. Li, "A nonlinear perturbation model based on artificial neural network," *Journal of Hydrology*, vol. 333, no. 2–4, pp. 504–516, 2007.

[31] A. P. Piotrowski and J. J. Napiorkowski, "A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling," *Journal of Hydrology*, vol. 476, pp. 97–111, 2013.

[32] C. Imrie, S. Durucan, and A. Korre, "River flow prediction using artificial neural networks: generalisation beyond the calibration range," *Journal of Hydrology*, vol. 233, no. 1–4, pp. 138–153, 2000.

[33] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++: The Art of Scientific Computing*, Cambridge University Press, 2002.

[34] T. Masters, *Practical Neural Network Recipes in C++*, Morgan Kaufmann, 1st edition, 1993.

[35] N. Andrei, "Scaled conjugate gradient algorithms for unconstrained optimization," *Computational Optimization and Applications. An International Journal*, vol. 38, no. 3, pp. 401–416, 2007.

[36] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, pp. 149–154, 1964.

[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[38] P. K. Kitanidis and R. L. Bras, "Real-time forecasting with a conceptual hydrologic model. 2: applications and results," *Water Resources Research*, vol. 16, no. 6, pp. 1034–1044, 1980.

[39] E. Anderson, Z. Bai, J. Dongarra et al., "Lapack: a portable linear algebra library for high-performance computers," in *Proceedings of the ACM/IEEE conference on Super- computing*, pp. 2–11, IEEE Computer Society Press, Los Alamitos, Calif, USA, November 1990.

[40] L. S. Blackford, J. Demmel, J. Dongarra et al., "An updated set of basic linear algebra subprograms (BLAS)," *ACM Transactions on Mathematical Software*, vol. 28, no. 2, pp. 135–151, 2002.

[41] C. Sanderson, "Armadillo: an open source C++ linear algebra library for fast prototyping and computationally intensive experiments," Tech. Rep., NICTA, Sydney, Australia, 2010.

[42] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of adaptive weights," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '90)*, vol. 1–3, pp. C21–C26, International Neural Network Society, San Diego, Calif, USA, June 1990.

[43] F. Stäger and M. Agarwal, "Three methods to speed up the training of feedforward and feedback perceptrons," *Neural Networks*, vol. 10, no. 8, pp. 1435–1443, 1997.

[44] Z. Huo, S. Feng, S. Kang, G. Huang, F. Wang, and P. Guo, "Integrated neural networks for monthly river flow estimation in arid inland basin of Northwest China," *Journal of Hydrology*, vol. 420-421, pp. 159–170, 2012.

[45] R. Durstenfeld, "Algorithm 235: random permutation," *Communications of the ACM*, vol. 7, no. 7, p. 420, 1964.

[46] J. Pavlasek, M. Tesar, P. Maca et al., "Ten years of hydrological monitoring in upland microcatchments in the bohemian forest, Czech Republic," in *Status and Perspectives of Hydrology in Small Basins*, pp. 213–219, IAHS, 2010.

[47] R. J. Hyndman and Y. Fan, "Sample quantiles in statistical packages," *American Statistician*, vol. 50, no. 4, pp. 361–365, 1996.

[48] G. J. Bowden, G. C. Dandy, and H. R. Maier, "Input determination for neural network models in water resources applications. Part 1: background and methodology," *Journal of Hydrology*, vol. 301, no. 1–4, pp. 75–92, 2005.

[49] R. J. May, H. R. Maier, G. C. Dandy, and T. M. K. G. Fernando, "Non-linear variable selection for artificial neural networks using partial mutual information," *Environmental Modelling and Software*, vol. 23, no. 10-11, pp. 1312–1326, 2008.

[50] R. J. May, G. C. Dandy, H. R. Maier, and J. B. Nixon, "Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems," *Environmental Modelling & Software*, vol. 23, no. 10-11, pp. 1289–1299, 2008.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 836137, 9 pages
http://dx.doi.org/10.1155/2014/836137

Research Article

Attribute Extended Algorithm of Lattice-Valued Concept Lattice Based on Congener Formal Context

Li Yang[1] and Yang Xu[2]

[1] *School of Mathematics and Information Science, North China University of Water Resources and Electric Power, Zhengzhou 450046, China*
[2] *Intelligent Control Development Center, Southwest Jiaotong University, Chengdu 610031, China*

Correspondence should be addressed to Li Yang; yangli6672@sina.com

Received 9 April 2014; Revised 9 June 2014; Accepted 11 June 2014; Published 30 June 2014

Academic Editor: Ker-Wei Yu

This paper is the continuation of our research work about lattice-valued concept lattice based on lattice implication algebra. For a better application of lattice-valued concept lattice into data distributed storage and parallel processing, it is necessary to research attribute extended algorithm based on congener formal context. The definitions of attribute extended formal context and congener formal context are proposed. On condition that the extent set stays invariable when the new attribute is increased, the necessary and sufficient conditions of forming attribute values are researched. Based on these conditions, the algorithms of generating lattice-valued congener formal context and establishing concept lattice are given, by which we can provide a useful basis for union algorithm and constructing algorithm of lattice-valued concept lattices in distributed and parallel system.

1. Introduction

Concept lattice (also called formal concept analysis FCA) was proposed by Wille [1–3] in 1982, and its ideological core is constructing the binary relation between objects and attributes based on bivalent logic. Facing the massive fuzzy information existed in reality, fuzzy concept lattice has appeared [4–9], which is used to describe the fuzzy relation between objects and attributes. As a conceptual clustering method, concept lattices have been proved to benefit machine learning, information retrieval, and knowledge discovery.

Lattice-valued concept lattice talked about in this paper, which can be looked as another different kind of fuzzy concept lattice, is constructed on the structure of lattice implication algebra. Its key point different from the classical fuzzy concept lattice is that its values range is not general interval $[0, 1]$ but a complete lattice structure, on which incomparability fuzzy information can be dealt with very well. This selection of lattice implication algebra has two advantages in contrast to general structure. The first computational process of lattice-valued formal concepts is closed under the fuzzy operation; the second complicated inference is avoided in setting up Galois connection of lattice-valued concept lattice. A large amount of research on lattice-valued concept lattice may be referred to [10–12]. While for lattice implication algebra, Xu et al. [13–15] proposed this concept by combining lattice and implication algebra in the 1980s in order to depict uncertainty information more factually.

With the rapid development of network technology, especial in the internet area, distributed storage and parallel processing of data are urgently needed. Distributed construction idea [16, 17] of concept lattice is firstly to form several subcontexts by dividing formal context and then construct the corresponding sublattices and get required concept lattice through combing these sublattices in the end; simply speaking, this idea is to construct concept lattice through combing several sublattices. Based on the horizontal and vertical resolution in formal contexts [18, 19], the union of multiple classical concept lattices includes the horizontal union and vertical union. And these two kinds of union algorithms have their respective preconditions required for formal contexts, which are the same objects sets for the horizontal union and the same attributes sets for the vertical union. However, for lattice-valued concept lattice, it is impossible to execute union operation on the formal contexts by only guaranteeing the same object sets or the same attribute sets, because the formal concepts derived from different formal contexts only with the same objects sets or the same attributes sets will belong to the different types. Therefore, if we want to combine, distributed, and construct the lattice-valued concept lattices, the different formal contexts should be guaranteed to possess both the same object sets and the same attribute sets. According to the three different situations that existed in formal contexts: the same object sets but the different attribute sets, the same attribute sets but the different object sets, the different object sets, and the different attributes, this paper mainly researches the required conditions with attribute increasing, under which the new established lattice-valued concept lattice has the invariable extent set—it is called as congener formal context and then obtains attribute extended algorithm of lattice-valued concept lattices based on congener formal context. The advantages of this algorithm are (i) it is helpful for multiple contexts with different attribute sets to have the same object sets and the same attribute sets and very convenient to execute union operation on multiple concept lattices; (ii) it is also used to improve the reduction ability of lattice-valued concept lattice [20–22], that is, based on the idea of this algorithm; a complicated lattice-valued concept lattice can be divided into several easier sublattices in order to realize reduction, and these sublattices can also be combined by the relevant union algorithm.

Based on these analyses, this paper puts forward attribute extended algorithm of lattice-valued concept lattice based on congener formal context. In Section 2, we give an overview of classical concept lattice and lattice implication algebra. In Section 3, the related works of lattice-valued concept lattice are briefly summarized. Successively, the definitions of attribute extended formal context and congener formal context are proposed in Section 4, where we show the relevant attribute extended judgment theorems based on congener formal context and give the generation algorithms of attribute extended formal context and lattice-valued formal concepts and make the algorithm analysis among the Bordat algorithm, extended algorithm and union algorithm, respectively. Concluding remarks are presented in Section 5.

2. Concept Lattice and Lattice Implication Algebra

In this section, we review briefly the classical concept lattices and lattice implication algebra and they are the foundations of constructing lattice-valued concept lattice.

Definition 1 (Birkhoff [1]). A partial ordered set (poset) is a set, in which a binary relation $\leq$ is defined, which satisfies the following conditions: for any x, y, z,

(1) $x \leq x$, for any x (reflexive),

(2) $x \leq y$ and $y \leq x$ implies $x = y$ (antisymmetry),

(3) $x \leq y$ and $y \leq z$ implies $x \leq z$ (transitivity).

Definition 2 (Birkhoff [1]). Let L be an arbitrary set, and let there be given two binary operations on L, denoted by $\wedge$ and $\vee$. Then the structure $(L, \wedge, \vee)$ is an algebraic structure with two binary operations. We call the structure $(L, \wedge, \vee)$ a lattice provided that it satisfies the following properties:

(1) for any $x, y, z \in L$, $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ and $x \vee (y \vee z) = (x \vee y) \vee z$;

(2) for any $x, y \in L$, $x \wedge y = y \wedge x$ and $x \vee y = y \vee x$;

(3) for any $x \in L$, $x \wedge x = x$ and $x \vee x = x$;

(4) for any $x, y \in L$, $x \wedge (x \vee y) = x$ and $x \vee (x \wedge y) = x$.

Definition 3 (Ganter and Wille [3]). The formal context of classical concept lattice is defined as a set structure (G, M, I) consisting of sets G and M and a binary relation $I \subseteq G \times M$. The elements of G and M are called objects and attributes, respectively, and the relationship gIm is read: the object g has the attribute m. For a set of objects $A \subseteq G$, A^* is defined as the set of features shared by all the objects in A; that is,

$$A^* = \{m \in M \mid gIm \ \forall g \in A\}. \tag{1}$$

Similarly, for $B \subseteq M$, B^* is defined as the set of objects that possess all the features in B; that is,

$$B^* = \{g \in G \mid gIm \ \forall m \in B\}. \tag{2}$$

Definition 4 (Ganter and Wille [3]). A formal concept of the context (G, M, I) is defined as a pair (A, B) with $A \subseteq G$, $B \subseteq M$ and $A^* = B$, $B^* = A$. The set A is called the extent and B the intent of the concept (A, B).

Definition 5 (Xu [13]). Let $(L, \wedge, \vee, O, I)$ be a bounded lattice with an order-reversing involution $'$, I, and O the greatest and the smallest element of L, respectively, and let $\rightarrow : L \times L \rightarrow L$ be a mapping. If the following conditions hold for any $x, y, z \in L$:

(1) $x \rightarrow (y \rightarrow z) = y \rightarrow (x \rightarrow z)$,

(2) $x \rightarrow x = I$,

(3) $x \rightarrow y = y' \rightarrow x'$,

(4) $x \rightarrow y = y \rightarrow x = I$ implies $x = y$,

(5) $(x \rightarrow y) \rightarrow y = (y \rightarrow x) \rightarrow x$,

(6) $(x \vee y) \rightarrow z = (x \rightarrow z) \wedge (y \rightarrow z)$,

(7) $(x \wedge y) \rightarrow z = (x \rightarrow z) \vee (y \rightarrow z)$,

then $(L, \wedge, \vee, ', \rightarrow, O, I)$ is called a lattice implication algebra (LIA).

Example 6 (Xu [13]). Let $L = \{O, a, b, c, d, I\}$, $O' = I$, $a' = c$, $b' = d$, $c' = a$, $d' = b$, $I' = O$, the Hasse diagram of L be defined as Figure 1, and its implication operation be defined as Table 1, then $(L, \wedge, \vee, ', \rightarrow, O, I)$ is a lattice implication algebra.

3. Lattice-Valued Concept Lattice

Lattice-valued concept lattice is the combination of classical concept lattice and lattice implication algebra and its ideological core is constructing lattice-valued relation between

Table 1: Implication Operation of $L = \{O, a, b, c, d, I\}$.

$\longrightarrow$	O	a	b	c	d	I
O	I	I	I	I	I	I
a	c	I	b	c	b	I
b	d	a	I	b	a	I
c	a	a	I	I	a	I
d	b	I	I	b	I	I
I	O	a	b	c	d	I

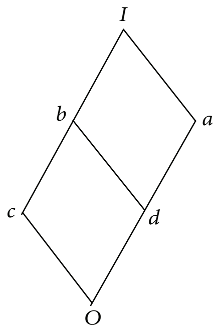


Figure 1: Hasse diagram of $(L, \wedge, \vee, ', \rightarrow, O, I)$.

objects and attributes. It can be used to directly deal with incomparability fuzzy information and is totally different from classical fuzzy concept lattice. In this section, we study the definitions and theorems of lattice-valued concept lattice and give an example to illustrate it.

Definition 7. A four-tuple $K = (G, M, L_n, \widetilde{I})$ is called an n-ary lattice-valued formal context, where $G = \{g_1, g_2, \ldots, g_r\}$ is the set of objects, $M = \{m_1, m_2, \ldots, m_s\}$ is the set of attributes, L_n is an n-ary lattice implication algebra, and $\widetilde{I}$ is a fuzzy relation between G and M; that is, $\widetilde{I} : G \times M \rightarrow L_n$.

Let G be a nonempty objects set and $(L_n, \vee, \wedge, ', \rightarrow)$ an n-ary lattice implication algebra. Denote the set of all the L_n-fuzzy subsets on G as L_n^G, for any $\widetilde{A}_1, \widetilde{A}_2 \in L_n^G$,

$$\widetilde{A}_1 \subseteq \widetilde{A}_2 \Longleftrightarrow \widetilde{A}_1(g) \leq \widetilde{A}_2(g), \quad g \in G; \tag{3}$$

then $(L_n^G, \subseteq)$ is a partial ordered set.

Let M be a nonempty attributes set and $(L_n, \vee, \wedge, ', \rightarrow)$ an n-ary lattice implication algebra. Denote the set of all the L_n-fuzzy subsets on Mas L_n^M, for any $\widetilde{B}_1, \widetilde{B}_2 \in L_n^M$,

$$\widetilde{B}_1 \subseteq \widetilde{B}_2 \Longleftrightarrow \widetilde{B}_1(m) \leq \widetilde{B}_2(m), \quad m \in M, \tag{4}$$

then $(L_n^M, \subseteq)$ is a partial ordered set.

Theorem 8 (Yang and Xu [10]). *Let $K = (G, M, L_n, \widetilde{I})$ be an n-ary lattice-valued formal context and let L_n be an n-ary lattice implication algebra; define mappings f_1, f_2 between L_n^G and L_n^M:*

$$\begin{gathered} f_1 : L_n^G \longrightarrow L_n^M, \\ f_1(\widetilde{A})(m) = \bigwedge_{g \in G} (\widetilde{A}(g) \longrightarrow \widetilde{I}(g, m)), \\ f_2 : L_n^M \longrightarrow L_n^G, \\ f_2(\widetilde{B})(g) = \bigwedge_{m \in M} (\widetilde{B}(m) \longrightarrow \widetilde{I}(g, m)); \end{gathered} \tag{5}$$

then for any $\widetilde{A} \in L^G$, $\widetilde{B} \in L^M$, (f_1, f_2) is a Galois connection based on lattice implication algebra.

Theorem 9 (Yang and Xu [10]). *Let $K = (G, M, L_n, \widetilde{I})$ be a lattice-valued formal context and (f_1, f_2) the Galois connection, for any $\widetilde{A}_1, \widetilde{A}_2, A \in L_n^G$, $\widetilde{B}_1, \widetilde{B}_2, B \in L_n^M$; there are the following properties:*

(1) $\widetilde{A}_1 \subseteq \widetilde{A}_2 \Rightarrow f_1(\widetilde{A}_2) \subseteq f_1(\widetilde{A}_1)$, $\widetilde{B}_1 \subseteq \widetilde{B}_2 \Rightarrow f_2(\widetilde{B}_2) \subseteq f_2(\widetilde{B}_1)$;

(2) $\widetilde{A} \subseteq f_2 f_1(\widetilde{A})$, $\widetilde{B} \subseteq f_1 f_2(\widetilde{B})$.

Definition 10. Let $K = (G, M, L_n, \widetilde{I})$ be an n-ary lattice-valued formal context; denote the set

$$L(K) = \{(\widetilde{A}, \widetilde{B}) \mid f_1(\widetilde{A}) = \widetilde{B}, f_2(\widetilde{B}) = \widetilde{A}\}, \tag{6}$$

and define

$$(\widetilde{A}_1, \widetilde{B}_1) \leq (\widetilde{A}_2, \widetilde{B}_2) \Longleftrightarrow \widetilde{A}_1 \subseteq \widetilde{A}_2 \left(\text{or } \widetilde{B}_2 \subseteq \widetilde{B}_1\right). \tag{7}$$

Theorem 11. *Let $K = (G, M, L_n, \widetilde{I})$ be an n-ary lattice-valued formal context; define the operations $\wedge$ and $\vee$ on $L(G, M, L_n, \widetilde{I})$ as*

$$\begin{gathered} \bigwedge_j (\widetilde{A}_j, \widetilde{B}_j) = \left(\bigcap_j \widetilde{A}_j, f_1 f_2\left(\bigcup_j \widetilde{B}_j\right)\right), \\ \bigvee_j (\widetilde{A}_j, \widetilde{B}_j) = \left(f_2 f_1\left(\bigcup_j \widetilde{A}_j\right), \bigcap_j \widetilde{B}_j\right); \end{gathered} \tag{8}$$

then $L(K)$ is a complete lattice.

Proof. For any $(\widetilde{A}_j, \widetilde{B}_j), (\widetilde{A}_k, \widetilde{B}_k) \in L(K)$, $k \in J$, and by Theorem 9(1),

$$\begin{aligned}\bigcap_{j\in J}\widetilde{A}_j \subseteq \widetilde{A}_k &\Longrightarrow f_2 f_1\left(\bigcap_{j\in J}\widetilde{A}_j\right) \subseteq f_2 f_1\left(\widetilde{A}_k\right) = \widetilde{A}_k \\ &\Longrightarrow f_2 f_1\left(\bigcap_{j\in J}\widetilde{A}_j\right) \subseteq \bigcap_{k\in J}\widetilde{A}_k = \bigcap_{j\in J}\widetilde{A}_j.\end{aligned} \tag{9}$$

By Theorem 9(2), we can get

$$\bigcap_{j\in J}\widetilde{A}_j \subseteq f_2 f_1\left(\bigcap_{j\in J}\widetilde{A}_j\right); \tag{10}$$

thus,

$$\bigcap_{j\in J}\widetilde{A}_j = f_2 f_1\left(\bigcap_{j\in J}\widetilde{A}_j\right); \tag{11}$$

by Theorem 9,

$$f_1\left(\bigcap_{j\in J}\widetilde{A}_j\right) = f_1\left(\bigcap_{j\in J} f_2\left(\widetilde{B}_j\right)\right) = f_1 f_2\left(\bigcup_{j\in J}\widetilde{B}_j\right), \tag{12}$$

it follows that $(\cap_{j\in J}\widetilde{A}_j, f_1 f_2(\cup_{j\in J}\widetilde{B}_j)) = (\cap_{j\in J}\widetilde{A}_j, f_1(\cap_{j\in J}\widetilde{A}_j)) \in L(K)$; obviously, $(\cap_{j\in J}\widetilde{A}_j, f_1 f_2(\cup_{j\in J}\widetilde{B}_j))$ is the lower bound of $L(K)$; on the other hand, suppose that $(\widetilde{A}, \widetilde{B})$ is a lower bound of $L(K)$; then

$$\widetilde{A} \subseteq \widetilde{A}_j \Rightarrow \widetilde{A} \subseteq \bigcap_{j\in J}\widetilde{A}_j; \tag{13}$$

that is,

$$\left(\widetilde{A}, \widetilde{B}\right) \leq \left(\bigcap_{j\in J}\widetilde{A}_j, f_1 f_2\left(\bigcup_{j\in J}\widetilde{B}_j\right)\right), \tag{14}$$

so

$$\begin{aligned}\underset{j\in J}{\wedge}\left(\widetilde{A}_j, \widetilde{B}_j\right) &= \left(\bigcap_{j\in J}\widetilde{A}_j, f_1 f_2\left(\bigcup_{j\in J}\widetilde{B}_j\right)\right) \\ \underset{j\in J}{\vee}\left(\widetilde{A}_j, \widetilde{B}_j\right) &= \left(f_2 f_1\left(\bigcup_{j\in J}\widetilde{A}_j\right), \bigcap_{j\in J}\widetilde{B}_j\right)\end{aligned} \tag{15}$$

can be proved similarly.

Thus, $(L(K), \leq)$ is a complete lattice. □

Example 12. A lattice-valued formal context and its concept lattice based on the 4-ary lattice implication algebra are shown in Figures 2 and 3 and Tables 2 and 3.

In this Hasse diagram (see Figures 2 and 3), the fuzzy concepts are shown as follows:

0#: $(\{I, I\}, \{O, O, a\})$,
1#: $(\{a, I\}, \{b, O, a\})$,
2#: $(\{I, a\}, \{O, b, I\})$,

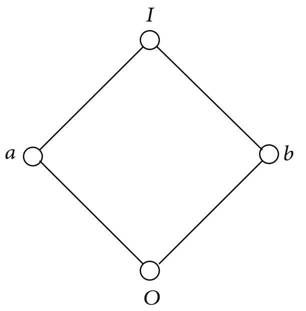


Figure 2: Hasse diagram of 4-ary lattice implication algebra $L = \{O, a, b, I\}$.

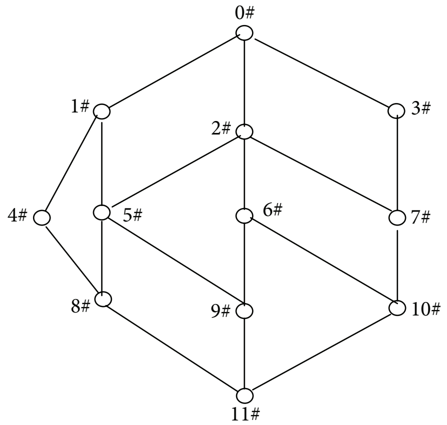


Figure 3: Hasse diagram of lattice-valued concept lattice of K.

3#: $(\{I, b\}, \{a, O, a\})$,
4#: $(\{a, a\}, \{b, b, I\})$,
5#: $(\{a, b\}, \{I, O, a\})$,
6#: $(\{b, b\}, \{a, a, a\})$,
7#: $(\{I, O\}, \{a, b, I\})$,
8#: $(\{a, O\}, \{I, b, I\})$,
9#: $(\{O, b\}, \{I, a, a\})$,
10#: $(\{b, O\}, \{a, I, I\})$,
11#: $(\{O, O\}, \{I, I, I\})$.

4. Attribute Extended Algorithm of Lattice-Valued Concept Lattice

In the pioneering work of extended algorithm of concept lattice, many researchers put emphasis on reconstructing algorithm of concept lattice in adding attributes, and in this reconstructing process, the structure of original concept lattice generally may be changed. Up to now, there have not been any researches on extended algorithm of fuzzy concept lattice, because attribute increase will bound to be leading to the fundamental changes in the fuzzy concepts, and so will be bringing for the structure of lattice-valued concept lattice. In this section, we propose the definitions

Table 2: Implication operator of $L = \{O, a, b, I\}$.

$\longrightarrow$	I	a	b	O
I	I	a	b	O
a	I	I	b	b
b	I	a	I	a
O	I	I	I	I

Table 3: Lattice-valued formal context $K = (G, M, L_4, \widetilde{I})$.

	m_1	m_2	m_3
g_1	a	b	I
g_2	b	O	a

of attribute extended context and congener context. The relevant judgment theorems and algorithms are to be talked about.

4.1. Attribute Extended Theory of Lattice-Valued Fuzzy Concept Lattice

Definition 13. Let $K = (G, M, L_n, \widetilde{I})$ be an n-ary lattice-valued formal context; the four-tuple $K_M = (G, M_+, L_n, \widetilde{I}_+)$ is called an n-ary lattice-valued attribute extended formal context of K, where G is a set of objects, $M_+ \supseteq M$ is an extended set of attributes, L_n is an n-ary lattice implication algebra, and $\widetilde{I}_+$ is the fuzzy relation of G and M_+; that is, $\widetilde{I}_+ : G \times M_+ \rightarrow L_n$ and $\forall g \in G, m \in M \cap M_+$, satisfying

$$g\widetilde{I}m = g\widetilde{I}_+m. \tag{16}$$

For the above formal contexts, there are relevant concept lattices that can be denoted by $L(G, M, L_n, \widetilde{I})$, $L(G, M_+, L_n, \widetilde{I})$, respectively. And denote the set

$$\widetilde{\mathbf{A}} = \left\{\widetilde{A} \mid \left(\widetilde{A}, \widetilde{B}\right) \in L\left(G, M, L_n, \widetilde{I}\right)\right\}, \tag{17}$$

$$\widetilde{\mathbf{B}} = \left\{\widetilde{B} \mid \left(\widetilde{A}, \widetilde{B}\right) \in L\left(G, M, L_n, \widetilde{I}\right)\right\}; \tag{18}$$

$$\widetilde{\mathbf{A}}_M = \left\{\widetilde{A} \mid \left(\widetilde{A}, \widetilde{B}_+\right) \in L\left(G, M_+, L_n, \widetilde{I}\right)\right\}, \tag{19}$$

$$\widetilde{\mathbf{B}}_M = \left\{\widetilde{B}_+ \mid \left(\widetilde{A}, \widetilde{B}_+\right) \in L\left(G, M_+, L_n, \widetilde{I}\right)\right\}. \tag{20}$$

Definition 14. Let $K = (G, M, L_n, \widetilde{I})$ be an n-ary lattice-valued formal context and $K_M = (G, M_+, L_n, \widetilde{I}_+)$ an n-ary lattice-valued attribute extended forma context. K_M is called congener formal context of K if

$$\widetilde{\mathbf{A}}_M = \widetilde{\mathbf{A}}; \tag{21}$$

accordingly, $L(K_M)$ is called congener concept lattice of $L(K)$.

For an n-ary lattice-valued attribute extended formal context $K_M = (G, M_+, L_n, \widetilde{I}_+)$, (f_1, f_2) is a Galois connection between G and M_+, $\forall \widetilde{A} \in L_n^G$, $m \in M$, and $m_+ \in M_+$; denote

$$\begin{aligned} f_1\left(\widetilde{A}\right)(m) &= \widetilde{B} \in L_n^M, \\ f_1\left(\widetilde{A}\right)(m_+) &= \widetilde{B}_+ \in L_n^{M_+}. \end{aligned} \tag{22}$$

Theorem 15. *Let $K_M = (G, M_+, L_n, \widetilde{I}_+)$ be an n-ary lattice-valued attribute extended formal context of K, $\forall \widetilde{A} \in L_n^G$, $m \in M$, $m_+ \in M_+$,and $g \in G$; then K_M is the congener formal context of K if and only if*

$$f_2\left(\widetilde{B}_+\right)(g) = f_2\left(\widetilde{B}\right)(g). \tag{23}$$

Proof. Consider that $\forall \widetilde{A} \in \widetilde{\mathbf{A}}_M$, $\exists (\widetilde{A}, \widetilde{B}_+) \in L(G, M_+, L_n, \widetilde{I})$; that is, $(f_2(\widetilde{B}_+)(g), \widetilde{B}_+) \in L(G, M_+, L_n, \widetilde{I})$; $\forall \widetilde{A} \in \widetilde{\mathbf{A}}$, $\exists (\widetilde{A}, \widetilde{B}) \in L(G, M, L_n, \widetilde{I})$; that is, $(f_2(\widetilde{B})(g), \widetilde{B}) \in L(G, M, L_n, \widetilde{I})$; by Definition 14, K_M is the congener formal context of K $\Leftrightarrow$ $\widetilde{\mathbf{A}}_M = \widetilde{\mathbf{A}} \Leftrightarrow f_2(\widetilde{B}_+)(g) = f_2(\widetilde{B})(g)$. □

Theorem 16. *Let $K_M = (G, M_+, L_n, \widetilde{I}_+)$ be an n-ary lattice-valued attribute extended formal context of K, $\forall \widetilde{A} \in L_n^G$, $n \in M_+/M$,and $g \in G$; then K_M is the congener formal context of K if and only if*

$$f_2\left(\widetilde{B}\right)(g) \leq f_2\left(f_1\left(\widetilde{A}\right)(n)\right)(g). \tag{24}$$

Proof. By Theorem 15, K_M is the congener formal context of K:

$$\begin{aligned} &\Longleftrightarrow f_2\left(\widetilde{B}_+\right)(g) = f_2\left(\widetilde{B}\right)(g) \\ &\Longleftrightarrow f_2\left(\widetilde{B}\right)(g) \wedge f_2\left(f_1\left(\widetilde{A}\right)(n)\right)(g) = f_2\left(\widetilde{B}\right)(g) \\ &\Longleftrightarrow f_2\left(\widetilde{B}\right)(g) \leq f_2\left(f_1\left(\widetilde{A}\right)(n)\right)(g). \end{aligned} \tag{25}$$

□

Corollary 17. *Let $K_M = (G, M_+, L_n, \widetilde{I}_+)$ be an n-ary lattice-valued attribute extended formal context of K, $\forall \widetilde{A} \in L_n^G$, $n \in M_+/M$, and $g \in G$; then K_M is the congener formal context of K, if $\exists m_j \in M$, $1 \leq j \leq s$, such that*

$$f_2\left(f_1\left(\widetilde{A}\right)\left(m_j\right)\right)(g) \leq f_2\left(f_1\left(\widetilde{A}\right)(n)\right)(g). \tag{26}$$

Theorem 18. *Let $K_M = (G, M_+, L_n, \widetilde{I}_+)$ be an n-ary lattice-valued attribute extended formal context of K, $\forall \widetilde{A} \in L_n^G$, $n \in M_+/M$, and $g \in G$; then K_M is the congener fuzzy context of K if $\exists m_{j_1}, m_{j_2} \in M$, $1 \leq j_1, j_2 \leq s$, such that*

$$\widetilde{I}_+(g, n) = \widetilde{I}\left(g, m_{j_1}\right) \wedge \widetilde{I}\left(g, m_{j_2}\right). \tag{27}$$

Proof. By Theorem 16,

$$\begin{aligned}
&f_2\left(f_1\left(\widetilde{A}\right)(n)\right)(g)\\
&\quad= \bigwedge_{n\in M_+/M}\left(f_1\left(\widetilde{A}\right)(n) \longrightarrow \widetilde{I}_+(g,n)\right)\\
&\quad= \bigwedge_{n\in M_+/M}\left(\bigwedge_{g\in G}\left(\widetilde{A}(g) \longrightarrow \widetilde{I}_+(g,n)\right) \longrightarrow \widetilde{I}_+(g,n)\right)\\
&\quad= \bigwedge_{g\in G}\left(\widetilde{A}(g) \longrightarrow \widetilde{I}\left(g,m_{j_1}\right) \wedge \widetilde{I}\left(g,m_{j_2}\right)\right)\\
&\qquad\longrightarrow \left(\widetilde{I}\left(g,m_{j_1}\right) \wedge \widetilde{I}\left(g,m_{j_2}\right)\right))\\
&\quad= \left(\left(f_1\left(\widetilde{A}\right)\left(m_{j_1}\right) \wedge f_1\left(\widetilde{A}\right)\left(m_{j_2}\right)\right) \longrightarrow \widetilde{I}\left(g,m_{j_1}\right)\right)\\
&\qquad\wedge \left(\left(f_1\left(\widetilde{A}\right)\left(m_{j_1}\right) \wedge f_1\left(\widetilde{A}\right)\left(m_{j_2}\right)\right) \longrightarrow \widetilde{I}\left(g,m_{j_2}\right)\right)\\
&\quad\geq \left(f_1\left(\widetilde{A}\right)\left(m_{j_1}\right) \longrightarrow \widetilde{I}\left(g,m_{j_1}\right)\right)\\
&\qquad\wedge \left(f_1\left(\widetilde{A}\right)\left(m_{j_2}\right) \longrightarrow \widetilde{I}\left(g,m_{j_2}\right)\right)\\
&\quad\geq \bigwedge_{m\in M}\left(f_1\left(\widetilde{A}\right)(m) \longrightarrow \widetilde{I}(g,m)\right)\\
&\quad= f_2\left(\widetilde{B}\right)(g),
\end{aligned} \tag{28}$$

so K_M is the congener fuzzy context of K. □

Corollary 19. *Let $K_M = (G, M_+, L_n, \widetilde{I}_+)$ an n-ary lattice-valued attribute extended formal context of K, $\forall \widetilde{A} \in L_n^G$, $n \in M_+/M$, and $g \in G$; then K_M is the congener formal context of K if $\exists m_{j_1}, m_{j_2}, \ldots, m_{j_k} \in M$, $1 \leq j_1, j_2, \ldots, j_k \leq s$, such that*

$$\widetilde{I}_+(g,n) = \bigwedge_{p=1}^{k} \widetilde{I}\left(g, m_{j_p}\right). \tag{29}$$

Theorem 20. *Let $K_M = (G, M_+, L_n, \widetilde{I}_+)$ be an n-ary lattice-valued attribute extended formal context of K, $\forall \widetilde{A} \in L_n^G$, $n \in M_+/M$, and $g \in G$; then K_M is the congener formal context of K if*

$$\widetilde{I}_+(g,n) = I. \tag{30}$$

Proof. By Theorem 16,

$$\begin{aligned}
f_2\left(f_1\left(\widetilde{A}\right)(n)\right)(g) &= \bigwedge_{n\in M_+/M}\left(f_1\left(\widetilde{A}\right)(n) \longrightarrow \widetilde{I}_+(g,n)\right)\\
&= \bigwedge_{n\in M_+/M}\left(f_1\left(\widetilde{A}\right)(n) \longrightarrow I\right)\\
&= I\\
&\geq f_2\left(\widetilde{B}\right)(g),
\end{aligned} \tag{31}$$

so K_M is the congener fuzzy context of K. □

4.2. Attribute Extended Algorithm of Lattice-Valued Concept Lattice. For general fuzzy concept lattice which is constructed on the interval $[0,1]$, attribute effective increased will inevitably change the number of fuzzy concepts and the structure of fuzzy concept lattice. But, for lattice-valued concept lattice, attribute conditional increased which is researched in this paper and will not change the number of formal concepts and the structure of concept lattice. We explain this through Algorithms 1 and 2.

Example 21. For Table 3, a 4-ary lattice-valued formal context $K = (G, M, L_4, \widetilde{I})$, we can compute its congener formal context as Table 4.

In this formal context K_M, $M_+ = \{m_1, m_2, \ldots, m_5\}$, $\widetilde{I}_+ : G \times M_+ \rightarrow L_4$, and it follows that $\forall g \in G$, $\widetilde{I}_+(g, m_4) = \widetilde{I}(g, m_1) \wedge \widetilde{I}(g, m_2)$, $\widetilde{I}_+(g, m_5) = I$. According to the theorems and algorithms of attribute increased, the formal concepts of K_M are directly derived from the relevant formal concepts of K as follows:

0#: $(\{I, I\}, \{O, O, a, O, I\})$,
1#: $(\{a, I\}, \{b, O, a, O, I\})$,
2#: $(\{I, a\}, \{O, b, I, O, I\})$,
3#: $(\{I, b\}, \{a, O, a, O, I\})$,
4#: $(\{a, a\}, \{b, b, I, b, I\})$,
5#: $(\{a, b\}, \{I, O, a, O, I\})$,
6#: $(\{b, b\}, \{a, a, a, a, I\})$,
7#: $(\{I, O\}, \{a, b, I, O, I\})$,
8#: $(\{a, O\}, \{I, b, I, b, I\})$,
9#: $(\{O, b\}, \{I, a, a, a, I\})$,
10#: $(\{b, O\}, \{a, I, I, a, I\})$,
11#: $(\{O, O\}, \{I, I, I, I, I\})$.

4.3. Algorithm Analysis. The time complexity of creating concept lattices is the major factor in analyzing the complexity of the algorithm, and computing the formal concepts plays the key role in the whole process of constructing concept lattices. Suppose that the lattice-valued formal context is $K = (G, M, L_n, \widetilde{I})$, where $G = \{g_1, g_2, \ldots, g_r\}$, $M = \{m_1, m_2, \ldots, m_{s_1}\}$, r, s_1, and n are the positive integers. If we compute the formal concepts under Bordat algorithm, the calculation times of the formal concepts are $r \times n^{s_1}(1 + s_1)$. When increasing s_2 attributes, the calculation times of the formal concepts are changed into $r \times n^{s_1+s_2}(1 + s_1 + s_2)$.

If we utilize the extended algorithm, the increased s_2 attributes can be directly obtained by the preliminary s_1 attributes, so the calculation times of the formal concepts are $r \times n^{s_1}(1 + s_1) + s_2$. Compared with Bordat algorithm, the calculation times of the formal concepts is $r \times n^{s_1}(1 + s_1) + s_2 \ll r \times n^{s_1+s_2}(1 + s_1 + s_2)$. And under the action of the extended algorithm, we can also firstly decompose the lattice-valued formal context $K = (G, M, L_n, \widetilde{I})$ into several subformal contexts; that is, $\{m_1, \ldots, m_{s_1}\} = \{m_1, \ldots, m_{s_{11}}\} \cup \{m_{s_{11}+1}, \ldots, m_{s_{11}+s_{12}}\} \cup \cdots \cup \{m_{s_{1(p-1)}+1}, \ldots, m_{s_{1(p-1)}+s_{1p}}\}$, where $s_1 = s_{11} + s_{12} + \cdots + s_{1p}$, $s_{11}, s_{12}, \ldots, s_{1p}$ are also the positive

Table 4: The congener formal context $K_M = (G, M_+, L_4, \widetilde{I}_+)$ of K.

	m_1	m_2	m_3	m_4	m_5
g_1	a	b	I	a	I
g_2	b	O	a	O	I

```
Input: the $n$-ary lattice-valued formal context $K = (G, M, L_n, \widetilde{I})$, let $M_+ = M$
Output: the $n$-ary lattice-valued attribute extended formal context $K_M = (G, M_+, L_n, \widetilde{I}_+)$
Begin
   while $((G, M, L_n, \widetilde{I}) \neq \Phi)$ do
     Calculate the attribute values $\widetilde{I}(g, m_j)$ of each attribute $m_j \in M$
     for each $\widetilde{I}(g, m_{j_1}), \widetilde{I}(g, m_{j_2}) \in K$ do
         if $\widetilde{I}(g, m_{j_1}) \wedge \widetilde{I}(g, m_{j_2}) \neq \Phi$ then
            $\widetilde{I}_+(g, n) := \widetilde{I}(g, m_{j_1}) \wedge \widetilde{I}(g, m_{j_2})$
            $M_+ := M \cup \{n\}$
         else
            $M_+ = M$
         endif;
         for each $\widetilde{I}(g, m_{j_1}), \widetilde{I}(g, m_{j_2}), \ldots, \widetilde{I}(g, m_{j_k}) \in K$ do
            if $\bigwedge_{p=1}^{k} \widetilde{I}(g, m_{j_p}) \neq \Phi$ then
               $\widetilde{I}_+(g, n) := \bigwedge_{p=1}^{k} \widetilde{I}(g, m_{j_p})$
               $M_+ := M \cup \{n\}$
            else
               $M_+ = M$
            endif;
            for $L_n$ do
               if $\exists I \in L_n$ then
                  $\widetilde{I}_+(g, n) := I$
                  $M_+ := M \cup \{n\}$
               else
                  $M_+ = M$
               endif;
            endif
         endfor;
     endfor;
end
```

Algorithm 1: Generation algorithm of $K_M = (G, M_+, L_n, \widetilde{I}_+)$.

integers, and the combination algorithm is executed on them; then the calculation times of the formal concepts is $\sum_{j=1}^{p} r \times n^{s_{1j}}(1 + s_{1j})$. Because of $s_{11}, s_{12}, \ldots, s_{1p} < s_1$, we can get $r \times n^{s_{1j}}(1 + s_{1j}) \ll r \times n^{s_1}(1 + s_1)$, $j = 1, 2, \ldots, p$; then $\sum_{j=1}^{p} r \times n^{s_{1j}}(1 + s_{1j}) + s_2 \ll r \times n^{s_1}(1 + s_1) + s_2$; that is to say, the complexity of this algorithm is significantly decreased.

Suppose that $K = (G, M, L_6, \widetilde{I})$ is the lattice-valued formal context based on 6-ary lattice implication algebra, where $G = \{g_1, g_2, \ldots, g_{10}\}$, $M = \{m_1, m_2, \ldots, m_{50}\}$. When increasing 10, 20, 30, 40, and 50 attributes into M, we can obtain attributes set $M_1 = \{m_1, m_2, \ldots, m_{60}\}$, $M_2 = \{m_1, m_2, \ldots, m_{70}\}$, $M_3 = \{m_1, m_2, \ldots, m_{80}\}$, $M_4 = \{m_1, m_2, \ldots, m_{90}\}$, and $M_5 = \{m_1, m_2, \ldots, m_{100}\}$. And furthermore, with the help of the extended algorithm, if we decompose $K = (G, M, L_6, \widetilde{I})$ into 5 subcontexts, the calculation times will be greatly reduced. The following experiments, respectively, compare to Bordat algorithm, as Figure 4.

In Figure 4, the curve B (Bordat algorithm), E (extended algorithm), and U (union algorithm), respectively, presents the calculation times of the formal concepts. By the description of the curves in the experiments, we can draw the conclusion that Bordat algorithm cause the calculation times absolutely increased and its growth rate is far more than that of extended algorithm and union algorithm, and with the help of extended algorithm, we also can see that the calculation times are significantly decreased.

```
Input: the $n$-ary lattice-valued concept lattice $L(G, M, L_n, \widetilde{I})$
Output: the $n$-ary lattice-valued attribute extended concept lattice $L(G, M_+, L_n, \widetilde{I}_+)$
Begin
  while $((G, M, L_n, \widetilde{I}) \neq \Phi)$ do
    Calculate the formal concepts $(\widetilde{A}, \widetilde{B})$ of $K = (G, M, L_n, \widetilde{I})$
    for each $(\widetilde{A}, \widetilde{B}) \in L(G, M, L_n, \widetilde{I})$, denote $\widetilde{A} = \left(\widetilde{A}(g_1), \widetilde{A}(g_2), \ldots, \widetilde{A}(g_r)\right) \in \widetilde{\mathbf{A}}$,
      $\widetilde{B} = \left(\widetilde{B}(m_1), \widetilde{B}(m_2), \ldots, \widetilde{B}(m_s)\right) \in \widetilde{\mathbf{A}}$ do
        if $\exists n \in M_+$, s.t., $\widetilde{I}_+(g, n) = \widetilde{I}(g, m_{j_1}) \wedge \widetilde{I}(g, m_{j_2}), m_{j_1}, m_{j_2} \in M$ then
          $\widetilde{B}_+ = \left(\widetilde{B}(m_1), \ldots, \widetilde{B}(m_s), \widetilde{B}(m_{j_1}) \wedge \widetilde{B}(m_{j_2})\right)$
          $(\widetilde{A}, \widetilde{B}_+) \in L(G, M_+, L_n, \widetilde{I}_+)$
        else
          $(\widetilde{A}, \widetilde{B}_+) \notin L(G, M_+, L_n, \widetilde{I}_+)$
        endif;
          if $\exists n \in M_+$, s.t., $\widetilde{I}_+(g, n) = \bigwedge_{p=1}^{k} \widetilde{I}(g, m_{j_p}), m_{j_1}, m_{j_2}, \ldots, m_{j_k} \in M$ then
            $\widetilde{B}_+ = \left(\widetilde{B}(m_1), \ldots, \widetilde{B}(m_s), \bigwedge_{p=1}^{k} \widetilde{I}(g, m_{j_p})\right)$
            $(\widetilde{A}, \widetilde{B}_+) \in L(G, M_+, L_n, \widetilde{I}_+)$
          else
            $(\widetilde{A}, \widetilde{B}_+) \notin L(G, M_+, L_n, \widetilde{I}_+)$
          endif;
            if $\exists n \in M_+$, s.t., $\widetilde{I}_+(g, n) = I, I \in L_n$
            then
              $(\widetilde{A}, \widetilde{B}_+) \in L(G, M_+, L_n, \widetilde{I}_+)$
            else
              $(\widetilde{A}, \widetilde{B}_+) \notin L(G, M_+, L_n, \widetilde{I}_+)$
            endif;
          endif
        endif;
    endfor;
end
```

Algorithm 2: Generation algorithm of formal concepts.

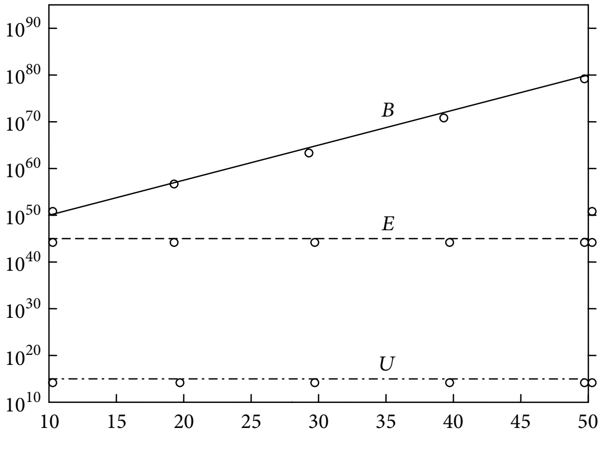


Figure 4: Algorithms comparison of Bordat, extended, and union algorithm.

5. Conclusions

As the theoretical premise of the union operation of lattice-valued concept lattice, our research about attribute extended algorithm of lattice-valued concept lattice is helpful to provide useful conditions not only to distribute various databases but also to make an important progress toward practical application of concept lattice. From the definitions of attribute extended formal context and congener formal context, this paper researches the necessary and sufficient conditions of forming attribute values under the condition that the extent set keeps invariable when the new attribute is increased. The algorithms of generating lattice-valued formal context and building concept lattice based on these conditions are proposed.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of People's Republic of China (Grant no. 60875034) and the Research Fund for the Doctoral Program of Higher Education (Grant no. 20060613007).

References

[1] G. Birkhoff, *Lattice Theory: American Mathematical Society Colloquium Publications Volume XXV*, American Mathematical Society, Providence, RI, USA, 1967.

[2] R. Wille, “Restructuring lattice theory: an approach based on hierarchies of concepts. I,” in *Ordered Sets (Banff, Alta., 1981)*, I. Rival, Ed., vol. 83 of *NATO Advanced Study Institute Series C: Mathematical and Physical Sciences*, pp. 445–470, Reidel, Dordrecht, The Netherlands, 1982.

[3] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer, Berlin, Germany, 1999.

[4] R. Bělohlávek, V. Sklenář, and J. Zacpal, “Crisply generated fuzzy concepts,” in *Proceedings of the 3rd International Conference on Formal Concept Analysis (ICFCA ’05)*, Lecture Notes in Artificial Intelligence, pp. 269–284, February 2005.

[5] W. X. Zhang, Y. Y. Yao, and Y. Liang, *Rough Set and Concept Lattice*, Xi’an Jiaotong University Press, Xi’an, China, 2006.

[6] M. Shao, M. Liu, and W. Zhang, “Set approximations in fuzzy formal concept analysis,” *Fuzzy Sets and Systems*, vol. 158, no. 23, pp. 2627–2640, 2007.

[7] M. H. Hu, L. Zhang, and F. L. Ren, “Fuzzy formal concept analysis and fuzzy concept lattices,” *Journal of Northeastern University (Natural Science)*, vol. 28, no. 9, pp. 1274–1277, 2007.

[8] J. Z. Pang, X. Y. Zhang, and W. H. Xu, “Attribute reduction in intuitionistic fuzzy concept lattices,” *Abstract and Applied Analysis*, vol. 2013, Article ID 271398, 12 pages, 2013.

[9] P. Butka, J. Pocs, and J. Pocsova, “Representation of fuzzy concept lattices in the framework of classical FCA,” *Journal of Applied Mathematics*, vol. 2013, Article ID 236725, 7 pages, 2013.

[10] L. Yang and Y. Xu, “Decision making with uncertainty information based on lattice-valued fuzzy concept lattice,” *Journal of Universal Computer Science*, vol. 16, no. 1, pp. 159–177, 2010.

[11] L. Yang and Y. Xu, “A decision method based on uncertainty reasoning of linguistic truth-valued concept lattice,” *International Journal of General Systems*, vol. 39, no. 3, pp. 235–253, 2010.

[12] L. Yang, Y. H. Wang, and Y. Xu, “A combination algorithm of multiple lattice-valued concept lattices,” *International Journal of Computational Intelligence Systems*, vol. 6, no. 5, pp. 881–892, 2013.

[13] Y. Xu, “Lattice implication algebras,” *Journal of Southwest Jiaotong University*, vol. 289, pp. 20–27, 1993 (Chinese).

[14] Y. Xu, D. Ruan, K. Qin, and J. Liu, *Lattice-Valued Logic-An Alternative Approach to Treat Fuzziness and Incomparability*, vol. 132, Springer, Berlin, Germany, 2003.

[15] Y. Xu, S. Chen, and J. Ma, “Linguistic truth-valued lattice implication algebra and its properties,” in *Proceedings of the IMACS Multiconference on Computational Engineering in Systems Applications (CESA ’06)*, pp. 1413–1418, Beijing, China, October 2006.

[16] Y. Li and Z. T. Liu, “Theoretical research on the distributed construction of concept lattices,” in *Proceedings of the 2nd International Conference on Machine Learning and Cybernetics*, pp. 474–479, Xian, China, 2003.

[17] R. Xie, Z. Pei, and C. L. He, “Reconstructing algorithm of concept lattice in adding attribute process,” *Journal of Systems Engineering*, vol. 22, no. 4, pp. 426–431, 2007.

[18] Y. Li, Z. T. Liu, L. C. Chen, X. H. Xu, and W. C. Cheng, “Horizontal union algorithm of multiple concept lattices,” *Acta Electronica Sinica*, vol. 32, no. 11, pp. 1849–1854, 2004.

[19] L. Zhang, X. J. Shen, D. J. Han et al., “Vertical union algorithm of concept lattices based on synonymous concept,” *Computer Engineering and Applications*, vol. 43, no. 2, pp. 95–98, 2007.

[20] M. Liu, M. Shao, W. Zhang, and C. Wu, “Reduction method for concept lattices based on rough set theory and its application,” *Computers & Mathematics with Applications*, vol. 53, no. 9, pp. 1390–1410, 2007.

[21] X. Wang and W. X. Zhang, “Relations of attribute reduction between object and property oriented concept lattices,” *Knowledge-Based Systems*, vol. 21, no. 5, pp. 398–403, 2008.

[22] S. Y. Zhao and E. C. C. Tsang, “On fuzzy approximation operators in attribute reduction with fuzzy rough sets,” *Information Sciences*, vol. 178, no. 16, pp. 3163–3176, 2008.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 936374, 21 pages
http://dx.doi.org/10.1155/2014/936374

Research Article

A Free Search Krill Herd Algorithm for Functions Optimization

Liangliang Li, Yongquan Zhou, and Jian Xie

College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

Correspondence should be addressed to Yongquan Zhou; yongquanzhou@126.com

Received 5 April 2014; Revised 13 May 2014; Accepted 21 May 2014; Published 19 June 2014

Academic Editor: Yang Xu

To simulate the freedom and uncertain individual behavior of krill herd, this paper introduces the opposition based learning (OBL) strategy and free search operator into krill herd optimization algorithm (KH) and proposes a novel opposition-based free search krill herd optimization algorithm (FSKH). In FSKH, each krill individual can search according to its own perception and scope of activities. The free search strategy highly encourages the individuals to escape from being trapped in local optimal solution. So the diversity and exploration ability of krill population are improved. And FSKH can achieve a better balance between local search and global search. The experiment results of fourteen benchmark functions indicate that the proposed algorithm can be effective and feasible in both low-dimensional and high-dimensional cases. And the convergence speed and precision of FSKH are higher. Compared to PSO, DE, KH, HS, FS, and BA algorithms, the proposed algorithm shows a better optimization performance and robustness.

1. Introduction

As many optimization problems cannot be solved by the traditional mathematical programming methods, the metaheuristic algorithms have been widely used to obtain global optimum solutions. And the aim of developing modern metaheuristic algorithms is to increase the accessibility of the global optimum. Inspired by nature, many successful algorithms are proposed, for example, Genetic Algorithm (GA) [1], Particle Swarm Optimization (PSO) [2, 3], Ant Colony Optimization (ACO) [4], Differential Evolution (DE) [5], Harmony Search (HS) [6], Artificial Bee Colony Optimization (ABC) [7], Firefly Algorithm (FA) [8], Artificial Fish Swarm Algorithm (AFSA) [9], Cuckoo Search (CS) [10, 11], Monkey Algorithm (MA) [12], Bat Algorithm (BA) [13], Charged System Search (CSS) [14], and Flower Pollination Algorithm (FPA) [15]. Nature-inspired algorithms can effectively solve the problems which traditional methods cannot solve and have shown excellent performance in many respects. So its application scope has been greatly expanded. In recent years, the metaheuristic algorithms mentioned above have been applied to solve the application problems. For example, Xu et al. (2010) solve the UCAV path planning problems by chaotic artificial bee colony approach [16]. Hasançebi et al. (2013) applied the bat algorithm in structural optimization problems [17]. Askarzadeh (2013) developed a discrete harmony search algorithm for size optimization of wind-photovoltaic hybrid energy system [18]. Basu and Chowdhury (2013) used cuckoo search algorithm in economic dispatch [19].

Based on the simulation of the herding behavior of krill individuals, Gandomi and Alavi proposed the krill herd algorithm (KH) in 2012 [20]. And KH algorithm is a novel biologically inspired algorithm to solve the optimization problems. In KH, the time-dependent position of the krill individuals is formulated by three main factors: (1) motion induced by the presence of other individuals; (2) foraging motion; (3) physical diffusion. Only time interval (C_t) should be fine-tuned in the KH algorithm which is a remarkable advantage in comparison with other nature-inspired algorithms. Therefore, it can be efficient for many optimization and engineering problems. To improve the krill herd algorithm, Wang and Guo (2013) proposed a hybrid krill herd algorithm with differential evolution for global numerical optimization [21]. The introduced HDE operator let the krill perform local search within the defined region. And the optimization performance of the DEKH was better than the KH. Then, in order to accelerate convergence speed, thus making the approach more feasible for a wider range of real-world engineering applications while keeping the desirable characteristics of the original KH, an effective Lévy-Flight KH (LKH) method was proposed by Wang et al. in 2013 [22]. And Wang also proposed a new improved metaheuristic simulated annealing-based krill herd (SKH) method for global optimization tasks [23]. The KH algorithm has been applied to solve some application problems. In 2014, a discrete Krill Herd Algorithm was proposed for graph based network route optimization by Sur [24]. KH algorithm was further validated against various engineering optimization problems by Gandomi et al. [25]. Inspired from the animals' behavior, free search (FS) [26] is firstly proposed by Penev and Littlefair. In FS, each animal has original peculiarities called sense and mobility. And each animal can operate either with small precise steps for local search or with large steps for global exploration. Moreover, the individual decides how to search personally.

In order to overcome the limited performance of standard KH on complex problems, a novel free search krill herd algorithm is proposed in this paper. The free search strategy has been introduced into the standard KH to avoid all krill individuals getting trapped into the local optima. The proposed algorithm can greatly enrich the diversity of krill population and improve the calculation accuracy, which leads to a good optimization performance. What is more, the new method can enhance the quality of solutions without losing the robustness.

The proposed FSKH algorithm is different from standard KH in two aspects. Firstly, in FSKH, the population of individuals is initialized using opposition based learning (OBL) strategy [27]. By using OBL strategy, the proposed algorithm can make a more uniform distribution of the krill populations. What is more, we can obtain fitter starting candidate solutions even when there is no knowledge about the solutions.

And secondly, the krill can do freedom and uncertain action using free search strategy. In standard KH, krill is influenced by its "neighbors" and the optimal krill, and the sensing distance of each krill is fixed. But in nature, even for the same krill, its sensitivity and range of activities will also change in different environment and different period. The proposed algorithm can simulate this freedom, uncertain individual behavior of the krill. The free search strategy allows nonzero probability for access to any location of the search space and highly encourages the individuals to escape from trapping in local optimal solution.

The remainder of this paper is organized as follows. In the Section 2, the standard krill herd algorithm and free search strategy are described, respectively. In Section 3, the concept of opposition based learning (OBL) strategy is briefly explained. And the proposed algorithm (FSKH) is described in detail. The simulation experiments of the proposed algorithm are presented in Section 4, compared to PSO, DE, KH, HS, FS and BA algorithms. Finally, some remarks and conclusions are provided in Section 5.

2. Preliminary

2.1. Krill Herd Algorithm. Krill herd (KH) is a novel metaheuristic swarm intelligence optimization method for solving optimization problems, which is based on the simulation of the herding behavior of krill individuals. The time-dependent position of an individual krill in two-dimensional surface is determined by the following three main actions:

(1) movement induced by other krill individuals;

(2) foraging activity;

(3) physical diffusion.

KH algorithm used the Lagrangian model as follows:

$$\frac{dX_i}{dt} = N_i + F_i + D_i, \tag{1}$$

where N_i is the motion induced by other krill individuals; F_i is the foraging motion; and D_i is the physical diffusion of the ith krill individuals.

2.1.1. Motion Induced by Other Krill Individuals. For a krill individual, the motion induced by other krill individuals can be determined as follows:

$$\begin{gathered} N_i^{\text{new}} = N^{\max}\alpha_i + \omega_n N_i^{\text{old}}, \\ \alpha_i = \alpha_i^{\text{local}} + \alpha_i^{\text{target}}, \end{gathered} \tag{2}$$

where $N^{\max}$ is the maximum induced speed, ω_n is the inertia weight of the motion induced in $[0, 1]$, N_i^{old} is the last motion induced, α_i^{local} is the local effect provided by the neighbors, and α_i^{target} is the target direction effect provided by the best krill individual. And the effect of the neighbors can be defined as

$$\begin{gathered} \alpha_i^{\text{local}} = \sum_{j=1}^{NN} \widehat{K}_{ij}\widehat{X}_{ij}, \\ \widehat{X}_{ij} = \frac{X_j - X_i}{\left\| X_j - X_i \right\| + \varepsilon}, \\ \widehat{K}_{ij} = \frac{K_j - K_i}{K^{\text{worst}} - K^{\text{best}}}, \end{gathered} \tag{3}$$

where K_i is the fitness value of the ith krill individual. K^{best} and K^{worst} are the best and worst fitness values of the krill individuals so far. K_j is the fitness of jth ($j = 1, 2, \ldots, \text{NN}$) neighbor. And NN is the number of the neighbors. X represents the related positions.

The sensing distance for each krill individual is determined as follows:

$$d_{s,i} = \frac{1}{5N} \sum_{j=1}^{N} \left\| X_i - X_j \right\|, \tag{4}$$

where N is the number of the krill individuals.

The effect of the individual krill with the best fitness on the ith individual krill is taken into account using

$$\partial_i^{\text{target}} = C^{\text{best}} \widehat{K}_{i,\text{best}} \widehat{X}_{i,\text{best}}, \tag{5}$$

where the value of C^{best} is defined as

$$C^{\text{best}} = 2\left(\text{rand} + \frac{I}{I_{\max}}\right). \tag{6}$$

2.1.2. Foraging Motion. The foraging motion is formulated in terms of two main effective parameters. The first is the food location and the second one is the previous experience about the food location. This motion can be expressed for the ith krill individual as follows:

$$F_i = V_f \beta_i + \omega_f F_i^{\text{old}},$$
$$\beta_i = \beta_i^{\text{food}} + \beta_i^{\text{best}}, \tag{7}$$

where V_f is the foraging speed, ω_f is the inertia weight of the foraging motion between 0 and 1, and F_i^{old} is the last foraging motion. β_i^{food} is the attraction of the food and β_i^{best} is the effect of the best fitness of the ith krill so far. In our paper, we set $V_f = 0.02$.

In KH, the virtual center of food concentration is approximately calculated according to the fitness distribution of the krill individuals, which is inspired from "*center of mass.*" The center of food for each iteration is formulated as follows:

$$X^{\text{food}} = \frac{\sum_{i=1}^{N} (1/K_i) X_i}{\sum_{i=1}^{N} (1/K_i)}. \tag{8}$$

Therefore, the food attraction for the ith krill individual can be determined as follows:

$$\beta_i^{\text{food}} = C^{\text{food}} \widehat{K}_{i,\text{food}} \widehat{X}_{i,\text{food}}, \tag{9}$$

where C^{food} is the food coefficient defined as follows:

$$C^{\text{food}} = 2\left(1 - \frac{I}{I_{\max}}\right). \tag{10}$$

The effect of the best fitness of the ith krill individual is also handled using the following equation:

$$\beta_i^{\text{best}} = \widehat{K}_{i,i\text{best}} \widehat{X}_{i,i\text{best}}, \tag{11}$$

where $\widehat{K}_{i,i\text{best}}$ is the best previously visited position of the ith krill individual.

2.1.3. Physical Diffusion. The random diffusion of the krill individuals can be considered to be a random process in essence. This motion can be described in terms of a maximum diffusion speed and a random directional vector. It can be indicated as follows:

$$D_i = D^{\max} \delta, \tag{12}$$

where $D^{\max}$ is the maximum diffusion speed, and δ is the random directional vector, and its arrays are random values in $[-1, 1]$. The better the position is, the less random the motion is. The effects of the motion induced by other krill individuals and foraging motion gradually decrease with increasing the time (iterations). Thus, another term equation (12) into equation (13). This term linearly decreases the random speed with the time and performs on the basis of a geometrical annealing schedule:

$$D_i = D^{\max}\left(1 - \frac{I}{I_{\max}}\right)\delta. \tag{13}$$

2.1.4. Crossover. The crossover is controlled by a crossover probability Cr, and the mth component of X_i (i.e., $x_{i,m}$) is defined as follows:

$$x_{i,m} = \begin{cases} x_{r,m} & \text{rand}_{i,m} < \text{Cr} \\ x_{i,m} & \text{else}, \end{cases}$$
$$\text{Cr} = 0.2\widehat{K}_{i,\text{best}}, \tag{14}$$

where $r \in \{1, 2, \ldots, i-1, i+1, \ldots, N\}$.

2.1.5. Mutation. The mutation is controlled by a mutation probability Mu, and the adaptive mutation scheme is determined as follows:

$$x_{i,m} = \begin{cases} x_{g\text{best},m} + \mu\left(x_{p,m} - x_{q,m}\right) & \text{rand}_{i,m} < \text{Mu} \\ x_{i,m} & \text{else}, \end{cases}$$
$$\text{Mu} = 0.05\widehat{K}_{i,\text{best}}, \tag{15}$$

where $p, q \in \{1, 2, \ldots, i-1, i+1, \ldots, K\}$, $\mu \in [0, 1]$.

2.1.6. Main Procedure of the Krill Herd Algorithm. In general, the defined motions frequently change the position of a krill individual toward the best fitness. The foraging motion and the motion induced by other krill individuals contain two global and two local strategies. These are working in parallel which make KH a powerful algorithm. Using different effective parameters of the motion during the time, the position vector of a krill individual during the interval t to $t + \Delta t$ is given by the following equation:

$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt}. \tag{16}$$

It should be noted that Δt is one of the most important constants and should be carefully set according to the optimization problem. This is because this parameter works as a scale factor of the speed vector. Δt can be simply obtained from the following formula:

$$\Delta t = C_t \sum_{j=1}^{\text{NV}} \left(\text{UB}_j - \text{LB}_j\right), \tag{17}$$

where NV is the total number of variables and LB_j, UB_j are lower and upper bounds of the$j$th variables ($j = 1, 2, \ldots, \text{NV}$), respectively. C_t is a constant number between $[0, 2]$ (Algorithm 1).

2.2. Free Search Operator. Free search (FS) [26] models the behavior of animals and operates on a set of solutions called population. In the algorithm each animal has original peculiarities called sense and mobility. The sense is an ability of the animal for orientation within the search space. The animal uses its sense for selection of location for the next step. Different animals can have different sensibilities. It also varies during the optimization process, and one animal can have different sensibilities before different walks. The animal can select, for start of the exploration walk, any location

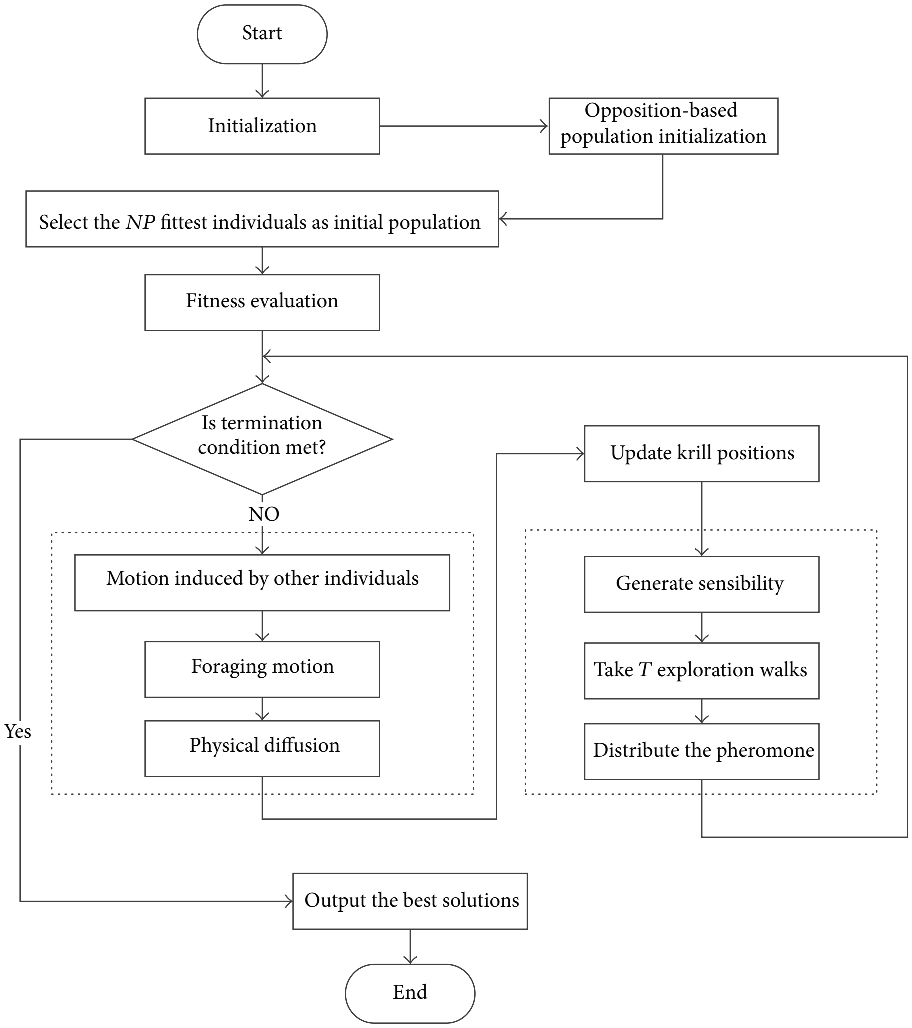

Start
Initialization
Opposition-based population initialization
Select the *NP* fittest individuals as initial population
Fitness evaluation
Is termination condition met?
NO
Yes
Motion induced by other individuals
Foraging motion
Physical diffusion
Update krill positions
Generate sensibility
Take *T* exploration walks
Distribute the pheromone
Output the best solutions
End

FIGURE 1: Flowchart of the free search krill algorithm.

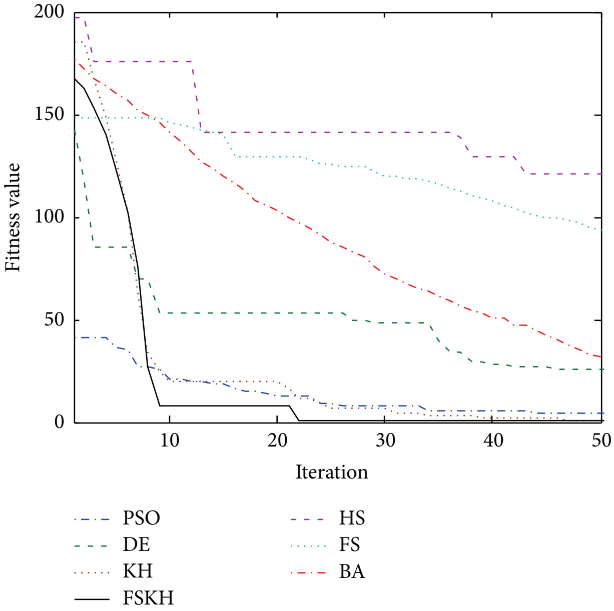


FIGURE 2: Convergence curves for F_1 $(D = 30)$.

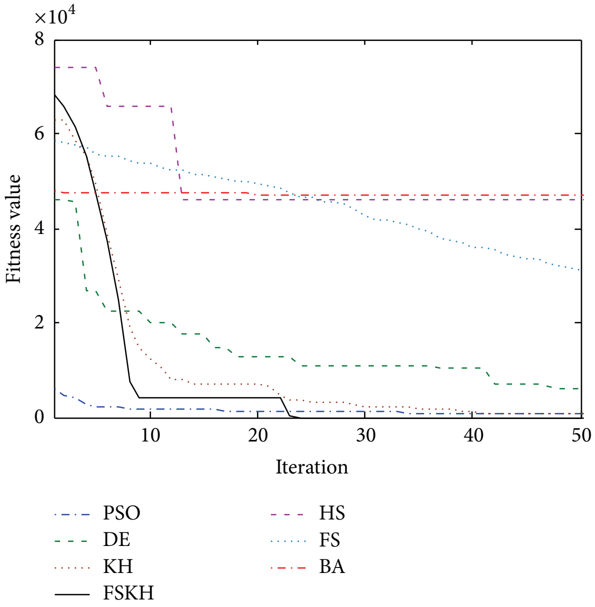


FIGURE 3: Convergence curves for F_2 $(D = 30)$

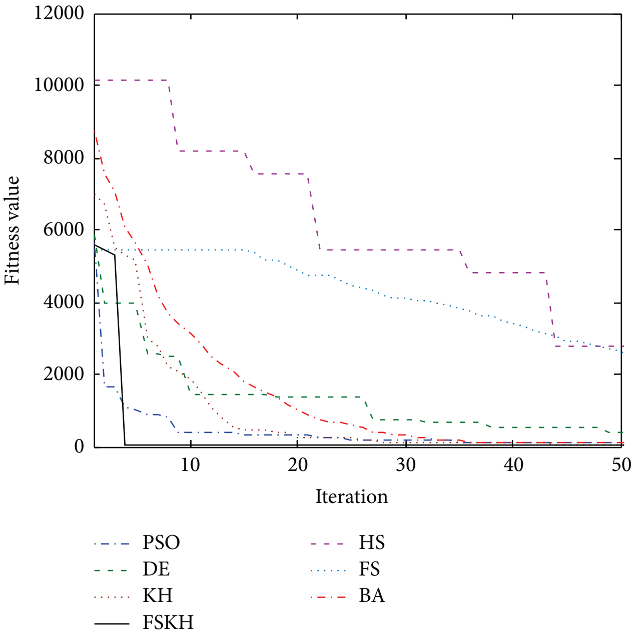


Figure 4: Convergence curves for F_3 $(D = 30)$.

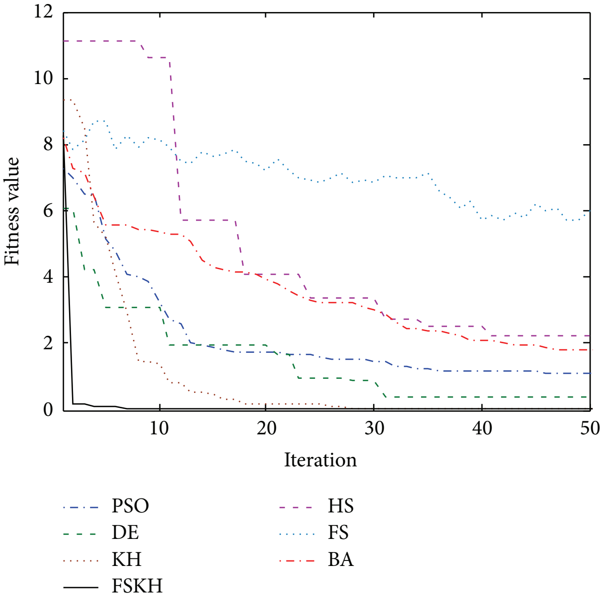


Figure 5: Convergence curves for F_4 $(D = 30)$.

marked with pheromone, which fits its sense. During the exploration walk the animals step within the neighbor space. The neighbor space also varies for the different animals. Therefore, the probability for access to any location of the search space is nonzero.

During the exploration, each krill achieves some favor (an objective function solution) and distributes a pheromone in amount proportional to the amount of the found favor (the quality of the solution). The pheromone is fully replaced with a new one after each walk.

Particularly, the animals in the algorithm are mobile. Each animal can operate either with small precise steps for local search or with large steps for global exploration. And each animal decides how to search (with small or with large steps) by itself. Explicit restrictions do not exist. The previous experience can be taken into account, but it is not compulsory.

2.2.1. The Structure of the FS Operator. The structure of the algorithm consists of three major events: initialization, exploration, and termination.

(1) Initialization. In this paper, the initialization strategy is $x_{0ji} = x_{i\min} + (x_{i\max} - x_{i\min}) * \text{random}_{tji}(0,1)$, where random$(0,1)$ is a random value in $[0,1]$ and $x_{i\min}$, $x_{i\max}$ are the space borders.

(2) Exploration. The exploration walk generates coordinates of a new location x_{tji} as

$$x_{tji} = x_{0ji} - \Delta x_{tji} + 2 * \Delta x_{tji} * \text{random}_{tji}(0,1). \quad (18)$$

The modification strategy is

$$\Delta x_{tji} = R_{ji} * (x_{i\max} - x_{i\min}) * \text{random}_{tji}(0,1), \quad (19)$$

where T is step limit per walk and $t = 1, \ldots, T$ is current step.

The individual behavior, during the walk, is modeled and described as $f_{tj} = f(x_{tji})$, $f_j = \max(f_{tj})$, where f_j is the only location marked with pheromone from an animal after the walk.

The pheromone generation is

$$P_j = \frac{f_j}{\max(f_j)}. \quad (20)$$

The sensibility generation is

$$S_j = S_{\min} + \Delta S_j,$$
$$\Delta S_j = (S_{\max} - S_{\min}) * \text{random}_j(0,1), \quad (21)$$

where $S_{\min}$ and $S_{\max}$ are minimal and maximal possible values of the sensibility. $P_{\min}$ and $P_{\max}$ are minimal and maximal possible values of the pheromone trials. And $P_{\max} = S_{\max}$, $P_{\min} = S_{\min}$.

Selection and decision making for a start location x'_{0ji} for an exploration walk is

$$x'_{0ji} = \begin{cases} x'_{0ji}, & (P_k < S_j) \\ x_{ji}, & (P_k \geq S_j), \end{cases} \quad (22)$$

where $j = 1, \ldots, \text{NK}$, $k = 1, \ldots, \text{NK}$, and k is the marked locations number.

(3) Termination. In this paper, the criterion for termination is *iter* > *iterMax*, where *iterMax* is the maximum number of iterations.

The steps are imitation of the ability for motion and action. The steps can be large or small and can vary. In the search process, the neighbor space is a tool for tuning rough and precise searches. So, search radius R_j is a parameter related to individual j search space; the value of R_j decides the optimization quality during the search process.

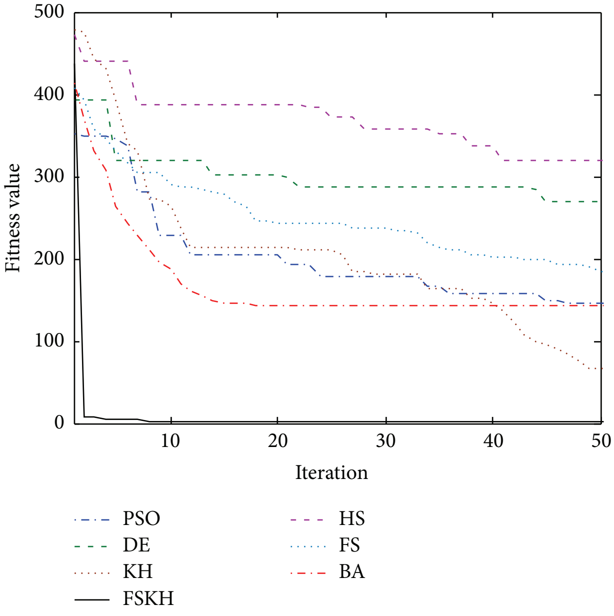


Figure 6: Convergence curves for F_5 $(D = 30)$.

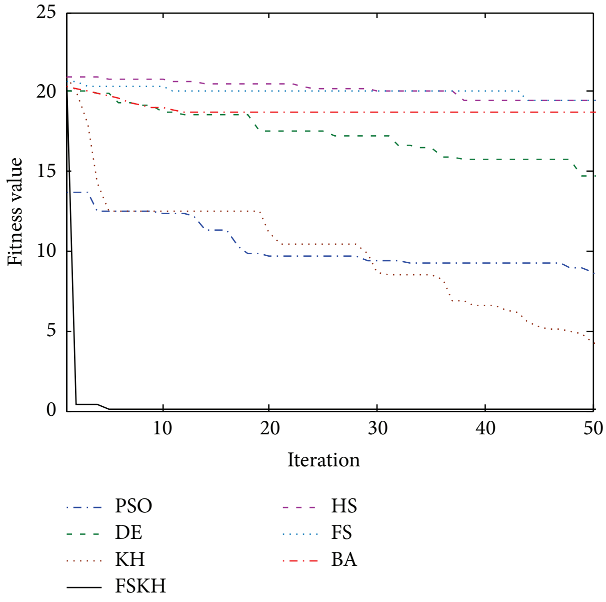


Figure 7: Convergence curves for F_6 $(D = 30)$.

There are two methods to set the value of search radius. The first one is that R_j is constant. If the value is higher, the individual search space is wider, search time is longer, and the convergence precision is lower. If the value is lower, the individual search space is smaller and the convergence precision is higher.

The second method is that changing neighbor space R_j adaptively. And during the search process, R_j is decreasing gradually. The rule is as follows:

$$R_j(t) = \rho * R_j(t-1), \tag{23}$$

where t is the exploration generation and $0 \le \rho \le 1$ is the radius contract coefficient, which is an important parameter. In this paper, we adopt the first approach.

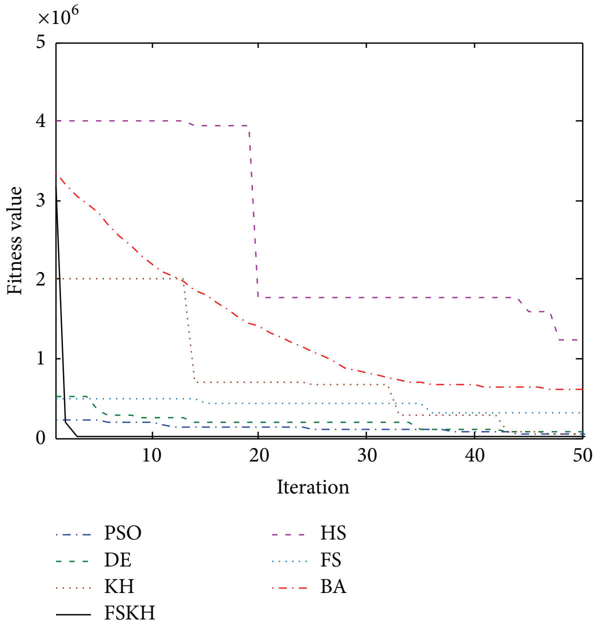


Figure 8: Convergence curves for F_7 $(D = 30)$.

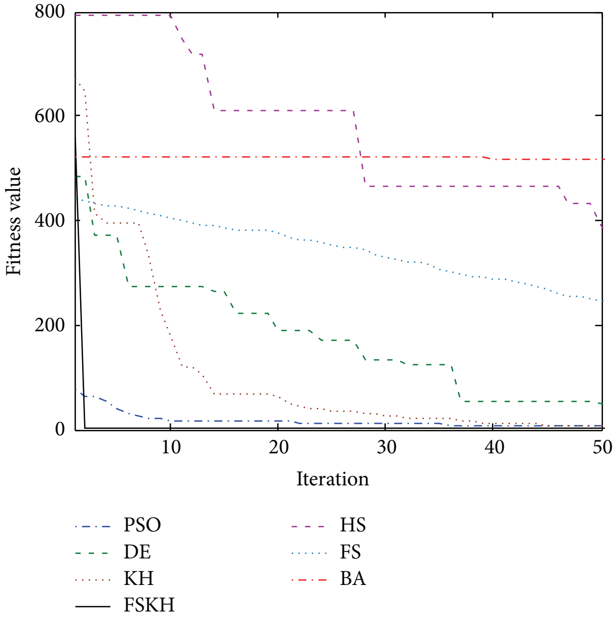


Figure 9: Convergence curves for F_8 $(D = 30)$.

2.2.2. Procedure of FS Operator. The detailed process of FS operator is described as in Algorithm 2.

3. Free Search krill Herd Algorithm

In KH algorithm, krill is influenced by its “neighbors” and the optimal krill, and the sensing distance of each krill is fixed. But in nature, the action of each krill is free and uncertain. In order to simulate this freedom, uncertain individual behavior of the krill, this paper introduces the free search strategy into

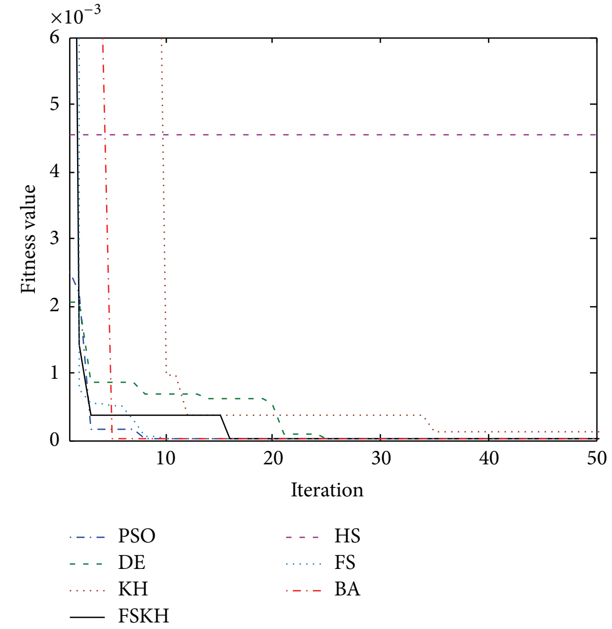


Figure 10: Convergence curves for F_9 ($D = 30$).

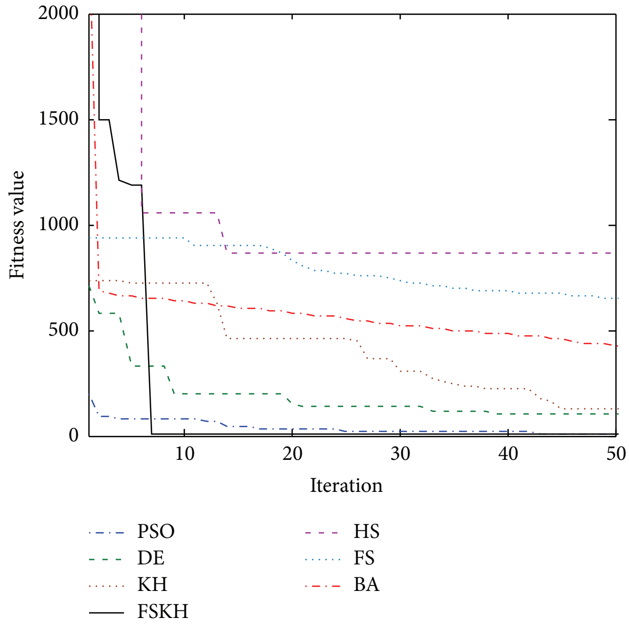


Figure 11: Convergence curves for F_{10} ($D = 30$).

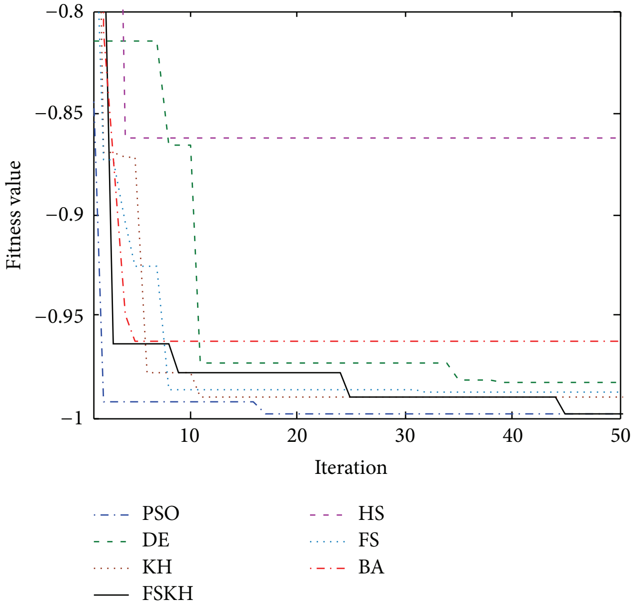


Figure 12: Convergence curves for F_{11} ($D = 2$).

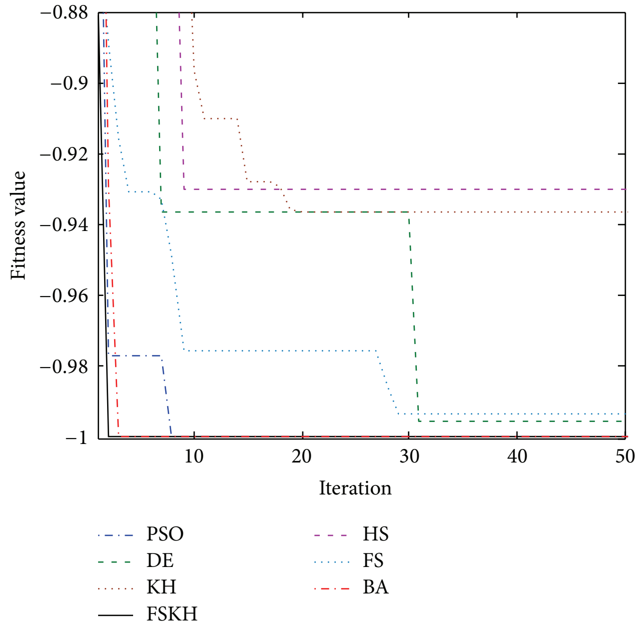


Figure 13: Convergence curves for F_{12} ($D = 2$).

the krill herd algorithm and proposes a novel free search krill herd algorithm (FSKH).

3.1. Opposition-Based Population Initialization. Population initialization has an important impact on the optimization results and global convergence; this paper introduces the initialization method of opposition based learning (OBL) strategy [27] to generate initial krill populations (Algorithm 3).

By utilizing OBL we can obtain fitter starting candidate solutions even when there is no knowledge about the solutions. This initialization method can make a more uniform distribution of the krill populations. Therefore, it is good for the method to get better optimization results. And by utilizing free search strategy, each krill individual in FSKH can decide how to search by itself (Algorithm 4). The strategy allows nonzero probability to approach to any location of the search space and highly encourages the individuals to escape from trapping in local optimal solution. During the search process, each krill takes exploration walks according to different search radius.

In general, three main actions in standard KH algorithm can guide the krill individuals to search the promising

Table 1: Benchmark functions.

Category	Number	Name	Benchmark function	Scope	$f_{\min}$
I	F_1	Sphere	$f(x)=\sum_{i=1}^{n}x_i^2$	$[-5.12, 5.12]$	0
	F_2	Step	$f(x)=\sum_{i=1}^{n-1}(\lfloor x_i+0.5\rfloor)^2$	$[-100, 100]$	0
	F_3	Rosenbrock	$f(x)=\sum_{i=1}^{n-1}\left[(x_i-1)^2+100(x_i^2-x_{i+1})^2\right]$	$[-2.048, 2.048]$	0
	F_4	Quartic	$f(x)=\sum_{i=1}^{n}x_i^4+\text{random}\,[0,1)$	$[-1.28, 1.28]$	0
	F_5	Rastrigin	$f(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	$[-5.12, 5.12]$	0
	F_6	Ackley	$f(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right)\right)+20+e$	$[-32.768, 32.768]$	0
	F_7	Quadric	$f(x)=\sum_{j=1}^{n}\left(\sum_{i=1}^{j}x_i\right)^2$	$[-100, 100]$	0
	F_8	Griewank	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}(x_i^2)-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-600, 600]$	0
	F_9	Alpine	$f(x)=\sum_{i=1}^{n}\lvert x_i\sin(x_i)+0.1x_i\rvert$	$[-10, 10]$	0
	F_{10}	Zakharov	$f(x)=\sum_{i=1}^{n}x_i^2+\left(\sum_{i=1}^{n}0.5ix_i\right)^2+\left(\sum_{i=1}^{n}0.5ix_i\right)^4$	$[-5, 10]$	0
II	F_{11}	Schaffer's F6	$f(x)=\frac{\sin^2\sqrt{(x_1^2+x_2^2)}-0.5}{\left[1+0.001\,(x_1^2+x_2^2)\right]^2}-0.5$	$[-100, 100]$	-1
	F_{12}	Drop wave	$f(x)=-\frac{1+\cos\left(12\sqrt{x_1^2+x_2^2}\right)}{(1/2)(x_1^2+x_2^2)+2}$	$[-5.12, 5.12]$	-1
	F_{13}	Six-hump camel back	$f(x_1,x_2)=\left(4-2.1x_1^2+\frac{x_1^4}{3}\right)x_1^2+x_1x_2+(-4+4x_2^2)\,x_2^2$	$[-3, 3]$	-1.0316
	F_{14}	Easom	$f(x)=-(-1)^n\left(\prod_{i=1}^{n}\cos^2(x_i)\right)\exp\left(-\sum_{i=1}^{n}(x_i-\pi)\right)^2$	$[-2\pi, 2\pi]$	-1

solution space. But it is easy for the standard KH algorithm to be trapped into local optima, and the performance in high-dimensional cases is unsatisfied. In the FSKH algorithm, the individual can search the promising area with small or large steps. So, the krill individuals can move step by step through multidimensional search space. In nature, the activity range of krill individuals is different. R_{ji} can adjust the activity range of the individual, and there is no explicit restrictions.

Using the free search strategy, the krill individual can search any region of the search space. Each krill individual can search according to their perception and the scope of activities and can not only search around the global optimum, but also search around local optimum. When using larger step, it takes global search which can strengthen the weak global search ability of KH. Therefore, the proposed algorithm has better population diversity and convergence speed and can enhance the global searching ability of the algorithm. To achieve a better balance between local search and global search, FSKH algorithm includes both "exploration" process of FS and "exploitation" process of KH. When increasing the sensitivity, the krill individual will approach the whole population's current best value (i.e., local search). While reducing the sensitivity, the krill individual can search around other neighborhood (i.e., global search) (Figure 1).

4. Simulation Experiments

4.1. Simulation Platform. All the algorithms compared in this section are implemented in Matlab R2012a (7.14). And experiments are performed on a PC with a 3.01 GHz, AMD Athlon (tm) II X4 640 Processor, 3 GB of RAM, and Windows XP operating system. In the tests, population size is NP = 50. The experiment results are obtained in 50 trials.

4.2. Benchmark Functions. In order to verify the effectiveness of the proposed algorithm, we select 14 standard benchmark functions [28–30] to detect the searching capability of the proposed algorithm. The proposed algorithm in this paper (i.e., FSKH) is compared with PSO, DE, KH, HS, FS, and BA.

4.3. Parameter Setting. Generally, the choice of parameters requires some experimenting. In this paper, after a lot of experimental comparison, the parameters of the algorithm are set as follows.

In KH and FSKH, the maximum induced speed $N^{\max}$ = 0.01, the foraging speed V_f = 0.02, and the maximum

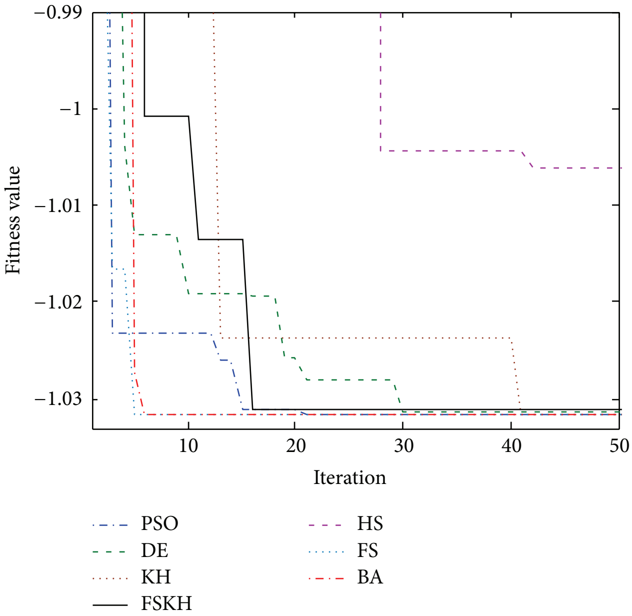


Figure 14: Convergence curves for F_{13} $(D = 2)$.

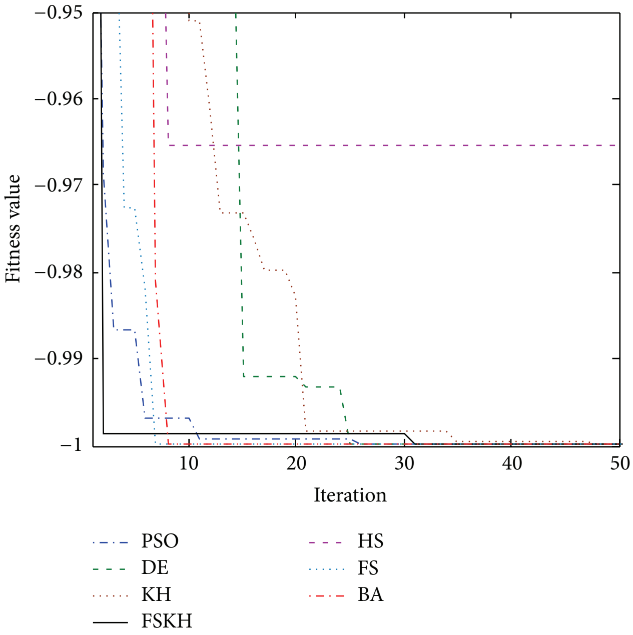


Figure 15: Convergence curves for F_{14} $(D = 2)$.

diffusion speed $D^{\max} = 0.005$. In FSKH and FS, the search radius

$$R_j = \begin{cases} 1, & j = 1, \dots, \dfrac{\text{NP}}{3} \\ 0.5, & j = \dfrac{\text{NP}}{3} + 1, \dots, \dfrac{2\text{NP}}{3}; \\ 0.1, & j = \dfrac{2\text{NP}}{3} + 1, \dots, \text{NP}. \end{cases} \quad (24)$$

But in the FSKH, the search step is $T = 5$. In FS, the search step is $T = 50$.

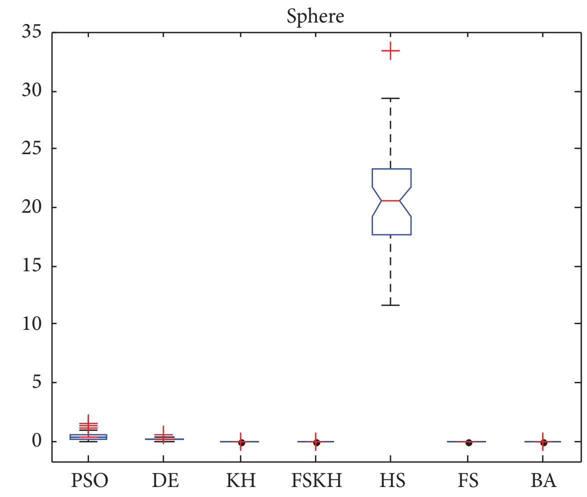


Figure 16: ANOVA tests of the global minimum for F_1 $(D = 30)$.

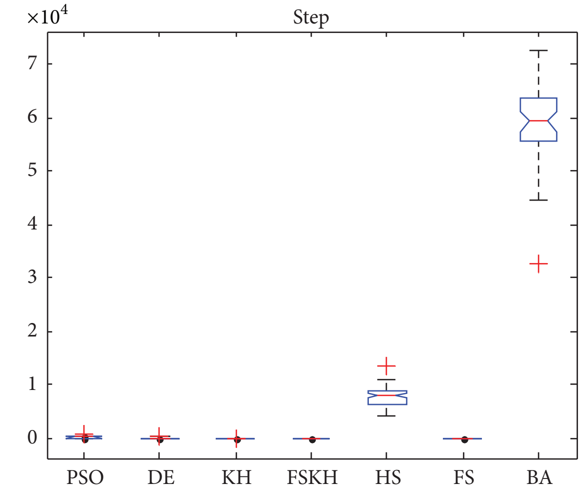


Figure 17: ANOVA tests of the global minimum for F_2 $(D = 30)$.

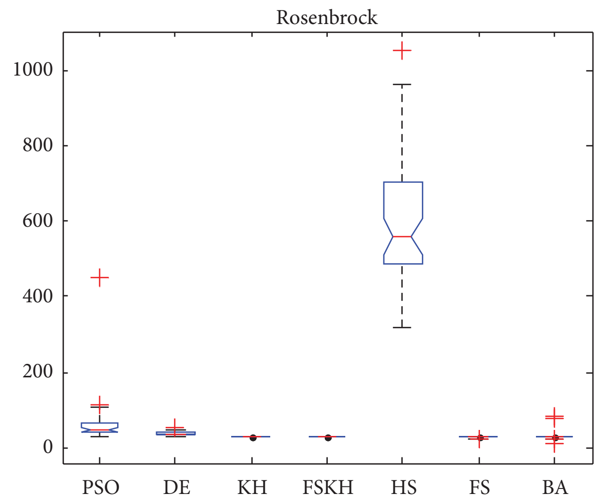


Figure 18: ANOVA tests of the global minimum for F_3 $(D = 30)$.

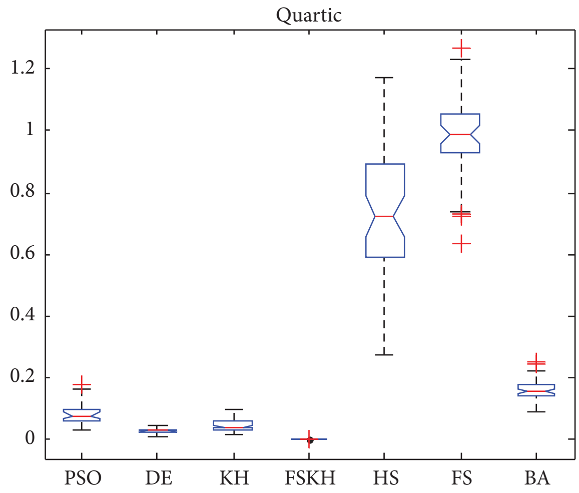


Figure 19: ANOVA tests of the global minimum for F_4 $(D = 30)$.

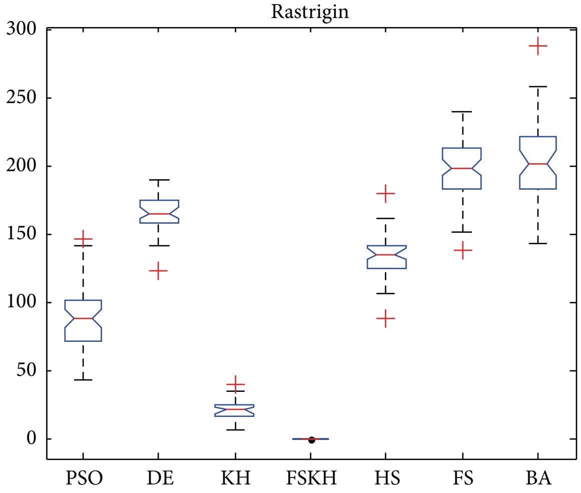


Figure 20: ANOVA tests of the global minimum for F_5 $(D = 30)$.

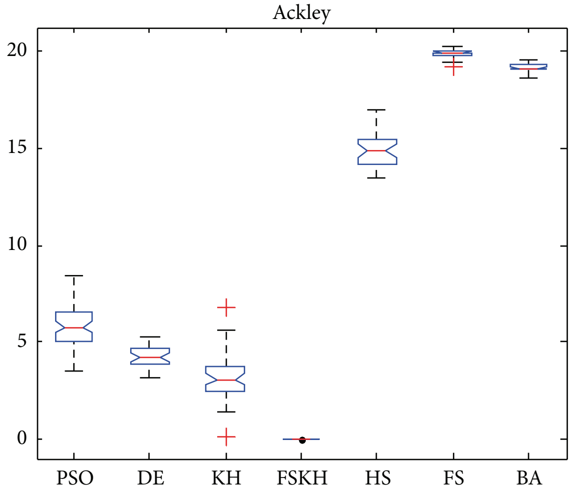


Figure 21: ANOVA tests of the global minimum for F_6 $(D = 30)$.

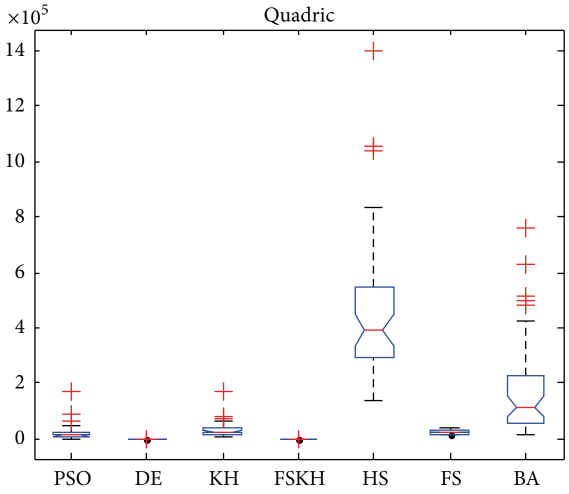


Figure 22: ANOVA tests of the global minimum for F_7 $(D = 30)$.

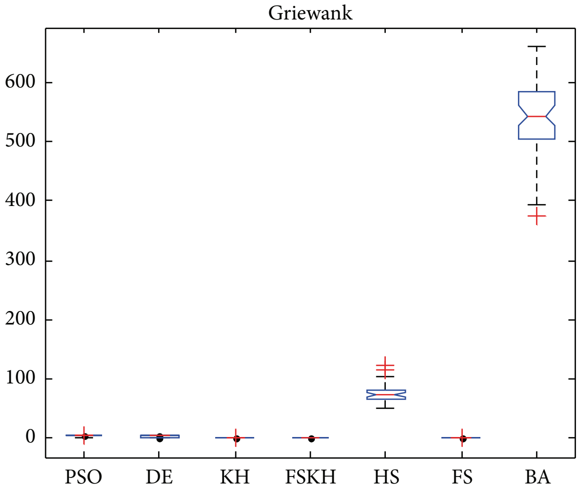


Figure 23: ANOVA tests of the global minimum for F_8 $(D = 30)$.

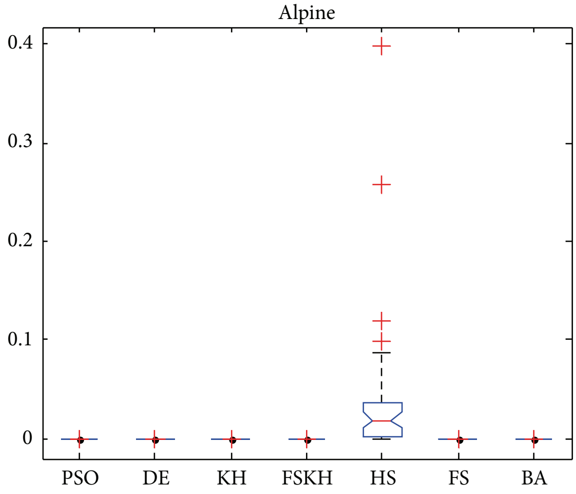


Figure 24: ANOVA tests of the global minimum for F_9 $(D = 30)$.

```
Begin
    Initialization. Set the generation counter $iter = 1$; initialize the population $P$ of NP krill individuals randomly and each
            krill corresponds to a potential solution to the given problem; set the foraging speed $V_f$, maximum diffusion
            speed $D^{\max}$, and maximum induced speed $N^{\max}$;
    Fitness evaluation. Evaluate each krill individual according to its position.
        While the termination criterion is not satisfied or $iter < iterMax$ Generation do
            Sort the population krill from best to worst.
            for $i = 1 : \mathrm{NP}$ (all krill) do
                Perform the following motion calculation.
                    Motion induced by the presence of other individuals
                    Foraging motion
                    Physical diffusion
                Implement the genetic operators.
                Update the krill individual position in the search space.
                Evaluate each krill individual according to its new position.
            end for
            Sort the population krill from best to worst and find the current best.
            $iter = iter + 1$.
        End While
            Post-processing the results and visualization.
End
```

Algorithm 1: Krill herd algorithm.

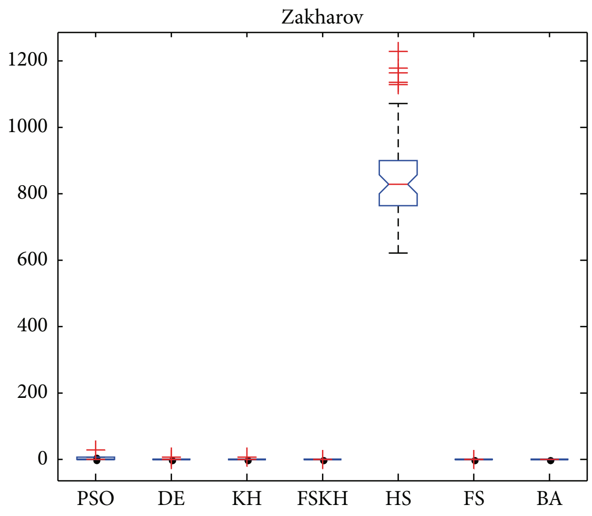


Figure 25: ANOVA tests of the global minimum for F_{10} ($D = 30$).

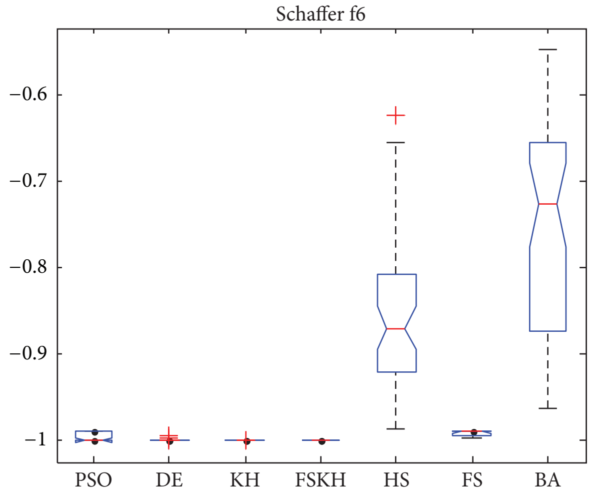


Figure 26: ANOVA tests of the global minimum for F_{11} ($D = 2$).

In BA, the parameters are generally set as follows: pulse frequency range is $Q_i \in [0, 2]$, the maximum loudness is $A^0 = 0.5$, maximum pulse emission is $r^0 = 0.5$, attenuation coefficient of loudness is $\alpha = 0.95$, and increasing coefficient of pulse emission is $\gamma = 0.05$. In PSO, we use linear decreasing inertia weight $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, and learning factor is $C_1 = C_2 = 1.4962$. In HS, the harmony consideration rate is HMCR = 0.9, the minimum pitch adjusting rate is $\mathrm{PAR}_{\min} = 0.4$, the maximum pitch adjusting rate is $\mathrm{PAR}_{\max} = 0.9$, the minimum bandwidth is $\mathrm{bw}_{\min} = 0.0001$, and the maximum bandwidth is $\mathrm{bw}_{\max} = 1$. In DE, the crossover constant is $P_{\mathrm{cr}} = 0.5$.

4.4. Comparison of Experiment Results. The best, mean, worst, and Std. represent the optimal fitness value, mean fitness value, worst fitness value, and standard deviation, respectively. Bold and italicized results mean that FSKH is better, while the $*$ results means that other algorithm is better.

For the low-dimensional case, the maximum number of iterations of each algorithm is $iterMax = 500$. As seen from Tables 2 and 3, FSKH provides better results than other algorithms except F_3, F_{13}, and F_{14}. What is more, FSKH can find the theoretical optimum solutions for nine benchmark functions (F_1, F_2, F_5, $F_7 \sim F_{12}$) and has a very strong robustness. For other algorithms, only PSO can find the theoretical optimum solution for three functions

Table 2: Simulation results for F_1 ~ F_9 in low dimension.

Benchmark function	Result	Method						
		PSO	DE	KH II	FSKH	HS	FS	BA
F_1 (D = 30)	Best	1.8151E − 02	1.2075E − 01	1.1148E − 04	**0**	1.2072E + 01	4.5486E − 02	9.4991E − 04
	Mean	4.5954E − 01	2.4943E − 01	1.6110E − 03	**6.1254E − 316**	3.4612E + 01	5.9487E − 02	1.4434E − 03
	Worst	1.6052E + 00	3.7078E − 01	6.4270E − 03	**1.1599E − 314**	5.1108E + 00	7.0424E − 02	1.7796E − 03
	Std.	3.2409E − 01	6.1704E − 02	1.6590E − 03	**0**	1.7649E + 01	6.7912E − 03	1.8339E − 04
	Rank	6	5	3	**1**	7	4	2
F_2 (D = 30)	Best	7.9000E + 01	2.1000E + 01	**0**	**0**	4.0710E + 03	1.6000E + 01	3.2450E + 04
	Mean	2.4706E + 02	8.8740E + 01	2.8200E + 00	**0**	7.8597E + 03	2.3780E + 01	5.9598E + 04
	Worst	6.2900E + 02	2.7900E + 02	8.4000E + 01	**0**	1.8688E + 03	3.0000E + 01	7.2507E + 04
	Std.	1.2618E + 02	5.3506E + 01	1.1845E + 01	**0**	7.1600E + 03	2.9017E + 00	6.9512E + 03
	Rank	5	4	3	**1**	7	2	6
F_3 (D = 30)	Best	2.8953E + 01	3.0199E + 01	2.6378E + 01	2.8703E + 01	3.1922E + 02	2.2637E + 01	**1.0961E + 01**
	Mean	6.2541E + 01	3.6796E + 01	2.7980E + 01	2.8912E + 01	5.8759E + 02	**2.6906E + 01**	3.0216E + 01
	Worst	4.5307E + 02	4.9489E + 01	2.9430E + 01	**2.8991E + 01**	1.0549E + 03	2.9389E + 01	8.3654E + 01
	Std.	6.0248E + 01	3.7560E + 00	8.1537E − 01	**6.1287E − 02**	1.5737E + 02	1.4196E + 00	1.3356E + 01
	Rank	5	6	3	4	7	2	**1**
F_4 (D = 30)	Best	3.0162E − 02	8.6779E − 03	1.1579E − 02	**2.2393E − 05**	2.7531E − 01	6.3723E − 01	8.6847E − 02
	Mean	8.0366E − 02	2.5195E − 02	4.4238E − 02	**7.6115E − 04**	7.3426E − 01	9.7518E − 01	1.6032E − 01
	Worst	1.7465E − 01	4.5602E − 02	9.5740E − 02	**2.1624E − 03**	1.1753E + 00	1.2684E + 00	2.4878E − 01
	Std.	3.3563E − 02	7.4389E − 03	2.0059E − 02	**5.0720E − 04**	2.1038E − 01	1.3336E − 01	3.4310E − 02
	Rank	4	2	3	**1**	7	6	5
F_5 (D = 30)	Best	4.3907E + 01	1.2231E + 02	7.2421E + 00	**0**	8.8691E + 01	1.3828E + 02	1.4257E + 02
	Mean	8.8471E + 01	1.6497E + 02	2.1243E + 01	**0**	1.3383E + 02	1.9791E + 02	2.0262E + 02
	Worst	1.4702E + 02	1.8931E + 02	4.0592E + 01	**0**	1.7953E + 02	2.4035E + 02	2.8786E + 02
	Std.	2.3543E + 01	1.4949E + 01	6.7613E + 00	**0**	1.5616E + 01	2.2151E + 01	2.8986E + 01
	Rank	6	3	2	**1**	4	5	7
F_6 (D = 30)	Best	3.4508E + 00	3.1954E + 00	8.0527E − 02	**8.8818E − 16**	1.3509E + 01	1.9172E + 01	1.8667E + 01
	Mean	5.8518E + 00	4.2090E + 00	3.1470E + 00	**8.8818E − 16**	1.4935E + 01	1.9908E + 01	1.9148E + 01
	Worst	8.4440E + 00	5.2716E + 00	6.8219E + 00	**8.8818E − 16**	1.6955E + 01	2.0247E + 01	1.9503E + 01
	Std.	1.2143E + 00	5.3016E − 01	1.1229E + 00	**0**	8.6116E − 01	2.2907E − 01	1.9351E − 01
	Rank	7	4	6	**1**	5	3	2
F_7 (D = 30)	Best	3.5384E + 03	5.6171E + 01	7.0436E + 03	**0**	1.3848E + 05	1.4414E + 04	1.8780E + 04
	Mean	2.1959E + 04	1.4336E + 02	3.2982E + 04	**4.3250E − 180**	4.6076E + 05	2.3871E + 04	1.7660E + 05
	Worst	1.7426E + 05	4.3311E + 02	1.7281E + 05	**2.1625E − 178**	1.3967E + 06	3.6638E + 04	7.5561E + 05
	Std.	2.7411E + 04	7.5588E + 01	2.7450E + 04	**0**	2.4220E + 05	5.4373E + 03	1.6793E + 05
	Rank	4	2	5	**1**	7	3	6
F_8 (D = 30)	Best	1.0973E + 00	1.1225E + 00	2.0611E − 02	**0**	5.0342E + 01	1.1446E + 00	3.7618E + 02
	Mean	2.0579E + 00	1.6510E + 00	1.2319E − 01	**0**	7.3791E + 01	1.2110E + 00	5.3973E + 00
	Worst	3.9297E + 00	2.5709E + 00	5.1307E − 01	**0**	1.2387E + 02	1.2616E + 00	6.61453177 + 02
	Std.	5.5487E − 01	3.5593E − 01	8.0788E − 02	**0**	1.4831E + 01	2.4008E − 02	6.4097E + 01
	Rank	5	4	3	**1**	6	2	7
F_9 (D = 30)	Best	**0**	1.9429E − 16	1.0908E − 09	**0**	5.3806E − 05	2.8243E − 08	1.2522E − 09
	Mean	9.1221E − 41	1.7080E − 13	6.3405E − 08	**3.7231E − 168**	3.6484E − 02	1.4745E − 06	1.2981E − 06
	Worst	1.9689E − 39	1.1459E − 12	2.5349E − 07	**8.9590E − 167**	3.9778E − 01	1.3508E − 05	2.5807E − 05
	Std.	3.4654E − 40	2.5917E − 13	6.0990E − 08	**0**	6.7720E − 02	2.4162E − 06	3.7830E − 06
	Rank	2	3	4	**1**	7	5	6
F_{10} (D = 30)	Best	4.7818E − 01	2.7822E − 01	4.0704E − 02	**0**	6.2320E + 02	1.6179E − 01	3.2702E − 03
	Mean	3.5156E + 00	9.6176E − 01	1.5103E + 00	**8.6510E − 318**	8.5918E + 02	2.7912E − 01	5.2075E − 03
	Worst	3.1501E + 01	3.1197E + 00	9.2295E + 00	**3.3739E − 316**	1.2293E + 03	3.8473E − 01	7.4336E − 03
	Std.	4.3846E + 00	5.1776E − 01	1.9643E + 00	**0**	1.0523E + 03	4.0208E − 02	1.0260E − 03
	Rank	6	4	5	**1**	7	3	2

Note: In this paper, both KH and KH II represent the KH with crossover operator.

Table 3: Simulation results for $F_{10} \sim F_{13}$.

Benchmark function	Result	Method						
		PSO	DE	KH II	FSKH	HS	FS	BA
F_{11} $(D = 2)$	Best	**−1**	−9.999984E − 01	−9.999997E − 01	**−1**	−9.880465E − 01	−9.982745E − 01	−9.627759E − 01
	Mean	−9.961136E − 01	−9.996264E − 01	−9.999894E − 01	**−1**	−8.445035E − 01	−9.921680E − 01	−7.407210E − 01
	Worst	−9.902840E − 01	−9.941573E − 01	−9.999449E − 01	**−1**	−5.830062E − 01	−9.902681E − 01	−5.482243E − 01
	Std.	4.808128E − 03	8.981117E − 04	1.209463E − 05	**0**	9.752777E − 02	2.869464E − 03	1.250904E − 01
	Rank	2	4	3	**1**	6	5	7
F_{12} $(D = 2)$	Best	**−1**	**−1**	**−1**	**−1**	−9.756697E − 01	−9.999931E − 01	−9.999997E − 01
	Mean	−9.961747E − 01	−9.999996E − 01	−9.821486E − 01	**−1**	−8.443158E − 01	−9.983692E − 01	−7.368404E − 01
	Worst	−9.362453E − 01	−9.999938E − 01	−9.362453E − 01	**−1**	−5.856036E − 01	−9.948806E − 01	−2.888120E − 01
	Std.	1.529461E − 02	1.113912E − 06	2.891639E − 02	**0**	9.704192E − 02	1.421345E − 03	1.667622E − 01
	Rank	3	2	4	**1**	7	6	5
F_{13} $(D = 2)$	Best	**−1.031628E + 00**	**−1.031628E + 00**	**−1.031628E + 00**	**−1.031628E + 00**	−1.031587E + 00	**−1.031628E + 00**	**−1.031628E + 00**
	Mean	**−1.031628E + 00**	**−1.031628E + 00**	**−1.031628E + 00**	−1.031510E + 00	−9.719758E − 01	−1.031628E + 00	−9.989818E − 01
	Worst	**−1.031628E + 00**	**−1.031628E + 00**	**−1.031628E + 00**	−1.031071E + 00	−6.378180E − 01	−1.031627E + 00	−2.154637E − 01
	Std.	**6.728967E − 16**	8.535156E − 16	2.176352E − 09	1.420660E − 04	7.340586E − 02	1.353951E − 07	1.615586E − 01
	Rank	**1**	2	3	5	7	4	6
F_{14} $(D = 2)$	Best	**−1**	**−1**	**−1**	**−1**	−9.938758E − 01	−9.999999E − 01	−1
	Mean	**−1**	**−1**	−9.999999E − 01	−9.998070E − 01	−5.496989E − 01	−9.997912E − 01	−6.000308E − 01
	Worst	**−1**	**−1**	−9.999999E − 01	−9.995538E − 01	−1.236216E − 05	−9.984373E − 01	−8.743099E − 12
	Std.	**0**	**0**	2.026218E − 09	1.291856E − 03	4.099494E − 01	2.968278E − 04	4.948335E − 01
	Rank	**1**	**1**	3	4	7	6	5

Table 4: Simulation results for $F_1 \sim F_9$ in high dimension.

Benchmark function	Result	Method						
		PSO	DE	KH II	FSKH	HS	FS	BA
F_1 ($D = 1000$)	Best	1.7539E + 02	1.1046E + 02	2.2325E + 02	**0**	5.5839E + 03	1.9398E + 03	4.6377E + 00
	Mean	4.2055E + 02	1.2990E + 02	2.3419E + 02	**4.6082E − 316**	5.8593E + 03	2.1829E + 03	5.1666E + 00
	Worst	8.7083E + 02	1.4698E + 02	2.4475E + 02	**4.4691E − 315**	6.0746E + 03	2.2895E + 03	5.7013E + 00
	Std.	1.5335E + 02	8.4319E + 00	8.0663E + 00	**0**	9.3024E + 01	7.1875E + 01	2.4083E − 01
	Rank	4	3	5	**1**	7	6	2
F_2 ($D = 200$)	Best	1.7420E + 03	5.8630E + 03	3.8140E + 03	**0**	2.0879E + 05	3.6160E + 03	5.1312E + 05
	Mean	5.6810E + 03	7.6012E + 03	6.4365E + 03	**0**	2.4108E + 05	4.2361E + 03	5.5451E + 05
	Worst	1.2432E + 04	9.4600E + 03	9.4830E + 03	**0**	1.3406E + 04	4.8140E + 03	6.0191E + 05
	Std.	2.4260E + 03	7.9567E + 02	1.2702E + 03	**0**	2.5040E + 05	2.8971E + 02	1.8030E + 04
	Rank	2	5	4	**1**	6	3	7
F_3 ($D = 100$)	Best	2.2607E + 02	1.8958E + 02	1.0216E + 02	**9.8070E + 01**	4.8892E + 03	1.1008E + 02	9.9091E + 01
	Mean	1.2251E + 03	2.3327E + 02	1.1403E + 02	**9.8701E + 01**	6.6060E + 03	1.1743E + 02	1.0747E + 02
	Worst	4.3896E + 03	2.8370E + 02	1.3766E + 02	**9.8942E + 01**	8.4203E + 03	1.2626E + 02	1.6127E + 02
	Std.	1.1701E + 03	2.0992E + 01	9.4142E + 00	**2.1724E − 01**	9.0378E + 02	2.8202E + 00	2.0822E + 01
	Rank	6	5	3	**1**	7	4	2
F_4 ($D = 1000$)	Best	2.9336E + 00	2.1517E − 01	1.3101E + 00	**6.3543E − 06**	2.6746E + 02	7.7482E + 01	6.9964E + 00
	Mean	2.7344E + 01	3.0597E − 01	1.5941E + 00	**4.4048E − 04**	2.9365E + 02	8.4210E + 01	8.4533E + 00
	Worst	6.4757E + 01	4.0618E − 01	2.0862E + 00	**9.8040E − 04**	3.1329E + 02	8.9597E + 01	9.7956E + 00
	Std.	1.3793E + 01	4.7306E − 02	2.5818E − 01	**3.0041E − 04**	9.2166E + 00	3.2374E + 00	5.4824E − 01
	Rank	4	2	3	**1**	7	6	5
F_5 ($D = 500$)	Best	2.7456E + 03	4.0397E + 03	2.0311E + 03	**0**	6.1828E + 03	6.3281E + 03	3.8981E + 03
	Mean	3.9835E + 03	4.2471E + 03	2.3252E + 03	**0**	6.5147E + 03	6.5574E + 03	4.2476E + 03
	Worst	4.6952E + 03	4.4312E + 03	2.5811E + 03	**0**	6.7919E + 03	6.7360E + 03	4.5534E + 03
	Std.	4.2126E + 02	8.3027E + 01	1.2453E + 02	**0**	1.2874E + 02	9.1246E + 01	1.3686E + 02
	Rank	3	5	2	**1**	6	7	4
F_6 ($D = 300$)	Best	6.3274E + 00	8.0299E + 00	7.0415E + 00	**8.8818E − 16**	1.9882E + 01	2.0898E + 01	1.9386E + 01
	Mean	8.2684E + 00	8.7293E + 00	8.6721E + 00	**8.8818E − 16**	2.0099E + 01	2.1003E + 01	1.9495E + 01
	Worst	1.0209E + 01	9.4457E + 00	1.0111E + 01	**8.8818E − 16**	2.0345E + 01	2.1056E + 01	1.9589E + 01
	Std.	8.7857E − 01	3.0761E − 01	6.4227E − 01	**0**	7.7479E − 02	3.3659E − 02	3.5615E − 02
	Rank	2	4	3	**1**	6	7	5
F_7 ($D = 100$)	Best	2.1725E + 05	2.7428E + 03	4.3171E + 05	**3.8339E − 319**	9.4228E + 06	7.1039E + 05	7.9254E + 05
	Mean	1.5219E + 06	5.7789E + 03	2.8233E + 06	**2.2442E − 280**	3.0414E + 07	1.3329E + 06	4.5801E + 06
	Worst	6.8641E + 06	1.2878E + 04	7.0375E + 06	**1.1221E − 278**	7.1585E + 07	1.8894E + 06	1.6408E + 07
	Std.	1.3202E + 06	2.5486E + 03	1.6109E + 06	**0**	1.3820E + 07	2.6259E + 05	3.7611E + 06
	Rank	3	2	4	**1**	7	5	6
F_8 ($D = 500$)	Best	4.7748E + 01	1.6294E + 02	2.1827E + 02	**0**	7.7044E + 03	1.0245E + 03	1.1878E + 04
	Mean	1.5434E + 02	2.0567E + 02	2.7742E + 02	**0**	8.3385E + 03	1.2015E + 03	1.3091E + 04
	Worst	3.1407E + 02	2.4796E + 02	3.3449E + 02	**0**	8.9522E + 03	1.4065E + 03	1.3825E + 04
	Std.	5.9349E + 01	1.7027E + 01	2.6215E + 01	**0**	2.7756E + 02	7.9418E + 01	3.3184E + 02
	Rank	2	3	4	**1**	6	5	7
F_9 ($D = 200$)	Best	**0**	**0**	3.4191E − 10	**0**	8.9839E − 05	7.9967E − 09	2.4396E − 10
	Mean	7.9602E − 78	9.7855E − 19	5.6362E − 05	**6.1471E − 167**	1.0189E − 02	4.9600E − 07	3.5023E − 07
	Worst	2.6973E − 76	1.3878E − 17	1.7294E − 03	**2.3564E − 165**	1.1317E − 01	2.5864E − 06	6.8099E − 06
	Std.	3.8349E − 77	2.9411E − 18	2.5904E − 04	**0**	1.9049E − 02	5.7052E − 07	1.1021E − 06
	Rank	2	3	5	**1**	7	6	4
F_{10} ($D = 300$)	Best	1.2447E + 02	1.2043E + 02	5.5973E + 02	**0**	9.0756E + 03	2.9821E + 02	7.3081E + 03
	Mean	3.9115E + 02	1.5201E + 02	8.4831E + 02	**1.3901E − 317**	1.0803E + 05	3.6677E + 02	8.3861E + 03
	Worst	8.1348E + 02	1.9321E + 02	1.1894E + 03	**6.4647E − 316**	1.9276E + 06	4.7007E + 02	9.4975E + 03
	Std.	1.8186E + 02	1.6427E + 01	1.6003E + 02	**0**	3.6921E + 05	3.4810E + 01	4.8766E + 02
	Rank	3	2	5	**1**	7	4	6

```
Begin
        Initialize $x_{i\min}$, $x_{i\max}$, NK, NP, $G$, $x_{0ji}$, $T$, $R_{ji}$;
        Take initial walks $f_{tj}(x_{0ji} + \Delta x_t)$;
        Generate the initial pheromone $P_k$;
        Distribute the initial pheromone $P_k \rightarrow x_{kP}$;
        Learn the initial achievements $P_k \rightarrow x_{jkP}$;
    While $(t \leq T)$
        Generate sensibility $S_j$
        Select start locations for a walk $x'_{0j} = x_k(S_j, P_k)$;
        Take exploration walks $f_{tj}(x_{0ji} + \Delta x_t)$;
        Generate a pheromone $P_k$;
        Distribute the pheromone $P_k \rightarrow x_{kP}$;
        Learn the achievements $P_k \rightarrow x_{jkP}$;
    End While
End
```

Algorithm 2: Free search operator.

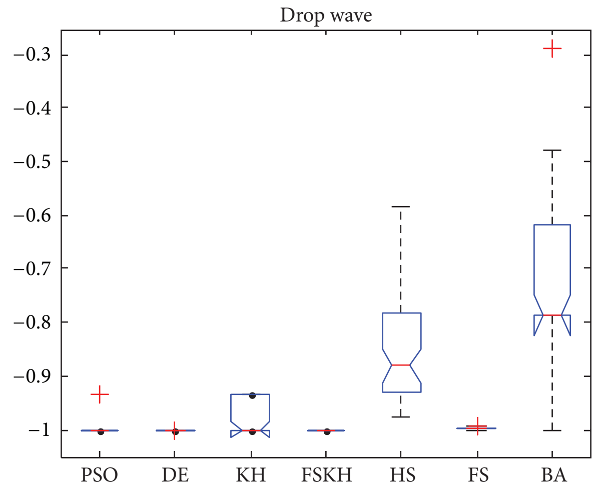


Figure 27: ANOVA tests of the global minimum for F_{12} $(D = 2)$.

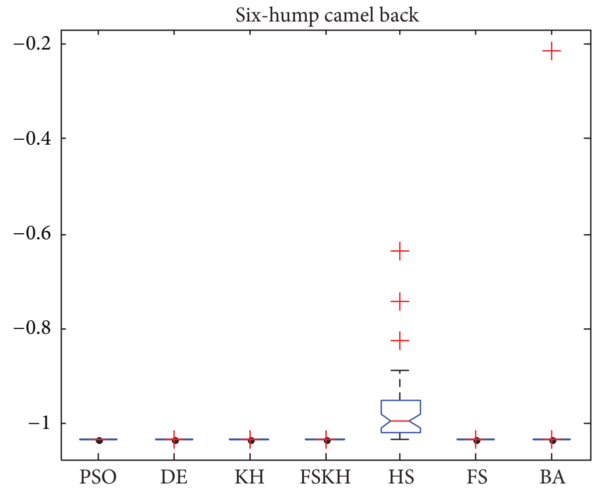


Figure 28: ANOVA tests of the global minimum for F_{13} $(D = 2)$.

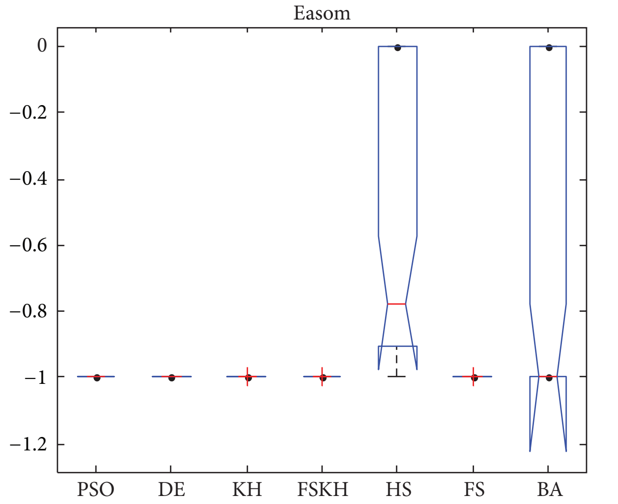


Figure 29: ANOVA tests of the global minimum for F_{14} $(D = 2)$.

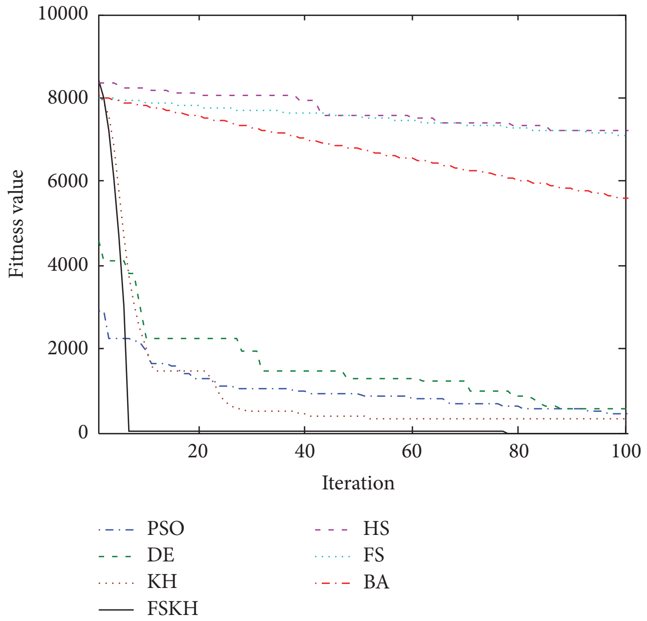


Figure 30: Convergence curves for F_1 $(D = 1000)$.

(F_9, F_{11}, F_{12}), DE can find the theoretical optimum solution for one function (F_{14}), and KH II can find the theoretical optimum solution for one function (F_2). The number of finding the optimal solution for FSKH is more than that of the other six algorithms. For F_4 and F_6 FSKH has a higher precision of optimization. The accuracy of FSKH can be higher than that of other algorithms for 2 and 14 orders of magnitude, respectively, at least. For F_3, F_{13}, and F_{14}, we can find that there is at least one algorithm that can perform better than FSKH. But in general, even for these three functions, the performance of FSKH is highly competitive with other algorithms. For all functions, the standard deviations of FSKH are very small are very small which indicates that FSKH is very robust and efficient.

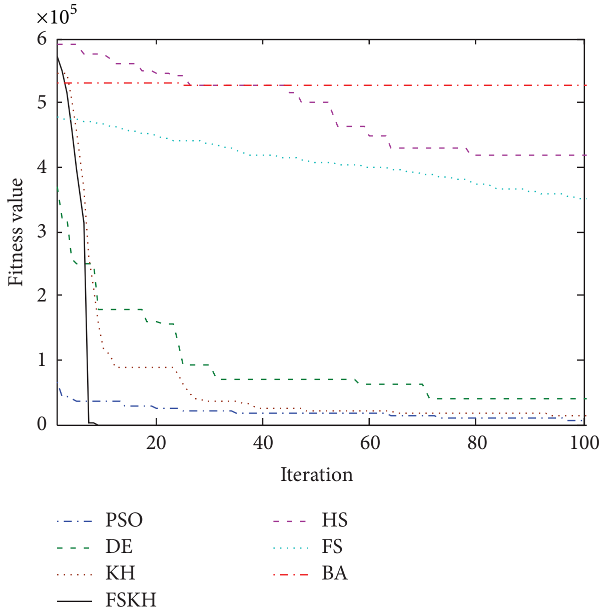


Figure 31: Convergence curves for F_2 ($D = 200$).

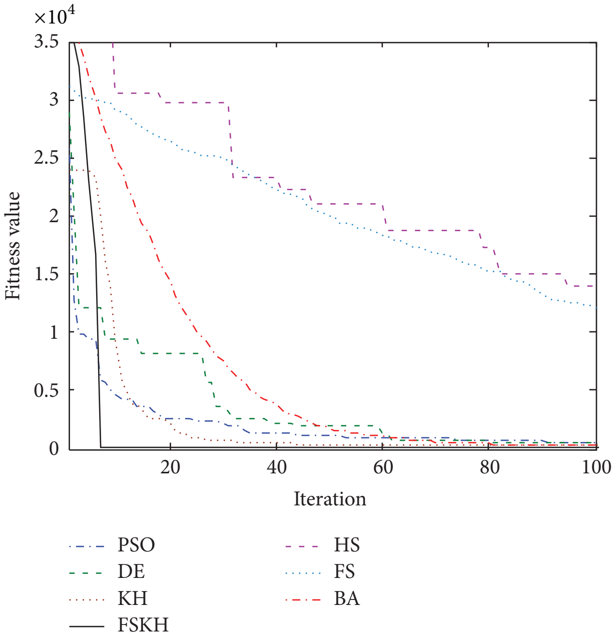


Figure 32: Convergence curves for F_3 ($D = 100$).

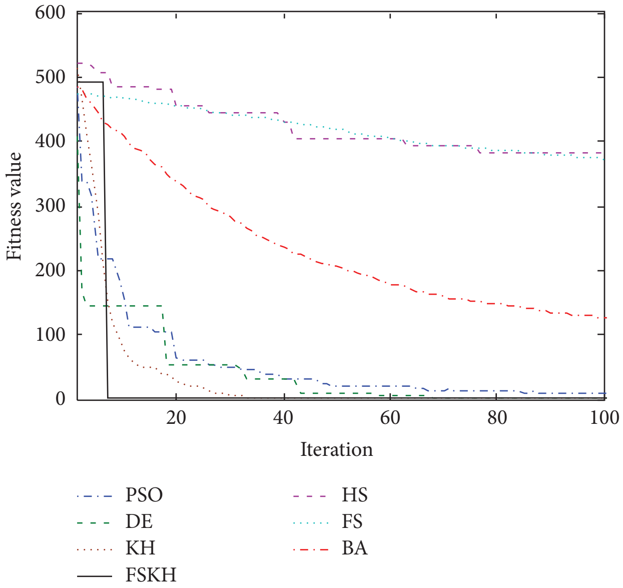


Figure 33: Convergence curves for F_4 ($D = 1000$).

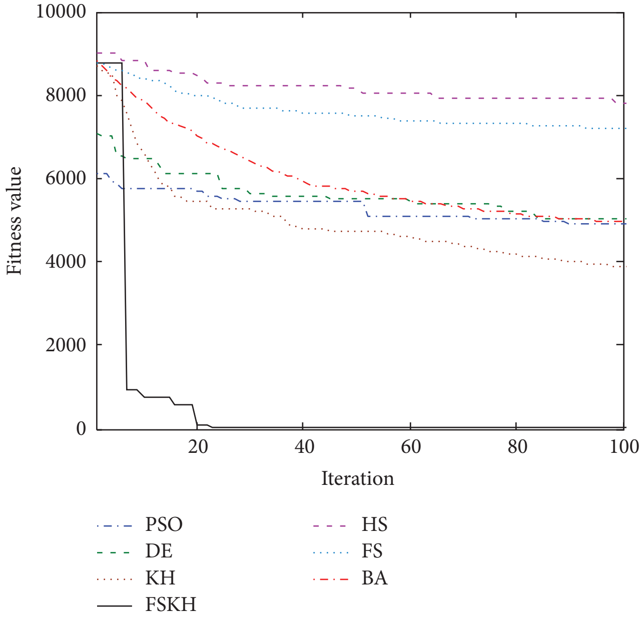


Figure 34: Convergence curves for F_5 ($D = 500$).

For benchmark functions in Table 1, Figures 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15 are the convergence curves, Figures 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, and 29 are the ANOVA tests of the global minimum. Figures 2, 3, 4, 5, 6, 7, 8, 9, and 11 show that the convergence speed of FSKH is quicker than other algorithms. FSKH is the best algorithm for most functions. Moreover, as seen from the ANOVA tests of the global minimum, we can find that FSKH is the most robust method. For the fourteen functions, other algorithms compared (i.e., PSO, DE, KH, HS, FS, and BA) can only be robust for a few functions, but cannot be robust for all functions. Therefore, FSKH is an effective and feasible method for optimization problems in low-dimensional case.

In order to validate the optimization ability of algorithms in high-dimensional case, this paper set F_3 and F_7 to 100-dimension, F_2 and F_9 to 200-dimension, F_6 and F_{10} to 300-dimension, F_5 and F_8 to 500-dimension, F_1 and F_4 to 1000-dimension. The maximum number of iterations of each algorithm is adjusted to $iterMax = 1000$, and other parameters are the same.

The comparison results for high-dimensional case are shown in Table 4. As seen from the results, the optimization

```
For $i = 1$ to NP do
    For $j = 1$ to NK do
        $X_{i,j} = X_{\min,j} + \left(X_{\max,j} - X_{\min,j}\right) * \text{rand};$
    End For
  End For
      /* opposition-based population initialization */
  For $i = 1$ to NP do
      For $j = 1$ to NK do
          $X'_{i,j} = X_{\min,j} + \left(X_{\max,j} - X_{i,j}\right);$
      End For
  End For
      Select the NP fittest individuals from $\{P_0 \cup OP_0\}$ as initial population.
```

Algorithm 3: Opposition-based population initialization.

```
Begin
        Initialization. Set the generation counter $iter = 1$; set the foraging speed $V_f$, maximum diffusion speed $D^{\max}$,
          and the maximum induced speed $N^{\max}$;
              Generate uniformly distributed random population $P_0$;
                Opposition-based population initialization
          For $i = 1$ to NP do
              For $j = 1$ to NK do
                  $X'_{i,j} = X_{\min,j} + \left(X_{\max,j} - X_{i,j}\right);$
              End For
          End For
          Select the NP fittest individuals from $\{P_0 \cup OP_0\}$ as initial population.
        Fitness evaluation. Evaluate each krill individual according to its position.
          While the termination criterion is not satisfied or $iter < iterMax$ Generation do
              Sort the population/krill from best to worst.
              For $i = 1 : \text{NP}$ (all krill) do
                  Perform the following motion calculation.
                          Motion induced by other individuals
                          Foraging motion
                          Free search operation
                          While $(t \le T)$
                                Take exploration walks
                                Modification strategy
                          End While
                  Update the krill individual position in the search space.
                  Evaluate each krill individual according to its new position.
              End For
              Sort the population/krill from best to worst and find the current best.
              $iter = iter + 1$.
        End While
              Post-processing the results and visualization.
End
```

Algorithm 4: Free search krill herd algorithm.

performance of FSKH is the best. Although the dimension of the functions is very high, the proposed algorithm (FSKH) can also find optimum solutions for six benchmark functions (F_1, F_2, F_5, $F_8 \sim F_{10}$). Some algorithms can show a good performance in low-dimensional case, but in the high-dimensional case it cannot get a good result for most benchmark functions. The optimization ability of FSKH does not show a significant decline with increasing the dimension.

Figures 30, 31, 32, 33, 34, 35, 36, 37, 38, and 39 are the convergence curves for high-dimensional case and Figures 40, 41, 42, 43, 44, 45, 46, 47, 48, and 49 are the ANOVA tests of the global minimum for high-dimensional case. As seen from the ANOVA tests of the global minimum, we can find that FSKH is still the most robust method, and superiority is even more obvious. Therefore, in high-dimensional case, FSKH can also be an effective and feasible method for optimization problems. For metaheuristic algorithm, it is important to tune its related parameters. The proposed algorithm is not a very complex metaheuristic algorithm. Only time interval (C_t) and search radius (R_{ji}) should be fine-tuned in FSKH.

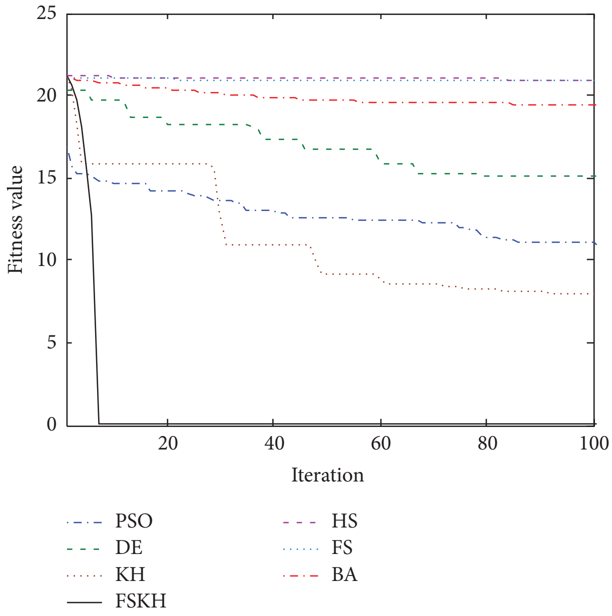


Figure 35: Convergence curves for F_6 $(D = 300)$.

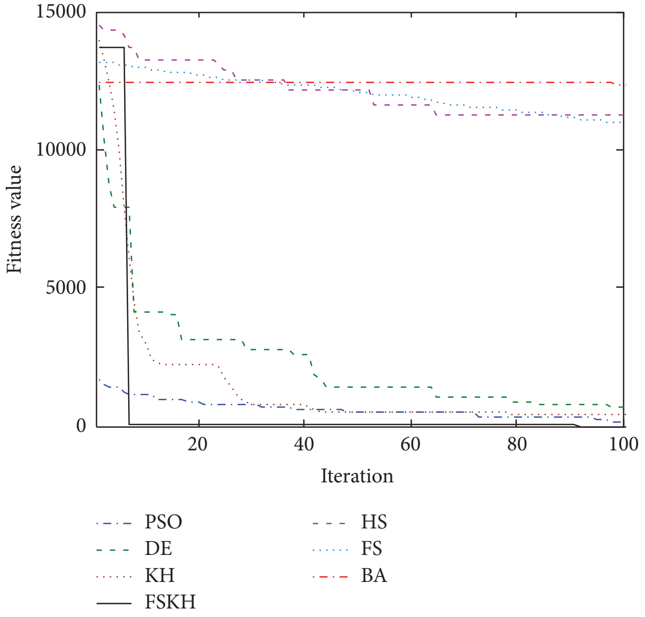


Figure 37: Convergence curves for F_8 $(D = 500)$.

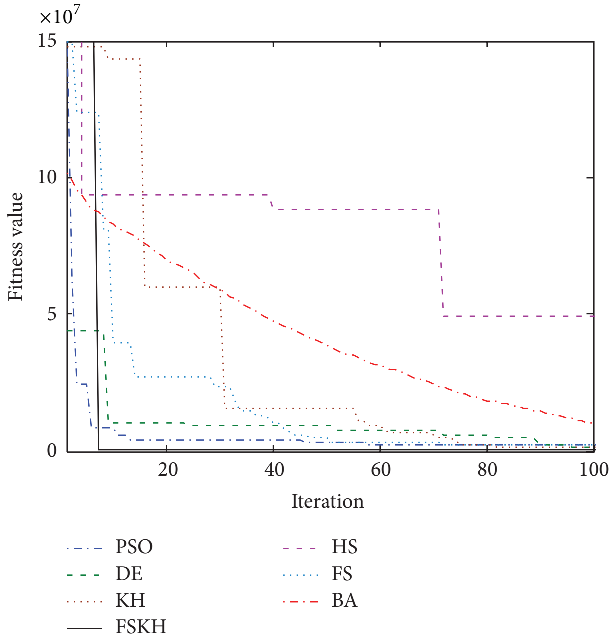


Figure 36: Convergence curves for F_7 $(D = 100)$.

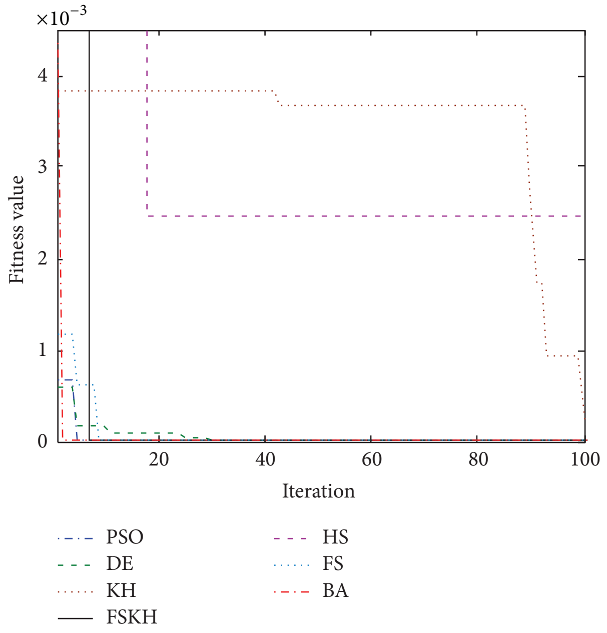


Figure 38: Convergence curves for F_9 $(D = 200)$.

It is a remarkable advantage of the proposed algorithm compared with other algorithms. But in order to get high accuracy solutions, the time complexity of the proposed algorithm is a little high. How to reduce the time complexity of the proposed algorithm by some strategies is our main work in the future.

5. Conclusions

In order to overcome the shortcomings of krill herd algorithm (e.g., poor population diversity, low precision of optimization, and poor optimization performance in high dimensional case). This paper introduces the free search strategy into krill herd algorithm and proposes a novel free search krill herd algorithm (FSKH). The main improvement is that the krill individual can search freely and the diversity of krill population is enriched. The proposed algorithm (FSKH) achieves a better balance between the global search and local search. Experiment simulation and comparison results with other algorithms show that the optimization precision, convergence speed, and robustness of FSKH are all better

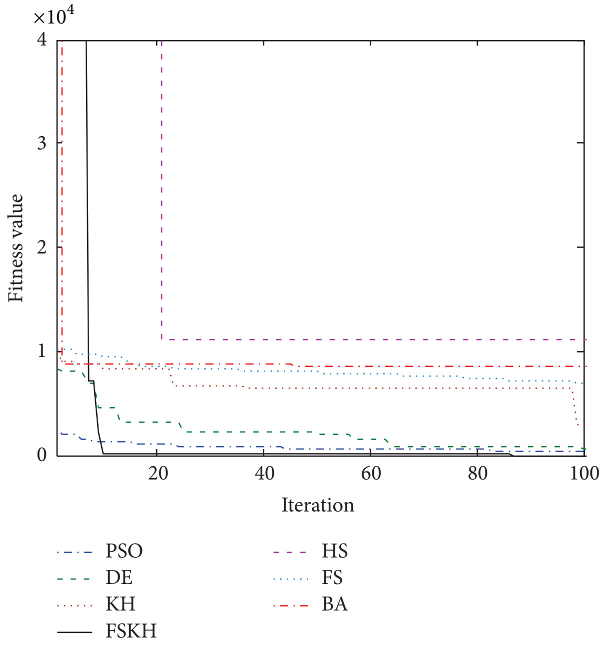


Figure 39: Convergence curves for F_{10} $(D = 300)$.

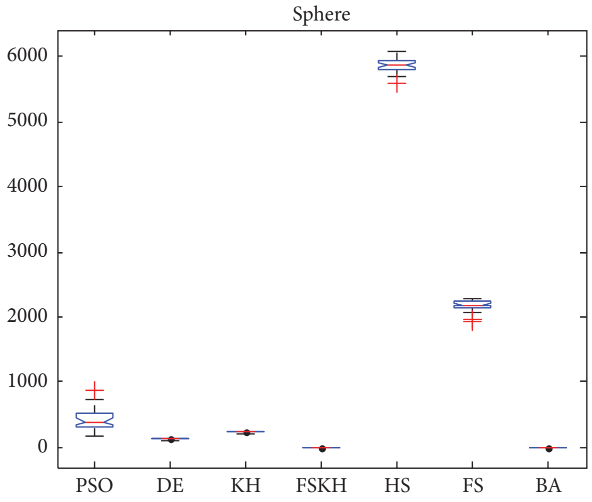


Figure 40: ANOVA tests of the global minimum for F_1 $(D = 1000)$.

than other algorithms for most benchmark functions. So FSKH is a more feasible and effective way for optimization problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Science Foundation of China under Grant no. 61165015, the Key Project of Guangxi

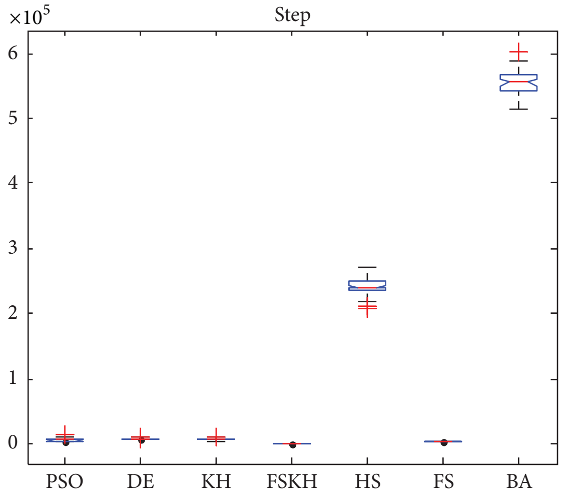


Figure 41: ANOVA tests of the global minimum for F_2 $(D = 200)$.

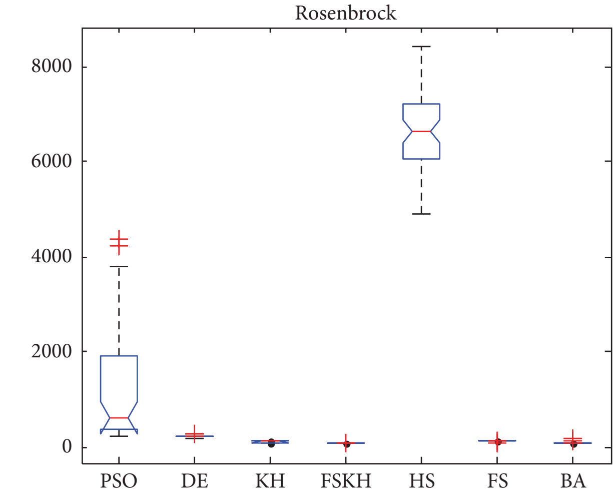


Figure 42: ANOVA tests of the global minimum for F_3 $(D = 100)$.

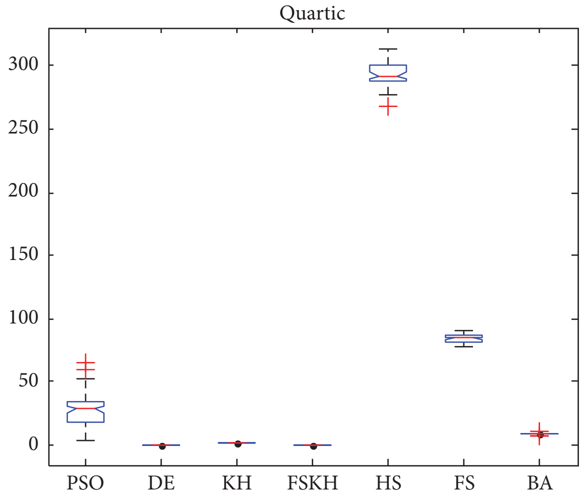


Figure 43: ANOVA tests of the global minimum for F_4 $(D = 1000)$.

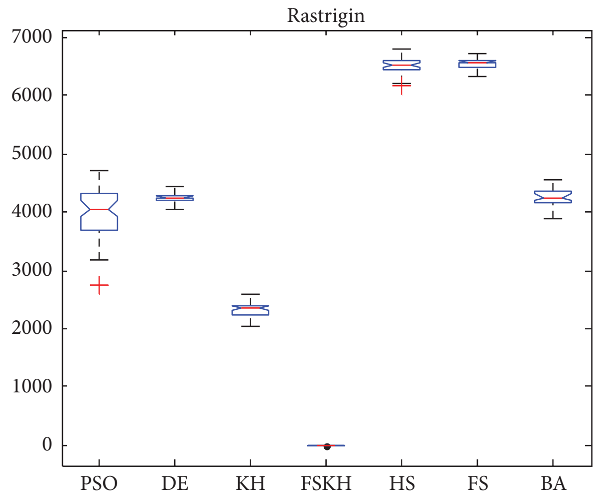


Figure 44: ANOVA tests of the global minimum for F_5 ($D = 500$).

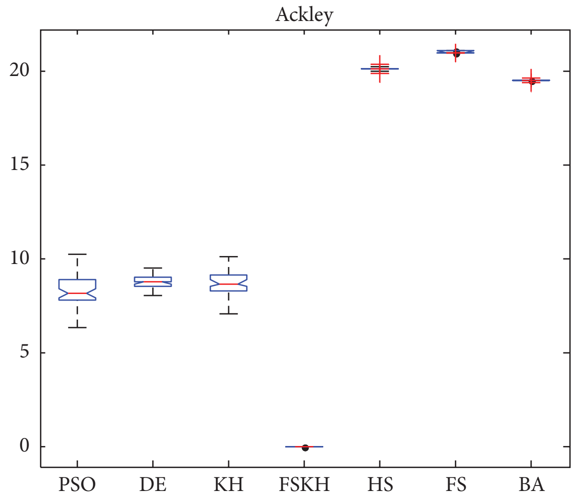


Figure 45: ANOVA tests of the global minimum for F_6 ($D = 300$).

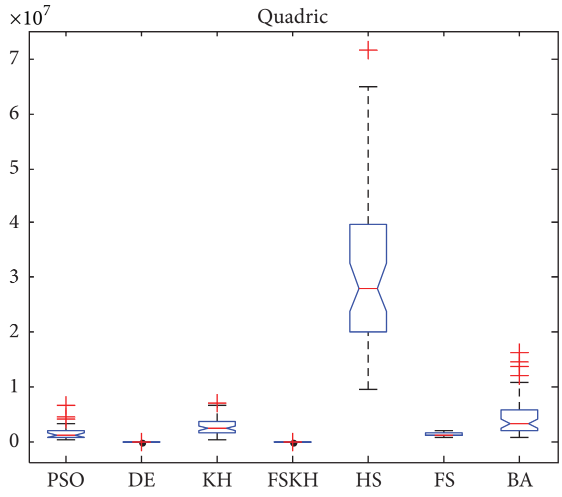


Figure 46: ANOVA tests of the global minimum for F_7 ($D = 100$).

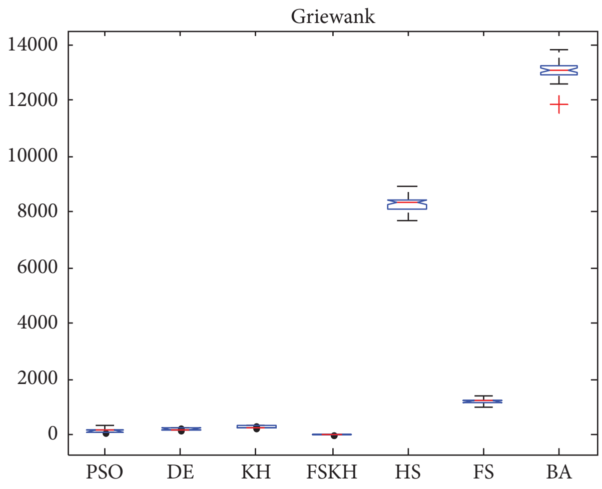


Figure 47: ANOVA tests of the global minimum for F_8 ($D = 500$).

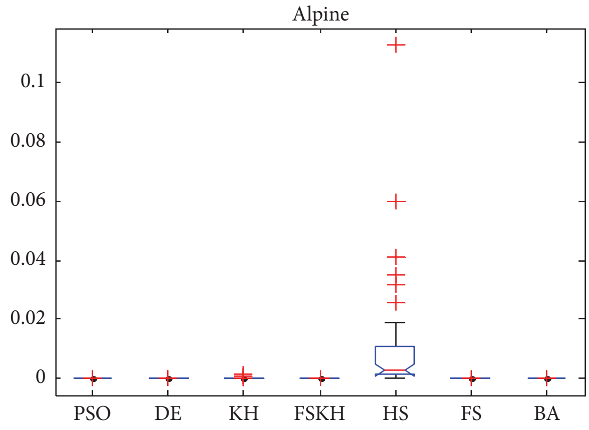


Figure 48: ANOVA tests of the global minimum for F_9 ($D = 200$).

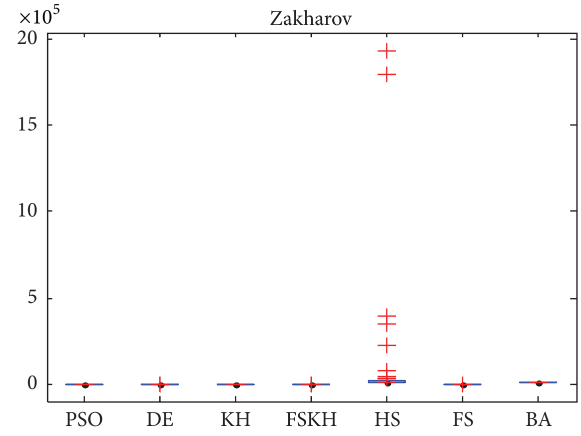


Figure 49: ANOVA tests of the global minimum for F_{10} ($D = 300$).

Science Foundation under Grant no. 2012GXNSFDA053028, the Key Project of Guangxi High School Science Foundation under Grant no. 20121ZD008, and the Funded by Open Research Fund Program of Key Lab of Intelligent Perception and Image Understanding of Ministry of Education of China under Grant no. IPIU01201100.

References

[1] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.

[2] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufman, San Francisco, Calif, USA, 2001.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.

[4] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[5] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[6] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[7] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[8] X. S. Yang, "Firefly algorithms for multimodal optimization," in *System for Automated Geoscientific Analyses*, O. Watanabe and T. Zeug-mann, Eds., vol. 5792, pp. 169–178, LNCS, 2009.

[9] X.-L. Li, Z.-J. Shao, and J.-X. Qian, "Optimizing method based on autonomous animats: fish-swarm Algorithm," *Xitong Gongcheng Lilun yu Shijian/System Engineering Theory and Practice*, vol. 22, no. 11, p. 32, 2002.

[10] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[11] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, December 2009.

[12] R. Q. Zhao and W. S. Tang, "Monkey algorithm for global numerical optimization," *Journal of Uncertain Systems*, vol. 2, no. 3, pp. 164–175, 2008.

[13] X. S. Yang, "A new meta-heuristic bat-inspired algorithm," in *Proceedings of the International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO '10)*, pp. 65–74.

[14] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," *Acta Mechanica*, vol. 213, pp. 267–289, 2010.

[15] X.-S. Yang, "Flower pollination algorithm for global optimization," *Lecture Notes in Computer Science*, vol. 7445, pp. 240–249, 2012.

[16] C. Xu, H. Duan, and F. Liu, "Chaotic artificial bee colony approach to Uninhabited Combat Air Vehicle (UCAV) path planning," *Aerospace Science and Technology*, vol. 14, no. 8, pp. 535–541, 2010.

[17] O. Hasançebi, T. Teke, and O. Pekcan, "A bat-inspired algorithm for structural optimization," *Computers & Structures*, vol. 128, pp. 77–90, 2013.

[18] A. Askarzadeh, "Developing a discrete harmony search algorithm for size optimization of wind-photovoltaic hybrid energy system," *Solar Energy*, vol. 98, pp. 190–195, 2013.

[19] M. Basu and A. Chowdhury, "Cuckoo search algorithm for economic dispatch," *Energy*, vol. 60, pp. 99–108, 2013.

[20] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.

[21] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics*, vol. 2013, Article ID 696491, 21 pages, 2013.

[22] G. Wang, L. Guo, A. H. Gandomi et al., "Lévy-flight krill herd algorithm," *Mathematical Problems in Engineering*, vol. 2013, Article ID 682073, 14 pages, 2013.

[23] G.-G. Wang, L. Guo, A. H. Gandomi, A. H. Alavi, and H. Duan, "Simulated annealing-based krill herd algorithm for global Optimization," *Abstract and Applied Analysis*, vol. 2013, Article ID 213853, 11 pages, 2013.

[24] C. Sur, "Discrete krill herd algorithm—a bio-inspired metaheuristics for graph based network route optimization," in *Distributed Computing and Internet Technology*, R. Natarajan, Ed., vol. 8337 of *Lecture Notes in Computer Science*, pp. 152–163, 2014.

[25] A. H. Gandomi, A. H. Alavi, and S. Talatahari, "Structural Optimization Using Krill Herd Algorithm.," in *Swarm Intelligence and Bio-Inspired Computation Theory and Applications*, X.-S. Yang, Z. Cui, R. Xiao et al., Eds., pp. 335–349, Elsevier, 2013.

[26] K. Penev and G. Littlefair, "Free search—a comparative analysis," *Information Sciences*, vol. 172, no. 1-2, pp. 173–193, 2005.

[27] R. S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.

[28] K. Tang, X. Yao, P. N. Suganthan et al., *Benchmark Functions For the CEC'2008 Special Session and Competition on Large Scale Global Optimization*, University of Science and Technology of China, Hefei, China, 2007.

[29] A. H. Gandomi, X.-S. Yang, S. Talatahari, and S. Deb, "Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization," *Computers and Mathematics with Applications*, vol. 63, no. 1, pp. 191–200, 2012.

[30] A. H. Gandomi, "Benchmark problems in structural optimization," in *Computational Optimization, Methods and Algorithms*, S. Koziel and X. S. Yang, Eds., Study in Computational Intelligence, SCI 356, pp. 259–281, Springer, 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 470821, 11 pages
http://dx.doi.org/10.1155/2014/470821

Research Article

Extracting Credible Dependencies for Averaged One-Dependence Estimator Analysis

LiMin Wang,[1,2] ShuangCheng Wang,[3] XiongFei Li,[1] and BaoRong Chi[4]

[1] *Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China*
[2] *State Key Laboratory of Computer Science, Beijing 100080, China*
[3] *School of Mathematics and Information, Shanghai Lixin University of Commerce, Shanghai 210620, China*
[4] *Medical College, Jilin University, Changchun 130021, China*

Correspondence should be addressed to XiongFei Li; lxf@jlu.edu.cn and BaoRong Chi; chibr@jlu.edu.cn

Received 8 April 2014; Revised 25 May 2014; Accepted 26 May 2014; Published 17 June 2014

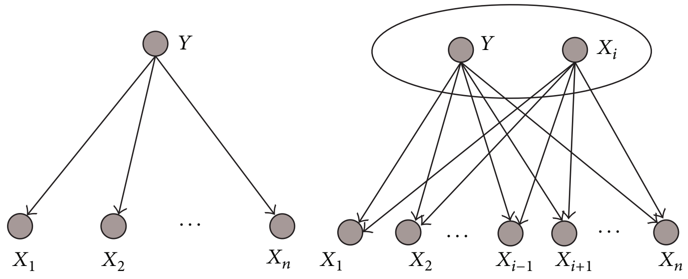
Academic Editor: Yang Xu

Of the numerous proposals to improve the accuracy of naive Bayes (NB) by weakening the conditional independence assumption, averaged one-dependence estimator (AODE) demonstrates remarkable zero-one loss performance. However, indiscriminate superparent attributes will bring both considerable computational cost and negative effect on classification accuracy. In this paper, to extract the most credible dependencies we present a new type of seminaive Bayesian operation, which selects superparent attributes by building maximum weighted spanning tree and removes highly correlated children attributes by functional dependency and canonical cover analysis. Our extensive experimental comparison on UCI data sets shows that this operation efficiently identifies possible superparent attributes at training time and eliminates redundant children attributes at classification time.

1. Introduction

Bayesian networks (BNs) are a key research area of knowledge discovery and machine learning. A BN consists of two parts: a qualitative part and a quantitative part. The qualitative part denotes the graphical structure of the network, while the quantitative part consists of the conditional probability tables (CPTs) in the network. Although BNs are considered efficient inference algorithms, the quantitative part is considered a complex component and learning an optimal BN structure from existing data has been proven to be an NP-hard problem. The graphical structure of naive Bayes (NB) is simple and definite because of the conditional independence assumption between attributes, making NB efficient and effective [1, 2]. However, violations of this conditional independence assumption can make the classification of NB suboptimal. Numerous algorithms have been proposed to retain the desirable simplicity and efficiency of NB while alleviating the problems of the independence assumption. Averaged one-dependence estimator (AODE) [3, 4] utilizes a restricted class of one-dependence estimators (ODEs) and aggregates the predictions of all qualified estimators within this class. A superparent attribute is indiscriminately selected from an attribute set as the parent of all the other attributes in each ODE. By averaging the estimates of all of the three-dimensional estimators, AODE makes a weaker conditional independence assumption than NB. Previous studies that compared different variations of NB techniques prove that AODE is significantly better than other NB techniques in terms of zero-one loss reduction [5]. Since its introduction in 2005, AODE has enjoyed considerable popularity because of its capability to improve the accuracy of NB [5].

Another strategy to remedy violations of the attribute independence assumption is to eliminate highly correlated attributes. Backward sequential elimination (BSE) [6] uses a simple heuristic wrapper approach that selects a subset of the available attributes to minimize zero-one loss on the training set. BSE is effective especially for data sets with highly correlated attributes. Forward sequential selection (FSS) [7] uses the reverse search direction to BSE. However, both FSS



(a) (b)

FIGURE 1: The network structure of NB and AODE.

and BSE have high computational overheads, especially on learning algorithms with high classification time complexity, because they apply the algorithms repeatedly until no accuracy improvement occurs. Subsumption resolution (SR) [8] identifies pairs of attribute values, such that one appears to subsume (be a generalization of) the other and deletes the generalization. Near-subsumption resolution (NSR) [8] is a variant of SR; it extends SR by deleting not only generalizations but also near-generalizations. For different instances, SR and NSR may find different attributes to remove, making them much more flexible than BSE and FSS. Since generalization mainly deals with pairs of attribute, there is no solution for more complicated situation and loop relationship. In this paper, we present a new type of seminaive Bayesian operation, which selects parent (SP) attributes by building maximum weighted spanning tree and removes children (RC) attributes by functional dependency and canonical cover analysis. Thus this algorithm has the advantages of BSE, FSS, and SR.

The remainder of the paper is organized as follows. Section 2 introduces the basic ideas of NB, AODE, and related background theory. Section 3 introduces the SP and RC techniques for attribute selection and elimination with AODE and presents the theoretical justification. Section 4 shows the experimental results on UCI data sets and a detailed analysis of different attribute selection techniques. The final section concludes the paper.

2. Related Research Work

2.1. NB and AODE. The aim of supervised learning is to predict from a training set the class of a testing instance $x = \{x_1, \ldots, x_n\}$, where x_i is the value of the ith attribute. We estimate the conditional probability of $P(y \mid x)$ by selecting $\arg\max_y P(y \mid x)$, where $y \in \{c_1, \ldots, c_k\}$ are the k classes. From Bayes theorem, we have

$$P(yx) = \frac{P(y, x)}{P(x)} \propto P(y, x) \propto P(x \mid y) P(y). \quad (1)$$

NB simplified the estimation of $P(x|y)$ by conditional independence assumption

$$P(xy) = \prod_{i=1}^{n} P(x_i \mid y). \quad (2)$$

Then, the following equation is often calculated in practice rather than (1):

$$P(yx) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y). \quad (3)$$

The corresponding network structure is depicted in Figure 1(a). One advantage of NB is avoiding model selection because selecting between alternative models can be expected to increase variance and allow a learning system to overfit the training data [3]. In consequence, changes in the training data will not lead to any change in NB, which leads in turn to lower variance [4]. By contrast, the underlying conditional probability tables will change correspondingly for those approaches (e.g., NB) with a definite model form when the training data changes, resulting in relatively gradual changes in the pattern of classification.

Numerous techniques have sought to enhance the accuracy of NB by relaxing the conditional independence assumption while attaining the efficiency and efficacy of one-dependence classifiers. Among them, averaged one-dependence estimator (AODE) [3, 4] utilizes a restricted class of one-dependence estimators (ODEs) and aggregates the predictions of all qualified estimators within this class. A superparent attribute (e.g., x_i) is selected as the parent of all the other attributes in each ODE, since

$$\begin{aligned} P(x, y) &= P(x_1, \ldots, x_i, \ldots, x_n, y) \\ &= P(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n, x_i, y) \\ &= P(x_i, y) P(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \mid x_i, y). \end{aligned} \quad (4)$$

Independence is assumed among the remaining attributes given x_i and y. Consider

$$P(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n x_i, y) = \prod_{j=1, j \neq i}^{n} P(x_j \mid x_i, y). \quad (5)$$

Hence, x can be classified by selecting

$$\arg\max_y P(x, y) = \arg\max_y P(x_i, y) \prod_{j=1, j \neq i}^{n} P(x_j \mid x_i, y). \quad (6)$$

The corresponding network structure of AODE is depicted in Figure 1(b). AODE maintains the robustness and much of the efficiency of NB and at the same time exhibits significantly higher classification accuracy for many data sets. Therefore, it has the potential to be a valuable substitute for NB over a considerable range of classification tasks.

2.2. Related Background Theory. In the following discussion, Greek letters ($\alpha, \beta, \gamma, \ldots$) are used to denote sets of attributes. Lowercase letters represent the specific values used by corresponding attributes (e.g., x_i represents $X_i = x_i$). $P(\cdot)$ denotes the probability and $\widehat{P}(\cdot)$ denotes the probability estimation of $P(\cdot)$. Given a relation R (in a relational database), attribute Y of R is functionally dependent on attribute X of R and X of R functionally determines Y of R (in symbols $X \rightarrow Y$). Armstrong (1974) proposed in [9] a set of axioms (or, more precisely, inference rules) to infer all the functional dependencies (FDs) on a relational database, which represent the expert knowledge of the organizational data and their interrelationships. The axioms mainly include the following rules.

(i) Augmentation rule: if $\alpha \rightarrow \beta$ holds and γ is a set of attributes, then $\alpha\gamma \rightarrow \beta\gamma$.

(ii) Transitivity rule: if $\alpha \rightarrow \beta$ holds and $\beta \rightarrow \gamma$ holds, then $\alpha \rightarrow \gamma$.

(iii) Union rule: if $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta\gamma$.

(iv) Decomposition rule: if $\alpha \rightarrow \beta\gamma$ holds, then $\alpha \rightarrow \beta$, $\alpha \rightarrow \gamma$.

(v) Pseudotransitivity rule: if $\alpha \rightarrow \beta$ holds and $\gamma\beta \rightarrow \delta$ holds, then $\alpha\gamma \rightarrow \delta$ holds.

Based on the aforementioned rules, we use the FD rules of probability in [10, 11] to link FD and probability theory. The following rules are included in the FD-probability theory link.

(i) Representation equivalence rule of probability: suppose data set S consists of two attribute sets $\{\alpha, \beta\}$ and β can be inferred by α; that is, the FD $\alpha \rightarrow \beta$ holds; then the following joint probability distribution holds:

$$P(\alpha) = P(\alpha, \beta). \tag{7}$$

(ii) Augmentation rule of probability: if $\alpha \rightarrow \beta$ holds and γ is a set of attributes, then the following joint probability distribution holds:

$$P(\alpha, \gamma) = P(\alpha, \beta, \gamma). \tag{8}$$

(iii) Transitivity rule of probability: if $\alpha \rightarrow \beta$ and $\beta \rightarrow \gamma$ hold, then the following joint probability distribution holds:

$$P(\alpha) = P(\alpha, \gamma). \tag{9}$$

(iv) Pseudotransitivity rule of probability: if $\alpha \rightarrow \beta$ and $\gamma\beta \rightarrow \delta$ hold, then the following joint probability distribution holds:

$$P(\alpha, \gamma) = P(\alpha, \gamma, \delta). \tag{10}$$

In the 1940s, Claude E. Shannon introduced information theory, the theoretical foundation of modern digital communication. Although Shannon was principally concerned with the problem of electronic communications, the theory has much broader applicability. Two commonly used definitions of information theory are described as follows.

Definition 1. Mutual information $I(Y; X)$ measures the information quantity that is transferred between attributes X and Y:

$$I(Y; X) = \sum_{x \in X} \sum_{y \in Y} P(x, y) \log_2 \frac{P(x, y)}{P(x) P(y)}, \tag{11}$$

where $P(x, y)$ is the joint probability distribution function of X and Y, and $P(x)$ and $P(y)$ are the marginal probability distribution functions of X and Y, respectively. High mutual information indicates a great relationship between X and Y; and zero mutual information between two random variables means they are independent.

Definition 2. Conditional mutual information (CMI) $I(X_1; X_2 \mid Y)$ measures the dependence between each pair of attributes $\{X_1, X_2\}$ given Y, which is shown as follows:

$$\begin{aligned} I(X_1; X_2 \mid Y) &= \sum_{y \in Y} P(y) \sum_{x_1 \in X_1} \sum_{x_2 \in X_2} P(x_1, x_2 \mid y) \\ &\quad \times \log_2 \frac{P(x_1, x_2 \mid y)}{P(x_1 \mid y) P(x_2 \mid y)}. \end{aligned} \tag{12}$$

Theorem 3. *Given instance $x = \{x_1, \ldots, x_n\}$ and class label y, if there exists FD $x_2 \rightarrow x_1$, then x_1 is extraneous for classification. That means $P(y \mid x) = P(y \mid x \rightharpoonup x_1)$, where "$\rightharpoonup$" represents the set difference.*

Proof. By applying the augmentation rule and the decomposition rule, from $x_2 \rightarrow x_1$ we can obtain

$$\begin{aligned} \{x \rightharpoonup x_1\} &\longrightarrow x_1, \\ \{x \rightharpoonup x_1, y\} &\longrightarrow x_1. \end{aligned} \tag{13}$$

By applying the representation equivalence rule of probability and the augmentation rule of probability, we can obtain

$$\begin{aligned} P(x) &= P(x \rightharpoonup x_1), \\ P(x, y) &= P(x \rightharpoonup x_1, y). \end{aligned} \tag{14}$$

Then,

$$\begin{aligned} P(y \mid x) &= \frac{P(x, y)}{P(x)} = \frac{P(x \rightharpoonup x_1, y)}{P(x \rightharpoonup x_1)} \\ &= P(y \mid x \rightharpoonup x_1). \end{aligned} \tag{15}$$

We can also prove Theorem 3 from the viewpoint of information theory, since

$$\begin{aligned} I(Y;X) &= -\sum_{x\in X}\sum_{y\in Y}\log_2 P(x,y)\frac{P(x,y)}{P(x)P(y)} \\ &= -\sum_{x\rightharpoonup x_1\in X\rightharpoonup X_1}\sum_{y\in Y}\log_2 P(x\rightharpoonup x_1,y) \\ &\qquad\times\frac{P(x\rightharpoonup x_1,y)}{P(x\rightharpoonup x_1)P(y)} \\ &= I(Y;X\rightharpoonup X_1). \end{aligned} \tag{16}$$

End of the proof. □

As for AODE, (4) will turn to be

$$\begin{aligned} &P(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n \mid x_i,y) \\ &\quad= P(x_2,\ldots,x_{i-1},x_{i+1},\ldots,x_n \mid x_i,y) \\ &\quad= \prod_{j=2,j\neq i}^{n} P(x_j \mid x_i,y). \end{aligned} \tag{17}$$

Thus, if FD $x_2 \rightarrow x_1$ is neglected, the contribution of x_1 to classification will be calculated repeatedly for each ODE and the classification result may be wrong.

3. Attribute Selection and Elimination

AODE makes a weaker attribute conditional independence assumption than that of NB. It selects one attribute as superparent in turn for each ODE submodel, and the other attributes are supposed to be conditionally independent. Previous studies have demonstrated that AODE has a considerably lower bias than that of NB with moderate increases in variance and time complexity [5]. The same attribute may play different roles (either parent or child) in different ODE submodels. In the following discussion, we will repair harmful interdependencies from two viewpoints: (1) select parent (p) attributes (SP) by building maximum weighted spanning tree; (2) remove children (c) attributes (RC) by functional dependency analysis.

3.1. How to Select Parent Attributes. SP selects the branch nodes as the p attributes from maximum weighted spanning tree (MST), the learning procedure of which can be summarized as follows.

(1) Use CMI to measure the weights of edges between each pair of attributes. Sort the edges into descending order by CMI. Let T be the set of edges comprising the MST. Set $T = \{\Phi\}$.

(2) Find the edge with the greatest weight and add this edge to T if and only if it does not form a cycle in T. If no remaining edges exist, exit and report MST to be disconnected.

(3) If T has $n-1$ edges (where n is the number of vertices in MST) stop and output T. Otherwise go to step (2).

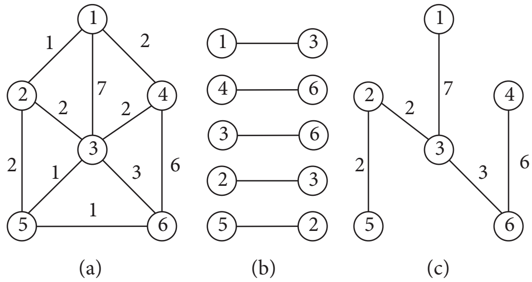


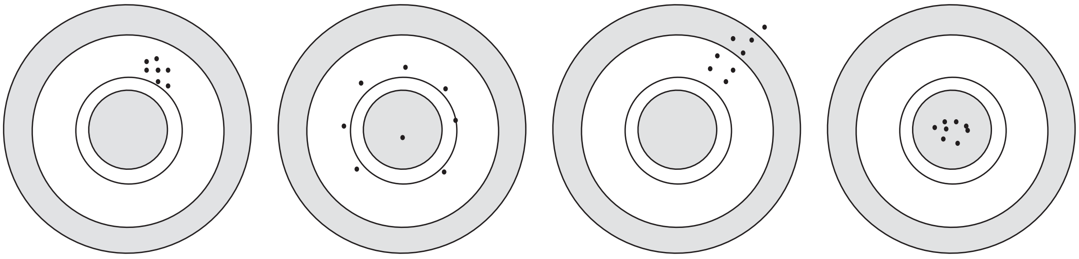
Figure 2: Example of building MST.

The p attributes selected must satisfy the criterion that they either appear as the branch nodes in MST or as the leaf nodes but with stronger relationship with other attributes. Figures 2(a), 2(b), and 2(c) show the original spanning tree, procedure of selecting edges, and final MST, respectively. As shown in Figure 2(c), attributes ②, ③, and ⑥ are branch nodes and can be used as p attributes. In addition, ①, ④, and ⑤ are leaf nodes with corresponding CMIs of 7, 6, and 2, respectively. The CMIs are then sorted into descending order. In this paper, if the sum of CMIs of the first k leaf nodes is greater than 85% of the sum of CMIs of all leaf nodes, we suppose that they represent the most important marginal relationships and can also be selected as p attributes. For example, since

$$\frac{7+6}{7+6+2} = 86.7\% > 85\%, \tag{18}$$

then ① and ④ can be used as p attributes. This criterion helps to ensure that strong, and only strong, relationships among attributes will be retained. By contrast, AODE [4] indiscriminately uses each attribute as superparent even if some attributes may be independent of others. Besides, SP supports incremental learning because it may reselect the subset of attributes when a new training instance becomes available.

At training time SP needs only to form the tables of joint attribute value, class frequencies to estimate the probabilities $\widehat{P}(y)$, $\widehat{P}(y,x_i)$, and $\widehat{P}(y,x_i,x_j)$, which are required for estimating $\widehat{P}(x_i \mid y)$, $\widehat{P}(x_i,x_j \mid y)$, and $\widehat{P}(x_j \mid y,x_i)$ in turn. Calculating the estimates requires a simple scan through the data, an operation of time complexity $O(tn^2)$, where t is the number of training instances and n is the number of attributes. To build maximum weighted spanning tree, SP must first calculate the CMI, requiring consideration for each pair of attributes, every pairwise combination of their respective values in conjunction with each class value. The time complexity to build a MST is $O(n^2\log n)$. The resulting time complexity is $O(tn^2 + k(nv)^2 + n^2\log n)$ and space complexity is $O(k(nv)^2)$, where k is the number of classes and v is the average number of values for an attribute. At classification time SP needs only to store the probability tables, space complexity $O(k(nv)^2)$. This compression over the table required at training time is achieved by storing probability estimates for each attribute value conditioned by



(a) High bias, low variance (b) Low bias, high variance (c) High bias, high variance (d) Low bias, low variance

Figure 3: Bias and variance in shooting arrows at a target.

the parent selected for that attribute and the class. The time complexity of classifying a single instance is $O(kn^2)$.

3.2. How to Eliminate Children Attributes. Kohavi and Wolpert [12] presented a bias-variance decomposition of expected misclassification rate, which is a powerful tool from sampling theory statistics for analyzing supervised learning scenarios. Suppose y and $\hat{y}$ are the true class label and that class generated by a learning algorithm, respectively; the zero-one loss function is defined as

$$\xi(y,\hat{y}) = 1 - \delta(y,\hat{y}), \tag{19}$$

where $\delta(y,\hat{y}) = 1$ if $\hat{y} = y$ and 0 otherwise. The bias term measures the squared difference between the average output of the target and the algorithm. This term is defined as follows:

$$\text{bias} = \frac{1}{2}\sum_{\hat{y},y\in Y}\left[P(\hat{y}\mid x) - P(y\mid x)\right]^2, \tag{20}$$

where x is the combination of any attribute value. The variance term is a real valued nonnegative quantity and equals zero for an algorithm that always makes the same guess regardless of the training set. The variance increases as the algorithm becomes more sensitive to changes in the training set. It is defined as follows:

$$\text{variance} = \frac{1}{2}\left[1 - \sum_{\hat{y}\in Y}P(\hat{y}\mid x)^2\right]. \tag{21}$$

Moore and McCabe [13] illustrated bias and variance through shooting arrows at a target, as described in Figure 3. The perfect model can be regarded as the bull's eye on a target and the learned classifier as an arrow fired at the bull's eye. Bias and variance describe what happens when an archer fires many arrows at the target. Bias means that the aim is off and the arrows land consistently off the bull's eye in the same direction. Variance means that the arrows are scattered. Large variance means that repeated shots are widely scattered on the target. They do not give similar results but differ widely among themselves.

It is reported that removing redundant children attributes from within ODEs can help to decrease both bias and zero-one loss [3, 14]. Subsumption resolution (SR) [8] identifies pairs of attribute values such that one can replace the other. Deleting x_j from a Bayesian classifier should not be harmful when x_j is a generalization of x_i; that is, $P(x_j \mid x_i) = 1.0$. Only the attribute value x_i is necessary for classification; that is, $P(y \mid x_1,\dots,x_n) = P(y|x_1,\dots,x_{j-1},x_{j+1},\dots,x_n)$. Such deletion may improve a classifier's estimate if it makes unwarranted assumptions about the relationship of x_j to the other attributes when estimating intermediate probability values, such as NB's independence assumption. Since $P(x_j|x_i) = 1.0$ can be represented as FD: $x_i \rightarrow x_j$, SR and FD have the same meaning but from different viewpoints.

Table 1: A loop relationship example.

X_1	X_2	X_3	X_4	Y
a_1	b_1	c_1	d_1	y_1
a_2	b_2	c_1	d_2	y_2
a_1	b_1	c_2	d_1	y_2
a_2	b_1	c_2	d_3	y_1
a_2	b_2	c_1	d_3	y_2

SR mainly considers pair relationship of one-one. However, four basic relationships exist in the real world: one-one, one-many, many-many, and many-one. These four relationships can be grouped into two sets: one-one and many-one. Thus, SR cannot resolve interdependencies when a loop appears in the many-one relationship. The data presented in Table 1 shows a loop example with four attributes $\{X_1, X_2, X_3, X_4\}$ and class label Y. For the first instance $\{X_1 = a_1, X_2 = b_1, X_3 = c_1, X_4 = d_1\}$, suppose $\{X_2 = b_1\}$, $\{X_4 = d_1\}$, and $\{X_1 = a_1\}$ are generalizations of $\{X_1 = a_1\}$, $\{X_2 = b_1, X_3 = c_1\}$, and $\{X_4 = d_1\}$, respectively. The loop relationship is described in Figure 4(a), where "$\rightarrow$" represents the one-one relationship and "$\dashrightarrow$" represents the many-one relationship. After SR, only attributes X_3 are used for classification and NB will misclassify the first instance as "$Y = y_2$", even though it occurs in the training data. For different testing instances, different correlated attributes will be deleted. These instances will be illustrated from the viewpoint of FD. $\{X_1 = a_1, X_2 = b_1, X_3 = c_1, X_4 = d_1\}$ can be replaced by three FDs:

(1) $\{X_1 = a_1\} \rightarrow \{X_2 = b_1\}$,

(2) $\{X_2 = b_1, X_3 = c_1\} \rightarrow \{X_4 = d_1\}$,

(3) $\{X_4 = d_1\} \rightarrow \{X_1 = a_1\}$.

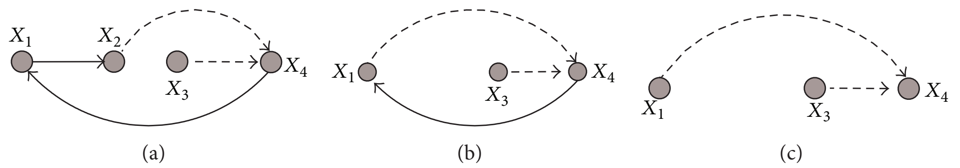


Figure 4: Loop relationship between attributes of training instance.

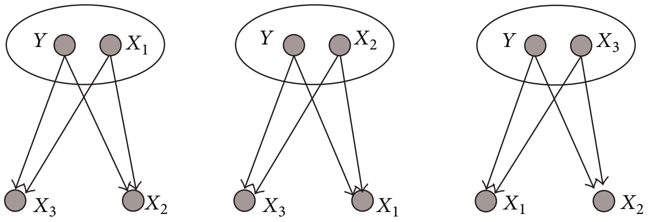


Figure 5: Original network structures of AODE submodels.

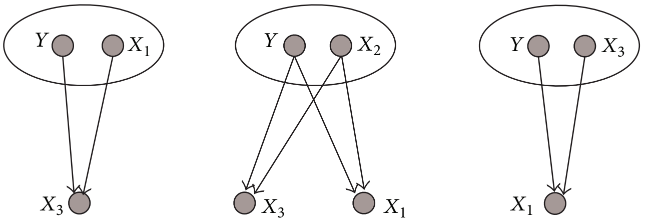


Figure 6: Network structures corresponding to instance-1.

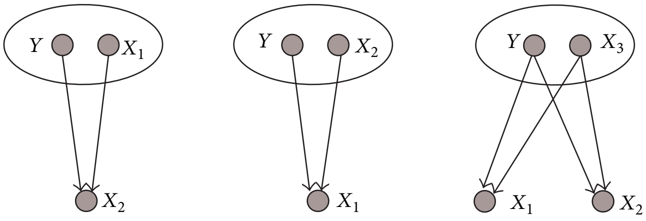


Figure 7: Network structures corresponding to instance-2.

The following results can be generated. We can obtain $\{X_1 = a_1\} \rightarrow \{X_1 = a_1, X_2 = b_1\}$ from FD (1) by applying union rule. As shown in Figure 4(b), X_2 disappears and the arc that once connects X_2 and X_4 now extends to connect X_1 and X_4.

We can obtain $\{X_1 = a_1, X_3 = c_1\} \rightarrow \{X_1 = a_1, X_2 = b_1, X_3 = c_1, X_4 = d_1\}$ from FD (2) by applying augmentation rule. As Figure 4(c) shows, the arc from X_4 to X_1 is removed to avoid a loop relationship. Thus, we can infer and obtain the other two attribute values from two attribute values of $\{X_1 = a_1, X_3 = c_1\}$. Correspondingly the first instance will be correctly classified.

It should be noted that SP selects p attributes from the probabilistic viewpoint by calculating CMI while RC selects c attributes from the logical viewpoint by inferring FDs from the training data. That is, the learning procedure of SP + RC is divided into two parts; SP roughly describes the basic structure of each submodel which uses p attributes as superparents and other attributes as the children; and for different instances, RC further refines the model by deleting redundant children attributes and thus makes the final model much more flexible and robust. Suppose an extreme instance, the CMIs of all attributes are all small and equal, and then all the attributes will be selected as p attributes. The structure will be just the same as AODE after applying SP. But for different testing instances, different FDs can help to make each submodel express the key dependencies. For example, suppose $\text{FD}_1 = \{x_1 \rightarrow x_2\}$ holds for instance-1 and $\text{FD}_2 = \{x_2 \rightarrow x_3\}$ holds for instance-2; Figures 5, 6, and 7 show the original AODE structure after applying SP and corresponding structures for instance-1 and instance-2, respectively.

Discovering FD from existing databases is an important issue. This issue has long been investigated and has been recently addressed with a data mining viewpoint in a novel and efficient way. Rather than exhibiting the set of all functional dependencies which hold in a relation, related work aims to discover a smaller cover equivalent to this set. This problem is known as FD inference. Association rules can be used to discover the relationships and potential associations of items or attributes among huge data. These rules can be effective in uncovering unknown relationships, thereby providing results that can be the basis of forecast and decision. They have proven to be useful tools for an enterprise as they strive to improve their competitiveness and profitability.

4. Experimental Study

We expect AODE with SP and RC to exhibit low zero-one loss and low bias. Thus, we compare the performance of the system with the following attribute selection methods. First is p attribute addition (PAA), which starts with p and c initialized to the empty and full sets, respectively. It adds one p attribute to each ODE at each step. Second is c attribute addition (CAA), which begins with p and c initialized to the full and empty sets, respectively. It adds one c attribute to every ODE at each step. Third is p attribute elimination (PAE), which starts with p and c initialized to the full set and deletes one p attribute from every ODE at each step. Fourth is c attribute elimination (CAE), which deletes one c attribute from every ODE at each step. Fifth and sixth are SR and NSR, respectively.

Table 2 summarizes the characteristics of each data set, including the numbers of instances, attributes, and classes. Missing values for qualitative attributes are replaced with modes and those for quantitative attributes are replaced

Table 2: Data sets.

Number	Data set	Number of instances	Attribute	Class
1	Abalone	4177	8	3
2	Adult	48842	14	2
3	Anneal	898	38	6
4	Audio	226	69	24
5	Car	1728	7	4
6	Chess	551	39	2
7	Cleveland	303	13	2
8	Connect 4	67557	42	3
9	Contact lenses	24	4	3
10	Donation	5749132	11	2
11	Heart	303	13	2
12	Hepatitis	155	19	2
13	Hungarian	294	13	2
14	Hypothyroid	3163	25	2
15	Iris	150	4	3
16	kr versus kp	3196	36	2
17	Labor	57	16	2
18	Mushroom	8124	22	2
19	Nursery	12960	8	5
20	Optdigits	5620	64	10
21	Satellite	6435	37	6
22	Shuttle	58000	9	7

with means from the training data. We estimate the base probabilities $P(y)$, $P(y, x_i)$, and $P(y, x_i, x_j)$ using the Laplace estimate as follows [15]:

$$\widehat{P}(y) = \frac{F(y)+1}{K+k},$$
$$\widehat{P}(y, x_i) = \frac{F(y, x_i)+1}{K_i+k_i}, \qquad (22)$$
$$\widehat{P}(y, x_i, x_j) = \frac{F(y, x_i, x_j)+1}{K_{ij}+k_{ij}},$$

where $F(\cdot)$ is the frequency with which a combination of terms appears in the training data, K is the number of training instances for which the class value is known, K_i is the number of training instances for which both the class and attribute X_i are known, and K_{ij} is the number of training instances for which all of the class and attributes X_i and X_j are known. k is the number of attribute values of class Y, k_i is the number of attribute value combinations of Y and X_i, and k_{ij} is the number of attribute value combinations of Y, X_j, and X_i. As NB and AODE require discrete valued data, all data were discretized using minimum description length (MDL) discretization [16]. Classifier is formed from data set and bias, variance, and zero-one loss estimated from the performance of those classifiers on the same data set. Experiments are performed on a dual-processor 3.1 GHz Windows XP computer with 3.16 Gb RAM. All algorithms are applied to the 22 data sets described in Table 2.

4.1. Zero-One Loss, Bias, and Variance Results. Table 3 presents for each data set the zero-one loss, which is estimated by 50 runs of twofold cross-validation to give an accurate estimation of the average performance of an algorithm. The advantage of this technique is that it uses the full training data as the training set and the testing set. Moreover, every case in the training data is used for the same number of times in each of the roles of training and testing data. Tables 4 and 5 provide the bias and variance results, respectively. The zero-one loss, bias, or variance across multiple data sets provides a gross measure of relative performance.

The basic relationships among attributes can be clearly observed by building MST. If one attribute is connected with several other attributes, the attribute is supposed to have crossed functional zones and will be selected as p attribute to retain complementarity. If one attribute is connected with only one other attribute, its independence characteristics may be obvious and will be reconsidered by the weight of CMI. Besides, RC helps to detect the situation in which the relationships that hold in MST need to be refined. Table 3 shows that the advantage of SP + RC is significant compared with SR and NSR in terms of zero-one loss. However, SR and NSR have a significant advantage over CAA, PAA, CAE, and PAE. The disappointing performances of CAA, PAA, CAE, and PAE can be ascribed to their susceptibility to getting trapped into poor selections by local minima during the first several additions or deletions.

The records in Table 4 show that all the c attribute selection algorithms applying SR, NSR, or FDs have a significant advantage in bias over CAE and PAE. In addition, CAE and PAE outperform CAA and PAA. However, comparing SP + RC with SP again does not show obvious difference. This result indicates that SP takes the main role for classification and its effect differs greatly in different data sets. The same result can also be inferred by comparing SP + RC with SR. The training sets containing only 25% of each data set for bias-variance evaluation are small because the data sets are primarily small. The bias of SP + RC decreases as training set size increases because more data will lead to more accurate probability distribution estimates and hence to more appropriate p attribute selection. Of these algorithms, RC, SR, and NSR have the weakest sensitivity to the changes in training data because they can utilize the testing set to infer rules for c attribute elimination. By contrast PAA, PAE, CAA, and CAE perform model selection and their biases differ greatly with different training data.

With respect to variance, as Table 5 shows, the variance of SP + RC does not show obvious advantage to other algorithms. Low-variance algorithms tend to enjoy an advantage with small data sets, whereas low-bias algorithms tend to enjoy an advantage with large data sets. Cross data set experimental studies of the traditional form presented above also support this hypothesis. The main reason may be that the relationship inferred based on MST may overfit the training data because SP needs to calculate CMI to construct MST. This requires enough instances to achieve precise probability estimation.

In the following discussion, canonical cover analysis [17], which can use limited number (e.g., 100) of instances to

Table 3: Experimental results of zero-one loss.

Data set	PAE	CAE	PAA	CAA	SR	NSR	SP	SP + RC
Abalone	0.512	0.528	0.491	0.519	0.489	0.493	0.396	0.402
Adult	0.212	0.187	0.182	0.158	0.149	0.152	0.137	0.131
Anneal	0.017	0.014	0.013	0.013	0.012	0.011	0.017	0.008
Audio	0.272	0.263	0.247	0.258	0.258	0.217	0.261	0.158
Car	0.236	0.222	0.117	0.212	0.132	0.249	0.127	0.082
Chess	0.147	0.135	0.121	0.129	0.099	0.142	0.128	0.102
Cleveland	0.259	0.262	0.251	0.252	0.247	0.196	0.247	0.178
Connect 4	0.277	0.268	0.289	0.271	0.275	0.267	0.264	0.238
Contact lenses	0.422	0.415	0.409	0.411	0.395	0.445	0.401	0.375
Donation	0.002	0.002	0.002	0.001	0.000	0.001	0.000	0.000
Heart	0.192	0.187	0.182	0.189	0.191	0.197	0.189	0.174
Hepatitis	0.216	0.213	0.209	0.211	0.198	0.208	0.187	0.163
Hungarian	0.193	0.192	0.187	0.190	0.194	0.196	0.217	0.163
Hypothyroid	0.031	0.026	0.021	0.029	0.024	0.021	0.037	0.012
Iris	0.118	0.128	0.125	0.121	0.113	0.102	0.077	0.087
kr versus kp	0.070	0.071	0.091	0.075	0.068	0.067	0.061	0.072
Labor	0.062	0.051	0.047	0.060	0.059	0.061	0.057	0.053
Mushroom	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Nursery	0.088	0.091	0.085	0.081	0.085	0.083	0.089	0.072
Optdigits	0.051	0.055	0.053	0.052	0.051	0.049	0.049	0.030
Satellite	0.122	0.119	0.117	0.132	0.112	0.117	0.116	0.114
Shuttle	0.001	0.001	0.001	0.000	0.001	0.001	0.001	0.000

Table 4: Experimental results of bias.

Data set	PAE	CAE	PAA	CAA	SR	NSR	SP	SP + RC
Abalone	0.332	0.351	0.327	0.312	0.312	0.307	0.291	0.316
Adult	0.147	0.151	0.148	0.152	0.118	0.120	0.155	0.161
Anneal	0.235	0.187	0.236	0.224	0.211	0.200	0.188	0.179
Audio	0.181	0.201	0.183	0.191	0.184	0.127	0.207	0.212
Car	0.152	0.164	0.155	0.148	0.134	0.138	0.127	0.131
Chess	0.106	0.110	0.092	0.114	0.104	0.112	0.113	0.112
Cleveland	0.071	0.068	0.072	0.069	0.064	0.048	0.052	0.049
Connect 4	0.201	0.165	0.199	0.215	0.192	0.197	0.202	0.204
Contact lenses	0.132	0.129	0.133	0.142	0.154	0.112	0.127	0.126
Donation	0.207	0.219	0.211	0.212	0.214	0.206	0.183	0.179
Heart	0.133	0.131	0.135	0.140	0.159	0.162	0.130	0.132
Hepatitis	0.112	0.120	0.114	0.121	0.113	0.109	0.115	0.113
Hungarian	0.205	0.211	0.206	0.198	0.203	0.211	0.204	0.219
Hypothyroid	0.013	0.013	0.012	0.015	0.009	0.007	0.008	0.008
Iris	0.090	0.088	0.090	0.101	0.094	0.067	0.087	0.085
kr versus kp	0.067	0.070	0.071	0.085	0.069	0.075	0.061	0.058
Labor	0.102	0.099	0.098	0.101	0.104	0.107	0.103	0.099
Mushroom	0.002	0.002	0.002	0.003	0.001	0.002	0.001	0.001
Nursery	0.050	0.049	0.050	0.048	0.052	0.054	0.051	0.049
Optdigits	0.028	0.029	0.028	0.031	0.029	0.028	0.027	0.025
Satellite	0.077	0.075	0.074	0.078	0.080	0.078	0.081	0.078
Shuttle	0.003	0.004	0.003	0.005	0.002	0.003	0.002	0.002

Table 5: Experimental results of variance.

Data set	PAE	CAE	PAA	CAA	SR	NSR	SP	SP + RC
Abalone	0.127	0.134	0.142	0.152	0.165	0.161	0.151	0.153
Adult	0.045	0.043	0.046	0.043	0.044	0.042	0.040	0.048
Anneal	0.137	0.146	0.139	0.144	0.152	0.155	0.142	0.145
Audio	0.103	0.101	0.092	0.085	0.082	0.078	0.091	0.087
Car	0.101	0.108	0.118	0.122	0.132	0.133	0.112	0.115
Chess	0.108	0.112	0.103	0.089	0.093	0.088	0.102	0.096
Cleveland	0.152	0.162	0.172	0.179	0.192	0.187	0.207	0.198
Connect 4	0.110	0.105	0.108	0.112	0.110	0.107	0.098	0.092
Contact lenses	0.202	0.213	0.192	0.182	0.178	0.180	0.175	0.172
Donation	0.048	0.054	0.051	0.052	0.064	0.061	0.069	0.062
Heart	0.128	0.133	0.137	0.128	0.142	0.138	0.156	0.152
Hepatitis	0.055	0.058	0.059	0.061	0.063	0.060	0.058	0.053
Hungarian	0.207	0.217	0.196	0.180	0.183	0.177	0.195	0.191
Hypothyroid	0.004	0.004	0.005	0.005	0.004	0.006	0.004	0.004
Iris	0.092	0.096	0.090	0.094	0.092	0.088	0.085	0.081
kr versus kp	0.032	0.032	0.031	0.031	0.023	0.021	0.020	0.020
Labor	0.037	0.034	0.036	0.037	0.038	0.036	0.027	0.024
Mushroom	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Nursery	0.025	0.026	0.027	0.028	0.029	0.031	0.037	0.032
Optdigits	0.021	0.020	0.021	0.022	0.023	0.023	0.022	0.019
Satellite	0.045	0.047	0.049	0.050	0.049	0.045	0.051	0.048
Shuttle	0.003	0.003	0.004	0.003	0.002	0.002	0.002	0.002

Table 6: Variance results on 5 large data sets normalized with respect to AODE.

Data set	PAE	CAE	PAA	CAA	SR	NSR	SP	SP + RC
Adult	0.8653	0.8269	0.8846	0.8269	0.8461	0.8076	0.7692	0.7115
Connect 4	0.9091	0.8677	0.8925	0.9256	0.9078	0.8842	0.8099	0.7603
Donation	0.6575	0.7397	0.6986	0.7123	0.8767	0.8356	0.9452	0.7393
Nursery	0.6756	0.7027	0.7297	0.7567	0.7837	0.8378	0.9875	0.5945
Shuttle	0.7142	0.7142	0.9523	0.7142	0.4761	0.4761	0.4761	0.4761

infer credible FD set, is applied in tandem with functional dependency analysis. Let F_c be a canonical cover for a set of simple FDs and the procedure of computing F_c is described as in Algorithm 1.

The chosen canonical cover is the set of minimal dependencies. Such a cover is information lossless and is considerably smaller than the set of all valid dependencies. These qualities are particularly important to decrease variance while zero-one loss and bias are not affected negatively because of the provision of relevant knowledge wherein redundancy is minimized and extraneous information is discarded. Five large data sets with number of instances >10,000 are selected for variance comparison. The experimental results are shown in Table 6, and we can see that canonical cover analysis can help to project AODE to be competitive with other attribute selection strategies from the viewpoint of variance comparison.

4.2. Elimination Ratio. Statistically a win/draw/loss record (W/D/L) is calculated for each pair of competitors A and B with regard to a performance measure M. The record represents the number of data sets in which A, respectively, beats, loses to, or ties with B on M. Small improvements in leave-one-out error may be attributable to chance. Consequently, it may be beneficial to use a statistical test to assess whether an improvement is significant. A standard binomial sign test, assuming that wins and losses are equiprobable, is applied to these records. A difference is considered to be significant when the outcome of a two-tailed binomial sign test is less than 0.05. To observe the effect of p attribute elimination of SP on zero-one loss, we used the following criterion:

$$E^{p}_{\text{Ratio}} = \frac{E_p}{n}, \tag{23}$$

where E_p is the number of p attributes eliminated and n is the number of attributes. Table 7 shows the comparison results. The E^{p}_{Ratio} of SP is much higher than that of the other two p attribute elimination algorithms, PAA and PAE. The statistical records in Table 3 show that the corresponding zero-one loss of SP is lower generally. We suggest that

```
$F_c$ = FDs
repeat
    Use the union rule to replace any dependencies in $F_c$ of the form
        $\alpha_1 \rightarrow \beta_1$ and $\alpha_1 \rightarrow \beta_2$ with $\alpha_1 \rightarrow \beta_1\beta_2$.
    Find a functional dependency $\alpha \rightarrow \beta$ in $F_c$ with an extraneous
        attribute either in $\alpha$ or in $\beta$.
    If an extraneous attribute is found, delete it from $\alpha \rightarrow \beta$ in $F_c$.
until ($F_c$ does not change)
```

Algorithm 1

Table 7: Win/draw/loss comparison of elimination ratio of p attribute.

W/D/L	PAE	PAA
PAA	8/10/4	
SP	9/9/4	6/11/5

Table 8: Win/draw/loss comparison of elimination ratio of c attribute.

W/D/L	CAE	CAA	SR	NSR
CAA	6/12/4			
SR	9/9/4	6/12/4		
NSR	9/8/5	6/11/5	6/11/5	
RC	11/7/4	11/9/2	10/8/4	9/9/4

the reason for SP's outstanding performance on zero-one loss reduction is that it greatly utilizes the probabilistic dependency relationship on training data.

For example, the elimination ratio of SP is as high as 61.5% for data set "anneal." However, the corresponding zero-one loss is lower than that of PAA and PAE. The main reason may be that, for as many as 38 attributes and only 894 instances, some attributes may have cross-functional zones, and only a few attributes may play the decisive role. After calculating and comparing the sum of CMI between one attribute and all the other attributes, most of the eliminated attributes have weak relationships to other attributes or are even nearly independent of them.

SP selects p attributes based on MST; attributes with a strong relationship among them will be selected first. If any attribute is removed by mistake, the classification results will not be affected greatly. However, for different training sets, especially when their sizes are very small, the conditional distribution estimates may differ greatly and different structures of MST may be obtained. Different p attributes will be selected for classification. The number of FDs extracted from RC is less than that from SR because numerous p attributes are eliminated during SP, especially for data set "audio," with fewer p attributes and much more complicated FDs to remove c attributes. In the W/D/L records, the advantage in zero-one loss is significant with respect to SP versus PAE or SP versus PAA, but not SP + RC versus SP. This result shows that the advantage of SP + RC is from SP but not from RC.

With the increasing number of attributes, more RAM is needed to store joint probability distributions. An important restriction of our algorithm is that the number of the left side of FD should be no more than 2. To observe the effect of SP + RC and SR on each data set, we calculate the c attribute elimination ratios by the following criterion:

$$E^c_{\text{Ratio}} = \frac{\sum_{i=1}^{N} E^i_c}{N}, \tag{24}$$

where E^i_c is the number of c attributes eliminated for the ith instance, and N is the size of data set. Table 8 shows the comparison results of E^c_{Ratio} of SP + RC with the other three c attribute elimination algorithms, CAA, CAE, and SR. Table 3 shows that SP + RC has a significant advantage in zero-one loss over SR and NSR while SR and NSR outperform CAA and CAE. Comparing Table 3 with Table 8 reveals that both RC and SR can help to decrease zero-one loss. However, the effectiveness of RC relies greatly on SP while SR can always improve the performance of AODE. If SP removes p attributes by mistake, some valuable FDs will not be extracted by RC. However, if just redundant p nodes are eliminated, RC can extract more reliable FDs than SR because RC has considered all possible situations of SR.

For example, on data set "hypothyroid" the E^c_{Ratio} of RC is 32%, which indicates that RC just uses approximately 30% of all attributes as c attribute. The reason for this high ratio is that SP has eliminated 21% of the attributes from the data set. For data set "anneal," the E^c_{Ratio} is also as high as 34%, and the zero-one loss is much higher than that of the other three algorithms. This result means SP has removed some p attributes by mistake; RC cannot extract FDs that are dependent on those deleted p attributes. Hence, the experimental results of SP + RC can still be improved if we can find other methods to keep more valuable p attributes for classification.

5. Conclusion and Future Work

AODE provides an attractive framework by averaging all models from a restricted class of one-dependence classifiers. The class of all such classifiers that have all other attributes depends on a common attribute and the class attribute. The current work aims to improve the accuracy derived by MST and FD from weakening the attribute independence assumption without high computational overheads.

Overall, this study developed a classification learning technique that retains the simplicity and direct theoretical foundation of AODE while reducing computational overhead without incurring a negative effect on classification performance. The p attribute of AODE can also be considered the parent of the class attribute. Therefore, we hypothesize that the success of AODE and its variations may be attributed to the fact that AODE not only aggregates all other restricted classes of models but also extends NB to handle the parent of class attribute. If this hypothesis can be proven, we can design a novel and perhaps more effective Bayesian classifier than AODE by constructing the Markov blanket of class attributes.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Science Foundation of China (Grants nos. 61272209 and 61300145) and Postdoctoral Science Foundation of China (Grants nos. 20100481053 and 2013M530980).

References

[1] D. Dash and G. F. Cooper, "Exact model averaging with naive Bayesian classifiers," in *Proceedings of the 19th International Conference on Machine Learning*, pp. 91–98, Sydney, Australia, July 2002.

[2] E. Frank, M. Hall, and B. Pfahringer, "Locally weighted naive Bayes," in *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, pp. 249–256, Acapulco, NM, USA, August 2003.

[3] F. Zheng and G. I. Webb, "Finding the right family: parent and child selection for averaged one dependence estimators," in *Proceedings of the 18th European Conference on Machine Learning*, pp. 490–501, Warsaw, Poland, September 2007.

[4] F. Zheng and G. I. Webb, "Efficient lazy elimination for averaged one-dependence estimators," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 1113–1120, Pittsburgh, Pa, USA, June 2006.

[5] A. Z. Nayyar, C. Jesus, and G. I. Webb, "Alleviating naive bayes attribute independence assumption by attribute weighting," *The Journal of Maching Learning Research*, vol. 14, no. 6, pp. 1113–1120, 2006.

[6] P. Langley and S. Sage, "Induction of selective Bayesian classifiers," in *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pp. 399–406, Seattle, Wash, USA, July 1994.

[7] M. J. Pazzani, "Constructive induction of Cartesian product attributes," in *Proceedings of the Information, Statistics and Induction in Science Conference*, pp. 66–77, July 1996.

[8] F. Zheng, G. I. Webb, P. Suraweera, and L. Zhu, "Subsumption resolution: an efficient and effective technique for semi-naive Bayesian learning," *Machine Learning*, vol. 87, no. 1, pp. 93–125, 2012.

[9] W. W. Armstrong, "Dependency structures of data base relationships," in *Proceedings of the IFIP Congress*, pp. 580–583, 1974.

[10] L. M. Wang, G. F. Yao, and X. Li, "Extracting logical rules and attribute subset from confidence domain," *Information*, vol. 15, no. 1, pp. 173–180, 2012.

[11] L. M. Wang and G. F. Yao, "Bayesian network inference based on functional dependency mining of relational database," *Information*, vol. 15, no. 6, pp. 2441–2446, 2012.

[12] R. Kohavi and D. Wolpert, "Bias plus variance decomposition for zero-one loss functions," in *Proceedings of the 13th European Conference on Machine Learning*, pp. 275–283, June 1996.

[13] D. S. Moore and G. P. McCabe, *Introduction to the Practice of Statistics*, Michelle Julet, San Francisco, Calif, USA, 4th edition, 2002.

[14] A. Z. Nayyar and G. I. Webb, "Fast and effective single pass Bayesian learning," in *Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, vol. 7818 of *Lecture Notes in Computer Science*, pp. 149–160, Gold Coast, Australia, April 2013.

[15] B. Cestnik, "Estimating probabilities: a crucial task in machine learning," in *Proceedings of the 9th European Conference on Artificial Intelligence*, pp. 147–149, Pitman, Boston, Mass, USA, August 1990.

[16] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 1022–1029, August 1993.

[17] L. M. Wang and G. F. Yao, "Learning NT Bayesian classifier based on canonical cover analysis of relational database," *Information*, vol. 15, no. 1, pp. 165–172, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 457197, 10 pages
http://dx.doi.org/10.1155/2014/457197

Research Article

Real-Time Forecast of Tourists Distribution Based on the Improved *k*-Means Method

Peiyu Ren, Zhixue Liao, and Peng Ge

Business School, Sichuan University, Chengdu 610064, China

Correspondence should be addressed to Zhixue Liao; law1219@126.com

Received 29 March 2014; Revised 1 May 2014; Accepted 15 May 2014; Published 5 June 2014

Academic Editor: Ker-Wei Yu

Tourist distribution, a vector to reflect the tourist number of every scenic spot in a certain period of time, serves as the foundation for a scenic spots manager to make a schedule scheme. In this paper, a forecast model is offered to forecast tourist distribution. First of all, based on the analysis of changing mechanism of tourist distribution, it is believed that the possibility for a scenic spot to have the same tourist distribution next time is high. To conduct this forecast, we just need to research on the similar tourist distributions of which time and tourist scale are close. Considering that it is time-consuming, an improved K-means cluster method is put forward to classify the historical data into several clusters so that little time will be needed to search for the most similar historical data. In the end, the case study of Jiuzhai Valley is adopted to illustrate the effectiveness of this forecast model.

1. Introduction

Tourism, which benefits transportation, accommodation, catering, entertainment, and retailing sectors, has become a quite prosperous industry in the past few decades [1]. In 1960, the international tourist arrivals were only 69.3 million, while in 2010 the number was 935 million, and that meant an average growth rate of 5.3 per year [2]. The UNWTO forecasted that international tourism demand will double by 2020 to 1.6 billion visitors and it would generate nearly $2 trillion dollars in economic activities [3]. China's tourism industry develops faster than other countries as the Chinese people's living standards improved significantly along with the rapid development of national economy. As long as a scenic spot can successfully attract tourists and expand its bearing capacity, it will bring huge economic benefits. Therefore, in China, especially the western area which lacks natural resources, tourism has become a driving force of regional economic development. With the rapid development of tourism industry and the rapid increase of tourists, the scenic area has become increasingly congested, so that safety risk is increased and visitors experience is not so good. Congestion can be divided into two categories, true congestion and spurious congestion. The former means the tourist number is far beyond the capacity of scenic spot, while the latter means tourist number is about to surpass the capacity, and, because of tourist scheduling delay, tourist distribution is of disequilibrium; namely, some spots are crowded while some are vacant. For the first kind of congestion, the capacity should be expanded, while, for the latter, effective scheduling can be used. Nowadays, shunt scheduling has two main problems. The first one is delay. The existing scheduling mode belongs to the schedule that scheduling instruction is offered only after the crowd comes into being, so scheduling instruction is delayed because when the scheduling began to take effect, the congestion degree of the scenic spot has changed, which in the end greatly reduced its effect. The second is that the present scheduling lacks hierarchy so that it fails to effectively control the congestion. A good solution to this problem is the following: managers could accurately forecast the amount of tourists arrivals and forecast under different resolutions. Real-time forecast can help scheduling scheme design in different levels of each node and effectively improve the utilization rate of resources. As a result, the real-time forecast on the amount of tourists arrivals has become the research focus.

The researches on forecasting of tourists quantity began in the 1960s, and, in recent years, scholars have made great efforts to improve the accuracy of the forecast. Exponential model, ARIMA model, and GARCH model were built to produce accurate forecasting (Kim and Ngo [4], 2001; Chen [1], 2011; Law [5, 6], 2000a, 2004; Qu and Zhang [7], 1996). Some scholars think that nonlinear method can improve the forecasting accuracy because the data characteristics of the tourists quantity is not linear. Law and Au [8] (1999) built the first neural network model to forecast Japanese's demand to travel to Hong Kong, showing that neural network model performed well in forecasting the tourism demand. Since then neural network model had attracted lots of attention to forecasting such as Cho [9] (2003), Law [10] (2000), Taskaya-Temized and Casey [11] (2005), and Palmer et al. [12] (2006). It was demonstrated that no single forecasting model could produce the best forecast in all situations by empirical results (Cho [13], 2001). Furthermore, there were no certain criteria being used to choose a certain forecasting method which can produce the best performance when a particular tourism demand forecasting was performed. For example Smeral and Wüger [14] (2005) proposed that ARIMA model performed better than exponential model; however Yang et al. [15] (2011) confirmed that ARIMA model performed worse comparing with exponential model.

From the above analysis, it is found that researches on methods to forecast tourist arrivals have been done by lots of researchers, but the research on the real-time forecast is few. Qiu et al. [16] analyzed the space distribution features of tourist flow. Witt and Song [17] pointed out that the tourists distribution is imbalanced in terms of time and space. Yan and Meng [18], Lu et al. [19], and Yan et al. [20] analyzed time and space distribution features of the tourists. Liang and Bao [21] analyzed the seasonal features of theme parks visitors, its influencing factors, and the tourist flow fluctuation law in busy seasons. These studies only focused on the characteristics of tourists' distribution in terms of space and time so that in-depth dynamic forecast researches on nodes in scenic spots are needed.

Based on the analysis of the tourist number of each scenic node in a scenic spot, we initiate a clustering algorithm to search the nearest tourist distribution. Then the algorithm is used to forecast the future tourist distribution of a scenic spot so as to testify the previous forecast. At last, an empirical research is done on Jiuzhai Valley.

Section 2 describes the problem and analyzes tourist distribution forecast in a scenic spot. In Section 3, an improved K-means algorithm is proposed to solve the forecast model. Section 4 takes Jiuzhai Valley as an example to verify the effectiveness of the tourist distribution forecast model for a scenic spot and Section 5 draws a conclusion.

2. Problem Analysis

For a scenic spot with only one entrance, one exit, and many scenic nodes, if we can forecast the tourist number of each node at any time, then the schedule schemes of the scenic spot will become easy and timely. In order to make this real-time forecast, we need to study historical tourist distribution data of the scenic spot. Tourist distribution is defined as

$$N(t) = (N(t)_1, N(t)_2, \ldots, N(t)_n). \tag{1}$$

In (1), tourist distribution $N(t)$ at time t is a n-dimension vector which represents n scenic spots' tourists number $N(t)_i$, $i = 1, 2, \ldots, m$.

It is clear that the possibility for a scenic spot to have the same tourist distribution next time is high if time and tourist scale are close. For example, time t_1 and t_2 are similar times of different days and $|t_1 - t_2|$ is small. If their respective tourist distributions $N(t_1)$ and $N(t_2)$ are very similar and tourist scales $S(t_1)$ and $S(t_2)$ are also close, then the possibility for their next-time tourist distributions $N(t_1 + 1)$ and $N(t_2 + 1)$ to become similar will be high. So, if, at time t, we want to forecast tourist distribution $N(t + 1)$, we only need to know tourists distribution $N(t')^*$ is close to $N(t)$ and the next-time tourist distribution $N(t' + 1)^*$ of $N(t')^*$ can be taken as the forecast value of $N(t + 1)$.

However, as the purpose is to make real-time forecast of each minute and scenic spots' open duration is generally more than 10 hours which means there are 600 data a day, it is time-consuming to find the similar tourists distribution N_t^*. In order to make the search efficient, we can classify historical data into several clusters by clustering algorithm and then judge which cluster $N(t)$ is belonged to. Thanks to the cluster, we can find the similar tourists distribution $N(t)^*$ within time $(1/k)B$ on average (k is the cluster numbers and B is the time consumed to find $N(t)^*$ without clustering).

3. Modeling

3.1. Principles of Forecast. From the above analysis, to realize the real-time tourist distribution forecast, we should firstly classify the tourist distribution samples into several clusters and abide by the principle that in the same cluster the objects should be similar in respect of tourist distribution, time of that tourist distribution, and the tourist scale at that time. As a result, we should add elements (time of that tourist distribution and the tourist scale at that time) into the tourist distribution $N(t) = (N(t)_1, N(t)_2, \ldots, N(t)_n)$ and the new tourist distribution is defined as the time-scale distribution of the tourist distribution $N(t) = (N(t)_1, N(t)_2, \ldots, N(t)_n)$.

So let $O = (O^1, O^2, \ldots, O^{n+1}, O^{n+2})$ be the time-scale distribution of the tourist distribution $N(t)$ in which O_1 represents time t of the tourist distribution $N(t)$, O_{n+2} represents tourist scale, and $(O^2, \ldots, O^{n+1}) = N(t) = (N(t)_1, N(t)_2, \ldots, N(t)_n)$ represents tourist number of node n. So time-scale distribution O and tourist distribution $N(t)$ have the following connection:

$$N\left(\left(O^1\right)\right) = \left(O^2, O^3, \ldots, O^{n-1}\right). \tag{2}$$

$O = (O^1, O^2, \ldots, O^{n+1}, O^{n+2})$ can be classified into many clusters based on the cluster algorithm. In the same cluster, tourist distribution N_t, time t of the tourist distribution, and the tourist scale are similar. So, if a certain data $O(i)$ is judged to belong to cluster V_j, then the next-time tourist distribution $N(O^1(i) + 1)$ can be estimated ($O^1(i)$ is the first element of $O(i)$, so it refers to the time) in that it is highly possible that the similar distribution jumps to the similar cluster next time.

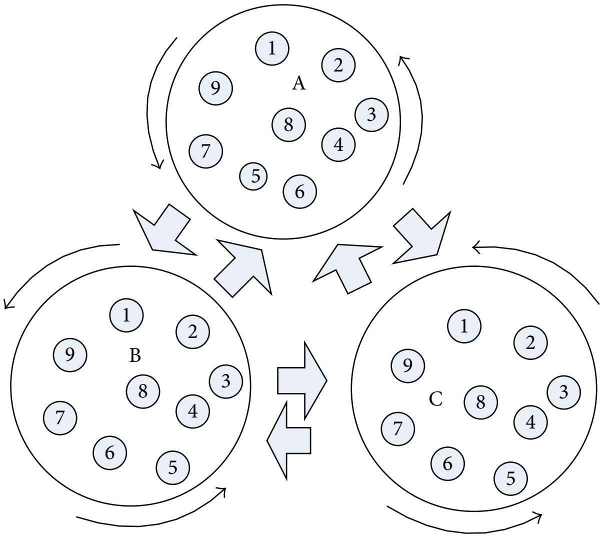


Figure 1: Changing mechanism of tourist distribution.

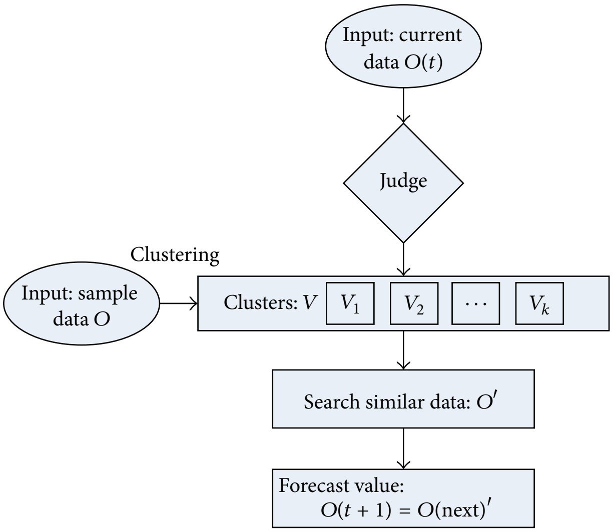

Input: current data $O(t)$
Judge
Clustering
Input: sample data O
Clusters: V
V_1
V_2
$\cdots$
V_k
Search similar data: O'
Forecast value: $O(t+1) = O(\text{next})'$

Figure 2: Forecast process.

For example, in Figure 1, a tourist distribution data $N = (N_1, N_2, \ldots, N_n)$ has 3 clusters $V = (A, B, C)$. Any distribution $N(t)_i$ which belongs to cluster A can be changed under three kinds of circumstances. Firstly, next-time distribution $N(t+1)$ may just remain in the same cluster A with the distribution $O(t)$. Secondly, next-time distribution $N(t+1)$ may jump to cluster B. Thirdly, next-time distribution $N(t+1)$ may jump to cluster C.

At time t, the tourist distribution is $N(t) = (N(t)_1, N(t)_2, \ldots, N(t)_n)$, and the sample distribution is $N(t)^* = (N(t)^*_1, N(t)^*_2, \ldots, N(t)^*_n)$, which is the most similar one to $N(t) = (N(t)_1, N(t)_2, \ldots, N(t)_n)$ in the same cluster. In this situation, tourist distribution $N(t+1) = (N(t+1)_1, N(t+1)_2, \ldots, N(t+1)_n)$ of next time $t+1$ should be similar to the $N(t+1)^* = (N(t+1)^*_1, N(t+1)^*_2, \ldots, N(t+1)^*_n)$. So $N(t+1)^* = (N(t+1)^*_1, N(t+1)^*_2, \ldots, N(t+1)^*_n)$ can be taken as the forecast value of $N(t+1) = (N(t+1)_1, N(t+1)_2, \ldots, N(t+1)_n)$. The detailed forecast process is shown in Figure 2.

3.2. K-Means Algorithms. Let $O = \{O_1, O_2, O_3, \ldots, O_m\}$ be a set of m data objects and each object has $n+2$ natures; then these data can also be represented by a profile data matrix $O_{m\times n+2}$ that has m row vectors and each row vector has $n+2$ dimensions to represent a data object. In this paper, a data object represents a time-scale distribution, so $O_{m\times n+2}$ is the dataset of the time-scale distribution. The ith row vector $O_i = \{O_i^1, O_i^2, O_i^3, \ldots, O_i^{n+2}\}$ denotes the ith time-scale distribution from the dataset $O_{m\times n+2}$ and in O_i the first element O_i^j, and it represents time t, the last element represents tourist scale and other elements represent tourist distribution $N(t)$ at time t. With $O_{m\times n+2}$, the aim of the k-means clustering algorithm is to find a partition $C = C_1, C_2, \ldots, C_k$ of k groups. The partition can generate time-scale distributions in the same group that are as similar to each other as possible, while time-scale distributions of different groups are as different as possible. The k-means clustering algorithm's main idea is stated as follows.

(1) Firstly, randomly select k initial centroids $C = C_1, C_2, \ldots, C_k$ and then assign each time-scale distributions O to the closest centroids C_i, $i = 1, 2, \ldots, k$. A given time-scale distribution O_i will be assigned to centroid C_t if and only if the following condition holds:

$$\left\|O_i - C_t\right\|^2 \geq \left\|O_i - C_j\right\|^2, \quad t, j = (1, 2, \ldots, k), \quad i = (1, 2, \ldots, m), \tag{3}$$

where $\|O_i - C_t\|^2$ is the distance between object O_i and centroid C_t.

The distance can be calculated by the following function: $\|O_i - C_t\|^2 = \sum_{\iota=1}^{n+2}(O_i^\iota - C_t^\iota)^2$, where $\iota = (1, 2, \ldots, n+2)$ denotes ιth component or feature of the corresponding data object.

(2) Secondly, generate new centroid by calculating the mean of the objects set assigned to each cluster. The new centroids $C' = C'_1, C'_2, \ldots, C'_k$ will be

$$C'_j = \frac{1}{M_j} \sum_{O_i \in C_j} O_i, \quad j = (1, 2, \ldots, k), \quad i = (1, 2, \ldots, n+2), \tag{4}$$

where M_j denotes the number of the objects belonging to cluster C_j, $j = (1, 2, \ldots, k)$.

Then repeating the above step until centroids $C = C_1, C_2, \ldots, C_k$ remains unchanged. The k-means clustering algorithm is shown in Algorithm 1.

```
Input: the dataset $O$, and the number of clusters $k$
Output: Set of $k$ clusters    $S = \{S_1, S_2, \ldots, S_K\}$
Initialization: $s = \{\}$, Randomly, select $k$ initial centroids
Begin
    Repeat
        Step 1. Vectors are assigned to the closest centroid
        Step 2. The objective function $J$ is computed
        Step 3. New centroids are calculated as the mean of the vectors assigned to each cluster
    Until Convergence (or no change in the objective function)
    Return S
    End
```

Algorithm 1: K-means clustering: $\mathrm{KM}(O, k)$.

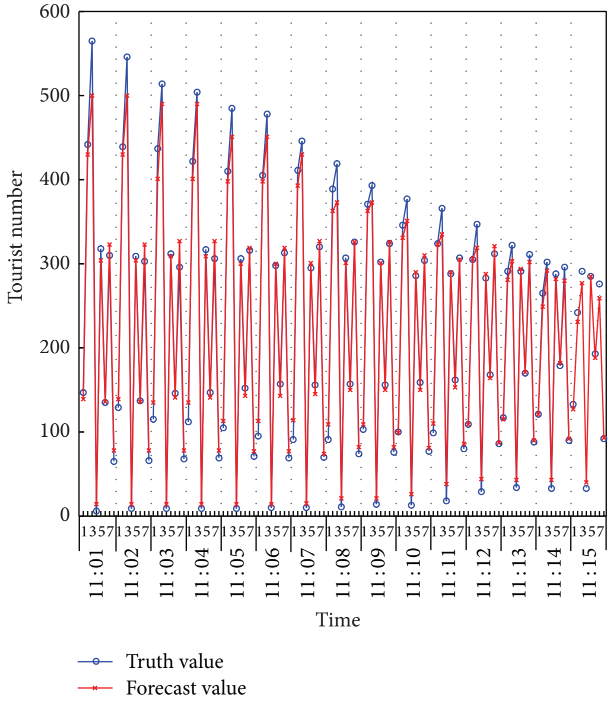


Figure 3: Truth value and forecast value from 11:01 to 11:15 on May 7, 2013.

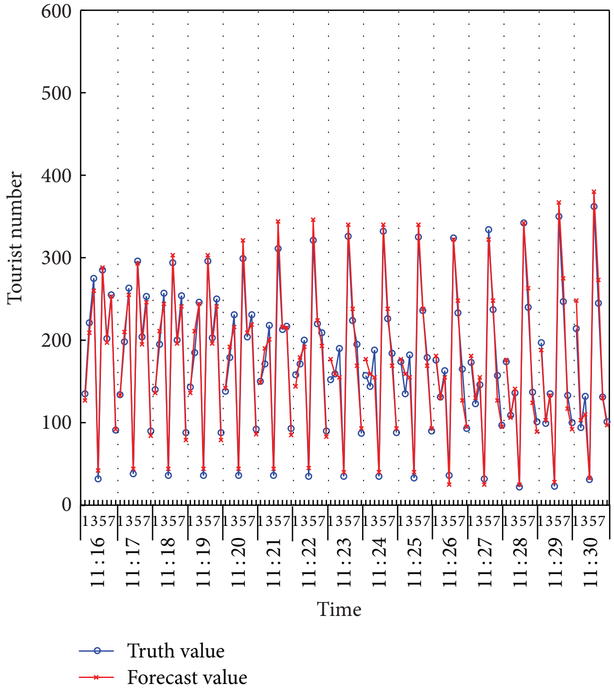


Figure 4: Truth value and forecast value from 11:16 to 11:30 on May 7, 2013.

3.3. The Improved K-Means Algorithms. In the forecast of tourist distribution, cluster number k has an impact on the accuracy of forecast which is the key to the k-means clustering algorithm. However, the k-means clustering algorithm cannot give cluster number k. To solve this problem, we will propose improved k-means algorithms which can ensure cluster number k to make a better forecast.

3.3.1. Select the Two Optimal Centroids. To make sure that objects in the same group are as similar to each other as possible and objects of different groups differ as much as possible, we should select two centroids whose distance is the farthest when compared with any other two objects. So they are the two optimal centroids that can be obtained through the method below.

(1) Randomly select an object O_t from the object data $O = \{O_1, O_2, \ldots, O_m\}$, and then calculate the distance d_{ti} between object O_t and any other object O_i:

$$d_{ti} = \sum_{\iota=1}^{n+2} \left(O_t^{\iota} - O_i^{\iota}\right)^2, \quad \iota = (1, 2, \ldots, n+2), \qquad i = (1, 2, \ldots, n), \quad \text{but } i \neq t. \tag{5}$$

(2) Select the $\max(d_{ti})$ and the farthest object O_i.

(3) Calculate distance d_{ij} between the object O_i and any other object O_j and then get the farthest object O_j.

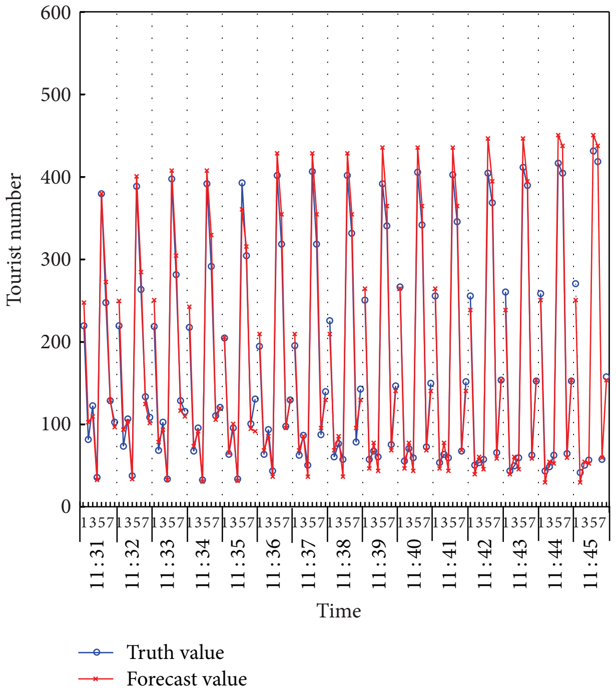


FIGURE 5: Truth value and forecast value from 11:31 to 11:45 on May 7, 2013.

(4) O_i, O_t are the two optimal centroids if $t = j$ or $d_{ij} < d_{ti}$; else, O_i, O_j are the two optimal centroids,

(5) C_1, C_2 represent the two optimal centroids.

3.3.2. Confirmation of the Cluster Number k. After obtaining the two centroids, we need to determine the cluster number k. The method is stated as follows.

(1) Select a new centroid.

By calculating the distance d_{ij} between the existing centroids C_j (the current cluster number) and another object O_i we can get each minimum distance $d_i = \min\{d_{i1}, d_{i2}, \dots, d_{ij}, \dots, d_{iq}\}$ of object O_i and object O_k may be taken as a new centroid (named C_{q+1}) if $d_k = \min\{d_1, d_2, \dots, d_i, \dots, d_m\}$.

(2) Judge whether the new centroid is good or not.

By assigning every other object O to the closest centroid with (1) and (2), the $q+1$ clusters V can be obtained. The new centroid $C' = C'_1, C'_2, \dots, C'_{q+1}$ can be calculated by

$$C_i^{\prime\iota} = \frac{1}{M_i}\sum_{O_j \in V_i} O_j^{\iota}, \quad \iota = (1, 2, \dots, n+2), \qquad i = (1, 2, \dots, q+1), \tag{6}$$

where $C_i^{\prime\iota}$ represents ιth feature of centroids C'_i in cluster V_i.

Definition 1. Let $R_{i,j}$ be the intercluster distance, which represents the distance between the ith cluster V_i and the jth cluster V_j. Thus, it can be calculated by

$$R_{ij} = \left\|\overline{V_i}, \overline{V_j}\right\| = \sum_{\iota=1}^{n+2}\left(C_i^{\prime\iota} - C_J^{\prime\iota}\right)^2, \quad i, j = (1, 2, \dots, q+1), \quad \text{but } i \neq j. \tag{7}$$

Definition 2. Let U_i be the intracluster distance which represents the average distance among the objects $O_i \in V_i$; then it can be calculated by

$$U_i = \frac{1}{M_i}\sum_{O_i \in V_i}\sum_{\iota=1}^{n+2}\left(O_i^{\iota} - C_i^{\prime\iota}\right)^2, \quad i = (1, 2, \dots, q+1). \tag{8}$$

Definition 3. Let I be evaluation index which indicates whether the new centroid C_{q+1} is necessary; then I is defined as

$$I_{q+1} = \frac{1}{q+1}\sum_{i=1}^{q}\max\left(\frac{U_i + U_j}{R_{ij}}\right), \quad j = (1, 2, \dots, q+1). \tag{9}$$

A better cluster algorithm must conform with two conditions. Firstly, objects in the same cluster should be as similar as possible which means intracluster distance U_i must be as small as possible. Secondly, objects in different clusters should be as different as possible which means intercluster

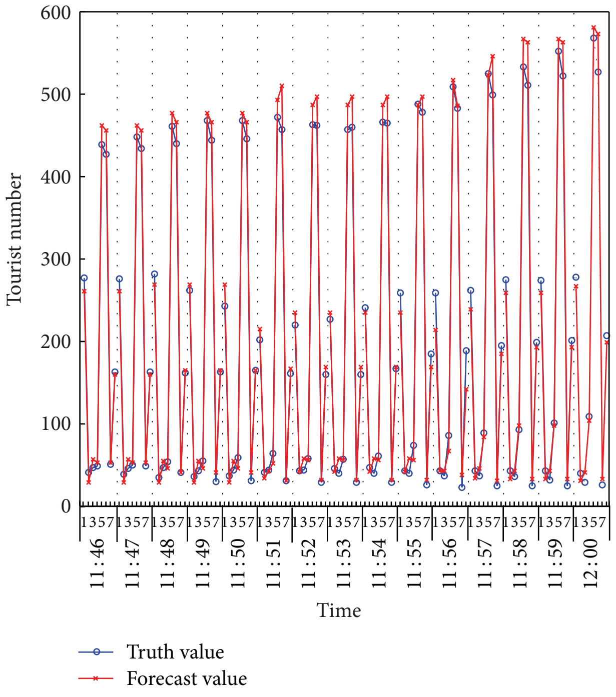


Figure 6: Truth value and forecast value from 11:46 to 12:00 on May 7, 2013.

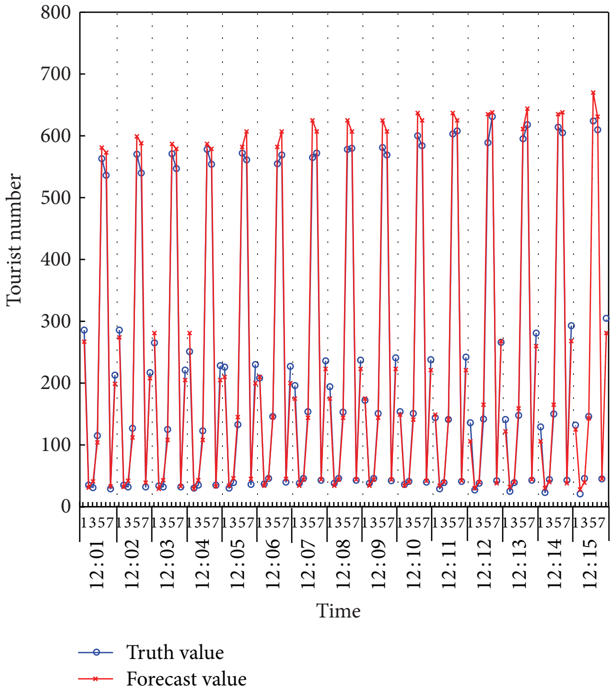


Figure 7: Truth value and forecast value from 12:01 to 12:15 on May 7, 2013.

distance R_{ij} must be as big as possible. In (7), we can find that I increases while both U_i and R_{ij} decrease. So I can be regarded as the evaluation index to reflect whether creating a new centroid is reasonable. In this case when centroid q has been created, we just need to compare I_q with I_{q+1}. If $I_q > I_{q+1}$, then the new centroid is reasonable, and a new cluster by above method. Else, the new centroid is unreasonable and the optimal $k = q$.

The improved k-means algorithm is based on the k-means algorithm, and it has not set the cluster number k, which is determined by the data. So the improved k-means algorithm can perform better than k-means algorithm. The comprehensive description of the improved k-means algorithm is shown in Algorithm 2.

3.4. Forecast Method. In order to forecast tourist distribution, we should judge which cluster the current tourist distribution's time-scale distribution O_i belongs to and then find out the most similar time-scale distribution O_j in that cluster. Therefore, we can take next-time tourist distribution $N(O_j^1 + 1)$ of the similar tourist distribution $N(O_j^1)$ as the forecasting value for the next minute. The detailed forecast process is shown in Figure 2.

4. Empirical Analysis

4.1. Jiuzhaigou Introduction and Errors Evaluation Method. Jiuzhai Valley scenic spot is located in western China and is complimented as a world-class natural scenic spot. The tourism industry there greatly promoted the economic development of western China. The number of tourists every day during statutory holidays, such as Chinese National Day Golden Week, can be as many as 20,000. In the past two years,

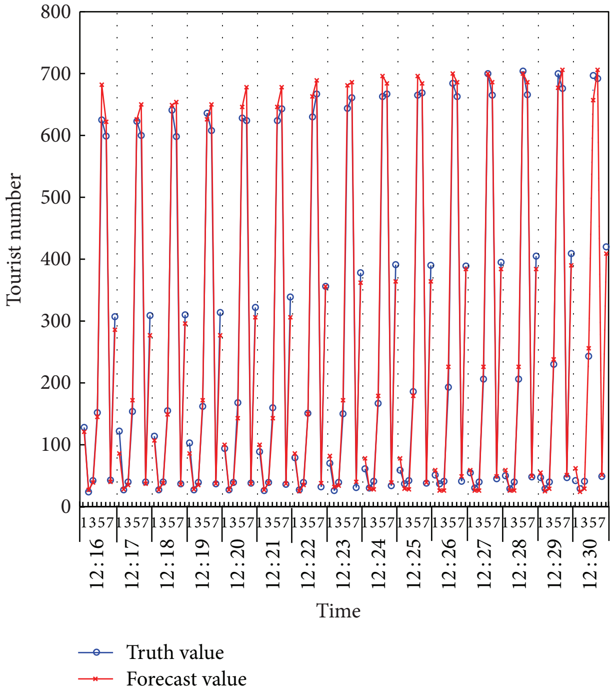


Figure 8: Truth value and forecast value from 12:16 to 12:30 on May 7, 2013.

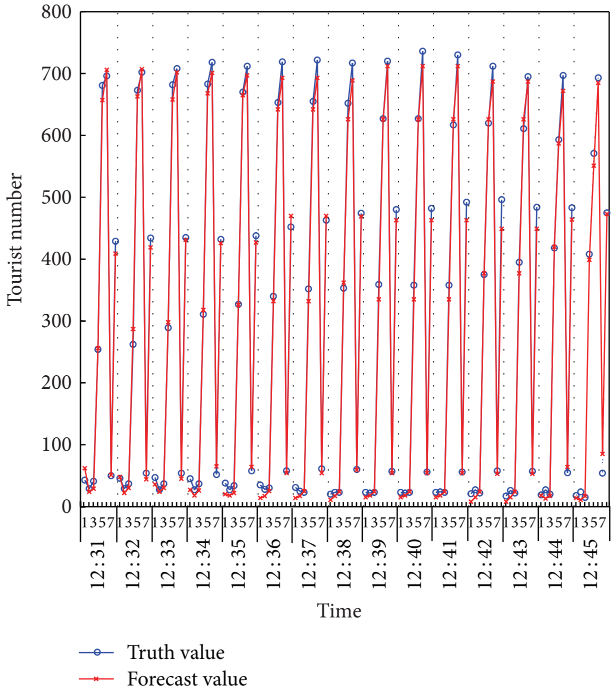


Figure 9: Truth value and forecast value from 12:31 to 12:45 on May 7, 2013.

as people's living standard improved and the governments vigorously support tourism, tourist number has increased greatly. For example, during the National Day Golden Week in 2012, the tourist number in Jiuzhai Valley per day was more than 40,000. The peak season has posed a great challenge to tourism management so that it is important to forecast tourist distribution.

Jiuzhai Valley Scenic Spot Administration has carried out Digital Jiuzhai Valley Comprehensive Demonstration Project (Digital Jiuzhai Valley for short). Technologies such as RFID and GIS are widely applied in the scenic spot; digitization systems such as Access Control System and RFID card reader have been distributed in each attraction of the scenic spot so that all kinds of data can be collected. In this paper, Jiuzhai Valley is taken as an example to testify the forecast model.

In order to evaluate the effectiveness of the forecast model, this paper takes mean absolute percentage error (MAPE) as evaluation statistics, and the calculating method is as follows:

$$\text{MAPE} = \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n}\left|\frac{y(t)^i - \hat{y}(t)^i}{y(t)^i}\right|. \tag{10}$$

As MAPE is the mean value of all times forecast errors, we should analyze each times error to find the major characteristics of the errors. We set relative error and its calculating method is as follows:

$$\text{RE}(t) = \sqrt{\frac{\sum_{i=1}^{n}\left(y(t)^i - \hat{y}(t)^i\right)^2}{\sum_{i=1}^{n}\left(y(t)^i\right)^2}}, \tag{11}$$

where $\hat{y}(t)$ represents forecasting result of the tourist distribution and $y(t)$ the truth tourists distribution at time t.

Through (10), it is clear that if the forecast value is closer to the real value, then the result of the equation will be smaller. So the equation can be used to evaluate the effectiveness of the forecast model.

4.2. Data. We have collected data of Jiuzhai Valley's tourist distribution and its corresponding time-scale distribution from May 1 to May 7 in 2012 and 2013, respectively. The data of May 7, 2013, has been used to test the accuracy of the forecast model and other data are taken as the training data to get the clusters and their centroids by the improved k-means cluster algorithm.

4.3. Clustering. With the help of the improved k-means cluster algorithm, we have obtained 9 clusters and their corresponding centroids C are shown in Table 1.

4.4. Forecast. Based on the forecast method proposed in this paper, we have forecast the tourist distribution of the 120 minutes (from 11:01 to 13:00) on May 7, 2013 (see Figures 3, 4, 5, 6, 7, 8, 9, and 10), and then obtained the MAPE = 5.32% by (10). In order to analyze the errors at different times, RE(t) is calculated by (11) (see Figure 11).

From Figure 11, we can find that 4 forecast errors are less than 3%, and they contribute to 2.9% of the total; 27 are less than 4% but more than 3%, and they contribute to 22.5% of the total; 20 are less than 5% but more than 4%, and they contribute to 16.7% of the total; 26 are less than 6% but more than 5%, and they contribute to 21.7% of the total; 16 are less

TABLE 1: The centroids C.

Cluster	Centroid									
C_1	349	23	13	16	436	618	682	62	450	2300
C_2	241	155	437	544	12	310	138	351	74	2021
C_3	281	225	65	75	27	386	358	74	109	1319
C_4	400	18	11	16	371	179	543	205	226	1569
C_5	311	163	31	37	138	619	648	44	215	1895
C_6	245	93	201	262	9	308	122	214	78	1287
C_7	345	33	21	26	257	420	455	64	230	1506
C_8	410	10	1	16	238	232	364	137	439	1437
C_9	395	15	9	16	377	322	804	176	298	2017

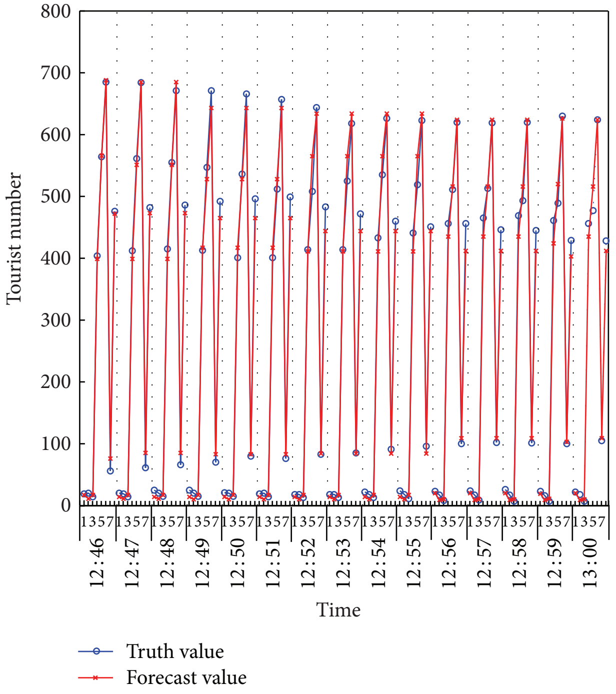


FIGURE 10: Truth value and forecast value from 12:46 to 13:00 on May 7, 2013.

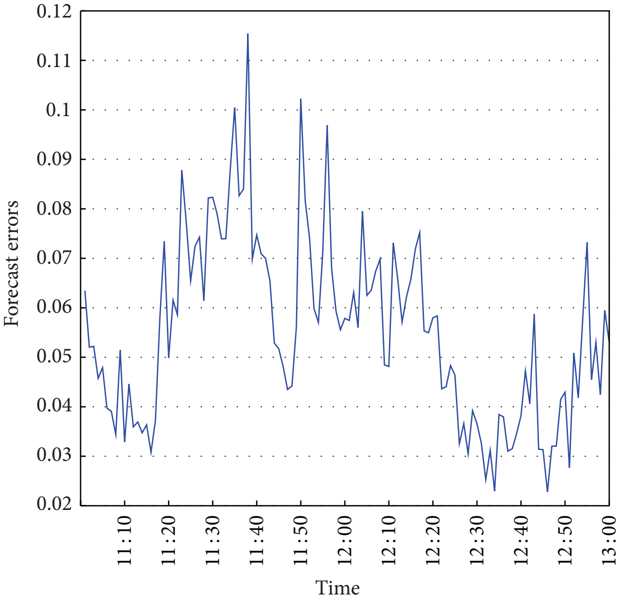


FIGURE 11: Forecast relative errors from 11:01 to 13:00 on May 7, 2013.

than 7% but more than 6%, and they contribute to 13.3% of the total; 16 are less than 8% but more than 7%, and they contribute to 13.3% of the total; 7 are less than 9% but more than 10%, and they contribute to 5.8% of the total; and 3 are (10.05%, 11.54%, and 10.22% at 11:35, 11:38, and 11:50, resp.) more than 10%, and they contribute to 2.5% of the total. So we can come to the conclusion that the forecast model is effective.

5. Conclusion

Based on the improved k-means cluster algorithm, a forecast model is put forward to forecast the tourist distribution which serves as the foundation for the scheduling. Firstly, with the improved k-means cluster algorithm, time-scale distribution data O is classified into several clusters and then we come to judge which cluster it belongs to. Because $N(t)$ consists of elements $[O_2, O_3, \ldots, O_{n-1}]$ in O_i of which the first element $O_i^1 = t$, tourist distribution $N(t)$ can also be classified into the same cluster with O_i, of which the first element $O_i^1 = t$ and $N(t)$ belongs to the same cluster with its time-scale distribution O_i, of which the first element $O_i^1 = t$. Secondly, research is made on the most similar historical data $N'(t')$ in the same cluster with tourist distribution $N(t)$ which is being used to test the forecast model, and tourist distribution $N'(t' + 1)$ is taken as the forecast value of the tourist distribution $N(t + 1)$. Finally, the empirical study in Jiuzhai Valley illustrates that the forecast model is effective.

```
Input: The dataset $O$
Output: Set of $k$ clusters $S = \{S_1, S_2, \ldots, S_K\}$ ($k$ is not a certain value)
Initialization: Select two objects with maximum distance, produce the initial centroids $S = \{S_1, S_2\}$
Begin
  Repeat
  Step 1. Find the longest distance between objects with centroids, and take the it as new centroids
  Step 2. Assign the other objects to the closest centroid
  Step 3. New centroids are calculated as the mean of the objects assigned to each cluster
  Step 4. Calculate the inter-cluster distance, intra-cluster distance and the evaluation index $I_{new}$
  Until $I_{new} > I_{last}$
  Return S
  End
```

Algorithm 2: Improved K-means clustering: IKM(O).

Conflict of Interests

The authors (Peiyu Ren, Zhixue Liao, and Peng Ge) declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Major International Joint Research Programme of the National Natural Science Foundation of China (Grant no. 71020107027), the National High Technology Research and Development Major Programme of China (863 Programme) (Grant no. 2008AA04A107), the National Natural Science Foundation of China (Grant no. 71001075), the National Natural Science Foundation of China (Grant no. 71371130), the China Postdoctoral Science Foundation (Grant no. 2012M521704), Doctoral Fund of Ministry of Education of China (20110181110034), 985 and 211 Projects of Sichuan University, and Central University Fund of Sichuan University.

References

[1] K. Y. Chen, "Combining linear and nonlinear model in forecasting tourism demand," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10368–10376, 2011.

[2] H. Y. Song and R. J. Hyndman, "Tourism forecasting: an introduction," *International Journal of Forecasting*, vol. 27, no. 3, pp. 817–821, 2011.

[3] S. E. Levy and D. E. Hawkins, "Peace through tourism: commerce based principles and practices," *Journal of Business Ethics*, vol. 89, no. 4, pp. 569–585, 2009.

[4] J. H. Kim and M. T. Ngo, "Modelling and forecasting monthly airline passenger flows among three major Australian cities," *Tourism Economics*, vol. 7, no. 4, pp. 397–412, 2001.

[5] R. Law, "Demand for hotel spending by visitors to Hong Kong: a study of various forecasting techniques," *Journal of Hospitality and Leisure Marketing*, vol. 6, pp. 17–29, 2000.

[6] R. Law, "Initially testing an improved extrapolative hotel room occupancy rate forecasting technique," *Journal of Travel and Tourism Marketing*, vol. 16, pp. 71–77, 2004.

[7] H. Qu and H. Q. Zhang, "Projecting international tourist arrivals in East Asia and the Pacific to the year 2005," *Journal of Travel Research*, vol. 35, no. 1, pp. 27–34, 1996.

[8] R. Law and N. Au, "A neural network model to forecast Japanese demand for travel to Hong Kong," *Tourism Management*, vol. 20, no. 1, pp. 89–97, 1999.

[9] V. Cho, "A comparison of three different approaches to tourist arrival forecasting," *Tourism Management*, vol. 24, no. 3, pp. 323–330, 2003.

[10] R. Law, "Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting," *Tourism Management*, vol. 21, no. 4, pp. 331–340, 2000.

[11] T. Taskaya-Temizel and M. C. Casey, "A comparative study of autoregressive neural network hybrids," *Neural Networks*, vol. 18, no. 5-6, pp. 781–789, 2005.

[12] A. Palmer, J. J. Montaño, and A. Sesé, "Designing an artificial neural network for forecasting tourism time series," *Tourism Management*, vol. 27, no. 5, pp. 781–790, 2006.

[13] V. Cho, "Tourism forecasting and its relationship with leading economic indicators," *Journal of Hospitality and Tourism Research*, vol. 25, pp. 399–420, 2001.

[14] E. Smeral and M. Wüger, "Does complexity matter? Methods for improving forecasting accuracy in tourism: the case of Austria," *Journal of Travel Research*, vol. 44, no. 1, pp. 100–110, 2005.

[15] X. Z. Yang, C. L. Gu, and Q. Wang, "Study on the driving force of tourist flows," *Geographical Research*, vol. 30, pp. 23–36, 2011.

[16] Y. Q. Qiu, P. Ge, and P. Y. Ren, "A study on temporal and spatial navigation based on the load-Balance of tourists in Jiuzhaigou Valley," *Resources Science*, vol. 32, pp. 118–123, 2010.

[17] S. F. Witt and H. Song, "Forecasting tourism flows," in *Tourism and Hospitality in the 21st Century*, vol. 3, pp. 106–118, 2002.

[18] F. Yan and J. D. Meng, "The application of logistic growth model in forecast of tourists amount: a case study of Suiyang, Guizhou Province," *Human Geography*, vol. 20, pp. 87–91, 2005.

[19] S. Lu, L. Lu, and L. Wang, “Temporal characteristics of tourist flows to ancient villages: a case study of two world cultural heritages, Xidi Village and Hongcun Village,” *Geographica Sinica*, vol. 24, pp. 250–256, 2004.

[20] L. Yan, X. G. Xu, and X. P. Zhang, “Analysis to temporal characteristics of tourist flows on Jiuzhaigou world natural heritage,” *Acta Scientiarum Naturalium Universitatis Pekinensis*, vol. 45, no. 1, pp. 171–177, 2009.

[21] Z. X. Liang and J. G. Bao, “A seasonal study on tourist flows in theme parks during golden weeks: a case of theme parks in Shenzhen Overseas Chinese Town,” *Tourism Tribune*, vol. 27, pp. 58–65, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 649630, 18 pages
http://dx.doi.org/10.1155/2014/649630

Research Article

Continuous Genetic Algorithm as a Novel Solver for Stokes and Nonlinear Navier Stokes Problems

Z. S. Abo-Hammour,[1] A. D. Samhouri,[2] and Y. Mubarak[2]

[1] *Mechatronics Engineering Department, Faculty of Engineering and Technology, The University of Jordan, Amman 11942, Jordan*
[2] *Chemical Engineering Department, Faculty of Engineering and Technology, The University of Jordan, Amman 11942, Jordan*

Correspondence should be addressed to Z. S. Abo-Hammour; zaer_hr@yahoo.com

Received 14 March 2014; Accepted 15 May 2014; Published 4 June 2014

Academic Editor: Yang Xu

The one-dimensional continuous genetic algorithm (CGA) previously developed by the principal author is extended and enhanced to deal with two-dimensional spaces in this paper. The enhanced CGA converts the partial differential equations into algebraic equations by replacing the derivatives appearing in the differential equation with their proper finite difference formula in 2D spaces. This optimization methodology is then applied for the solution of steady-state two-dimensional Stokes and nonlinear Navier Stokes problems. The main advantage of using CGA for the solution of partial differential equations is that the algorithm can be applied to linear and nonlinear equations without any modification in its structure. A comparison between the results obtained using the 2D CGA and the known Galerkin finite element method using COMSOL is presented in this paper. The results showed that CGA has an excellent accuracy as compared to other numerical solvers.

1. Introduction

Computational fluid dynamics (CFD), which has been commercialized for few decades, is the science of predicting the effect of different phenomena on the overall performance of the processes through solving sets of mathematical equations that govern these processes using numerical techniques. There are many CFD commercial codes available today, which have been originated by the 80s and 90s, such as Fluent, CFX, Fidap, Polyflow, Phoenix, Star CD, Flow 3D, ESI/CFDRC, and SCRYU [1].

The methodology of CFD is, briefly, applying numerical method (called discretization) to develop approximations of the governing equations of fluid mechanics in the fluid region of interest. This discretization involves the governing differential equations expressed algebraically and the collection of cells (the grid). These sets of equations are solved simultaneously for the flow field variables at each node or cell, which enormously contributes.

The discretized conservation equations are solved iteratively. A number of iterations are usually required to reach a converged solution. The convergence is reached when the changes in solution variables from one iteration to the next are negligible; in other words, the residuals provide a mechanism to help monitor this trend [1]. The major advantage of CFD is the relatively low cost that is continuing to decline as computers and software become more powerful and sophisticated. Moreover, CFD simulations can be executed in a short period of time. It has also the ability of simulating real conditions.

The principle methods currently used in CFD are finite difference method (FDM), finite element method (FEM), the Galerkin method, spectral method (SM), filter scheme methods (FSM), and boundary integral equation methods (BIEM) [1]. The first four methods mentioned above are applicable in principle to all types of fluid flow problems (hyperbolic, parabolic, elliptic, and mixed) whereas (BIEM) method has more restrictions and is usually valuable for elliptic type problems [1].

Fluid flows are encountered in the daily life applications including environmental hazards (air pollution, transport of contaminants, etc.), ventilation and air conditioning applications, combustion in automobile engines, complex flows in furnaces, heat exchangers, chemical reactors, and the human body processes (blood flow, breathing, drinking, etc.) [2]. These fluid flow applications are expressed mathematically in sets of partial differential equations that represent the conservation principle of mass, momentum, and energy.

The Navier Stokes equations are the basic governing equations for viscous fluid flow. These equations are obtained by applying Newton's law of motion to a fluid element; alternatively these equations are called the momentum equations. These equations are usually supplemented by the mass conservation equation, also called continuity equation. A combination of both momentum equation and continuity equation is usually termed as Navier Stokes equations. The viscous incompressible Navier Stokes equations (INSE) were firmly established in the 19th century as a system of nonlinear partial differential equations that describe the motion of most fluids. As a result, finding a solution with reasonable accuracy of INSE set of equations has attracted the research interest of many scientists around the world [1].

Genetic algorithm (GA) was invented by Holland in the 1960s at the University of Michigan [3]. In contrast to evolution strategies and programming, Holland's original objective was not to design algorithms that solve specific problems but rather to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems [4].

GA was recently reviewed by Goldberg [5], Davis [6], and others. In general, GA is implemented using computer simulations in which an optimization problem is specified. A genetic algorithm begins with a population of typically random parents in which members of a space of candidate solutions called individuals are represented using abstract representations called chromosomes [4]. In each generation, multiple individuals are randomly selected from the current population based upon some application of fitness and then using crossover and modified through mutation to form a new population. And finally it reaches the fit population with the exact one.

There is no rigorous definition of "genetic algorithm" accepted by all in the evolutionary computation community that differentiates GA from other evolutionary computation methods. However, it can be said that most methods called "GA" have at least the following elements in common: populations of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutation of new offspring [7].

Continuous genetic algorithms (CGAs) were developed by Abo-Hammour as an efficient method for finding the global solutions of smooth functions [7]. In CGAs, the smoothness of the solution curve is achieved. Their novel development has opened a wide venue for different engineering and mathematical applications. In the numerical field, CGAs have been applied for the numerical solution of boundary value problems [8–13], Laplace equations [14], differential-algebraic systems [15], fuzzy differential equations [16], and the solution of Stokes and nonlinear Navier Stokes problems [17]. CGAs have been also applied in robotics [18, 19] and optimal control problems [20, 21].

When using GA in optimization problems, there are two vital points that should be considered: firstly, seeking the interrelation between parameters of optimization and secondly whether there is some restriction on the smoothness of the resulting curve or not [8]. In the case of uncorrelated parameters or nonsmooth solution curves, the conventional genetic algorithm (GA) performs well and there is no need of the CGA. On the other hand, if the parameters are interrelated or if the smoothness of the solution curve is a must, then the continuous genetic algorithm (CGA) is preferable [18].

In summary, CGA has many advantages compared with conventional GA especially when it is applied to problems where smooth solution curve is a need and when more than one parameter is used [7].

(1) CGA needs smaller memory than what GA needs, because GA uses genotype and phenotype representations of the population's individuals while CGA utilizes only the phenotype data. This makes CGA more suitable to problems with larger number of parameters.

(2) Execution time for CGA is smaller than time needed by GA because there are no encoding/decoding processes in CGA.

(3) The conventional GA cannot be used in applications where the optimal solution is required.

The novel application of the (CGA) possesses several advantages: (i) it guarantees the smoothness of the solution curves, (ii) the results obtained using CGA are found to be in good agreement with the analytical solutions, and (iii) it can be applied to linear and nonlinear problems without any modification in the algorithm.

In this paper, the novel method based on continuous genetic algorithms is applied to solve two problems modeled by the steady state Navier Stokes equations in two-dimensional rectangular regions [17]. These problems varied in terms of linearity, that is, linear and nonlinear problems of Dirichlet boundary conditions. The main objective of this work is to solve algebraic equations that result from finite differencing of the partial differential equations (PDE) by CGA as a global optimization technique. The efficiency of CGA method is demonstrated by comparing its results to the results of other well-known methods. Furthermore, the exact solutions for both cases are already known, which make the judgment of the CGA method results possible. The usage of exact solutions here is only to reveal the efficiency of CGA in solving problems.

2. Theory of Continuous Genetic Algorithm

Continuous genetic algorithms (CGAs) were firstly developed to avoid the sharp jumps in the parameter values which can be resulted by using conventional genetic algorithm (GA) form in some optimization problems. That can be shown when the parameters of the optimization problem

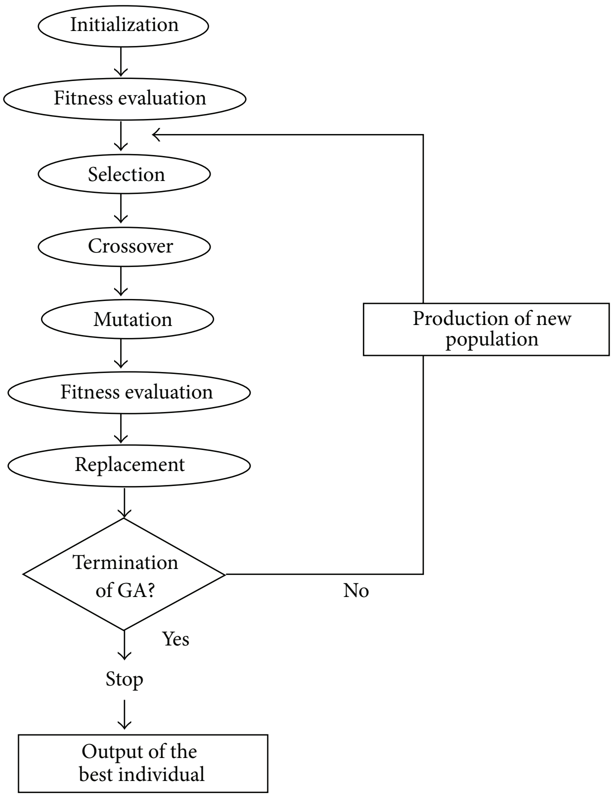


Figure 1: Block diagram for GA procedure.

are correlated with each other or if the smoothness of the solution curve is a must [8]. Continuous genetic algorithm (CGA) mainly depends on the evolution of curves in one-dimensional space, surfaces in two-dimensional space, and volumes in three-dimensional space [18].

The CGAs are of global nature; consequently, the operators of CGA result in smooth transitions but the results of the traditional GA are a step-function-like jump in the parameter values [8]. As discussed by Abo-Hammour et al. [14], the solution of two-dimensional equation in the Cartesian coordinates was found using continuous genetic algorithm (CGA). Following Laplace equation other equations such as Stokes type can be discretized in the same way [8].

The central difference discretization form of two-dimensional equations is

$$u_{i,j} = g\left(u_{i+1,j}, u_{i-1,j}, u_{i,j+1}, u_{i,j-1}\right). \quad (1)$$

The nodal residue for node (i, j) is defined as

$$r_{i,j} = u_{i,j} - g\left(u_{i+1,j}, u_{i-1,j}, u_{i,j+1}, u_{i,j-1}\right). \quad (2)$$

The overall residue for equation is the sum of squares of the nodal residues and is stated as

$$R = \sqrt{\sum_{i=1}^{n_x} \sum_{j=1}^{m_y} r_{i,j}^2}, \quad (3)$$

where n_x is the number of unknown nodes along the x-direction and m_y is the number of unknown nodes along the y-direction.

Then, an appropriate fitness function can be given as [7]

$$F = \frac{1}{1+R}. \quad (4)$$

This means that the larger the residues value the smaller the fitness one. For the exact solution $R \approx 0$ and $F \approx 1$.

The CGAs consist of general steps as discussed by [8] and are followed during solving problems and these steps are shown in Figure 1.

(1) Initialization. This step is just like an initial guess for the solution curve that modified functions can be used to randomly generate the initial population or parents [22]. In general, the initialization functions used in the algorithm should satisfy the following conditions.

(1) It should be smooth from one side and should satisfy any problem-specific constraints, if such constraints exist, from the other side [7]. That is, any smooth function that is close enough to the expected solution curve and satisfies the boundary values can be used.

(2) It should result in an information-rich initial population based on the population diversity, which is split into two main types.

(a) *Smooth Functions Diversity*. This is the diversity due to using a mixture of smooth functions in the initialization phase instead of one function. The expected solution curves are not known and correspondingly a mixture of smooth functions that satisfy the given boundary values will be beneficial to result in a diverse initial population.

(b) *Parameters Diversity*. This is the diversity due to the random generation of the initialization parameters, and in order to make the initial population as much diverse as we can, randomness should be there to remove any bias toward any solution.

It is to be noted that the closer the initialization function to the final solution, the faster the convergence speed [8–13]. However, this convergence speed improvement is minor. This means that the initialization functions have minor effect on the convergence speed of the algorithm because usually the effect of the initial population dies after few tens of generations and the convergence speed after that is governed by the selection mechanism, crossover, and mutation operators.

It is also worth mentioning that the population size is kept constant throughout the evolution process of the algorithm. This population size affects the convergence speed, the average fitness, and the corresponding errors of CGA [7–15]. Small population sizes suffer from larger number of generations required for convergence and the probability of being trapped in local minima, while large population sizes suffer from larger number of fitness evaluations that means larger execution time. Previous studies and analysis performed showed that a compromise population size for similar problems with similar number of unknown nodes is about 500 individuals [7–12].

The initial population in this case will be divided into four equal segments; each of them generated using a different set of smooth functions to obtain a diverse initial population. The first part of the initial population is generated using two-dimensional Gaussian functions, $n_1(x, y)$, the second segment of the initial population is generated using a correlated two-dimensional tangent hyperbolic function, $n_2(x, y)$, and the third part $n_3(x, y)$ is generated using an uncorrelated two-dimensional tangent hyperbolic, while the last segment $n_4(x, y)$ is generated using a mixture of the above three functions. The four functions are given as follows [22]:

$$\begin{aligned}
n_1(x, y) &= \sum_{i=1}^{k} B_{1,i} \times \exp\left[-\frac{1}{2\left(1 - l_i^2\right)} \times \left\{\frac{\left(x - c_{x,i}\right)^2}{\sigma_{x,i}^2} - 2l_i \frac{\left(x - c_{x,i}\right)\left(y - c_{y,i}\right)}{\sigma_{x,i}\sigma_{y,i}} + \frac{\left(y - c_{y,i}\right)^2}{\sigma_{y,i}^2}\right\}\right], \\
n_2(x, y) &= \sum_{i=1}^{k} B_{2,i} \times 0.5 \times \left(1 + \operatorname{hyp}\tan\left[\frac{q_{x,i} \times \left(x - c_{x,i}\right)}{\sigma_{x,i}} + 2l_i \frac{\left(x - c_{x,i}\right)\left(y - c_{y,i}\right)}{\sigma_{x,i}\sigma_{y,i}} + \frac{q_{y,i}\left(y - c_{y,i}\right)}{\sigma_{y,i}}\right]\right),
\end{aligned} \tag{5}$$

$$\begin{aligned}
n_3(x, y) &= B_3 + 0.5 \times \left(1 + \operatorname{hyp}\tan\left\{\frac{q_x\left(x - c_x\right)}{\sigma_x}\right\} \times \operatorname{hyp}\tan\left\{\frac{q_y\left(y - c_y\right)}{\sigma_y}\right\}\right), \\
n_4(x, y) &= n_1(x, y) + n_2(x, y) + n_3(x, y),
\end{aligned} \tag{6}$$

where k is a random in the range $[1, N/5]$ and N is the number of unknown nodes in the grid. B_1, B_2 are a random number in the range $[-2, 2]$ and B_3 is a random number in the range $[-0.25, 0.25]$. c_x is a random number in the range $[x_{\min}, x_{\max}]$. σ_x is a random number in the range $[0, x_{\max} - x_{\min}]$. c_y is a random number in the range $[y_{\min}, y_{\max}]$. σ_y is a random number in the range $[0, y_{\max} - y_{\min}]$. l is a random number in the range $[-0.5, 0.5]$. q_x, q_y are random numbers in the range $[0, 2]$ and in $n_4(x, y)$, $k = 1$, for both n_1 and n_2.

The upper functions $n(x, y)$ are then rescaled within the range $[I_{\min}, I_{\max}]$ representing the expected minimum and maximum initial nodal values.

It is to be noted that the parameter "l" appearing in $n_1(x, y)$ and $n_2(x, y)$ as given in (5) is chosen between -0.5 and $+0.5$ based on the second term of the argument of the exponential or the tangent hyperbolic function which is related to the dual effect of both coordinates (x, y). The coefficient of that term is “$2l$” and in order to have a contribution for that term within the range $[-1, 1]$, then the range of the “l” values should be within the range $[-0.5, 0.5]$.

(2) Fitness Evaluation. In this step, the measure of goodness of each individual in the population is calculated. Fitness will be applied to all populations until reaching the maximum criteria which is around one. This step is applied according to (2)–(4) [7].

(3) Selection. Selection process is to choose the individuals according to their relative fitness in order to enter the devotion process to create the new individuals for the next generation. This step ensures that the overall quality of the population increases from one generation to the next [7].

(4) Crossover. Crossover is a major operation that really empowers the GA. It operates on two randomly selected highly fitted individuals at a time and generates offspring by combining both parent individuals' features [14].

In two-dimensional cases, crossover function will also be expanded into the two-dimensional following form [14]:

$$O_1(x,y) = S(x,y) \times I_1(x,y) + (1 - S(x,y)) \times I_2(x,y), \tag{7a}$$

$$O_2(x,y) = (1 - S(x,y)) \times I_1(x,y) + S(x,y) \times I_2(x,y), \tag{7b}$$

where O_1, O_2 are the two children resulted from the crossover of the two parents I_1 and I_2. S is the mixing function in the range $[0, 1]$.

Four types of mixing function can be used: the first three types are tangent hyperbolic-related functions, while the last one is of Gaussian shape [14]:

$$\begin{aligned}
&S_1(x,y) \\
&= 0.5 \times \left(1 + \text{hyp tan}\left[\frac{q_x(x-c_x)}{\sigma_x} + \frac{q_y(y-c_y)}{\sigma_y}\right]\right), \\
&S_2(x,y) \\
&= 0.5 \times \left(1 + \text{hyp tan}\left[\frac{q_x(x-c_x)}{\sigma_x} \right.\right. \\
&\qquad + \frac{q_{x,y}(x-c_x)(y-c_y)}{\sigma_x\sigma_y} \\
&\qquad \left.\left. + \frac{q_y(y-c_y)}{\sigma_y}\right]\right), \\
&S_3(x,y) \\
&= 0.5 \times \left(1 + \text{hyp tan}\left[\frac{q_x(x-c_x)}{\sigma_x}\right] \right. \\
&\qquad \left. \times \text{hyp tan}\left[\frac{q_y(y-c_y)}{\sigma_y}\right]\right), \\
&S_4(x,y) \\
&= \sum_{i=1}^{N_c} \exp\left[-\frac{1}{2(1-l_i^2)} \right. \\
&\qquad \times \left\{\frac{(x-c_{x,i})^2}{\sigma_{x,i}^2} \right. \\
&\qquad \left.\left. - 2l_i\frac{(x-c_{x,i})(y-c_{y,i})}{\sigma_{x,i}\sigma_{y,i}} + \frac{(y-c_{y,i})^2}{\sigma_{y,i}^2}\right\}\right],
\end{aligned} \tag{8}$$

where

$$\begin{aligned}
c_x &= [x_{\min} + 0.1 \times (x_{\max} - x_{\min}), 0.9 \times (x_{\max} - x_{\min})], \\
c_y &= [y_{\min} + 0.1 \times (y_{\max} - y_{\min}), 0.9 \times (y_{\max} - y_{\min})].
\end{aligned} \tag{9}$$

σ_x, σ_y are chosen such that the tangent hyperbolic function achieves its complete transition from −1 to 1 within the given ranges of both x and y. q_x, q_y have values +1, −1 randomly. $q_{x,y}$ is random number in the range $[-0.2, 0.2]$. N_c represents the number of crossing points and it is chosen in the range $[1, N/20]$. l is a random number within the range $[-0.1, 0.1]$.

(5) Mutation. Mutation introduces random variations into the population. It is applied to a single chromosome only. It is usually performed with low probability. Genetic diversity of the population must be maintained so in order to guard against premature convergence, mutation is usually taken into consideration [14].

Two-dimensional mutation processes can be governed by the following formula [14]:

$$O^m(x,y) = O(x,y) + w \times M(x,y), \tag{10}$$

where $O(x, y)$ represents the child produced from the crossover process. $O^m(x, y)$ is the mutated child. M is the mutation function. w is a random number in the range

$[-R_{\text{ave}}, R_{\text{ave}}]$ and R_{ave} is average overall residue of the population. There are three types of mutation functions [7, 8]:

$$\begin{aligned}
M_1(x,y) &= m_{\text{dc1}} + \exp\left[-\frac{1}{2(1-l_i)} \times \left\{\frac{(x-c_x)^2}{\sigma_x^2} - 2l\frac{(x-c_x)(y-c_y)}{\sigma_x\sigma_y} + \frac{(y-c_y)^2}{\sigma_y^2}\right\}\right],\\
M_2(x,y) &= m_{\text{dc2}} + 0.5 \times \left(1 + \text{hyp tan}\left[\frac{q_x(x-c_x)}{\sigma_x} + \frac{q_{x,y}(x-c_x)(y-c_y)}{\sigma_x\sigma_y} + \frac{q_y(y-c_y)}{\sigma_y}\right]\right),\\
M_3(x,y) &= m_{\text{dc3}},
\end{aligned} \tag{11}$$

where c_x is a random number in the range $[x_{\min}, x_{\max}]$. σ_x is a random number in the range $[0, x_{\max} - x_{\min}]$. c_y is a random number in the range $[y_{\min}, y_{\max}]$. σ_y is a random number in the range $[0, y_{\max} - y_{\min}]$. q_x, q_y have values +1, −1 randomly. $q_{x,y}$ is random number in the range $[-0.2, 0.2]$. $m_{\text{dc1}}, m_{\text{dc2}}$ are some random dc offset values within the range $[-0.25, 0.75]$. l is a random number within the range $[0, 1]$. m_{dc3} is a random number within the range $[-0.5, 0.5]$.

There are three methods for selecting the value of c that is usually used in the mutation process [8].

(1) Random method: in this method c is randomly selected as given in the crossover step.

(2) Lamarckian method: this method depends on the value of residuals at certain mesh point that is the larger residual value at that point then the probability of choosing this point as a mutation center.

(3) Deterministic manner: in this method the mesh point with the maximum residual is chosen as the mutation center.

CGA also used "extinction and immigration" operator. This operator is applied when all individuals in the population are identical or when the improvement in the fitness value of the best individual over a certain number of generations is less than some threshold value. This means that no new information will be obtained through crossover process [7]. The CGA thus tends to stagnate; "extinction and immigration" operator is used to bypass this difficulty. This operator, as indicated by its name, consists of two stages; the first stage is the extinction process where all of the individuals in the current generation are removed except the best of generation individual [7]. The second stage is the mass-immigration process, where the extinct population is filled out again by generating $N_p - 1$ individuals to keep the population size fixed. The generated population is divided into two equal segments each of $N_p/2$ size; the first segment, with $2 \le j \le N_p/2$, is generated as in the initialization phase, while the other segment is generated by performing continuous mutations to the best-of-generation individual as given by the following formula [14]:

$$O_i(x,y) = O_1(x,y) + F \times G(x,y), \quad 2 \le i \le \frac{N_p}{2}, \tag{12}$$

where O_i is the ith parent generated using immigration operator. O_1 is the best of generation individual. F is a random number in the range $[-R_1, R_1]$, where R_1 is the overall residue of the best individual and $G(x, y)$ is one of the following functions [14]:

$$\begin{aligned}
G_1(x,y) &= m_{\text{dc}} + 0.5 \times \left(1 + \text{hyp tan}\left[\frac{q_x(x-c_x)}{\sigma_x} + \frac{q_{x,y}(x-c_x)(y-c_y)}{\sigma_x\sigma_y} + \frac{q_y(y-c_y)}{\sigma_y}\right]\right),\\
G_2(x,y) &= m_{\text{dc}} + 0.5 \times \left(1 + \text{hyp tan}\left[\frac{q_x(x-c_x)}{\sigma_x}\right] \times \text{hyp tan}\left[\frac{q_y(y-c_y)}{\sigma_y}\right]\right),\\
G_3(x,y) &= m_{\text{dc}}
\end{aligned}$$

$$\begin{aligned}&+\exp\left[-\frac{1}{2\left(1-l_i\right)}\right.\\&\quad\times\left\{\frac{\left(x-c_x\right)^2}{\sigma_x^2}\right.\\&\quad\left.\left.-2l\frac{\left(x-c_x\right)\left(y-c_y\right)}{\sigma_x\sigma_y}+\frac{\left(y-c_y\right)^2}{\sigma_y^2}\right\}\right]\\&G_4\left(x,y\right)=m_{\text{dc}4}.\end{aligned}\tag{13}$$

q_x, q_y have values +1, −1 randomly. $q_{x,y}$ is random number in the range $[-1,1]$. m_{dc} are some random dc offset values within the range $[-0.25,0.75]$. l is a random number within the range $[-0.5,0.5]$. $m_{\text{dc}4}$ is a random number within the range $[0,1]$.

(6) Replacement. Applying the genetic operators to the parent's population generates the offspring's population and then the parents' population is totally replaced by the offspring's population. This is called nonoverlapping, generational, or replacement. In this step the "life cycle" of the population can be completed [7].

(7) Termination. When some convergence criterion is met, the GA is terminated. The convergence criteria are as follows: (i) the fitness of the best individual so far found exceeds a threshold value, (ii) the maximum number of generations is reached or the progress limit, and (iii) the improvement in the fitness value of the best member of the population over a specified number of generations is less than some predefined threshold. The best solution of the problem is found after the termination of the algorithm [7].

3. Numerical Results for Stokes Equations

The general forms of Stokes equations are shown in the following:

$$\begin{aligned}-\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right]&=f_x,\\-\left[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right]&=f_y,\end{aligned}\tag{14}$$

in which the right hand side is the shear stress term and the left hand side is the external force applied on the fluid, with continuity equation

$$\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)=0.\tag{15}$$

Dennis and Hudson [23] tried to solve the nonlinear form of Navier Stokes equations but with no pressure term as shown in the following equations:

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right]+f_x,\tag{16}$$

$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=\left[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right]+f_y.\tag{17}$$

Continuity equation is as follows:

$$\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)=0,\tag{18}$$

within the range of $x=[0,1]$ and $y=[0,1]$.

Wang et al. [24] tried to solve Stokes equation using finite element method within the region $x=[0,1]$, $y=[0,1]$; the authors used the following boundary conditions:

$$\begin{aligned}u\left(0,y\right)&=0, & u\left(x,0\right)&=0,\\u\left(1,y\right)&=0, & u\left(x,1\right)&=0,\\v\left(0,y\right)&=0, & v\left(x,0\right)&=0,\\v\left(1,y\right)&=0, & v\left(x,1\right)&=0.\end{aligned}\tag{19}$$

The exact solutions as given by Wang et al. [24] are as follows:

$$\begin{aligned}u&=-2xy\left(x-1\right)\left(y-1\right)x\left(x-1\right)\left(2y-1\right),\\v&=2xy\left(x-1\right)\left(y-1\right)y\left(y-1\right)\left(2x-1\right).\end{aligned}\tag{20}$$

The right hand sides for (14) were given by Wang et al. [24] in the following forms:

$$\begin{aligned}f_x=\;&8x^2(x-1)^2\left(y-1\right)+4x^2\left(2y-1\right)\left(x-1\right)^2\\&+8x^2y(x-1)^2+4y\left(2y-1\right)\left(x-1\right)^2\left(y-1\right)\\&+4x^2y\left(2y-1\right)\left(y-1\right)\\&+8xy\left(2x-2\right)\left(2y-1\right)\left(y-1\right),\\f_y=\;&-8y^2\left(x-1\right)\left(y-1\right)^2-4y^2\left(2x-1\right)\left(y-1\right)^2\\&-8xy^2(y-1)^2-4x\left(2x-1\right)\left(x-1\right)\left(y-1\right)^2\\&-4xy^2\left(2x-1\right)\left(x-1\right)\\&-8xy\left(2x-1\right)\left(2y-2\right)\left(x-1\right).\end{aligned}\tag{21}$$

This right hand sides are resulted by substituting the exact solutions shown in (20) into (14). The nodal values of the exact solutions are shown in Tables 1 and 2.

Moreover, the graphical representations of those actual values are shown in Figures 3(a) and 3(b).

The boundary values are known from the main problems but the other nodal values are unknown values of u and must

Table 1: Actual u nodal values for Stokes equations including boundary values.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	−0.0012	−0.0037	−0.0064	−0.0083	−0.009	−0.0083	−0.0064	−0.0037	−0.0012	0
y_2	0	−0.0016	−0.0049	−0.0085	−0.0111	−0.012	−0.0111	−0.0085	−0.0049	−0.0016	0
y_3	0	−0.0014	−0.0043	−0.0074	−0.0097	−0.0105	−0.0097	−0.0074	−0.0043	−0.0014	0
y_4	0	−0.0008	−0.0025	−0.0042	−0.0055	−0.006	−0.0055	−0.0042	−0.0025	−0.0008	0
y_5	0	0	0	0	0	0	0	0	0	0	0
y_6	0	0.00078	0.00246	0.00423	0.0055	0.006	0.00553	0.0042	0.0025	0.00078	0
y_7	0	0.00136	0.0043	0.00741	0.0097	0.0105	0.00968	0.0074	0.0043	0.00136	0
y_8	0	0.00156	0.00492	0.00847	0.0111	0.012	0.01106	0.0085	0.0049	0.00156	0
y_9	0	0.00117	0.00369	0.00635	0.0083	0.009	0.00829	0.0064	0.0037	0.00117	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 2: Actual v nodal values for Stokes equation including boundary values.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.001166	0.001555	0.001361	0.000778	0	−0.00078	−0.00136	−0.00156	−0.00117	0
y_2	0	0.003686	0.004915	0.004301	0.002458	0	−0.00246	−0.0043	−0.00492	−0.00369	0
y_3	0	0.00635	0.008467	0.007409	0.004234	0	−0.00423	−0.00741	−0.00847	−0.00635	0
y_4	0	0.008294	0.011059	0.009677	0.00553	0	−0.00553	−0.00968	−0.01106	−0.00829	0
y_5	0	0.009	0.012	0.0105	0.006	0	−0.006	−0.0105	−0.012	−0.009	0
y_6	0	0.008294	0.011059	0.009677	0.00553	0	−0.00553	−0.00968	−0.01106	−0.00829	0
y_7	0	0.00635	0.008467	0.007409	0.004234	0	−0.00423	−0.00741	−0.00847	−0.00635	0
y_8	0	0.003686	0.004915	0.004301	0.002458	0	−0.00246	−0.0043	−0.00492	−0.00369	0
y_9	0	0.001166	0.001555	0.001361	0.000778	0	−0.00078	−0.00136	−0.00156	−0.00117	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

be solved. These nodes can be represented in the (10 × 10) mesh of 100 elements, as shown in Figure 2.

With the same previous conditions COMSOL solution was obtained and the nodal values of u and v variables are shown in Tables 3 and 4. Moreover, the graphical representations for u and v COMSOL solutions are shown in Figures 3(c) and 3(d).

Appling (22) and using the errors between the exact and COMSOL values, it is found that L_2 norm error equals to 1.29×10^{-5} for u variable and about 1.21×10^{-5} for v variable with average time for solving the problem of about 0.045:

$$L_2 \text{ norm error} = \sqrt{\sum_{i=1}^{n} \left\| u_{\text{exact}} - u_{\text{numerical}} \right\|^2}. \qquad (22)$$

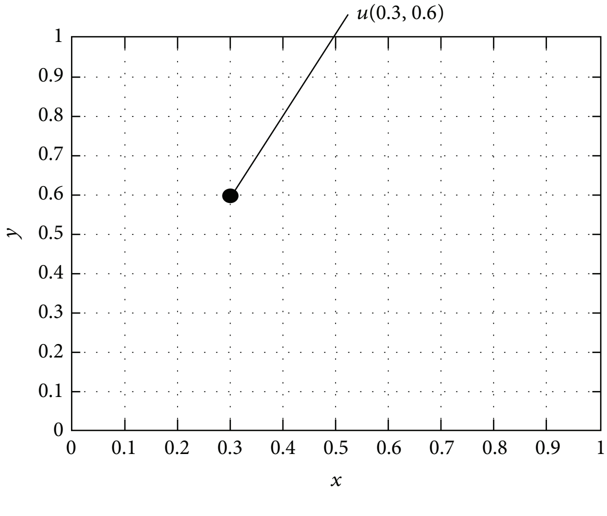


Figure 2: Rectangular mesh with step size equal 0.1.

3.1. Solution of Stokes Equations Using Continuous Genetic Algorithm. All steps of CGA are applied here and they are repeated till the convergence criterion is met (fitness = 0.99), where the best unknown values of nodes are reached. The central difference discretization form of Stokes equation was shown in (23) but for both u and v separately; also (24) and

Table 3: Nodal values of u variable using COMSOL for Stokes case solution.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	−0.0012	−0.0037	−0.0064	−0.0083	−0.009	−0.0083	−0.0064	−0.0037	−0.0012	0
y_2	0	−0.0016	−0.0049	−0.0085	−0.0111	−0.012	−0.0111	−0.0085	−0.0049	−0.0016	0
y_3	0	−0.0014	−0.0043	−0.0074	−0.0097	−0.0105	−0.0097	−0.0074	−0.0043	−0.0014	0
y_4	0	−0.0008	−0.0025	−0.0042	−0.0055	−0.006	−0.0055	−0.0042	−0.0025	-7.78×10^{-4}	0
y_5	0	0	0	0	0	0	0	0	0	0	0
y_6	0	7.79×10^{-4}	0.0025	0.0042	0.0055	0.006	0.0055	0.0042	0.0025	7.78×10^{-4}	0
y_7	0	0.0014	0.0043	0.0074	0.0097	0.0105	0.0097	0.0074	0.0043	0.0014	0
y_8	0	0.0016	0.0049	0.0085	0.0111	0.012	0.0111	0.0085	0.0049	0.0016	0
y_9	0	0.0012	0.0037	0.0064	0.0083	0.009	0.0083	0.0063	0.0037	0.0012	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 4: Nodal values of v variable using COMSOL for Stokes case solution.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.00117	0.00156	0.00136	7.79×10^{-4}	0	-7.79×10^{-4}	−0.00136	−0.00156	−0.00117	0
y_2	0	0.00369	0.00492	0.0043	0.00246	0	−0.00246	−0.0043	−0.00492	−0.00369	0
y_3	0	0.00635	0.00847	0.00741	0.00423	0	−0.00423	−0.00741	−0.00847	−0.00635	0
y_4	0	0.00829	0.01106	0.00967	0.00553	0	−0.00553	−0.00968	−0.01106	−0.00829	0
y_5	0	0.009	0.012	0.0105	0.006	0	−0.006	−0.0105	−0.012	−0.009	0
y_6	0	0.00829	0.01106	0.00967	0.00553	0	−0.00553	−0.00968	−0.01106	−0.00829	0
y_7	0	0.00635	0.00847	0.00741	0.00423	0	−0.00423	−0.00741	−0.00847	−0.00635	0
y_8	0	0.00369	0.00492	0.0043	0.00246	0	−0.00246	−0.0043	−0.00492	−0.00369	0
y_9	0	0.00117	0.00156	0.00136	7.79×10^{-4}	0	-7.79×10^{-4}	−0.00136	−0.00156	−0.00117	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 5: Nodal u values including boundaries for Stokes equation using CGA.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	−0.0012	−0.0037	−0.0064	−0.0083	−0.009	−0.0083	−0.0064	−0.0037	−0.0012	0
y_2	0	−0.0016	−0.0049	−0.0085	−0.0111	−0.012	−0.0111	−0.0085	−0.0049	−0.0016	0
y_3	0	−0.0014	−0.0043	−0.0074	−0.0097	−0.0105	−0.0097	−0.0074	−0.0043	−0.0014	0
y_4	0	−0.0008	−0.0025	−0.0042	−0.0056	−0.006	−0.0056	−0.0043	−0.0025	−0.0008	0
y_5	0	0	0	0	0	0	0	0	0	0	0
y_6	0	0.0008	0.0024	0.0042	0.0055	0.006	0.0055	0.0042	0.0024	0.0008	0
y_7	0	0.0013	0.0043	0.0074	0.0096	0.0105	0.0097	0.0074	0.0043	0.0014	0
y_8	0	0.0015	0.0049	0.0084	0.011	0.012	0.011	0.0085	0.0049	0.0016	0
y_9	0	0.0012	0.0037	0.0063	0.0083	0.009	0.0083	0.0064	0.0037	0.0012	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

(25) are used in order to define the unknown residual nodal u and v values:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f_x = \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}}{h^2} + f_x, \tag{23}$$

where h is the step size in which h_x is the step size in the x-direction and h_y is the step size in the y-direction. And in this case $h_x = 0.1$ and $h_y = 0.1$, again the problem is defined over the rectangular region $x = [0, 1]$, $y = [0, 1]$.

Then, after rearrangement, one has

$$u_{i,j} = u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} + \frac{f_x}{4} \times h^2. \tag{24}$$

The nodal residue for node (i, j) is defined as

$$r_{i,j} = u_{i,j} - \left[u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} + \frac{f_x}{4} \times h^2\right]. \tag{25}$$

Table 6: Nodal v values including boundaries for Stokes equation using CGA.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.0012	0.0015	0.0013	0.0008	0	−0.0008	−0.0014	−0.0016	−0.0012	0
y_2	0	0.0037	0.0049	0.0043	0.0024	0	−0.0025	−0.0043	−0.0049	−0.0037	0
y_3	0	0.0063	0.0085	0.0074	0.0042	0	−0.0042	−0.0074	−0.0085	−0.0064	0
y_4	0	0.0083	0.011	0.0097	0.0055	0	−0.0055	−0.0097	−0.0111	−0.0083	0
y_5	0	0.009	0.012	0.0105	0.006	0	−0.006	−0.0105	−0.012	−0.009	0
y_6	0	0.0083	0.0111	0.0097	0.0055	0	−0.0055	−0.0097	−0.0111	−0.0083	0
y_7	0	0.0063	0.0085	0.0074	0.0042	0	−0.0042	−0.0074	−0.0085	−0.0064	0
y_8	0	0.0037	0.0049	0.0043	0.0025	0	−0.0025	−0.0043	−0.0049	−0.0037	0
y_9	0	0.0011	0.0016	0.0014	0.0008	0	−0.0008	−0.0014	−0.0016	−0.0012	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 7: Residual u values for all nodes for Stokes equation after using CGA ($\times 10^{-4}$).

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	3.9746	1.6466	6.6173	4.3473	9.92845	0.3673	4.8303	2.7018	14.821	0
y_2	0	1.4584	9.9158	7.0978	2.2746	0.28946	3.0963	7.1133	8.0372	2.23402	0
y_3	0	6.9231	6.3902	5.4447	0.828	14.9252	4.7176	18.645	0.7471	12.2662	0
y_4	0	14.162	3.6126	13.076	11.32	9.05015	11.809	14.497	9.7418	0.92215	0
y_5	0	1.1706	0.4675	10.523	7.319	6.62552	3.2928	14.653	7.4677	0.74626	0
y_6	0	3.9953	0.8315	4.3266	2.8571	5.79388	12.773	2.5444	8.3358	0.7428	0
y_7	0	4.9107	4.7092	20.41	10.882	11.2205	8.4874	10.567	4.3102	5.81061	0
y_8	0	6.3028	1.5624	14.346	0.5647	0.08398	7.2523	7.6049	0.0954	2.45801	0
y_9	0	10.577	2.1018	5.8802	0.9785	1.1455	3.3996	3.2041	4.0743	1.87985	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 8: Residual v values for all nodes for Stokes equation after using CGA ($\times 10^{-4}$).

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	2.1016	8.8962	8.8272	5.753	5.4979	16.1784	12.78	0.0776	6.7879	0
y_2	0	5.1713	1.5391	2.8585	2.7778	4.2006	4.71092	4.7172	6.9823	5.42282	0
y_3	0	7.7604	6.4171	11.901	13.982	5.4556	15.6543	13.987	1.2369	14.1758	0
y_4	0	9.7749	8.5258	13.076	7.6596	7.2225	8.32927	8.4301	1.9981	13.6789	0
y_5	0	0.532	3.1775	0.0507	3.1278	8.7859	4.20619	4.328	2.846	5.58614	0
y_6	0	14.067	1.6875	15.331	0.6956	13.451	4.39109	11.482	6.6231	0.66414	0
y_7	0	8.3609	2.5554	6.8751	21.146	2.2431	11.059	11.517	8.1391	3.99419	0
y_8	0	8.1196	9.4279	12.997	0.6023	0.3942	4.97137	5.9129	5.6786	1.88925	0
y_9	0	0.793	2.9236	5.7456	4.52	5.8774	1.37201	0.4039	2.9202	1.80238	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Then, the overall residues were solved for (25) as the sum of squares of the nodal residues, as stated in (3); after that the appropriate fitness function resulted from (4).

The obtained solution for Stokes equation [Dirichlet boundary conditions] using continuous genetic algorithms (CGAs) are shown in Tables 5 and 6, including the nodal values of the boundaries which are equal to zero. And the graphical representations of these values are shown in Figures 3(e) and 3(f).

By calculating the errors between the actual nodal values and CGA one, the L_2 norm error is found to be equal 3.02×10^{-4} for u variable and 1.08×10^{-4} for v variable. The time needed to reach solution in CGA was about 7523 seconds.

Wang et al. [24] used finite element method with a mesh size of 8×8 for solving Stokes equations and then reported 5.05×10^{-4} as L_2 norm error of the solution.

Comparing the error values which resulted by applying CGA with the errors given by finite element method by Wang,

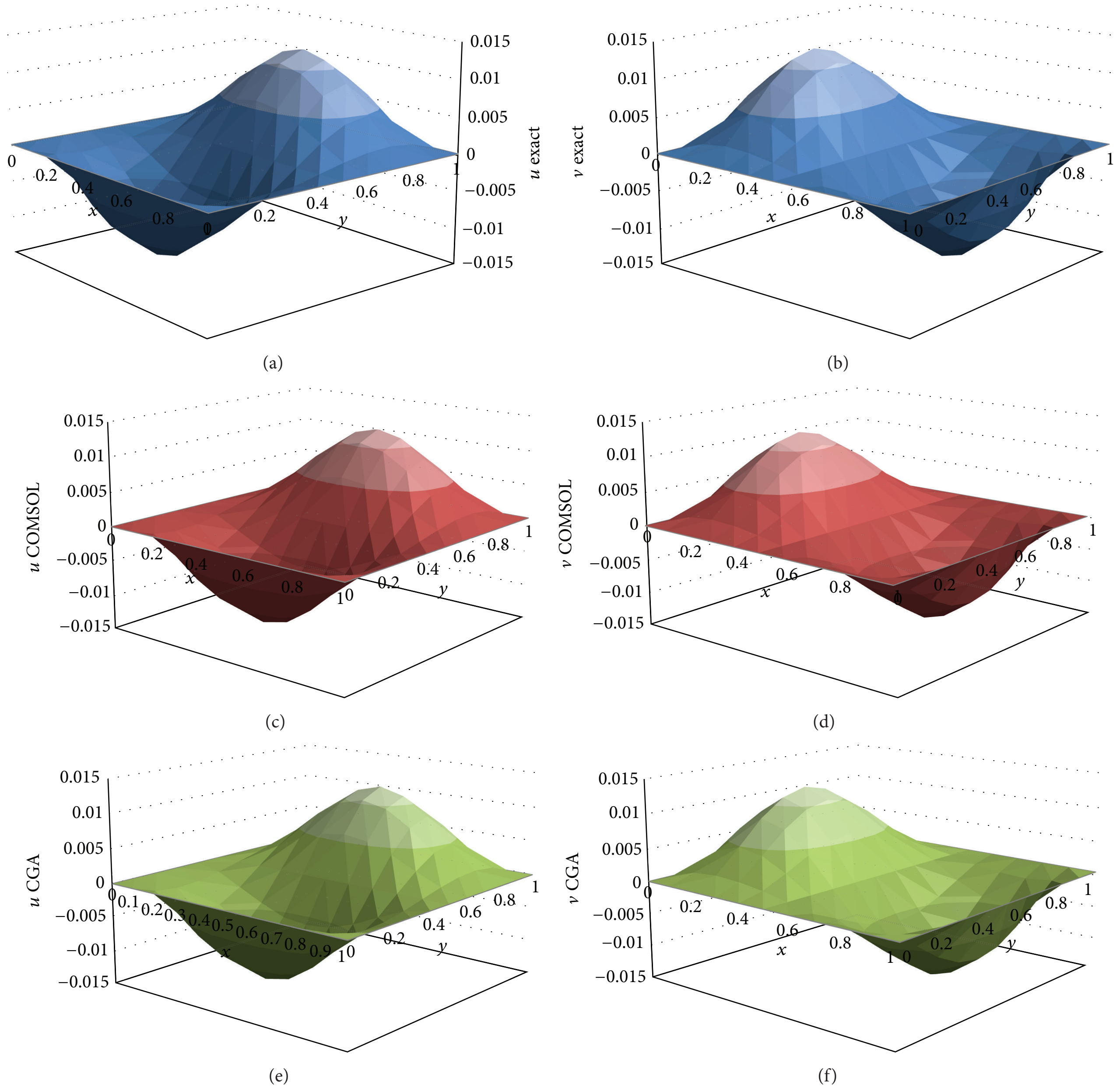


Figure 3: For Stokes case. (a) Graphical representation for actual values of u. (b) Graphical representation for actual values of v. (c) Finite element solution for u variable using COMSOL. (d) Finite element solution for v variable using COMSOL. (e) Graphical representation for nodal u values using CGA. (f) Graphical representation for nodal v values using CGA.

it is found that the error for solution obtained by CGA is close to that error for solution obtained by finite element. On the other hand, L_2 norm error for using COMSOL is less, that was about 1.29×10^{-5} for u variable and about 1.21×10^{-5} for v values and even these errors are less than the errors of CGAs method.

The average number of generations was 1139, and the average residual value was about 6.339×10^{-4}. Then, the average fitness according to (4) is very close to the maximum fitness value of 1.

The convergence state of nodal values of the nodes was presented along all generations such as an example for $[u(2,2), u(6,6)$, and $u(6,10)]$ nodes excluding the boundary values as shown in Figures 4, 5, and 6, respectively. It is concluded from the figures that steady state is reached for $u(2,2)$ nodes much earlier than that for $u(6,6)$ and $u(6,10)$. The reason for that is that the node $(2,2)$ location is closer to the boundaries compared with the other nodes locations. For the same reason, the convergence for node $(6,10)$ is faster than for $(6,6)$.

The nodal residues in this case were less than 14.925×10^{-4} for u values as shown in Table 7 and were less than 18.441×10^{-4} for v values as shown in Table 8. It can be concluded from these values that the solution is close to the exact solution.

Table 9: Actual u nodal values for nonlinear Navier Stokes equation including boundary values.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
y_2	0	0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.2
y_3	0	0.03	0.06	0.09	0.12	0.15	0.18	0.21	0.24	0.27	0.3
y_4	0	0.04	0.08	0.12	0.16	0.2	0.24	0.28	0.32	0.36	0.4
y_5	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
y_6	0	0.06	0.12	0.18	0.24	0.3	0.36	0.42	0.48	0.54	0.6
y_7	0	0.07	0.14	0.21	0.28	0.35	0.42	0.49	0.56	0.63	0.7
y_8	0	0.08	0.16	0.24	0.32	0.4	0.48	0.56	0.64	0.72	0.8
y_9	0	0.09	0.18	0.27	0.36	0.45	0.54	0.63	0.72	0.81	0.9
y_{10}	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

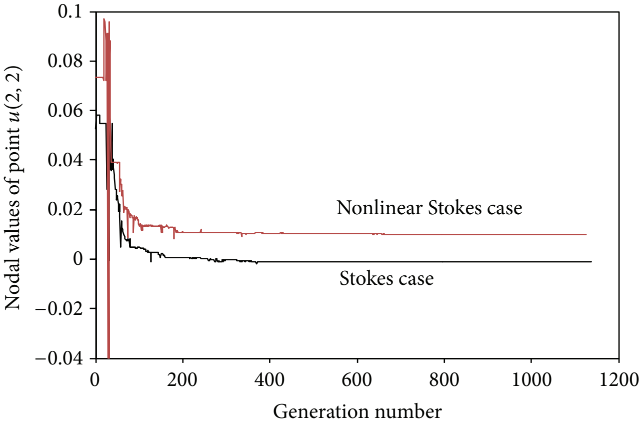


Figure 4: Nodal values of the point $u(2, 2)$ along all generations for Stokes and nonlinear Stokes cases.

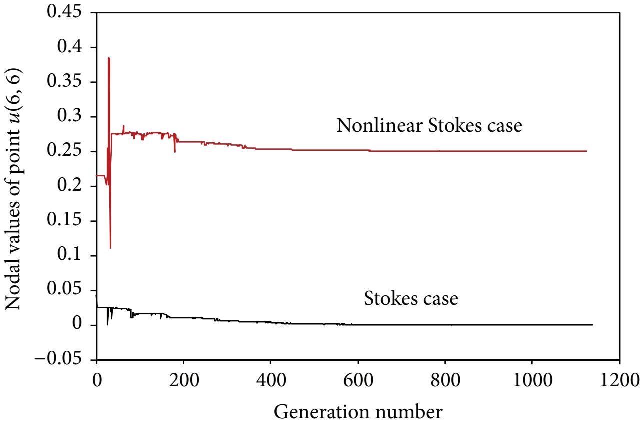


Figure 5: Nodal values of the point $u(6, 6)$ along all generations for Stokes and nonlinear Stokes cases.

Maximum fitness was reached with smooth increasing toward the maximum value along all generations and this assures how much the solution of CGA is close to the true solution. This behavior is shown in Figure 7(a).

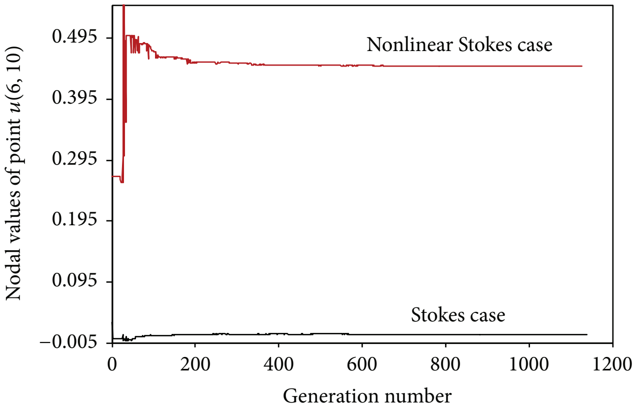


Figure 6: Nodal values of the point $u(6, 10)$ along all generations for Stokes and nonlinear Stokes cases.

4. Numerical Results for Nonlinear Navier Stokes Equations

The following conditions were taken into consideration while solving the problem.

Boundary conditions are as follows:

$$\begin{aligned} u(0,y)&=0, & u(x,0)&=0,\\ u(1,y)&=y, & u(x,1)&=x,\\ v(0,y)&=0, & v(x,0)&=0,\\ v(1,y)&=y^2, & v(x,1)&=x^2. \end{aligned} \tag{26}$$

The exact solutions are as follows:

$$\begin{aligned} u&=xy,\\ v&=x^2y^2. \end{aligned} \tag{27}$$

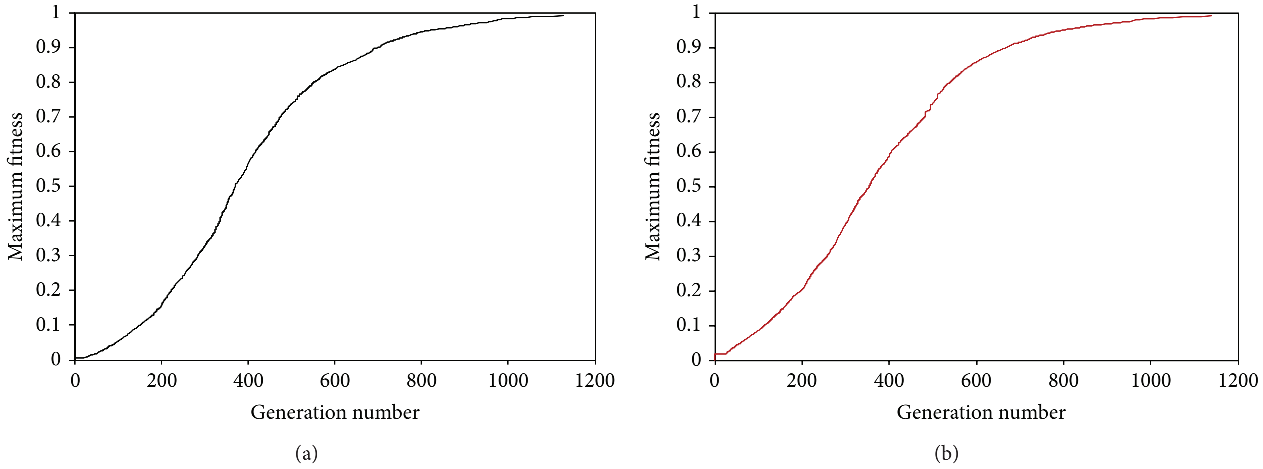


Figure 7: Maximum fitness values along all generations for Stokes and nonlinear Stokes cases.

Table 10: Actual v nodal values for nonlinear Navier Stokes equation including boundary values.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.0001	0.0004	0.0009	0.0016	0.0025	0.0036	0.0049	0.0064	0.0081	0.01
y_2	0	0.0004	0.0016	0.0036	0.0064	0.01	0.0144	0.0196	0.0256	0.0324	0.04
y_3	0	0.0009	0.0036	0.0081	0.0144	0.0225	0.0324	0.0441	0.0576	0.0729	0.09
y_4	0	0.0016	0.0064	0.0144	0.0256	0.04	0.0576	0.0784	0.1024	0.1296	0.16
y_5	0	0.0025	0.01	0.0225	0.04	0.0625	0.09	0.1225	0.16	0.2025	0.25
y_6	0	0.0036	0.0144	0.0324	0.0576	0.09	0.1296	0.1764	0.2304	0.2916	0.36
y_7	0	0.0049	0.0196	0.0441	0.0784	0.1225	0.1764	0.2401	0.3136	0.3969	0.49
y_8	0	0.0064	0.0256	0.0576	0.1024	0.16	0.2304	0.3136	0.4096	0.5184	0.64
y_9	0	0.0081	0.0324	0.0729	0.1296	0.2025	0.2916	0.3969	0.5184	0.6561	0.81
y_{10}	0	0.01	0.04	0.09	0.16	0.25	0.36	0.49	0.64	0.81	1

Table 11: Nodal values of u using finite element method in COMSOL for the nonlinear case.

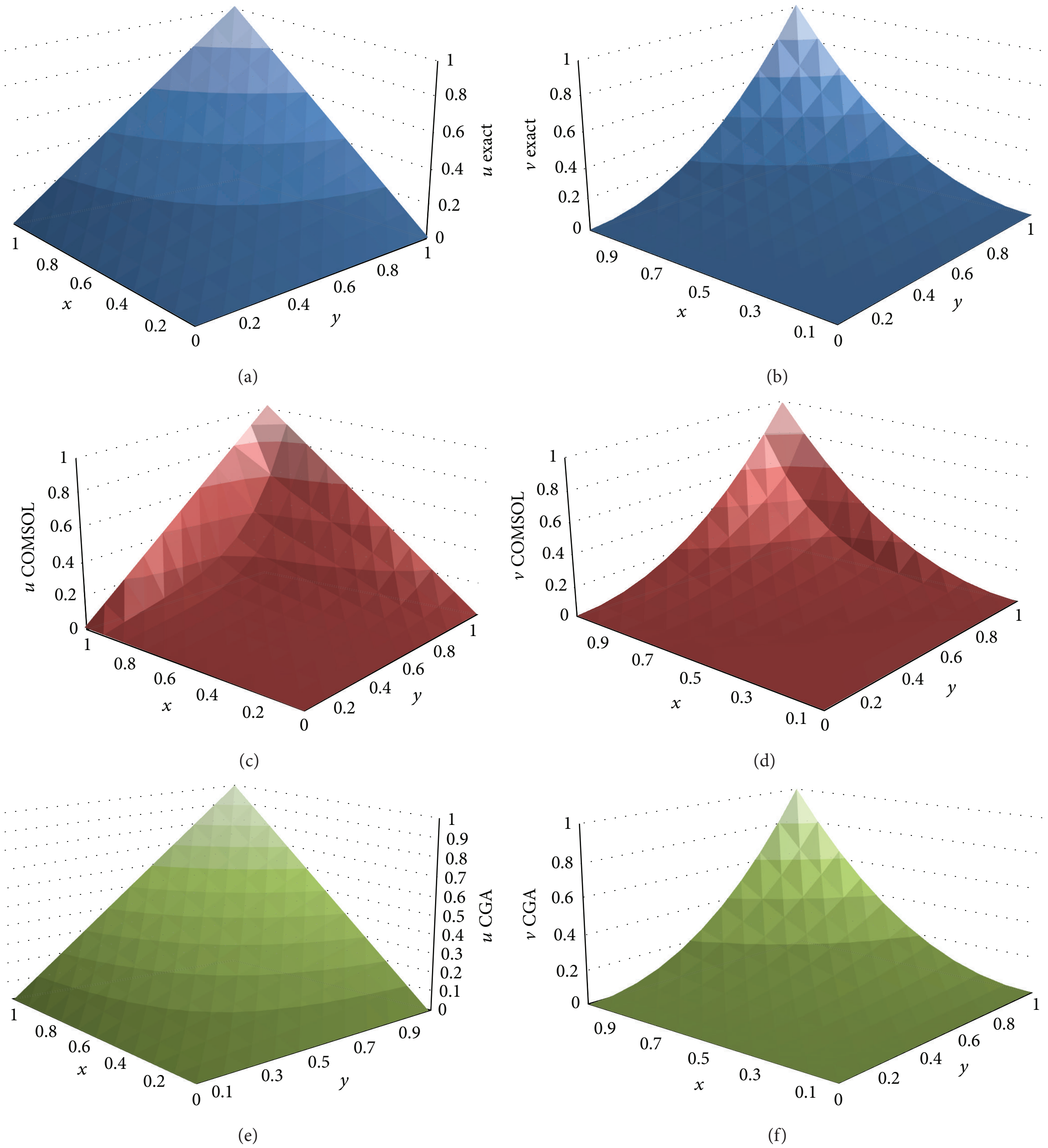
	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.00114	0.00238	0.00389	0.0059	0.00881	0.01335	0.0209	0.03432	0.06002	0.1
y_2	0	0.00214	0.00445	0.00723	0.01095	0.01641	0.02519	0.04074	0.07091	0.12767	0.2
y_3	0	0.003	0.00621	0.01008	0.01535	0.02337	0.03714	0.06365	0.11641	0.20307	0.3
y_4	0	0.00379	0.00781	0.01272	0.01956	0.03052	0.05059	0.09103	0.16659	0.27348	0.4
y_5	0	0.00461	0.00947	0.01544	0.02399	0.03828	0.06577	0.12076	0.21378	0.33949	0.5
y_6	0	0.00568	0.01155	0.01872	0.02913	0.04717	0.08281	0.15151	0.25716	0.40327	0.6
y_7	0	0.00757	0.01506	0.02385	0.03642	0.05846	0.10225	0.18249	0.29951	0.46673	0.7
y_8	0	0.01211	0.02323	0.03537	0.05133	0.07788	0.12894	0.21691	0.34411	0.53192	0.8
y_9	0	0.02659	0.04985	0.07414	0.10151	0.13769	0.19393	0.2804	0.41185	0.61123	0.9
y_{10}	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

The second terms in the right hand side of (16) and (17) are defined in (28) and they were obtained by substituting the exact solutions given in (27) into (16) and (17):

$$\begin{gathered} f_x = x^3 y^2 + x y^2, \\ f_y = 2x^4 y^3 + 2x^2 y^3 - 2x^2 - 2y^2. \end{gathered} \tag{28}$$

The actual nodal values of both u and v variables are shown in Tables 9 and 10. The graphical representations of those actual values are shown in Figures 8(a) and 8(b).

Tables 11 and 12 show the obtained nodal values of u and v variables as a result of applying COMSOL. Moreover, Figures 8(c) and 8(d) show the graphical representations of these solution values.



(a) (b) (c) (d) (e) (f)

Figure 8: For nonlinear Navier Stokes. (a) Graphical representation for actual values of u. (b) Graphical representation for actual values of v. (c) Finite element solution for u using COMSOL. (d) Finite element solution for v using COMSOL. (e) CGA solution for u. (f) CGA solution for v.

For solution obtained by COMSOL, the norm error was found to be about 1.6258 for u variable values and about 0.5845 for v variable. And the time needed to solve this problem was about 0.546 second. For the nonlinear Navier Stokes, the results of using COMSOL to solve u variable for different sizes of meshes are shown in Table 13.

4.1. Solution of Nonlinear Navier Stokes Using Advanced Continuous Genetic Algorithm. Firstly, the discretization of (16) is

$$u_{i,j}\left[\frac{u_{i+1,j}-u_{i-1,j}}{2h_x}\right]+v_{i,j}\left[\frac{u_{i,j+1}-u_{i,j-1}}{2h_y}\right]$$
$$=\left[\frac{u_{i-1,j}-2u_{i,j}+u_{i+1,j}}{h_x^2}+\frac{u_{i,j-1}-2u_{i,j}+u_{i,j-1}}{h_y^2}\right]+f_x \quad (29)$$

Table 12: Nodal values of v using finite element method in COMSOL for nonlinear case.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.00117	0.00205	0.00265	0.0031	0.00333	0.0033257	0.0027	0.00068	0.00413	0.01
y_2	0	0.00218	0.00351	0.00410	0.0042	0.00398	0.0032	0.0012	0.00363	0.01451	0.04
y_3	0	0.00302	0.00435	0.00441	0.0036	0.00229	4×10^{-6}	0.00404	0.0118	0.02991	0.09
y_4	0	0.00362	0.0045	0.00349	0.0013	0.0021	0.0066	0.01319	0.02459	0.05177	0.16
y_5	0	0.00385	0.00375	0.00101	0.0036	0.0096	0.01758	0.02864	0.04622	0.08582	0.25
y_6	0	0.00355	0.00171	0.00374	0.0116	0.0222	0.0368	0.05721	0.08729	0.14279	0.36
y_7	0	0.00243	0.00232	0.01169	0.0251	0.0446	0.07346	0.11277	0.16266	0.23196	0.49
y_8	0	6.4×10^{-5}	0.00922	0.02522	0.0495	0.0878	0.14193	0.20487	0.27293	0.35907	0.64
y_9	0	0.00398	0.02085	0.049881	0.0956	0.163	0.24296	0.32916	0.42668	0.54521	0.81
y_{10}	0	0.01	0.04	0.09	0.16	0.25	0.36	0.49	0.64	0.81	1

Table 13: COMSOL results for different meshes.

Number of elements in mesh	L_2 norm error	CPU time (s)
100	1.6258	0.55
228	0.0213	0.57
365	0.0207	0.702
798	0.0201	1.014
2736	0.0195	2.386
24865	0.019	18.034

and is the same for the second equation. Then, for residues, one has

$$r_{1i,j} = u_{i,j}\left[\frac{u_{i+1,j} - u_{i-1,j}}{2h_x}\right] + v_{i,j}\left[\frac{u_{i,j+1} - u_{i,j-1}}{2h_y}\right] - \left[\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h_x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j-1}}{h_y^2}\right] - f_x, \tag{30}$$

Overall residues are the summation for both first and second equations residues:

$$R1 = \sqrt{\sum_{i=1}^{n_x}\sum_{j=1}^{m_y} r_{1i,j}^2}, \qquad R2 = \sqrt{\sum_{i=1}^{n_x}\sum_{j=1}^{m_y} r_{2i,j}^2}, \tag{31}$$
$$R = R1 + R2.$$

A solution using continuous genetic algorithm (CGA) for case 2 (nonlinear Navier Stokes equation [Dirichlet boundary conditions]) is shown in Tables 14 and 15, including the nodal values of the boundaries. And the graphical representation of these values is shown in Figures 8(e) and 8(f).

The results of applying CGA can be summarized in calculating the L_2 norm error that was found to be about 1.42×10^{-4} for u variable values and about 9.76×10^{-5} for v variable values. The time that CGA needed to solve such kind of problems was about 7456 seconds.

It can be concluded from the results that solving nonlinear equations using finite element method in COMSOL application have a low accuracy. On the other hand, the trend of accuracy for CGA method is the same for both linear and nonlinear cases.

The CGAs process for solving nonlinear Navier Stokes equations in visual basic tools about 7456 seconds, in which the average number of generations was 1127 generations. Moreover, the average residual value was about 5.75433×10^{-4}. Then, the average fitness values according to the equation $F = 1/(1 + R) = 0.9994$ are very close to the maximum fitness value of unity.

The convergence state of nodal values of the nodes was presented along all generations such as an example for $[u(2,2), u(6,6)$, and $u(6,10)]$ nodes excluding the boundaries values as shown in Figures 4, 5, and 6, respectively. It is concluded from the figures that steady state is reached for $u(2,2)$ nodes much earlier than that for $u(6,6)$ and $u(6,10)$. The reason for that is that the node $(2,2)$ location is much closer to the boundaries compared with the other nodes locations. For the same reason, the convergence for node $(6,10)$ is faster than for $(6,6)$.

The nodal residues in this case were less than 22.1059×10^{-4} for u values as shown in Table 16 and were less than 25.686×10^{-4} for v values as shown in Table 17. Maximum fitness was reached with smooth increasing toward the maximum value along all generations and this assures how much the solution of CGA is close to the true solution. This behavior is shown in Figure 7(b).

Table 18 summarizes the overall performance of the used solvers solutions for the selected cases of Navier Stokes equations. As shown in the table, the CGA method has good results in solving both linear and nonlinear problems. It is worth mentioning that in the Stokes case the accuracy of COMSOL was higher than that of CGA and Wang et al. results. However, in the nonlinear case, COMSOL results' errors were higher than those of the CGA results.

As a result, it can be concluded that COMSOL software has good predictions and small errors for the linear cases

Table 14: Nodal u values including boundaries for nonlinear Navier Stokes equation using CGA.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.01002	0.020019	0.03002	0.04002	0.05001	0.06	0.07	0.08	0.09	0.1
y_2	0	0.02002	0.040022	0.06003	0.08002	0.10001	0.12001	0.14001	0.16	0.18	0.2
y_3	0	0.03002	0.06002	0.09003	0.12002	0.15002	0.18001	0.21001	0.24001	0.27	0.3
y_4	0	0.04	0.080009	0.12002	0.16002	0.20002	0.24002	0.28002	0.32002	0.36001	0.4
y_5	0	0.04999	0.1	0.15001	0.20002	0.25002	0.30002	0.35003	0.40002	0.45001	0.5
y_6	0	0.05998	0.119997	0.18	0.24002	0.30002	0.36002	0.42003	0.48003	0.54002	0.6
y_7	0	0.06999	0.139998	0.21001	0.28002	0.35002	0.42002	0.49003	0.56003	0.63002	0.7
y_8	0	0.07999	0.160001	0.24001	0.32001	0.40001	0.48002	0.56002	0.64002	0.72001	0.8
y_9	0	0.09001	0.180006	0.27001	0.36001	0.45001	0.54001	0.63001	0.72001	0.81001	0.9
y_{10}	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

Table 15: Nodal v values including boundaries for nonlinear Navier Stokes equation using CGA.

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	0.00011	0.0004	0.0009	0.0016	0.00251	0.0036	0.0049	0.0064	0.0081	0.01
y_2	0	0.00039	0.0016	0.0036	0.00641	0.01001	0.0144	0.0196	0.02561	0.0324	0.04
y_3	0	0.00088	0.00359	0.00809	0.01441	0.02251	0.0324	0.0441	0.05761	0.0729	0.09
y_4	0	0.00158	0.00638	0.01439	0.0256	0.04001	0.0576	0.0784	0.10241	0.1296	0.16
y_5	0	0.00249	0.00999	0.02249	0.04	0.06251	0.09	0.1225	0.16001	0.2025	0.25
y_6	0	0.0036	0.0144	0.0324	0.0576	0.09	0.1296	0.1764	0.23041	0.2916	0.36
y_7	0	0.00492	0.01961	0.04411	0.07839	0.12249	0.1764	0.2401	0.3136	0.3969	0.49
y_8	0	0.00642	0.02561	0.05761	0.10239	0.15999	0.2304	0.3136	0.4096	0.51841	0.64
y_9	0	0.00813	0.03242	0.07291	0.12959	0.20248	0.2916	0.3969	0.5184	0.65611	0.81
y_{10}	0	0.01	0.04	0.09	0.16	0.25	0.36	0.49	0.64	0.81	1

Table 16: Residual u values for all nodes for nonlinear Navier Stokes equation after using CGA ($\times 10^{-4}$).

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	3.2678	3.4335	2.66768	5.61671	2.83248	0.4064	3.16061	1.10452	2.13471	0
y_2	0	8.4514	1.4042	4.34961	1.19837	0.67391	2.3282	6.07949	4.44807	4.88446	0
y_3	0	4.3892	3.1142	2.07415	11.8851	1.81051	1.3184	22.1059	11.5826	5.51202	0
y_4	0	10.995	3.8016	13.5119	14.0723	0.54436	2.6779	9.73838	8.83235	0.27101	0
y_5	0	11.113	1.4708	6.96716	11.347	6.3083	4.7194	9.58696	7.05931	2.25748	0
y_6	0	7.9889	1.4367	0.50162	5.15927	1.85226	2.642	11.4612	3.56919	0.78898	0
y_7	0	8.3608	3.1107	30.6914	6.85968	3.82915	0.8626	0.45646	5.65133	0.51875	0
y_8	0	2.1557	6.875	18.7506	4.19258	0.02673	2.7288	5.30467	0.00834	3.6369	0
y_9	0	2.1154	4.5035	4.44288	5.03227	0.89008	2.042	0.79609	1.67729	0.46048	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

but not for nonlinear problems since its accuracy dramatically deceased. This makes CGA technique advantageous over the COMSOL technique because of its adaptability to the nonlinear nature of the problems with reasonable and consistent errors. However, it should be mentioned that the CGA method requires longer time periods than COMSOL in solving the problems. This is a new challenge for CGA, which could be overcome through the continuous development of the CGA algorithm or the computer hardware. However, it is worth mentioning that CGA has good accuracy even in the first run of solving such kind of problems.

5. Conclusions

In this paper, two-dimensional CGA was applied for the solution of steady-state two-dimensional Stokes and nonlinear Navier Stokes problems. The proposed CGA for the solution

Table 17: Residual v values for all nodes for nonlinear Navier Stokes equation after using CGA ($\times 10^{-4}$).

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_0	0	0	0	0	0	0	0	0	0	0	0
y_1	0	6.5086	4.0087	12.004	11.414	11.025	18.179	1.3779	0.3946	4.3496	0
y_2	0	0.2215	0.2655	2.231	2.344	2.9315	5.6304	7.1332	6.5507	1.8278	0
y_3	0	16.501	0.157	21.686	19.681	10.5	11.835	0.8445	3.0172	6.0424	0
y_4	0	6.6381	4.0749	0.8183	4.7228	8.8724	21.16	7.4542	8.3205	3.6702	0
y_5	0	2.9267	8.0611	13.552	2.0773	0.2907	10.927	5.7256	3.4446	1.6143	0
y_6	0	18.818	8.7243	25.686	3.0502	4.1499	3.4448	1.7468	2.8593	4.1917	0
y_7	0	5.8223	2.5064	11.64	9.513	7.5191	14.893	2.2233	2.9787	7.9392	0
y_8	0	12.419	6.9534	2.7893	10.201	13.735	1.7175	6.0922	4.7986	2.2204	0
y_9	0	2.2476	1.6574	3.4965	2.8279	3.9258	1.4725	0.6767	4.4832	0.8819	0
y_{10}	0	0	0	0	0	0	0	0	0	0	0

Table 18: Summary of all Navier Stokes cases results.

	Solution method	Average L_2 norm error (according to exact solution)	CPU time (seconds)	Tolerance and termination criteria
(Stokes case)	CGA	2.05×10^{-4}	7523	Fitness = 0.99 or number of generations = 3000
	Finite element Wang et al., (2009) [24]	5.05×10^{-4}	—	1×10^{-8}
	Galerkin finite element in COMSOL	1.25×10^{-5}	0.045	1×10^{-6}
(Nonlinear Navier Stokes case)	CGA	1.198×10^{-4}	7456	Fitness = 0.99 or number of generations = 3000
	Galerkin finite element in COMSOL	1.1052	0.546	1×10^{-6}

of partial differential equations has the following advantages: first, it does not require any modification while switching from the linear to the nonlinear case, only the discretization of the problem will be changed, while the solution procedure using CGA will be the same; second, the algorithm requires minimal knowledge about the aimed problem that is the discretized form of the partial differential equation in the residual form; third, the method is found to be simple, efficient, and attractive with great potential in engineering applications. The resulting solutions provided good accuracy for Stokes case and excellent accuracy as compared to other commercial packages for the nonlinear case.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] M. B. Abbott and D. R. Basco, *Computational Fluid Dynamics: An Introduction for Engineers*, Long Man Group, New York, NY, USA, 1989.

[2] J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics*, Springer, New York, NY, USA, 1996.

[3] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[4] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Mass, USA, 1996.

[5] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[6] L. Davis, *Hand Book of Genetic Algorithms*, Van Norstrand Reinhold, New York, NY, USA, 1991.

[7] Z. S. Abo-Hammour, *Advanced continuous genetic algorithms and their applications in the motion planning of robot manipulators and in numerical solution of boundary value problems [Ph.D. thesis]*, Quaid-i-Azam University, Islamabad, Pakistan, 2002.

[8] Z. S. Abo-Hammour, M. Yusuf, N. M. Mirza, S. M. Mirza, M. Arif, and J. Khurshid, "Numerical solution of second-order, two-point boundary value problems using continuous genetic algorithms," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 8, pp. 1219–1242, 2004.

[9] H. Jaradat, *Numerical solution of temporal two point boundary value problems using continuous genetic algorithms [Ph.D. thesis]*, University of Jordan, Amman, Jordan, 2006.

[10] O. Abu Arqub, Z. Abo-Hammour, S. Momani, and N. Shawagfeh, "Optimization solution of Troesch's and Bratu's problems of ordinary type using novel continuous genetic

algorithm," *Discrete Dynamics in Nature and Society*, vol. 2014, Article ID 401696, 15 pages, 2014.

[11] O. Abu Arqub, Z. Abo-Hammour, and S. Momani, "Application of continuous genetic algorithm for nonlinear system of second-order boundary value problems," *Applied Mathematics and Information Systems*, vol. 8, no. 1, Article ID 205391, pp. 235–248, 2014.

[12] O. Abu Arqub, Z. Abo-Hammour, and S. Momani, "An optimization algorithm for solving systems of singular boundary value problems," *Applied Mathematics and Information Systems*, 2014.

[13] O. Abu Arqub, Z. Abo-Hammour, and S. Momani, "Solving singular two-point boundary value problems using continuous genetic algorithm," *Abstract and Applied Analysis*, vol. 2012, Article ID 205391, 25 pages, 2012.

[14] Z. S. Abo-Hammour, R. B. Albadarneh, and M. S. Saraireh, "Solution of Laplace equation using continuous genetic algorithms," *Kuwait Journal of Science and Engineering A*, vol. 37, no. 2, pp. 1–15, 2010.

[15] O. F. Al-Sayyed, *Numerical solution of differential-algebraic equations using continuous genetic algorithms [Ph.D. thesis]*, University of Jordan, Amman, Jordan, 2006.

[16] O. M. Abo-Arqoub, *Numerical solution of fuzzy differential equations using continuous genetic algorithms [Ph.D. thesis]*, University of Jordan, Amman, Jordan, 2008.

[17] A. Samhouri, *Applicability of continuous genetic algorithm in solving navier stokes problems [M.S. thesis]*, University of Jordan, Amman, Jordan, 2012.

[18] Z. S. Abo-Hammour, N. M. Mirza, S. M. Mirza, and M. Arif, "Cartesian path generation of robot manipulators using continuous genetic algorithms," *Robotics and Autonomous Systems*, vol. 41, no. 4, pp. 179–223, 2002.

[19] Z. S. Abo-Hammour, O. M. Alsmadi, S. I. Bataineh, M. A. Al-Omari, and N. Affach, "Continuous genetic algorithms for collision-free Cartesian path planning of robot manipulators regular paper," *International Journal of Advanced Robotic Systems*, vol. 8, no. 6, pp. 14–36, 2011.

[20] O. M. K. Alsmadi, Z. S. Abo-Hammour, A. M. Al-Smadi, and M. S. Saraireh, "A robust and efficient genetic algorithm for solving a chemical reactor problem: theory, application and convergence analysis," *Transactions of the Institute of Measurement and Control*, vol. 34, no. 5, pp. 594–603, 2012.

[21] Z. S. Abo-Hammour, A. G. Asasfeh, A. M. Al-Smadi, and O. M. K. Alsmadi, "A novel continuous genetic algorithm for the solution of optimal control problems," *Optimal Control Applications and Methods*, vol. 32, no. 4, pp. 414–432, 2011.

[22] L. L. Lapin, *Probability and Statistics for Modern Engineering*, PWS-KENT, Boston, Mass, USA, 2nd edition, 1990.

[23] S. C. R. Dennis and J. D. Hudson, "Compact h4 finite-difference approximations to Operators of Navier-Stokes Type," *Journal of Computational Physics*, vol. 85, no. 2, pp. 390–416, 1989.

[24] J. Wang, Y. Wang, and X. Ye, "A robust numerical method for stokes equations based on divergence-free iidie(div) finite element methods," *SIAM Journal on Scientific Computing*, vol. 31, no. 4, pp. 2784–2802, 2009.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 761403, 14 pages
http://dx.doi.org/10.1155/2014/761403

Research Article

A Novel Adaptive Elite-Based Particle Swarm Optimization Applied to VAR Optimization in Electric Power Systems

Ying-Yi Hong,[1] Faa-Jeng Lin,[2] Syuan-Yi Chen,[3] Yu-Chun Lin,[2] and Fu-Yuan Hsu[1]

[1] *Department of Electrical Engineering, Chung Yuan Christian University, Chung Li City 320, Taiwan*
[2] *Department of Electrical Engineering, National Central University, Chung Li City 320, Taiwan*
[3] *Department of Applied Electronics Technology, National Taiwan Normal University, Taipei 320, Taiwan*

Correspondence should be addressed to Ying-Yi Hong; yyh10632@yahoo.com.tw

Received 5 February 2014; Accepted 30 March 2014; Published 22 May 2014

Academic Editor: Ker-Wei Yu

Particle swarm optimization (PSO) has been successfully applied to solve many practical engineering problems. However, more efficient strategies are needed to coordinate global and local searches in the solution space when the studied problem is extremely nonlinear and highly dimensional. This work proposes a novel adaptive elite-based PSO approach. The adaptive elite strategies involve the following two tasks: (1) appending the mean search to the original approach and (2) pruning/cloning particles. The mean search, leading to stable convergence, helps the iterative process coordinate between the global and local searches. The mean of the particles and standard deviation of the distances between pairs of particles are utilized to prune distant particles. The best particle is cloned and it replaces the pruned distant particles in the elite strategy. To evaluate the performance and generality of the proposed method, four benchmark functions were tested by traditional PSO, chaotic PSO, differential evolution, and genetic algorithm. Finally, a realistic loss minimization problem in an electric power system is studied to show the robustness of the proposed method.

1. Introduction

Particle swarm optimization (PSO) is a population-based optimization method [1]. PSO attempts to mimic the goal-searching behavior of biological swarms. A possible solution of the optimization problem is represented by a particle. The PSO algorithm requires iterative computation, which consists of updating the individual velocities of all particles at their corresponding positions. The PSO algorithm has some merits: (1) it does not need the crossover and mutation operations used in genetic algorithms; (2) it has a memory so that the elite solutions can be retained by all particles (solutions); and (3) the computer code of PSO is easy to develop. The PSO has been successfully applied to solve many problems, for example, reconfiguration of shipboard power system [2], economic dispatch [3], harmonic estimation [4], harmonic cancellation [5], and energy-saving load regulation [6].

PSO is designed to conduct global search (exploration) and local search (exploitation) in the solution space. The former explores different extreme points of the search space by jumping among extreme particles. In contrast, the latter searches for the extreme particle in a local region. The particles ultimately converge to the same extreme position. However, when a problem involves a great number of unknowns, the PSO generally requires a large number of particles in order to gain a global optimal solution (position). Consequently, achieving the coordination between the global search and the local search becomes more difficult.

To overcome the limitation described above, van den Bergh and Engelbrecht presented the cooperative PSO (COPSO) based on the dimension partition [7]. Nickabadi et al. focused on several well-known inertia weighting strategies and gave a first insight into various velocity control approaches [8]. The adaptive PSO (APSO) presented by Zhan et al. utilized the information on population distribution to identify the search status of particles; the learning strategy tries to find an elitist and then search the global best position in an iterative step [9]. Caponetto et al. presented a chaos algorithm to compute the inertia weight of preceding position of a particle and a self-tuning method to compute learning factors [10]. Their method introduces a chaos parameter, which is determined by a logistic map. The learning factors are adaptive according to the fitness value of the best solution during the iterations. To enhance the scalability of PSO, José and Enrique presented two mechanisms [11]. First, a velocity modulation is designed to guide the particles to the region of interest. Second, a restarting modulation redirects the particles to promising areas in the search space. Deng et al. presented a two-stage hybrid swarm intelligence optimization algorithm incorporating with the genetic algorithm, PSO, and ant colony optimization to conduct rough searching and detailed searching by making use of the advantages of the parallel and positive feedback [12]. Li et al. presented a cooperative quantum-behaved PSO, which produces several particles using Monte Carlo method first and then these particles cooperate with one another to converge to the global optimum [13]. Recently, Arasomwan and Adewumi tried many chaotic maps, in additional to the traditional logistic map, which attempted to enhance the performances of PSO [14]. Yin et al. incorporated the crossover and mutation operators in PSO to enhance diversity in the population; besides, a local search strategy based on constraint domination was proposed and incorporated into the proposed algorithm [15].

The variants of PSO generally have the following limitations.

(1) The solutions are likely to be trapped in the local optima and undesirable premature situations because it relies heavily on the learning factors and inertia weight.

(2) Compared with the traditional PSO, variants of PSO require longer CPU times to compute the global optimum due to the extra treatment in exploration and exploitation.

(3) The iterative process is unstable or even fails to converge because of the ergodic and dynamic properties, for example, chaos sequence.

Based on the above literature review, a novel PSO is proposed in this paper. This novel adaptive elite-based PSO employs the mean of all particles and standard deviation of distances between any two particles. In addition to global and local searches, a new mean search is introduced as an alternative method for finding the global optimum. The standard deviation of distances among all particles is utilized to prune the distant particles. The best particle is cloned to replace the discarded particles. This treatment ensures a stable iterative process. The increase in the computational burden of this enhancement is trivial.

The rest of this paper is organized as follows. Section 2 provides the general PSO algorithm. Section 3 proposes the novel elite-based PSO. In Section 4, four benchmark functions are utilized to validate the proposed method. Comparative studies among different optimization methods (traditional PSO, chaotic PSO [16], differential evolution [17], and genetic algorithm) are given. The results of its application to the real problem of loss minimization in an electric power system are presented. Section 5 draws conclusions.

2. General PSO Algorithm

PSO, which is an evolutionary optimization method for minimizing an objective $f(X)$, reflects the behavior of flocking birds or schooling fish. PSO comprises a population of particles iteratively updating the empirical information about a search space. The population consists of many individuals that represent possible solutions and are modeled as particles moving in a ψ-dimensional search space. Let the superscript t be the iteration index. The position and velocity of each particle p are updated as follows [1]:

$$V_p^{t+1} = \omega^t \times V_p^t + C_1 \times r_1^t \times \left(p_{\text{best}}^t - X_p^t\right) + C_2 \times r_2^t \times \left(g_{\text{best}}^t - X_p^t\right), \tag{1}$$

$$X_p^{t+1} = X_p^t + V_p^{t+1}, \tag{2}$$

where vectors X_p and V_p are the ψ-dimensional position and velocity of particle p, $p = 1, 2, \ldots, P$ (number of population size), respectively. The inertia weight ω is designed to copy the previously updated features to the next iteration. If $\omega = 1$, then the preceding V_p^t has a strong impact on V_p^{t+1}. p_{best} and g_{best} are the best position of a particle and the best known position found by any particle in the swarm so far, respectively. The random numbers r_1 and r_2 are between 0 and 1. Learning factors C_1 and C_2 are positive constants constrained within the range 0 and 2 such that $C_1 + C_2 \leq 4$. The products $C_1 \times r_1$ and $C_2 \times r_2$ thus stochastically control the overall velocity of a particle. In this paper, X_p^{t+1} denotes the pth vector (particle) at the $(t+1)$th iteration and V_p^{t+1} can be regarded as the $(t+1)$th updated value (ΔX_p^t) that is added to X_p^t.

The second term in (1) is designed to conduct the global search (exploration) and the third term in (1) describes the local search (exploitation) in the solution space, as described in Section 1. The global and local searches should be coordinated in order to gain the global optimum and avoid premature results. The inertia weight ω decreases linearly throughout the iterations and governs the momentum of the particle by weighting the contribution of the particle's previous velocity to its current velocity. A large ω is designed for global searches, whereas a small ω is used for local searches. Consequently, in the earlier stages of the search process, a large ω is preferred, while a small value is preferred in the later stages.

3. Proposed Adaptive Elite-Based PSO

The proposed novel PSO has the following three features. (1) An extra mean search is introduced to coordinate exploration and exploitation; (2) distant particles are pruned and the best particle is cloned to ensure that the iterative process is stable; and (3) complicated computation is avoided and CPU time is thus reduced. The proposed adaptive elite-based PSO, thus, performs 2 main tasks: mean search and particle pruning/cloning. The effects of these two tasks decline as the number of iterations increases. The limitations of the traditional variants of PSO described in Section 1 can therefore be eliminated.

3.1. Mean Search. The proposed method is based on the mean of the particles and standard deviation of the distances between any two particles in the tth iterations. Equation (1) is herein modified to the following:

$$V_p^{t+1} = \omega^t \times V_p^t + C_1^t \times r_1^t \times \left(p_{\text{best}}^t - X_p^t\right) + C_2^t \times r_2^t \times \left(g_{\text{best}}^t - X_p^t\right) + C_3^t \times r_3^t \times \left(g_{\text{best}}^t - X_m^t\right), \tag{3}$$

where the inertia weight ω^t is decreased linearly from 0.5 in the first iteration to 0.3 in the final iteration. In addition to r_1 and r_2, the term r_3 is also a random number between 0 and 1. X_m^t is a ψ-dimensional vector of the mean values of all particles. The last term introduced in (3) is used to coordinate the global and local searches as well as the particle pruning/cloning task later.

Because $C_1 + C_2 \leq 4$ [18], the new learning factor is defined by

$$C_3^t = 4 - C_1^t - C_2^t, \tag{4}$$

where

$$C_1^t = C_2^t = 1 + \frac{1}{1 + \exp\left(\alpha / F\left(g_{\text{best}}^t\right)\right)} \tag{5}$$

and $\alpha = F(g_{\text{best}}^1)$ [10]. Equation (4) ensures that C_3^t decreases gradually to zero and (3) becomes (1) when the iterative process converges.

3.2. Particle Pruning/Cloning. X_m^t introduced in Section 3.1 is used to develop the second task of the adaptive elite strategy: pruning the distant particles. The standard deviation σ^t of all distances among particles in the tth iteration is evaluated. Suppose that all distances follow a Gaussian distribution. Then $X_m^t \pm 3\sigma^t$ covers 99.7% of all particles. Let $\eta^t = 3 - 2C_3^t$. Because C_3^t in (4) decreases approximately from 1.48 to zero, the values of η^t will be increased from 0.04 to 3. Hence, the second task in the adaptive elite-based PSO is initially to prune distant particles and finally cover all particles. Restated, the particles outside the range of $X_m^t \pm \eta^t \sigma^t$ are pruned. The reduced number of particles are substituted by cloning g_{best}^t. That is,

$$X_p^{t+1} = g_{\text{best}}^t + V_p^{t+1}. \tag{6}$$

As shown in Figure 1, X_m^t is a virtual particle which includes the mean of all particles. Consider the range $X_m^t \pm \sigma^t$. Only seven particles are inside this range and three are outside it. Consider X_p^t which is outside the range and is pruned. X_p^t is hence substituted by g_{best}^t, which is denoted by $X_p'^t$.

3.3. Algorithmic Steps. The proposed method can be implemented using the following algorithmic steps.

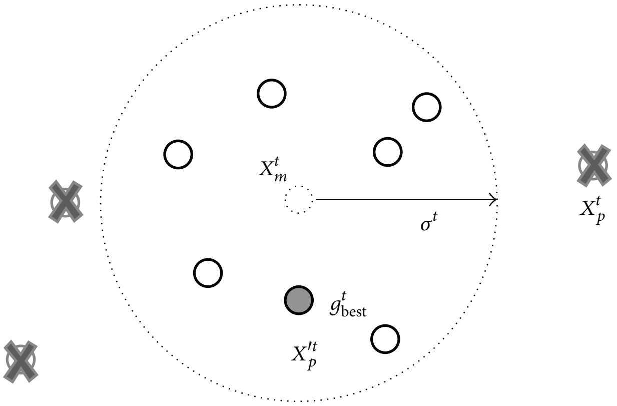


FIGURE 1: Pruning and cloning particles.

Step 1. Input the objective, unknowns, and the number of population size.

Step 2. Give randomly the particles (vectors of unknowns). Let $t = 1$.

Step 3. Calculate all objective values for all particles.

Step 4. If the iterative process meets the convergence criteria, then stop.

Step 5. Find the p_{best}^t and g_{best}^t from all particles according to the objective values of all particles.

Step 6. Calculate C_1^t and C_2^t using (5). Compute C_3^t using (4).

Step 7. Evaluate X_m^t, σ^t, and η^t.

Step 8. If a particle lies outside the range $X_m^t \pm \eta^t \sigma^t$, then it is pruned and $X_p^t = X_p'^t = g_{\text{best}}^t$.

Step 9. Calculate the velocities of all particles using (3).

Step 10. Update all particles using (2).

Step 11. Let $t = t + 1$ and go to Step 3.

3.4. Proof of Convergence of the Proposed Method. Let ω be a real number. Equations (2) and (3) are rewritten in the discrete-time matrix form as follows [19, 20]:

$$Y(t+1) = A(t)\,Y(t) + B(t)\,U(t), \tag{7}$$

where

$$Y(t) = \begin{bmatrix} X_p^t \\ V_p^t \end{bmatrix} = \begin{bmatrix} X_p(t) \\ V_p(t) \end{bmatrix};$$

$$A(t) = \begin{bmatrix} -\left(-1 + C_1^t r_1^t + C_2^t r_2^t + \dfrac{1}{P} C_3^t r_3^t\right) & \omega^t \\ -\left(C_1^t r_1^t + C_2^t r_2^t + \dfrac{1}{P} C_3^t r_3^t\right) & \omega^t \end{bmatrix};$$

$$B(t) = \begin{bmatrix} C_1^t r_1^t & C_2^t r_2^t + C_3^t r_3^t & -\frac{1}{P} C_3^t r_3^t \\ C_1^t r_1^t & C_2^t r_2^t + C_3^t r_3^t & -\frac{1}{P} C_3^t r_3^t \end{bmatrix};$$

$$U(t) = \begin{bmatrix} p_{\text{best}}^t \\ g_{\text{best}}^t \\ \sum_{k=1,k\neq p}^{P} X_k^t \end{bmatrix}. \tag{8}$$

Adding a constant vector $-[\delta_1 \;\; \delta_2]^T$ into both sides of (7), one can yield

$$\begin{aligned} Y(t+1) - [\delta_1 \;\; \delta_2]^T &= A(t) Y(t) + B(t) U(t) - [\delta_1 \;\; \delta_2]^T \\ &= A(t) \left[Y(t) - [\delta_1 \;\; \delta_2]^T \right] \\ &\quad + A(t) [\delta_1 \;\; \delta_2]^T + B(t) U(t) \\ &\quad - [\delta_1 \;\; \delta_2]^T. \end{aligned} \tag{9}$$

Define

$$\begin{aligned} \widehat{Y}(t+1) &\equiv Y(t+1) - \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} = \begin{bmatrix} X_p(t+1) - \delta_1 \\ V_p(t+1) - \delta_2 \end{bmatrix}, \\ \widehat{Y}(t) &\equiv Y(t) - \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} = \begin{bmatrix} X_p(t) - \delta_1 \\ V_p(t) - \delta_2 \end{bmatrix}. \end{aligned} \tag{10}$$

Thus, (9) can be rewritten as follows:

$$\begin{aligned} \widehat{Y}(t+1) \equiv{}& A(t) \widehat{Y}(t) + A(t) [\delta_1 \;\; \delta_2]^T + B(t) U(t) \\ & - [\delta_1 \;\; \delta_2]^T. \end{aligned} \tag{11}$$

If the condition $A(t)[\delta_1 \;\; \delta_2]^T + B(t)U(t) - [\delta_1 \;\; \delta_2]^T = 0$ holds, (11) becomes a linear system as

$$\widehat{Y}(t+1) \equiv A(t) \widehat{Y}(t). \tag{12}$$

The equilibrium point $\widehat{Y}_{\text{eq}}(t) = 0$ must occur at $\widehat{Y}(t+1) = \widehat{Y}(t)$ as $t \to \infty$ for any $A(t)$. According to the condition $A(t)[\delta_1 \;\; \delta_2]^T + B(t)U(t) - [\delta_1 \;\; \delta_2]^T = 0$, one can obtain

$$\begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} = \begin{bmatrix} \dfrac{C_1^t r_1^t p_{\text{best}}^t + (C_2^t r_2^t + C_3^t r_3^t)\, g_{\text{best}}^t - (C_3^t r_3^t / P) \sum_{k=1,k\neq p}^{P} X_k^t}{C_1^t r_1^t + C_2^t r_2^t + (C_3^t r_3^t / P)} \\ 0 \end{bmatrix}. \tag{13}$$

Since $\widehat{Y}(t) \equiv Y(t) - [\delta_1 \;\; \delta_2]^T$ and the equilibrium point $\widehat{Y}_{\text{eq}}(t) \equiv Y_{\text{eq}}(t) - [\delta_1 \;\; \delta_2]^T = 0$, this implies

$$\begin{aligned} Y_{\text{eq}}(t) &= \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} \\ &= \begin{bmatrix} \dfrac{C_1^t r_1^t p_{\text{best}}^t + (C_2^t r_2^t + C_3^t r_3^t)\, g_{\text{best}}^t - (C_3^t r_3^t / P) \sum_{k=1,k\neq p}^{P} X_k^t}{C_1^t r_1^t + C_2^t r_2^t + (C_3^t r_3^t / P)} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} g_{\text{best}}^t \\ 0 \end{bmatrix} \end{aligned} \tag{14}$$

which is true only when $p_{\text{best}}^t = g_{\text{best}}^t = X_k^t$ as $t \to \infty$. Hence, the equilibrium point is

$$Y_{\text{eq}}(t) = \begin{bmatrix} g_{\text{best}}^t \\ 0 \end{bmatrix}. \tag{15}$$

In order to guarantee $Y_{\text{eq}}(t)$ of the discrete-time linear system as shown in (12) to be asymptotically stable, the following necessary and sufficient conditions are obtained using the classical Jury criterion:

$$\begin{gathered} C_1^t r_1^t + C_2^t r_2^t + \frac{C_3^t r_3^t}{P} > 0, \\ 1 + \omega^t - \frac{C_1^t r_1^t + C_2^t r_2^t + (C_3^t r_3^t / P)}{2} > 0 \\ \left| \omega^t \right| < 1. \end{gathered} \tag{16}$$

Since random numbers r_1^t, r_2^t, and r_3^t are between 0 and 1, the above stability conditions are equivalent to the following set of parameter selection heuristics, which guarantee convergence for the proposed adaptive elite-based PSO algorithm:

$$\begin{gathered} 0 < C_1^t + C_2^t + \frac{C_3^t}{P} < 4, \\ \frac{C_1^t + C_2^t + (C_3^t / P)}{2} - 1 < \omega^t < 1. \end{gathered} \tag{17}$$

According to (17), the convergence of the proposed adaptive elite-based PSO can be guaranteed.

4. Simulation Results

The performance of the proposed adaptive elite-based PSO is verified by four benchmark test functions in this section. Traditional PSO, chaotic PSO (CPSO), differential evolution (DE), and genetic algorithm (GA) are used to investigate the benchmark functions for comparative studies [21]. The number of dimensions of each benchmark function is 20 (meaning that the number of unknowns $n = 20$). The number of particles (population size) is also 20 in all methods. The mutation rate and crossover rate in DE are 0.8 and 0.5, respectively; while those in GA are 0.01 and 1.0, respectively. Ten simulations are conducted by each method to verify the optimality and robustness of the algorithms. Two convergence criteria are adopted: convergence tolerance and fixed maximum number of iterations. Five manuscript codes are developed using MATLAB 7.12 to minimize the benchmark functions. The CPU time for studying these four test functions is evaluated using a PC with Intel (R) Core i7 (CPU@3.4 GHz, RAM 4 GB). Finally, the problem of loss minimization in an electric power system is used to validate the proposed method.

4.1. Benchmark Testing: Sphere Function. First, the sphere function is tested as follow:

$$f_1(\hat{x}) = \sum_{i=1}^{n} x_i^2 \qquad -100 < x_i < 100, \tag{18}$$

Table 1: Performance of different methods used to minimize f_1 based on 10 simulation results (convergence tolerance = 0.001).

		Proposed method	PSO	CPSO	DE	GA
Shortest time	f_1	$7.74e-4$	$9.16e-4$	$8.93e-4$	$7.16e-4$	—
	Iterations	203	218	203	294	—
	Time (s)	0.0343	0.0129	0.0146	0.1854	—
Mean	f_1	$8.80e-4$	$9.40e-4$	$9.06e-4$	$8.95e-4$	—
	Iterations	250.1	239.9	241	267.8	—
	Time (s)	0.0391	0.0141	0.0168	0.1935	—
Longest time	f_1	$8.68e-4$	$9.76e-4$	$9.72e-4$	$9.87e-4$	—
	Iterations	302	263	208	247	—
	Time (s)	0.0439	0.0149	0.0245	0.2062	—

Table 2: Performance of different methods used to minimize f_1 based on 10 simulation results (number of iterations = 500).

		Proposed method	PSO	CPSO	DE	GA
Best f_1	f_1	$3.30e-12$	$2.65e-11$	$1.02e-11$	$2.03e-09$	7.1996
	Time (s)	0.0616	0.0232	0.0271	0.6265	1.3834
Mean	f_1	$1.41e-10$	$1.70e-09$	$4.09e-07$	$1.22e-08$	18.4275
	Time (s)	0.0619	0.0240	0.0269	0.6209	1.3143
Worst f_1	f_1	$1.00e-09$	$8.20e-09$	$1.54e-06$	$4.10e-08$	45.4336
	Time (s)	0.0620	0.0234	0.0262	0.6140	1.3328

where $n = 20$ and $\hat{x} = (x_1, x_2, \ldots, x_{20}) \cdot f_1(\hat{x})$ is a unimodal test function whose optimal solution is $f_1(\hat{x}) = 0$ and $x_1 = x_2 = \cdots = x_{20} = 0$. First, the convergence tolerance is set to 0.001. Table 1 shows the shortest, mean and the longest CPU times obtained from 10 simulations by the five methods. As illustrated in Table 1, all methods can approach similar solutions except for GA (which fails to converge). DE requires the longest CPU times. In the conditions of mean and the longest CPU times, the proposed method yields smaller values of $f_1(\hat{x})$ than the other methods. Figure 2 shows the iteration excursions of the different methods with their shortest CPU times.

Table 2 shows the best, mean and the worst $f_1(\hat{x})$ obtained by running 500 iterations using all methods. It can be found that the proposed method always yields the smallest $f_1(\hat{x})$ among the five methods. Figures 3 and 4 present the parameter C_3 and the standard deviation σ obtained using the proposed method in the case with the best $f_1(\hat{x})$, respectively. The values of C_3 decrease quickly in the first 30 iterations and the standard deviation oscillates while decreasing to zero near the 200th iteration.

Generally, the CPU time required by the proposed method is longer than those taken by PSO and CPSO but shorter than those taken by DE and GA.

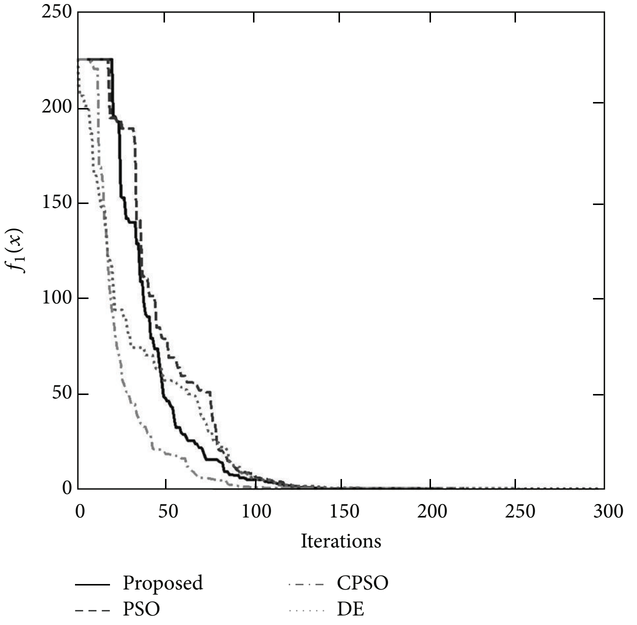


Figure 2: Iteration excursions with the shortest CPU times for different methods used to minimize f_1.

4.2. Benchmark Testing: Rosenbrock Function. This subsection employs the Rosenbrock function for testing as follows:

$$f_2(\hat{x}) = \sum_{i=1}^{n-1} 100 \times \left(x_{i+1} - x_i^2\right)^2 + \left(1 - x_i\right)^2 \qquad -2.048 < x_i < 2.048, \tag{19}$$

where $n = 20$ and $\hat{x} = (x_1, x_2, \ldots, x_{20})$. Each contour of the Rosenbrock function looks roughly parabolic. Its global minimum is located in the valley of the parabola (banana valley). Since the function changes little in the proximity of the global minimum, finding the global minimum is considerably difficult. $f_2(\hat{x})$ is also a unimodal test function whose optimal solution is $f_2(\hat{x}) = 0$ and $x_1 = x_2 = \cdots = x_{20} = 0$. First, the convergence tolerance is set to 0.001. Table 3

Table 3: Performance of different methods used to minimize f_2 based on 10 simulation results (convergence tolerance = 0.001).

		Proposed method	PSO	CPSO	DE	GA
Shortest time	f_2	$9.99e-04$	$9.99e-04$	—	$9.73e-04$	—
	Iterations	6543	6931	—	1167	—
	Time (s)	0.6641	0.3037	—	0.7361	—
Mean	f_2	$9.97e-04$	$9.99e-04$	—	$9.13e-04$	—
	Iterations	7681.3	8532.3	—	1274.7	—
	Time (s)	0.7787	0.4415	—	1.0411	—
Longest time	f_2	$9.99e-04$	$9.98e-04$	—	$8.68e-04$	—
	Iterations	8759	9099	—	1680	—
	Time (s)	0.8879	0.6239	—	1.4807	—

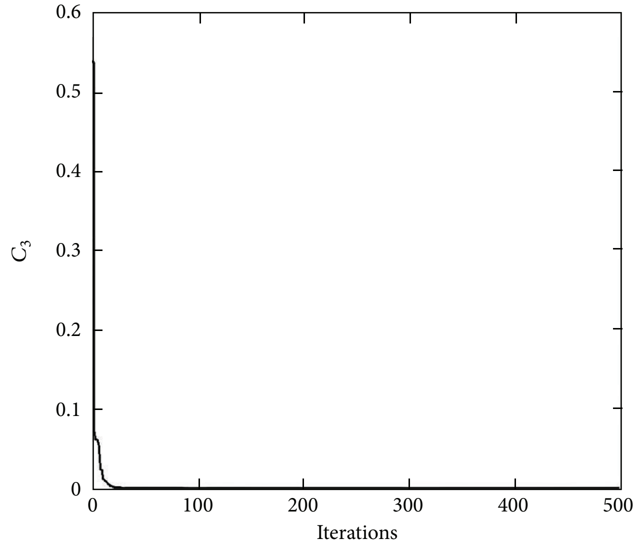


Figure 3: Iterative performance in terms of C_3 (sphere function).

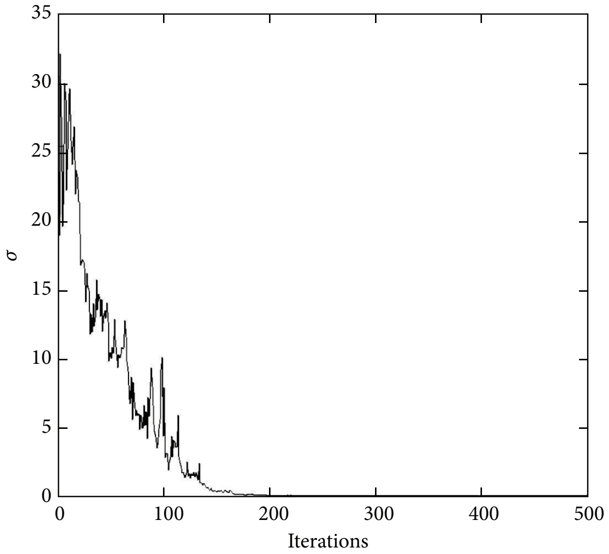


Figure 4: Iterative performance in terms of standard deviation (sphere function).

shows the shortest, mean and the longest CPU times obtained by performing 10 simulations using the five methods. As illustrated in Table 3, all methods yield similar solutions except for CPSO and GA (which fail to converge). The CPU times required by the proposed method are between those taken by PSO and DE. It could be found that DE requires fewer iterations but longer CPU times in all studies. All solutions (optimality), from the viewpoints of the shortest, mean and the longest CPU times, are similar. Figure 5 shows the iteration excursions of the different methods (the cases with their shortest CPU times).

Table 4 shows the best, mean and the worst $f_2(\hat{x})$ obtained by running 1000 iterations using all methods. Since the proposed AEPSO and PSO require 6500~8000 iterations to find the global optimum of $f_2(\hat{x})$, they only reach a near optimal solution in the 1000th iterations. Figures 6 and 7 display the parameter C_3 and standard deviation σ of the proposed method in the case with the best $f_2(\hat{x})$, respectively. The values of C_3 decrease quickly in the first 40 iterations. However, the standard deviation decreases to a very small positive value near the 220th iteration and then oscillates continuously.

4.3. Benchmark Testing: Griewank Function. In this subsection, the Griewank function is tested as follows:

$$f_3(\hat{x}) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \qquad -600 < x_i < 600, \tag{20}$$

where $n = 20$ and $\hat{x} = (x_1, x_2, \ldots, x_{20})$. If $n = 1$, then the function $f_3(x)$ has 191 minima, with the global minimum at $x = 0$ and local minima at $\pm x$ for $x \cong 6.28005$, 12.5601, 18.8401, and 25.1202,.... First, the convergence tolerance is set to 0.01. Table 5 shows the shortest, mean and the longest CPU times achieved using the five methods in 10 simulations. As illustrated in Table 5, CPSO, DE, and GA fail to converge. The optimality of the proposed method and PSO, as given by the shortest, mean and the longest CPU times, is similar. Figure 8 shows the iteration excursions of the methods with the shortest CPU times.

Table 6 shows the best, mean and the worst $f_3(\hat{x})$ obtained at the 1000th iteration by all methods. The proposed method still attains the best optimality compared with other methods.

Table 4: Performance of different methods used to minimize f_2 based on 10 simulation results (number of iterations = 1000).

		Proposed method	PSO	CPSO	DE	GA
Best f_2	f_2	0.0466	0.0258	65.8326	$4.19e-04$	1.9769
	Time (s)	0.1152	0.0488	0.0545	0.6335	2.0670
Mean	f_2	0.0633	0.0717	70.1498	0.0037	6.1595
	Time (s)	0.1162	0.0493	0.0535	0.6406	2.1172
Worst f_2	f_2	0.0816	0.1474	77.7252	0.0072	9.0414
	Time (s)	0.1157	0.0500	0.0543	0.6358	2.1768

Table 5: Performance of different methods used to minimize f_3 based on 10 simulation results (convergence tolerance = 0.01).

		Proposed method	PSO	CPSO	DE	GA
Min (Time)	f_3	0.0082	0.0098	—	—	—
	Generation	850	1416	—	—	—
	Time (s)	0.1349	0.1158	—	—	—
Mean (Time)	f_3	0.0093	0.0095	—	—	—
	Generation	5435	10347	—	—	—
	Time (s)	0.8773	0.8910	—	—	—
Max (Time)	f_3	0.0098	0.0100	—	—	—
	Generation	11553	30425	—	—	—
	Time (s)	1.6046	2.4117	—	—	—

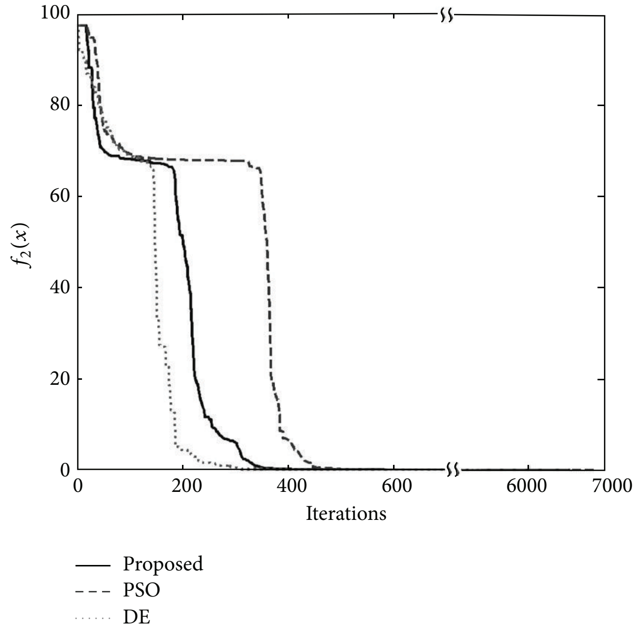


Figure 5: Iteration excursions with the shortest CPU times for different methods used to minimize f_2.

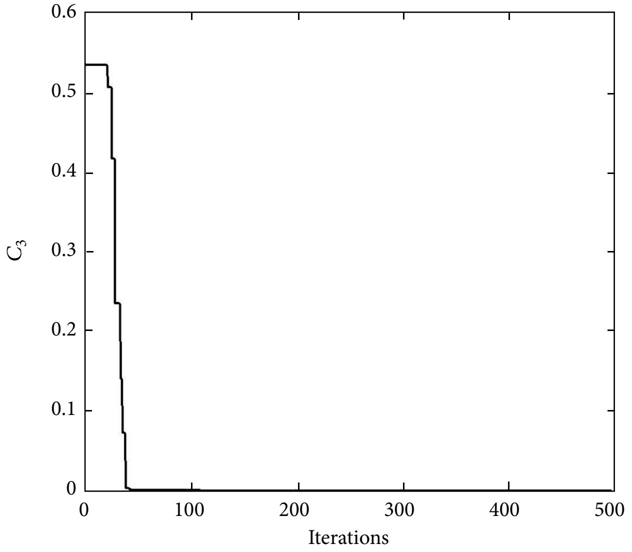


Figure 6: Iterative performance in terms of C_3 (Rosenbrock function).

4.4. Benchmark Testing: Ackley Function. The Ackley function is an n-dimensional highly multimodal function that has a great number of local minima, which look like noise but only one global minimum at $f_4(\hat{x}) = 0$ and $\hat{x} = (x_1, x_2, \ldots) = (0, 0, \ldots)$:

$$f_4(\hat{x}) = -20e^{-0.2\times\sqrt{(1/n)\sum_{i=1}^{n} x_i^2}} - e^{(1/n)\sum_{i=1}^{n}\cos(2\pi\times x_i)} + 20 + e^1 \qquad -32 < x_i < 32, \tag{21}$$

where $n = 20$ and $\hat{x} = (x_1, x_2, \ldots, x_{20})$ herein.

First, the convergence tolerance is set to 0.001. Table 7 shows the shortest, mean and the longest CPU times of 10 simulations obtained by the five methods. As illustrated in Table 7, all methods are able to find the global optimum, except for CPSO (which fails to converge). The CPU times required by the proposed method are between those required by PSO and DE, while GA needs very long CPU times. All solutions (optimality) are similar from the viewpoints of the shortest, mean and the longest CPU times. Figure 11 shows the iteration excursions of the various methods with the shortest CPU times. The proposed method converges the fastest.

Table 8 shows the best, mean and the worst $f_4(\hat{x})$ obtained at the 500th iteration by all methods. The proposed method and DE can gain the global optimum but PSO, CPSO,

Table 6: Performance of different methods used to minimize f_3 based on 10 simulation results (number of iterations = 1000).

		Proposed method	PSO	CPSO	DE	GA
Best f_3	f_3	0.0004	0.0195	$3.53e-02$	0.6151	0.3880
	Time (s)	0.1941	0.0820	0.1221	0.8560	2.7651
Mean	f_3	0.0129	0.0779	0.1293	0.6953	0.7267
	Time (s)	0.1927	0.1031	0.1157	0.8649	2.7097
Worst f_3	f_3	0.0472	0.1687	$1.87e-01$	0.8114	0.9426
	Time (s)	0.1922	0.1033	0.1144	0.8706	2.5987

Table 7: Performance of different methods used to minimize f_4 based on 10 simulation results (convergence tolerance = 0.001).

		Proposed method	PSO	CPSO	DE	GA
Shortest time	f_4	0.000926	0.000865	—	0.000963	0.000732
	Iterations	329	410	—	467	5604
	Time (s)	0.0698	0.0354	—	0.4140	313.2481
Mean	f_4	0.000956	0.000946	—	0.000944	0.000915
	Iterations	380.9	424.5	—	490.2	6222.1
	Time (s)	0.0775	0.0368	—	0.4371	322.7549
Longest time	f_4	0.000928	0.000959	—	0.000933	0.000972
	Iterations	398	431	—	503	7987
	Time (s)	0.0875	0.0397	—	0.4616	333.9159

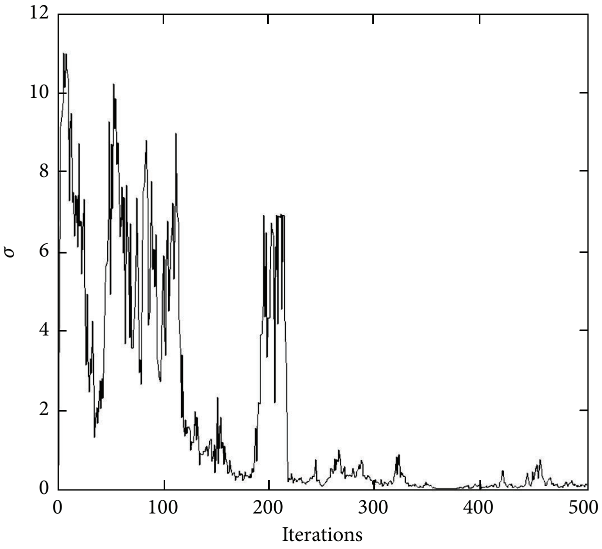


Figure 7: Iterative performance in terms of standard deviation (Rosenbrock function).

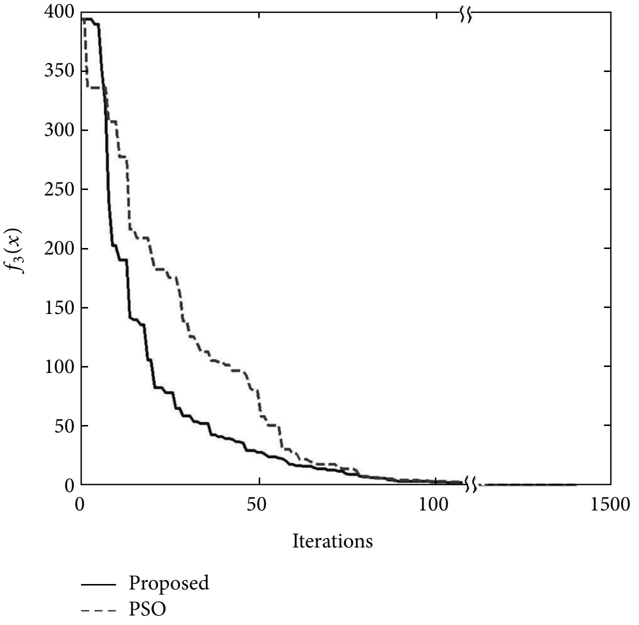


Figure 8: Iteration excursions with the shortest CPU times for different methods used to minimize f_3.

and GA cannot. The proposed method always gains the smallest values of $f_4(\hat{x})$ by inspecting the best, mean and the worst values of the $f_4(\hat{x})$. The proposed method also requires the shortest CPU times to converge to solutions. Figures 12 and 13 illustrate parameter C_3 and standard deviation σ, respectively, obtained using the proposed method in the case with the best $f_4(\hat{x})$. The values of C_3 decrease to small positive values close to the 220th iteration. The standard deviation decreases to very small positive values near the 400th iteration and thereafter oscillates (see Figures 9 and 10).

4.5. Studies on Loss Minimization in a Power System. Active power loss in the electric power system is caused by the resistance in the transmission/distribution lines. The losses can be evaluated as $\sum$ (each line current)^2 × (each line resistance) or simplified as (total active power generations − total active power demands). If the active power demands are constant and voltage magnitudes are increased, then the line currents will be reduced, leading to reduction of active power losses.

Table 8: Performance of different methods used to minimize f_4 based on 10 simulation results (number of iterations = 500).

		Proposed method	PSO	CPSO	DE	GA
Best f_4	f_4	$8.29e-06$	$1.93e-04$	1.423543	$4.46e-04$	0.780793
	Time (s)	0.0917	0.0394	0.0472	0.4336	1.8306
Mean	f_4	$7.83e-05$	0.000463	2.379854	0.000836	1.517645
	Time (s)	0.0921	0.0405	0.0475	0.4346	1.7197
Worst f_4	f_4	$3.35e-04$	$2.58e-03$	$4.10e+00$	$8.12e-04$	2.621082
	Time (s)	0.0919	0.0406	0.0475	0.4347	1.6464

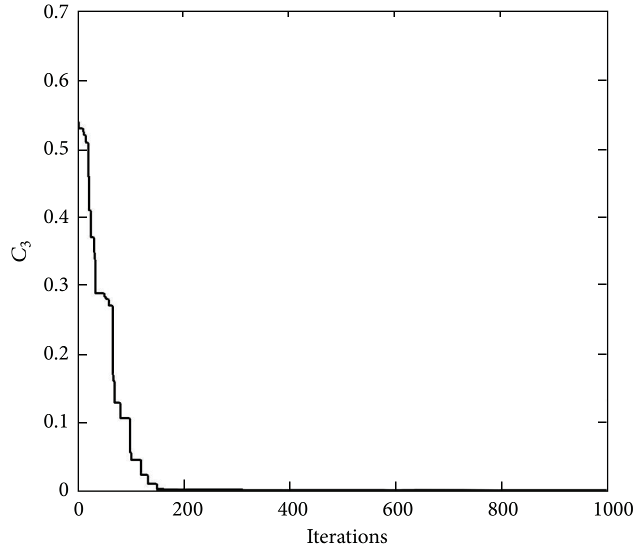


Figure 9: Iterative performance in terms of C_3 (Griewank function).

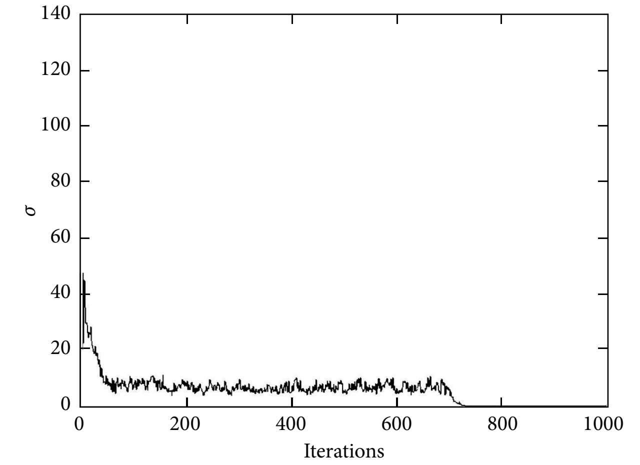


Figure 10: Iterative performance in terms of standard deviation (Griewank function).

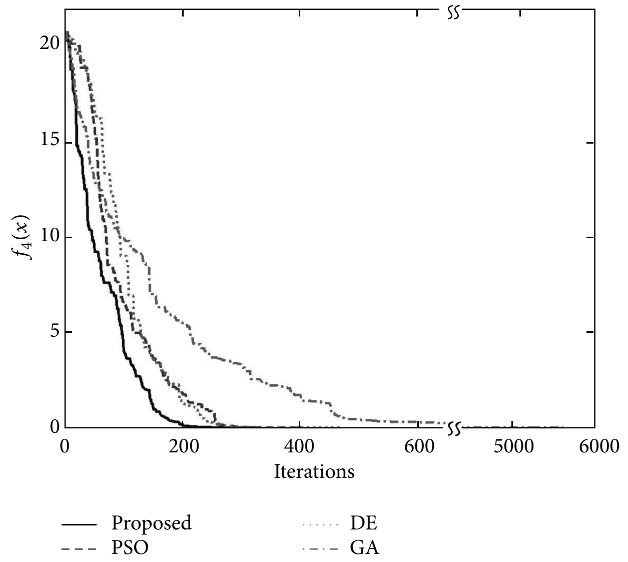


Figure 11: Iteration excursions with the shortest CPU times for different methods used to minimize f_4.

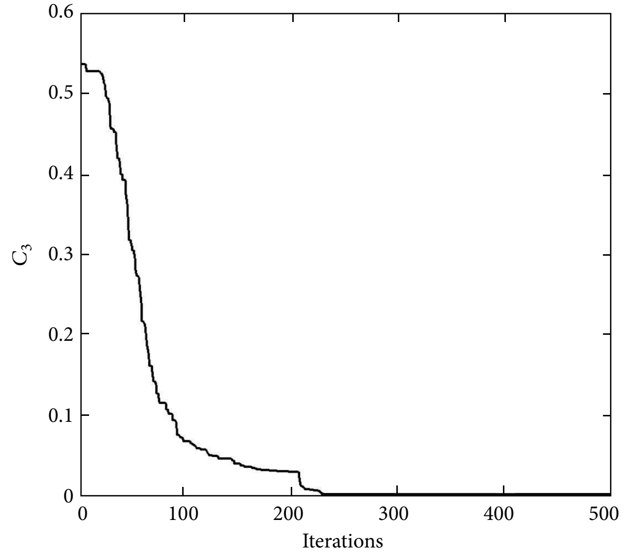


Figure 12: Iterative performance in terms of C_3 (Ackley function).

Therefore, engineers who work on electric power systems tend to utilize voltage controllers to increase the voltage profile in order to reduce the active power losses [22]. The voltage controllers include generator voltages, transformer taps, and shunt capacitors. The problem formulation of active power loss minimization is given in the appendix.

Figure 14 illustrates a 25-busbar power system with 3 wind farms at busbars 13, 14, and 25. The wind power generations in these three busbars are 2.4, 2.4, and 5.1 MW, respectively. The diesel generators are located at busbars 1~11. The shunt capacitors are at busbars 21, 22, and 23. Seven

Table 9: Optimality and CPU time required by different methods (10 runs).

		AEPSO	PSO	CPSO	DE	GA
Best objective	MW loss	1.783	1.865	1.829	1.857	1.862
	Time (s)	33.9461	43.4231	48.3487	42.0286	70.3614
Mean	MW loss	1.840	1.875	1.8654	1.915	1.886
	Time (s)	34.2277	46.8953	48.3953	44.9488	72.0624
Worst objective	MW loss	1.886	1.952	1.902	1.961	1.946
	Time (s)	34.6663	48.4029	48.7938	47.7389	75.1891

Table 10: Optimal solutions obtained by different methods (best solution).

Control variables	Proposed	PSO	CPSO	ED	GA
V_1–V_3	1.01	1.02	0.96	0.95	1.04
V_4–V_{11}	1.04	1.04	1.05	1.04	1.04
V_{13}	1.03	1.04	1.03	1.03	1.00
V_{14}	0.99	0.99	0.99	0.99	1.01
V_{25}	1.04	1.04	1.04	1.04	1.00
Tap_{1-12}	0.95	0.96	0.95	0.95	0.97
Tap_{4-12}	1.05	1.05	1.03	1.05	0.99
Tap_{12-17}	0.95	1.01	0.98	1.01	1.04
Tap_{12-18}	0.98	0.98	0.98	0.98	0.99
Tap_{12-19}	1.02	1.03	1.02	1.03	1.02
Tap_{17-15}	1.02	1.03	1.02	1.02	0.96
Tap_{14-16}	1.02	1.02	1.02	1.02	0.98
$\text{Tap}_{20\text{-}21}$	1.01	0.96	1.01	1.01	1.01
Tap_{20-22}	1.04	1.04	0.96	1.04	1.01
Tap_{20-23}	1.02	1.04	0.98	1.01	1.03
SC_{21}	0.05	0.05	0.1	0.1	0.09
SC_{22}	0.1	0.1	0.1	0.1	0.06
SC_{23}	0.15	0.15	0.1	0.1	0.09

demands can be seen at busbars 17~19 and 21~24. Consequently, the number of control variables is 39 (14 generator voltage magnitudes, 22 taps, and 3 shunt capaciotrs). The number of state variables is 36 (voltage magnitudes and phase angles at busbars 12, 15~19, and 20~24, phase angles and reactive power generations at generator busbars, and active power generation at reference busbar 1).

The proposed method, PSO, CPSO, DE, and GA are applied to the find the optimal 39 control variables in order to minimize the active power (MW) losses. Ten simulations, each with fixed 500 iterations are conducted using each method. Figure 15 illustrates the iteration performance of these five methods. It can be found the iteration process of the proposed method converges slowly at first but quickly after the 220th iteration. Table 9 reveals that the proposed method is able to converge the fastest and requires the shortest CPU times for the conditions of the best active power loss, mean values, and the worst active power loss in the 10 simulations. As shown in Table 9, GA still needs much more CPU time to solve the realistic problem and DE yields the worst, mean values over 10 simulation runs. Table 10 gives the results of optimal controls obtained using the different methods (best solution). Figure 16 displays the optimal voltage profile, which is obtained by the proposed method, of the electric power system.

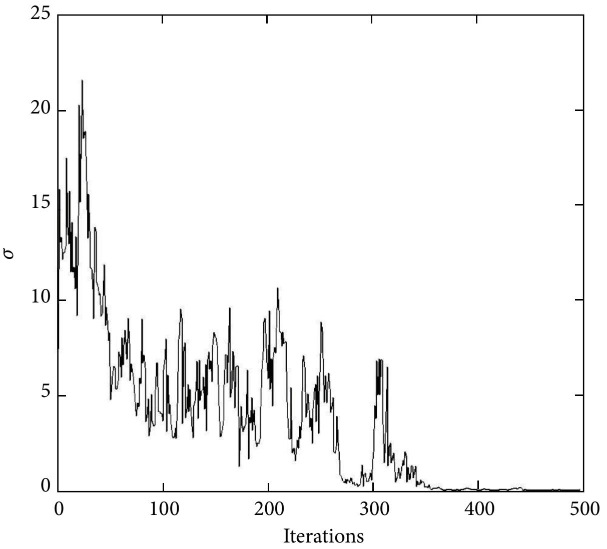


Figure 13: Iterative performance in terms of standard deviation (Ackley function).

4.6. Comparison of Time, Space Complexities, and Performance. Compared with the traditional PSO, the proposed novel mean search and particle pruning/cloning manners have been added to achieve stable iterative process and they avoid ineffective searches, respectively. Although the space complexity is increased a little bit due to these two tasks as shown in (3)–(6), the number of iterations can be mitigated

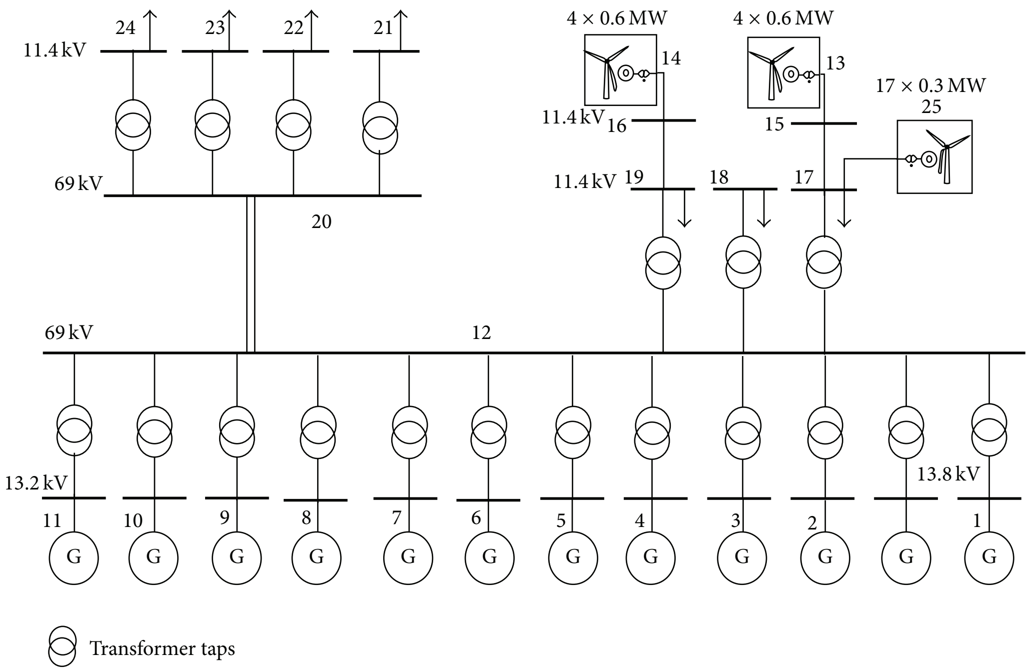


Figure 14: Oneline diagram of 25-busbar system.

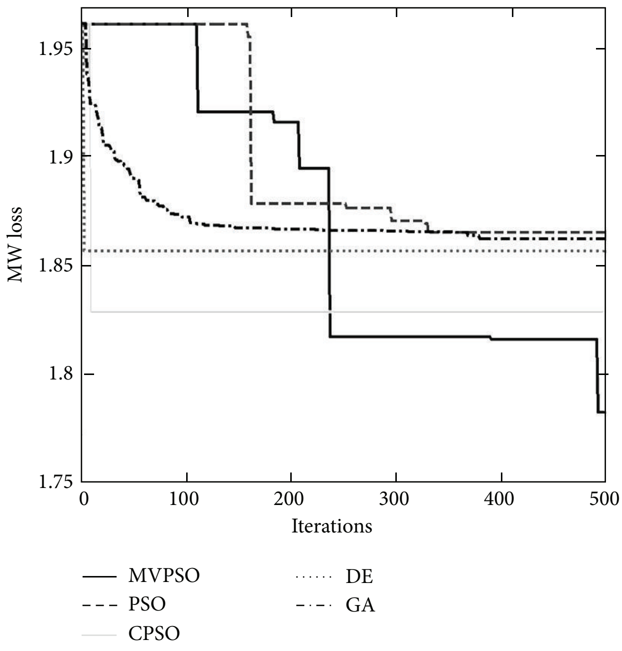


Figure 15: Iteration performance of five methods.

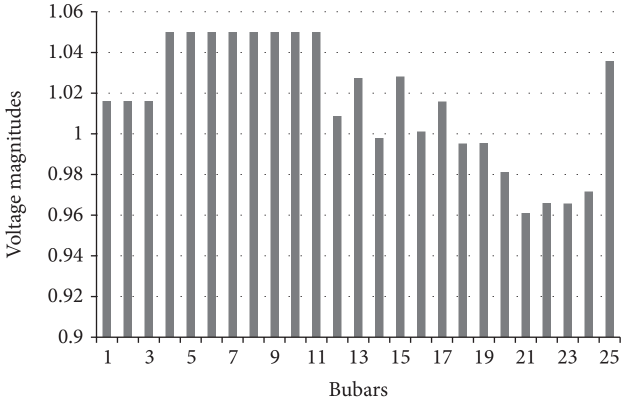


Figure 16: Voltage magnitudes in 25-busbar system (the proposed method).

and the computational burden can be reduced effectively. Thus, the time complexity is similar to that of the traditional PSO.

The performance comparisons among the proposed method, PSO, CPSO, DE, and GA are implemented using a set of four benchmark functions and one realistic loss minimization problem in an electric power system. The simulation results, as shown in Tables 1–10, imply that the CPSO and DE sometimes are unable to find the optimal solution or even diverge. Furthermore, both GA and DE always require long CPU times for computation. Additionally, the computational effort required by the proposed method to obtain high quality solutions is less than the effort required to obtain the same high quality solutions by the GA and DE. Therefore, the proposed method has the most stable convergence performance to find the optimums and has the best computational efficiency compared with the other methods considering the time complexity and space complexity.

5. Conclusions

In this paper, to provide a better tradeoff between global and local search in the iterative process than existing methods, a novel elite-based PSO approach is proposed for studying highly dimensional and extraordinarily nonlinear optimization problems. Two tasks have been developed: mean search and particle pruning/cloning. The mean of all particles and the standard deviation of the distances between pairs of particles are evaluated in each iteration. The mean search ensures that the iterative process is stable. Particle pruning/cloning avoids ineffective searches. All parameters are adaptively self-tuned during the iterations. The impacts of these two tasks are gradually mitigated in the iteration process. When the proposed process is near convergence, its computational burden is similar to that of the traditional PSO.

Based on the studies herein of benchmark functions, the CPSO and DE sometimes are unable to find the optimal solution or even diverge. GA is often unable to find the optimum if the benchmark functions are very nonlinear. In terms of optimality, the proposed method is generally better than PSO. In the loss minimization problem, the best, mean and the worst objective values obtained by the proposed method in 10 simulations are always the best among those obtained by all methods. The smallest mean loss value implies that the numerical process of proposed method is the more stable than all of the others. The proposed method also requires the least CPU time to solve the loss minimization problem.

Appendix

A.

The active power loss minimization problem can be formulated as follows. Let the number of transmission lines, the set of generator busbars, and the set of total busbars be T, K, and I, respectively. The objective is to minimize the line losses in all lines:

$$\min \sum_{t=1}^{T} P_{\text{loss}}^{t}. \tag{A.1}$$

Let busbar 1 be the reference with a phase-angle of zero. The loss can be expressed as

$$\begin{aligned} &V_1 \sum_{j=1}^{I} V_j \left(G_{1j}\cos\theta_{1j} + B_{1j}\sin\theta_{1j}\right) \\ &\quad + \left(\sum_{k\in K, k\neq 1} PG_k\right) - \left(\sum_{i\in I} PD_i\right), \end{aligned} \tag{A.2}$$

where P_{Gk} and P_{Di} are the active generation and demand at the kth generator and busbar i, respectively. The terms V_i and V_j represent the voltages of busbars i and j, respectively. $G_{ij} + jB_{ij}$ is the element at location (i, j) of the system admittance matrix. The term θ_{ij} represents the voltage phase-angle difference between busbars i and j.

The objective is subject to the following equality and inequality constraints.

A.1. Equality Constraints. The equality constraints covering the power flow equations are shown in (A.3) as follows:

$$\begin{aligned} P_{Gi} - P_{Di} - P_i &= 0, \quad i = 1, \ldots, I \\ Q_{Gi} - Q_{Ci} - Q_{Di} - Q_i &= 0, \quad i = 1, \ldots, I, \end{aligned} \tag{A.3}$$

where P_{Gi} and Q_{Gi} are the active and reactive power generation at busbar i, respectively. P_{Di} and Q_{Di} are the active and reactive power demands at busbar i, respectively. P_i and Q_i are the active and reactive power flow injections at busbar i, respectively. Q_{Ci} is the injected reactive power at busbar i where a capacitor is installed. P_i and Q_i can be represented as follows:

$$\begin{aligned} P_i &= V_i \sum_{j=1}^{I} V_j \left(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}\right) \\ Q_i &= V_i \sum_{j=1}^{I} V_j \left(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}\right). \end{aligned} \tag{A.4}$$

A.2. Inequality Constraints. In the following, the subscripts "max" and "min" denote the upper and lower limits.

Operational Limits of Voltage Magnitudes at Generator Busbars. Consider

$$V_{Gk,\min} \leq V_{Gk}^{t} \leq V_{Gk,\max}, \quad k \in K, \tag{A.5}$$

where V_{Gk} represents the voltage magnitude at generator busbar k. The symbol K denotes the set of generator busbars.

Limitations on Reactive Power at Generators. Consider

$$Q_{Gk,\min} \leq Q_{Gk}^{t} \leq Q_{Gk,\max}, \quad k \in K, \tag{A.6}$$

where Q_{Gk} represents the reactive power output at generator k.

Operational Limits at Transformers. Consider

$$T_{l,\min} \leq T_{l}^{t} \leq T_{l,\max}, \quad l \in L, \tag{A.7}$$

where T_l represents the tap at transformer l. The symbol L represents the set of transformers.

Operational Limits of Capacitors. Consider

$$Q_{Cm,\min} \leq Q_{Cm} \leq Q_{Cm,\max}, \quad m \in M, \tag{A.8}$$

where Q_{Cm} represents the injected reactive power at busbar m. The symbol M represents the set of capacitors.

Operational Limits of Voltage Magnitudes at Demand Busbars. Consider

$$V_{Dn,\min} \leq V_{Dn} \leq V_{Dn,\max}, \quad n \in N, \tag{A.9}$$

where V_{Dn} represents the voltage magnitude at busbar n. The symbol N indicates the set of demands.

The active/reactive power demands, active power generation, and line impedances $(R_{ij} + jX_{ij})$ of the 25-busbar system are given in Tables 11(a), 11(b), and 11(c), respectively. Please note that $G_{ij} + jB_{ij} = (R_{ij} + jX_{ij})^{-1}$.

Table 11: (a) Active and reactive demands. (b) Power generation (MW). (c) Line impedances of 25-busbar system.

(a)

Busbar	Active demand (MW)	Reactive demand (MVAR)
17	13.58	2.78
18	15.85	3.31
19	13.76	2.79
21	23.24	4.69
22	18.63	3.83
23	18.81	3.84
24	11.94	2.43

(b)

Busbar	Active power
2	10
3	10
4	11
5	11
6	11
7	11
8	11
9	11
10	11
11	11
13	1.55
14	1.59
25	1.52

(c)

Line impedance	R_{ij}	X_{ij}
L_{1-12}	0.0470	0.980
L_{2-12}	0.0470	0.980
L_{3-12}	0.0470	0.9840
L_{4-12}	0.0585	0.9820
L_{5-12}	0.0585	0.9820
L_{6-12}	0.0585	0.9800
L_{7-12}	0.0585	0.9750
L_{8-12}	0.0585	0.9820
L_{9-12}	0.0585	0.9870
L_{10-12}	0.0585	0.9900
$L_{11\text{-}12}$	0.0585	0.9820
L_{12-17}	$8e-05$	0.3548
L_{12-18}	$8e-05$	0.3544
L_{12-19}	$8e-05$	0.3544
L_{12-20}	0.0239	0.0514
L_{13-15}	0	0.4000
L_{14-16}	0	0.4000
L_{15-17}	0.9540	1.4070
L_{16-19}	0.9540	1.4070
L_{20-21}	$8e-05$	0.3440
L_{20-22}	$8e-05$	0.3444
L_{20-23}	$8e-05$	0.3444
L_{20-24}	$8e-05$	0.3444
L_{25-17}	0	0.4000

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank Professor C. R. Chen at NTUT (Taiwan) for his comments about this work. The authors are also grateful for financial support from the National Science Council (Taiwan) under Grant NSC 102-3113-P-008-001.

References

[1] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS '95)*, pp. 39–43, Nagoya, Japan, October 1995.

[2] C. Wang, Y. C. Liu, and Y. T. Zhao, "Application of dynamic neighborhood small population particles warm optimization for reconfiguration of shipboard power system," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, pp. 1255–1262, 2013.

[3] M. Neyestani, M. M. Farsangi, and H. Nezamabadi-Pour, "A modified particle swarm optimization for economic dispatch with non-smooth cost functions," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 7, pp. 1121–1126, 2010.

[4] B. Vasumathi and S. Moorthi, "Implementation of hybrid ANNPSO algorithm on FPGA for harmonic estimation," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 476–483, 2012.

[5] W. X. Liu, I.-Y. Chung, L. Liu, S. Y. Leng, and D. A. Cartes, "Real-time particle swarm optimization based current harmonic cancellation," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 132–141, 2011.

[6] R. J. Wai, Y. C. Huang, Y. C. Chen, and Y. W. Lin, "Performance comparisons of intelligent load forecasting structures and its application to energy-saving load regulation," *Soft Computing*, vol. 17, no. 10, pp. 1797–1815, 2013.

[7] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[8] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft Computing*, vol. 11, no. 4, pp. 3658–3670, 2011.

[9] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.

[10] R. Caponetto, L. Fortuna, S. Fazzino, and M. G. Xibilia, "Chaotic sequences to improve the performance of evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 289–304, 2003.

[11] G. N. José and A. Enrique, "Restart particle swarm optimization with velocity modulation: a scalability test," *Soft Computing*, vol. 15, no. 11, pp. 2221–2232, 2011.

[12] W. Deng, R. Chen, B. He, Y. Liu, L. Yin, and J. Guo, "A novel two-stage hybrid swarm intelligence optimization algorithm and application," *Soft Computing*, vol. 16, no. 10, pp. 1707–1722, 2012.

[13] Y. Y. Li, R. R. Xiang, L. C. Jiao, and R. C. Liu, "An improved cooperative quantum-behaved particle swarm optimization," *Soft Computing*, vol. 16, no. 6, pp. 1061–1069, 2012.

[14] A. M. Arasomwan and A. O. Adewumi, "An investigation into the performance of particle swarm optimization with various chaotic maps," *Mathematical Problems in Engineering*, vol. 2014, Article ID 178959, 17 pages, 2014.

[15] H. Yin, C. Zhang, B. Zhang, Y. Guo, and T. Liu, "A hybrid multiobjective discrete particle swarm optimization algorithm

for a SLA-aware service composition problem," *Mathematical Problems in Engineering*, vol. 2014, Article ID 252934, 14 pages, 2014.

[16] B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D.-X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons & Fractals*, vol. 25, no. 5, pp. 1261–1271, 2005.

[17] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[18] J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, San Francisco, Calif, USA, 2001.

[19] S. Bouallègue, J. Haggège, M. Ayadi, and M. Benrejeb, "PID-type fuzzy logic controller tuning based on particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 484–493, 2012.

[20] N. J. Li, W. J. Wang, C. C. J. Hsu, W. Chang, H. G. Chou, and J. W. Changa, "Enhanced particle swarm optimizer incorporating a weighted particle," *Neurocomputing*, vol. 124, pp. 218–227, 2014.

[21] G. A. Ortiz, "Evolution Strategies (ES)," Mathwork, 2012.

[22] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation and Control*, John Wiley & Son, New York, NY, USA, 2nd edition, 1996.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 604705, 8 pages
http://dx.doi.org/10.1155/2014/604705

Research Article

Receding Horizon Trajectory Optimization with Terminal Impact Specifications

Limin Zhang,[1,2] Mingwei Sun,[1] Zengqiang Chen,[1] Zenghui Wang,[3] and Yongkun Wang[1]

[1] *Tianjin Key Laboratory of Intelligent Robotics, College of Computer & Control Engineering, Nankai University, Tianjin 300071, China*
[2] *School of Mechanical Engineering, Liaoning Technical University, Fuxin, Liaoning 123000, China*
[3] *Department of Electrical and Mining Engineering, University of South Africa, Florida 1710, South Africa*

Correspondence should be addressed to Mingwei Sun; sunmw@nankai.edu.cn

Received 12 March 2014; Accepted 29 April 2014; Published 14 May 2014

Academic Editor: Jer-Guang Hsieh

The trajectory optimization problem subject to terminal impact time and angle specifications can be reformulated as a nonlinear programming problem using the Gauss pseudospectral method. The cost function of the trajectory optimization problem is modified to reduce the terminal control energy. A receding horizon optimization strategy is implemented to reject the errors caused by the motion of a surface target. Several simulations were performed to validate the proposed method via the C programming language. The simulation results demonstrate the effectiveness of the proposed algorithm and that the real-time requirement can be easily achieved if the C programming language is used to realize it.

1. Introduction

To enhance the penetration probability of a cruise missile, controlling both the impact angle and the impact time is of special significance in practice. In the past three decades, several guidance laws with terminal impact angle specifications were proposed within different frameworks [1–11]. It should be noted that the impact time is a crucial factor in the salvo attack against a highly valued surface target equipped with advanced air defense installations such as close-in weapon systems (CIWS). However the impact time specification is difficult to achieve and there were few reports in this field. Jeon et al. [7] proposed a 2-dimensional salvo attack strategy with terminal impact angle and time specifications, which was obtained by a linearized model based optimal control. An analytical guidance law was derived explicitly for this strategy. There are two control loops in this approach: the outer and inner loops are designed to regulate the time-to-go and the angle, respectively. When the time-to-go prediction is larger than the specified time, a trajectory curve will be planned online to consume the excessive time; otherwise, a comparatively straight line will be generated to meet the impact time requirement. Despite its simplicity, two severe pitfalls accompany this method: (1) the small heading angle assumption could cause large terminal errors when the practical angle is large and (2) the process constraints with the inequality forms, for example, the familiar no-fly zone constraints, are unable to be incorporated. To cope with the no-fly zone constraints, many trajectory planning methodologies based on A* algorithm [12–15] and intelligent optimization algorithms, such as genetic algorithm (GA) [16–18], ant colony optimization (ACO) [19, 20], and particle swarm optimization (PSO) [21–23], have been extensively investigated. Nevertheless, these approaches can only generate an open-loop optimal trajectory offline due to the computational complexity and it is not applicable to use them to impact a moving surface target. Therefore, it is urgent to solve the trajectory optimization problem with terminal impact angle and time specifications as well as process inequality constraints simultaneously in real-time.

Recently, the Gauss pseudospectral method (GPM) has attracted a wide attention within aerospace industry, especially in the fields of guidance and trajectory design [24–28]. GPM is a direct optimal control solving approach for the general nonlinear systems with various constraints. Nowadays, a mature GPM solving software GPOPS (Gauss Pseudospectral OPtimization Software) is available, which makes GPM a technology to be more than an art. In this paper, the GPM

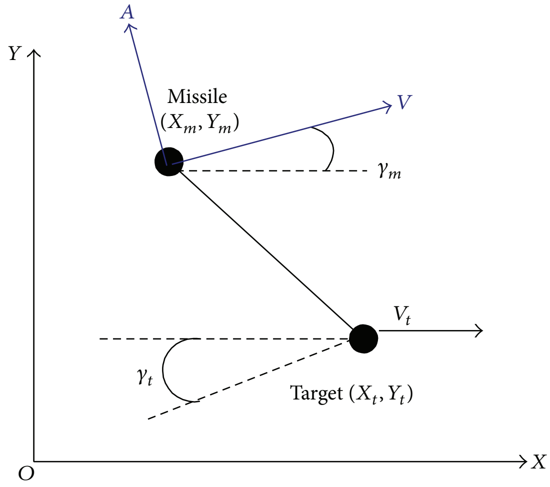


FIGURE 1: Missile-target relative kinematics.

is employed in a receding horizon way to obtain an adaptive trajectory to impact a moving surface target, which satisfies the terminal and process constraints and can be realized online. In the design, a time-to-go dependent penalty function is utilized to ensure a smooth trajectory at the terminal end and reduce the miss distance.

The remaining parts of this paper are organized as follows. The problem is formulated in Section 2. Section 3 gives a brief introduction about GPM. The receding horizon strategy to implement GPM is presented in Section 4. In Section 5, the simulation results are provided and analyzed. The concluding remarks are summarized in Section 6.

2. Problem Formulation

2.1. Kinematics Equations. The missile-target relative kinematics is shown in Figure 1. Without loss of generality, it is assumed that the velocity of the missile, V, is a constant. The earth coordinate system is chosen as XOY. The current states of the missile are the position (X_m, Y_m) and the heading angle γ_m. The control variable for the missile is the normal acceleration, A, being perpendicular to V, which can change the velocity direction. The corresponding states of the target are the position (X_t, Y_t) and the heading angle γ_t.

According to Figure 1, the mass-point dynamics of the missile and the target are [9, 29]

$$\begin{aligned} \frac{dX_m}{dt} &= V\cos(\gamma_m), \\ \frac{dY_m}{dt} &= V\sin(\gamma_m), \\ \frac{d\gamma_m}{dt} &= u_m = \frac{A}{V}, \\ \frac{dX_t}{dt} &= V_t\cos(\gamma_t), \\ \frac{dY_t}{dt} &= V_t\sin(\gamma_t). \end{aligned} \tag{1}$$

2.2. Optimal Trajectory Requirement

2.2.1. Terminal Constraints. The terminal impact conditions are

$$X_m(t_f) = X_t, \qquad Y_m(t_f) = Y_t, \qquad \gamma_m = \gamma_f, \tag{2}$$

where γ_f and t_f are the specified impact angle and time, respectively. When multiple missiles are synchronized by their onboard clocks separately, a salvo attack can be achieved automatically without any communication. Therefore, there is less risk when encountering sophisticated electronic jamming and deception environment.

2.2.2. Control Constraint. The acceleration is bounded as

$$|A| \le A_{\max}, \quad A_{\max} > 0, \tag{3}$$

which are determined by the airframe and the engine operating conditions of specific missiles.

2.2.3. No-Fly Zone Constraints. The cruise missile often flies over a wide range of area. There exist several types of no-fly zones, such as dangerous terrains and adverse air defense areas, which should be evaded completely. Without loss of generality, these no-fly zones can be represented by disks as

$$(X_m(t) - x_i)^2 + (Y_m(t) - y_i)^2 \le R_i^2, \quad i = 1, 2, \ldots n, \tag{4}$$

where (x_i, y_i) and R_i are the center and the radius, respectively, and n is the number of no-fly zones.

2.2.4. Cost Function. The accumulative control energy during the guidance course

$$J(t_0) = \frac{1}{2}\min\int_{t_0}^{t_f} u_m^2 dt \tag{5}$$

is often employed as the cost function to produce a more or less straight trajectory, where t_0 is the initial time instant.

For most guidance laws, the maximal control energy often occurs at the terminal end, which could cause large miss distance. To solve this problem, a time-weight function is introduced in the cost objective as

$$J(t_0) = \frac{1}{2}\min\int_{t_0}^{t_f} \left(f_p(t)\cdot u_m^2\right) dt. \tag{6}$$

As the impact time is fixed, $f_p(t)$ can be designed as a monotonically increasing function according to the terminal requirements.

3. A Brief Introduction to GPM

Consider a dynamic equation

$$\dot{x} = f(x(t), u(t), t), \tag{7}$$

where $x(t) \in R^n$ and $u(t) \in R^m$ are state and control variables, respectively, and the function $f\colon R^n \times R^m \times R \rightarrow R^n$.

The traditional optimal control problem can be formulated in a unified form as

$$J = \Phi\left(x\left(t_f\right), t_f\right) + \int_{t_0}^{t_f} g\left(x\left(t\right), u\left(t\right), t\right) dt, \tag{8}$$

subject to the boundary equality constraints

$$\phi\left(x\left(t_0\right), t_0, x\left(t_f\right), t_f\right) = 0, \tag{9}$$

and the process inequality constraints

$$C\left(x\left(t\right), u\left(t\right), t_0, t_f\right) \leq 0, \tag{10}$$

where $\phi : R^n \times R \times R^n \times R \rightarrow R^q$ and $C : R^n \rightarrow R^r$.

A brief review of pseudospectral method (PSM), especially Gauss PM (GPM), is presented here. The detailed description of GPM can be found in [24].

Firstly, the original time interval $[t_0, t_f]$ is transformed to the interval $[-1, 1]$ as

$$\tau = \frac{2t}{t_f - t_0} - \frac{t_f + t_0}{t_f - t_0}. \tag{11}$$

Using (11), (8) can be reformulated in terms of τ as

$$J = \Phi\left(x\left(1\right), t_f\right) + \frac{t_f - t_0}{2} \int_{-1}^{1} g\left(x\left(\tau\right), u\left(\tau\right), \tau\right) d\tau. \tag{12}$$

Similarly, the dynamic (7) and the boundary constraints (9) are replaced by

$$\frac{2}{t_f - t_0} \frac{dx}{d\tau} = f\left(x\left(\tau\right), u\left(\tau\right), \tau\right), \tag{13}$$

$$\phi\left(x\left(-1\right), t_0, x\left(1\right), t_f\right) = 0, \tag{14}$$

respectively.

In the GPM, the Legendre-Gauss (LG) points, τ_k $(1 \leq k \leq N - 1)$, distributed on the interval $[-1, 1]$, are defined as the roots of

$$P_N\left(\tau\right) = \frac{1}{2^N N!} \frac{d^N}{d\tau^N} \left[\left(\tau^2 - 1\right)^N\right], \tag{15}$$

while $\tau_0 = -1$, $\tau_N = 1$. The continuous state and control variables are approximated by the Nth polynomials as

$$x\left(\tau\right) \approx X\left(\tau\right) = \sum_{i=0}^{N} L_i\left(\tau\right) X\left(\tau_i\right), \tag{16}$$

$$u\left(\tau\right) \approx U\left(\tau\right) = \sum_{i=1}^{N} L_i\left(\tau\right) U\left(\tau_i\right), \tag{17}$$

where the Lagrange interpolation polynomial is

$$L_i\left(\tau\right) = \prod_{j=0, j \neq i}^{N} \frac{\tau - \tau_i}{\tau_i - \tau_j}. \tag{18}$$

Secondly, (16) is differentiated at the node points as [28]

$$\dot{x}\left(\tau\right) \approx \dot{X}\left(\tau\right) = \sum_{i=0}^{N} \dot{L}_i\left(\tau\right) X\left(\tau_i\right), \tag{19}$$

where

$$D_{ki} = \dot{L}_i\left(\tau_k\right) = \begin{cases} \dfrac{\left(1 + \tau_k\right) \dot{P}_K\left(\tau_k\right) + P_K\left(\tau_k\right)}{\left(\tau_k - \tau_i\right)\left[\left(1 + \tau_i\right) \dot{P}_K\left(\tau_i\right) + P_K\left(\tau_i\right)\right]}, & i \neq k, \\ \dfrac{\left(1 + \tau_i\right) \ddot{P}_K\left(\tau_i\right) + 2\dot{P}_K\left(\tau_i\right)}{2\left[\left(1 + \tau_i\right) \dot{P}_K\left(\tau_i\right) + P_K\left(\tau_i\right)\right]}, & i = k. \end{cases} \tag{20}$$

The static form of (7) can be obtained by equating the right side of both (19) and (7) at the discrete node points.

Thirdly, the continuous-time cost function (8) is approximated by using Gauss quadrature formula as

$$J = \Phi\left(X_0, t_0, X_f, t_f\right) + \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k g\left(X_k, U_k, \tau_k; t_0, t_f\right), \tag{21}$$

where w_k are the Gauss weights.

Similarly, the terminal and process constraints can also be reformulated.

According to the procedures described above, the original optimal control problem can be approximated as a static nonlinear programing (NLP) one. The essence of GPM is to replace the original infinite-dimension dynamic optimal control problem with a finite-dimension static NLP by eliminating differential and integral equations. There are many effective methods to solve NLP and among them the sequential quadratic programing (SQP) is a famous one which has been widely used because of its maturity. Nowadays, a kind of reliable software, SNOPT (Sparse Nonlinear OPTimizer), is available for solving SQP problems in a unified framework.

4. Receding Horizon GPM

The GPM provides us with an effective way to deal with nonlinear optimal control problem subject to various constraints directly, avoiding the shortcomings existing in the linearization approximation approach [7]. Moreover, a free GPOPS is available to seek GPM solution in real-time and reduce the huge computational complexity existing in the traditional trajectory planning [14, 19, 21]. However, the optimal solution obtained by using GPOPS is an open-loop one which is unable to cope with moving surface targets such as warships. Therefore, a closed-loop optimal solution should be generated instead. Enlightened by the mechanism of model predictive control [30], this obstacle can be eliminated by using receding horizon scheme which can be realized online.

The receding horizon GPM can be implemented as follows.

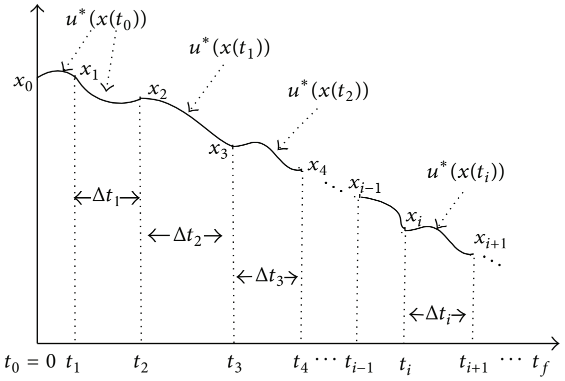


Figure 2: Receding horizon optimization schematic diagram.

Step 1. Input the number of LG points, N, the specified impact time t_f, and terminal heading angle γ_f. Set t_0 as the initial time instant and $i = 1$.

Step 2. According to the guidance measurement information, plan an optimal trajectory from the current position to the target and obtain a control law $u^*(t_i)$ using GPM. Assume that this computational period is ΔT_i. During the time interval $[t_i, t_i + \Delta T_i]$, or when the new optimization results are not available, the corresponding values of $u^*(t_{i-1})$ on this interval are implemented. And then $u^*(t_i)$ is used after the time instant of $t_i + \Delta T_i$.

Step 3. Let $i = i + 1$ and repeat Step 2 until $t = t_f$.

This method can be illustrated in Figure 2.

5. Numerical Simulations

In this section, the effectiveness of the proposed method is illustrated by several numerical simulations. The first simulation is to illustrate the effectiveness of the time-weight function $f_p(t)$, wherein a stationary target is employed without using receding horizon optimization. The second and the third simulations combined are designed to test the capacity of the proposed method to deal with no-fly zone constraints. Here a moving target is used to evaluate the receding horizon optimization strategy. In all the simulations, the velocities of missiles keep a constant of 250 m/s and the initial numbers of LG points, N, are universally selected as 20.

5.1. Selection of Time-Weight Function $f_p(t)$. For most guidance laws, the control energy almost reaches its peak at the terminal end due to the divergent line-of-sight (LOS) rate, which would lead to a large miss distance. Therefore, it would be preferred to ensure the terminal trajectory to be straight for a period of time. In order to find a better time-weight function, five types of time-dependent functions $f_p(t)$ are investigated and they are

(1) $f_p(t) = 1$;

(2) $f_p(t) = t^2$;

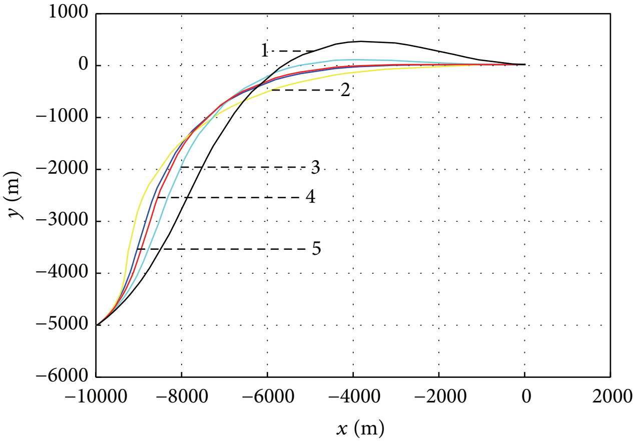


Figure 3: Flight trajectories.

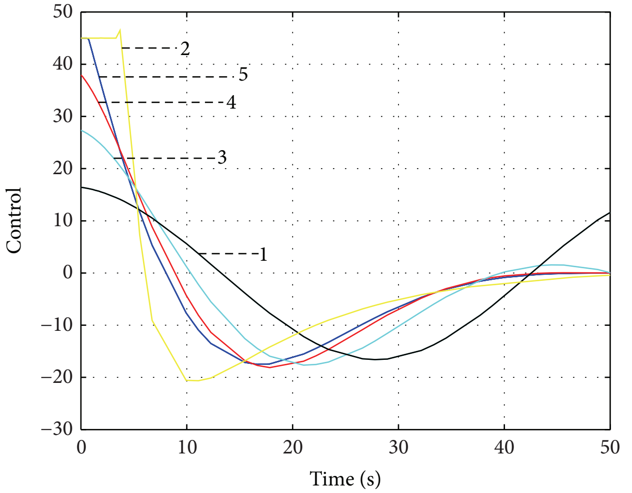


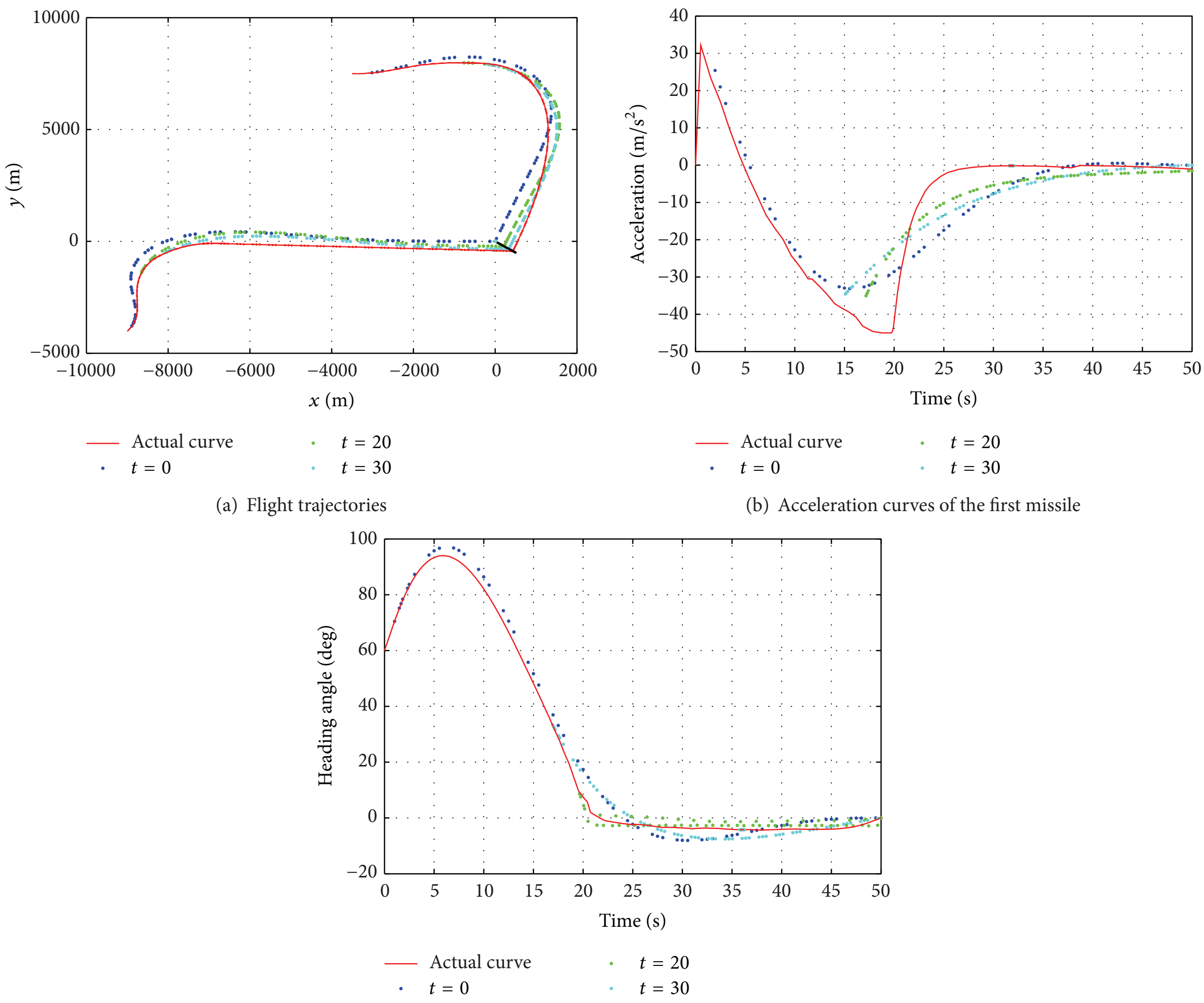
Figure 4: Acceleration curves.

(3) $f_p(t) = 1/(t_f - t + 0.5)$;

(4) $f_p(t) = 1/(t_f - t + 0.5)^2$;

(5) $f_p(t) = 1/(t_f - t + 0.5)^3$.

Because of the terminal impact time specification, these functions can be easily realized. Assume a scenario that the missile impacts a stationary target subject to $t_f = 50$ s and $\gamma_f = 0^\circ$. The simulation results are shown in Figures 3 and 4 when using a one-phase GPM. According to Figure 3, it can be seen that all the terminal constraints are not violated for each function. From Figure 4, it is clear that when $f_p(t) = 1$, the terminal control amplitude is the maximal one among all; when $f_p(t) = t^2$, the initial acceleration is large, and the control curve is similar to a bang-bang control. However, this function is free of t_f, which could be sensitive to the case configuration; when $f_p(t) = 1/(t_f - t + 0.5)^2$, the terminal flight trajectory is straight. But with the increment of power in the denominator, both the initial acceleration and the total energy consumption are increased. Therefore, a comprehensive tradeoff is made and $f_p(t) = 1/(t_f - t + 0.5)^2$ is chosen and fixed for the following simulations.



(a) Flight trajectories

(b) Acceleration curves of the first missile

(c) Heading angle curves of the second missile

Figure 5: Salvo attack without inequality constraints.

Table 1: Case configurations without inequality constraints.

Missile	(X_0, Y_0) (km)	γ_0 (deg)	γ_f (deg)	t_f (s)
1	(−3.5, 8.5)	0	−105	50
2	(−9, −4.5)	45	0	50

Table 2: Simulation results without inequality constraints.

Missile	ΔX (m)	ΔY (m)	$\Delta\gamma$ (deg)
1	$8.8e-08$	$4.1e-8$	0
2	$3.7e-08$	$-8.5e-8$	0

5.2. Without Process Inequality Constraint. In this scenario, two antiship missiles attack a moving warship without any no-fly zone constraint. The fixed sampling period is $\Delta T = 0.5$ s. The velocity of the warship is 15 m/s with $\gamma_t = -135^\circ$. The maximal feasible acceleration is $A_{\max} = 45\,\text{m/s}^2$. The case configurations are given in Table 1. The simulation results are shown in Figure 5 and Table 2. In Figure 5(a), the real lines are the actual trajectories and the point lines are the planned optimal trajectories at different sampling time instants. It can be found that both missiles fly along circular arcs during their early stages in order to consume the excessive time. According to Figures 5(b) and 5(c), the terminal trajectories are becoming straight gradually towards specified impact direction prior to impact instant; therefore tiny accelerations are needed in the final phase. In Table 2, $(\Delta X, \Delta Y)$ and $\Delta\gamma$ are the terminal errors of position and impact angle, respectively. It can be found that they are negligible from Table 2. This fact suggests that the proposed method can effectively impact a moving surface target subject to terminal constraints. It should also be noted that it is not necessary to pay a lot of attention to the selection of the initial values. In other words, this algorithm is insensitive to the quality of initial solution. The setup of the computer is i5 processor and 2G memory. The GPOPS is running on Matlab platform. When the algorithm is implemented in C code, the convergent time can be reduced by a factor of

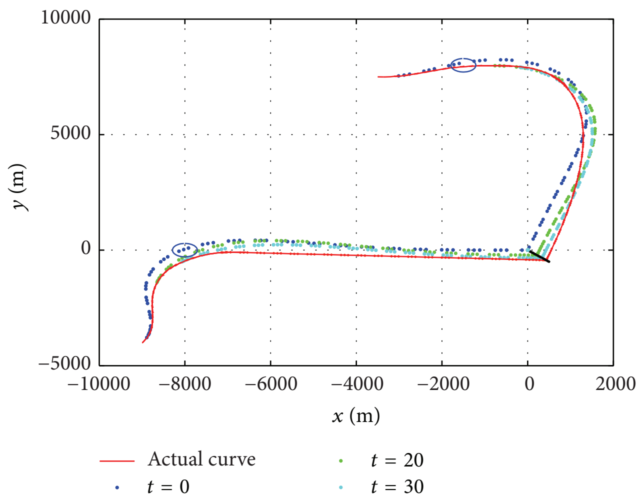


Figure 6: Location of the no-fly zones.

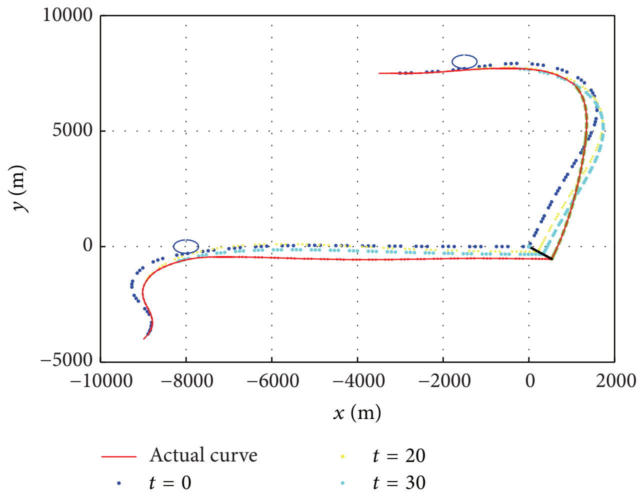


Figure 7: Salvo attack with no-fly zones.

up to 100. For this case, each trajectory can be obtained within 10 s by using Matlab; therefore it meets real-time requirement when using C programming language onboard. In fact, we rewrote the corresponding GPOPS functions used in the abovementioned path planning by the C programming language on the basis of the commercial SNOPT software for practitioners. When implemented on the practical DOS platform, the running time can be kept below 0.3 s for almost all situations.

5.3. With No-Fly Zone Constraints. In this case, we assume that there are two no-fly zones along the trajectories of the previous simulation and they are

(1) $(X_m + 1300)^2 + (Y_m - 8000)^2 \leq 122500$;

(2) $(X_m + 8000)^2 + (Y_m - 0)^2 \leq 122500$.

They are shown as two disks in Figure 6, which can be considered as the process constraints. It is obvious that the no-fly zones are just located along the current trajectories;

Table 3: Case configurations for comparison.

Missile	(X_0, Y_0) (km)	γ_0 (deg)	γ_f (deg)	t_f (s)
1	(−3, −3)	105	0	50
2	(−3, −3)	105	−25	50
3	(−3, −3)	105	−25	50

Table 4: Comparative precisions.

Missile	ΔX (m)	ΔY (m)	$\Delta\gamma$ (deg)
1	$3.7e - 08$	$-8.5e - 8$	−0.0002
2	−503.85	533.83	0.0699
3	$1.2e - 09$	$-2.1e - 9$	0

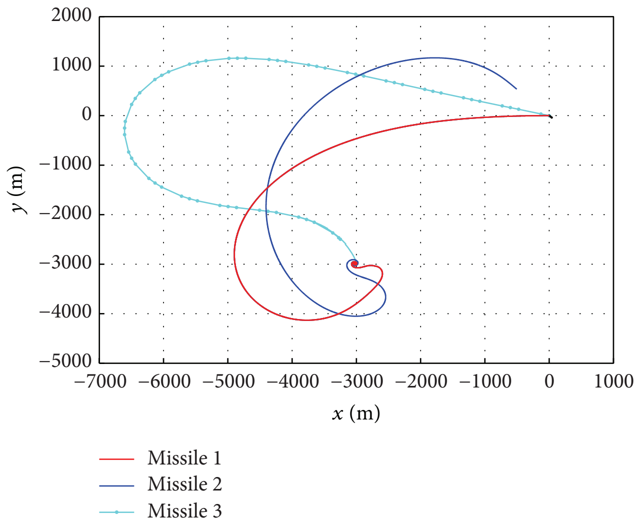


Figure 8: Comparative simulation results.

hence the trajectories need to be replanned to evade the no-fly zones. The new trajectories are shown in Figure 7 when adding the no-fly zone constraints. According to Figure 7, the two missiles evade the no-fly zones successfully in the simulations. On the other hand, the running time is almost unchanged compared with the one in the above section; that is, the process inequality constraints have little effect on the optimization time for GPM.

5.4. Comparison with the Analytical Guidance Law. In [7], an analytical guidance law with impact angle and time specifications was derived based on the linearization approximation using small heading angle assumption. In this section, the comparative simulation is performed. The case configurations are shown in Table 3. The first and second missiles are guided by the method in [7] while the trajectory of the third missile is obtained by using the proposed receding horizon GPM. The simulation results are shown in Figure 8 and Table 4. It is evident that poor miss distance is caused for the second missile due to the linearization approximation errors. However, the third missile can still guarantee impact accuracy.

6. Conclusions

This paper presented a trajectory optimization approach based on the receding horizon GPM to impact a moving surface target subject to the terminal impact angle and the time specifications. A time-dependent weighting function was introduced to reduce the control energy at the terminal end. The receding horizon strategy could be applied to the target in motion. When using GPOPS, this comprehensive scheme can handle the process inequality constraints without causing the increment of computational complexity and no linearization needs to be done. The optimization algorithm could be implemented real-time with an onboard computer. The effectiveness of the proposed method was illustrated by several numerical examples.

Conflict of Interests

The authors declare that they have no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to express their thanks to the anonymous reviewers of the paper. This work was supported in part by the Natural Science Foundation of China (nos. 61174094 and 61273138), the Tianjin Natural Science Foundation of China (nos. 14JCYBJC18700 and 13JCYBJC17400), South African National Research Foundation (no. 78673), and South African National Research Foundation Incentive Grant (no. 81705).

References

[1] M. Kim and K. V. Grider, "Terminal guidance for impact attitude angle constrained flight trajectories," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 9, no. 6, pp. 852–859, 1973.

[2] M. Idan, O. M. Golan, and M. Guelman, "Optimal planar interception with terminal constraints," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 6, pp. 1273–1279, 1995.

[3] B. S. Kim, J. G. Lee, and H. S. Han, "Biased PNG law for impact with angular constraint," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 277–288, 1998.

[4] C. K. Ryoo, H. J. Cho, and M. J. Tahk, "Closed-form solutions of optimal guidance with terminal impact angle constraint," in *Proceedings of the IEEE Conference on Control Applications*, vol. 1, pp. 504–509, June 2003.

[5] C. K. Ryoo, H. J. Cho, and M. J. Tahk, "Optimal guidance laws with terminal impact angle constraint," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, pp. 724–732, 2005.

[6] B. Y. Jung and Y. D. Kim, "Guidance laws for anti-ship missiles using impact angle and impact time," in *Proceeding of the AIAA Guidance, Navigation, and Control Conference*, pp. 3048–3060, August 2006.

[7] I. S. Jeon, J. I. Lee, and M. J. Tahk, "Impact-time-control guidance law for anti-ship missiles," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 2, pp. 260–266, 2006.

[8] M. G. Yoon, "Relative circular navigation guidance for the impact angle control problem," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 4, pp. 1449–1463, 2008.

[9] P. Wu and M. Yang, "Integrated guidance and control design for missile with terminal impact angle constraint based on sliding mode control," *Journal of Systems Engineering and Electronics*, vol. 21, no. 4, pp. 623–628, 2010.

[10] Y. Kim, J. Kim, and M. Park, "Guidance and control system design for impact angle control of guided bombs," in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '10)*, pp. 2138–2143, October 2010.

[11] H. B. Oza and R. Padhi, "Impact-angle-constrained suboptimal model predictive static programming guidance of air-to-ground missiles," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, pp. 153–164, 2012.

[12] R. J. Szczerba, P. Galkowski, I. S. Glickstein, and N. Ternullo, "Robust algorithm for real-time route planning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 3, pp. 869–878, 2000.

[13] B. M. Sathyaraj, L. C. Jain, A. Finn, and S. Drake, "Multiple UAVs path planning algorithms: a comparative study," *Fuzzy Optimization and Decision Making*, vol. 7, no. 3, pp. 257–267, 2008.

[14] D. Bodhale, N. Afzulpurkar, and N. T. Thanh, "Path planning for a mobile robot in a dynamic environment," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '09)*, pp. 2115–2120, February 2009.

[15] B. B. Meng and X. G. Gao, "UAV path planning based on bidirectional sparse A* search algorithm," in *Proceedings of the International Conference on Intelligent Computation Technology and Automation (ICICTA '10)*, vol. 3, pp. 1106–1109, Changsha, China, May 2010.

[16] Y. V. Pehlivanoglu, O. Baysal, and A. Hacioglu, "Path planning for autonomous UAV via vibrational genetic algorithm," *Aircraft Engineering and Aerospace Technology*, vol. 79, no. 4, pp. 352–359, 2007.

[17] K. J. Obermeyer, "Path planning for a UAV performing reconnaissance of static ground targets in terrain," in *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, AIAA-2009-5888, August 2009.

[18] M. Kapanoglu, M. Alikalfa, M. Ozkan et al., "A pattern-based genetic algorithm for multi-robot coverage path planning minimizing completion time," *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1035–1045, 2012.

[19] A. Fridman, S. Weber, V. Kumary et al., "Distributed path planning for connectivity under uncertainty by ant colony optimization," in *Proceedings of the American Control Conference (ACC '08)*, pp. 1952–1958, Seattle, Wash, USA, June 2008.

[20] M. A. P. Garcia, O. Montiel, O. Castillo et al., "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Applied Soft Computing*, vol. 9, no. 3, pp. 1102–1110, 2009.

[21] J. L. Foo, J. S. Knutzon, J. H. Oliver, and E. H. Winer, "Three-dimensional multi-objective path planning of unmanned aerial vehicles using particle swarm optimization," in *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pp. 2170–2179, April 2007.

[22] J. L. Foo, J. Knutzon, V. Kalivarapu, J. Oliver, and E. Winer, "Path planning of unmanned aerial vehicles using B-splines and particle swarm optimization," *Journal of Aerospace Computing, Information and Communication*, vol. 6, no. 4, pp. 271–290, 2009.

[23] M. P. Ran, H. B. Duan, X. G. Gao, and Z. Mao, "Improved particle swarm optimization approach to path planning of amphibious mouse robot," in *Proceedings of the 6th IEEE Conference on*

Industrial Electronics and Applications (ICIEA '11), pp. 1146–1149, Beijing, China, June 2011.

[24] D. Benson, *A Gauss pseudospectral transcription for optimal control [Ph.D. dissertation]*, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2004.

[25] I. M. Ross and F. Fahroo, "Pseudospectral knotting methods for solving optimal control problems," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 3, pp. 397–405, 2004.

[26] G. T. Huntington and A. V. Rao, "Optimal configuration of spacecraft formations via a gauss pseudospectral method," in *Proceedings of the AAS/AIAA 15th Space Flight Mechanics Meeting*, pp. 33–50, January 2005.

[27] G. T. Huntington and A. V. Rao, "Optimal reconfiguration of spacecraft formations using the gauss pseudospectral method," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 3, pp. 689–698, 2008.

[28] R. Tawfiqur, H. Zhou, Y. Z. Sheng, Y. M. Younis, and K. N. Zhang, "Trajectory optimization of hypersonic vehicle using Gauss pseudospectral method," in *Proceedings of the 2nd International Conference on Mechanical and Aerospace Engineering*, pp. 5232–5239, 2011.

[29] Y. H. Kim, C. K. Ryoo, and M. J. Tahk, "Guidance synthesis for evasive maneuver of anti-ship missiles against close-in weapon systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 3, pp. 1376–1388, 2010.

[30] E. N. Hartley, P. A. Trodden, and A. G. Richards, "Model predictive control system design and implementation for spacecraft rendezvous," *Control Engineering Practice*, vol. 20, no. 7, pp. 695–713, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 291870, 6 pages
http://dx.doi.org/10.1155/2014/291870

Research Article

The Lattice-Valued Turing Machines and the Lattice-Valued Type 0 Grammars

Juan Tang,[1] Yong Fang,[1] and Jian-Gang Tang[2]

[1] *College of Electronics and Information Engineering, Sichuan University, Chengdu, Sichuan 610064, China*
[2] *College of Mathematics and Statistics, Yili Normal University, Yining, Xinjiang 835000, China*

Correspondence should be addressed to Yong Fang; yfang2013@qq.com

Received 25 December 2013; Accepted 10 April 2014; Published 8 May 2014

Academic Editor: Yang Xu

Purpose. The purpose of this paper is to study a class of the natural languages called the lattice-valued phrase structure languages, which can be generated by the lattice-valued type 0 grammars and recognized by the lattice-valued Turing machines. *Design/Methodology/Approach.* From the characteristic of natural language, this paper puts forward a new concept of the l-valued Turing machine. It can be used to characterize recognition, natural language processing, and dynamic characteristics. *Findings.* The mechanisms of both the generation of grammars for the lattice-valued type 0 grammar and the dynamic transformation of the lattice-valued Turing machines were given. *Originality/Value.* This paper gives a new approach to study a class of natural languages by using lattice-valued logic theory.

1. Introduction

According to Chomsky's rationalist theory, the language can be divided into types by the grammar, and then, types can be recognized and processed, respectively. The fuzziness is the typical trait of the natural languages, and then, how to recognize and deal with the fuzzy natural language is becoming the more and more important subject. From the point of the reorganization for the natural language, in 1967, Wee [1] firstly introduced the fuzzy finite automata, which generated many interesting explorations. Its important applications in learning systems, automatic control, pattern recognition and database, and so on were also studied by many researchers; the more details and the fuzzy finite automata were studied by Wee and Fu [2], Santos [3–6], Lee and Zadeh [7], Kumbhojkar and Chaudhari [8, 9], Malik et al. [10–15], and so on; the authors can also refer to [16, 17].

In this paper, our main purpose is to consider a class of the natural languages which is called the lattice-valued phrase structure languages. In fact, these natural languages can be generated by the lattice-valued type 0 grammars and recognized by the lattice-valued Turing machines. Moreover, the natural languages can be described by the formation mechanism and the transferred mechanism of the grammar.

This paper is organized as follows. Some basic concepts such as the complete Heyting algebra, the lattice-valued language, and the corresponding membership were recalled in Section 2. In Section 3, we mainly studied the lattice-valued Turing machine and the lattice-valued recursively enumerable language. Especially, the mechanisms of both the generation of grammars for the lattice-valued type 0 grammar and the dynamic transformation of the lattice-valued Turing machines were given; we used the lattice-valued logic theory to study the class of the natural languages; it seems to be that the approach which we used is new as far as we know. Finally, we study the equivalence between the lattice-valued type 0 grammar and the lattice-valued Turing machine in Section 4.

2. Preliminaries

Throughout this paper, a nonempty finite set Σ is called an alphabet. The element in the alphabet is called a symbol or a letter. And, Σ^* denotes the free monoid, that is, the set of all strings with letters from Σ and the empty string, where the empty string is denoted by ϵ; moreover, the length of a string x is denoted by $|x|$, which is the number of symbols in the string; for the empty string ε, we set $|\varepsilon| = 0$.

Given an alphabet Σ. The concatenation of two strings x and y is the string which was denoted by xy and was obtained by appending the symbols of y to the right end of x. Assume x is a string, and then x^n stands for the string obtained by repeating the string x by n times. As a special case, we can define $x^0 = \varepsilon$.

Let Σ_1, Σ_2 be two alphabets; the product of Σ_1 and Σ_2 is given by

$$\Sigma_1\Sigma_2 = \{ab \mid a \in \Sigma_1, b \in \Sigma_2\}. \tag{1}$$

Define n power of an alphabet Σ by

$$\begin{gathered}\Sigma^0 = \epsilon,\\ \Sigma^n = \Sigma^{n-1}\Sigma, \quad n \geq 1.\end{gathered} \tag{2}$$

The *positive closure* of Σ is

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \Sigma^4 \ldots, \tag{3}$$

and the *star-closure* of Σ is

$$\Sigma^* = \Sigma^0 \cup \Sigma^+ = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \Sigma^4 \ldots. \tag{4}$$

Suppose L is a complete lattice; the least element and the greatest element are 0 and 1, respectively, and also satisfy the infinite distributivity law. That is, $\forall \alpha \in L$, $\beta_i \in L$, $i \in I$, we have

$$\begin{gathered}\alpha \vee (\wedge_{i\in I}\beta_i) = \wedge_{i\in I}(\alpha \vee \beta_i),\\ \alpha \wedge (\vee_{i\in I}\beta_i) = \vee_{i\in I}(\alpha \vee \beta_i).\end{gathered} \tag{5}$$

Then L is called a complete Heyting algebra.

Let L be a complete Heyting algebra and Σ an alphabet. We call a map $\varphi : \Sigma^* \to L$ a lattice-value language. $\forall x \in \Sigma^*$, $\varphi(x)$ denotes the membership degree of x belonging to the lattice-valued language.

Let φ, φ_1, and φ_2 be the lattice-valued languages on Σ. Then

(1) $\varphi_1 \vee \varphi_2$ (resp., $\varphi_1 \wedge \varphi_2$) is a lattice-valued language on Σ via

$$\begin{gathered}(\varphi_1 \vee \varphi_2)(x) = \varphi_1(x) \vee \varphi_2(x)\\ (\text{resp.}, (\varphi_1 \wedge \varphi_2)(x) = \varphi_1(x) \wedge \varphi_2(x)).\end{gathered} \tag{6}$$

Generally, if $\varphi_1, \varphi_2, \ldots, \varphi_n$ are the lattice-valued languages on Σ, then

$$\begin{gathered}\vee_{i=1}^n \varphi_i = (\varphi_1 \vee \varphi_2 \vee \cdots \varphi_{n-1}) \vee \varphi_n\\ (\text{resp. } \wedge_{i=1}^n \varphi_i = (\varphi_1 \vee \varphi_2 \vee \cdots \varphi_{n-1}) \ \vee \varphi_n).\end{gathered} \tag{7}$$

(2) $\varphi_1 \circ \varphi_2$ is a lattice-valued language on Σ which is defined by

$$\varphi_1 \circ \varphi_2(x) = \vee\{\varphi_1(y) \wedge \varphi_2(z) \mid x = yz\}. \tag{8}$$

(3) Let $\theta \in L$; then $\theta\varphi$ is a lattice-valued language on Σ which is defined by

$$(\theta\varphi)(x) = \theta \wedge \varphi(x). \tag{9}$$

(4) φ^* is a lattice-valued language on Σ which is defined by

$$\varphi^*(x) = \begin{cases}\varphi(\epsilon) & \text{if } x = \epsilon\\ \vee\{\varphi(b_1) \wedge \varphi(b_2) \wedge \cdots \wedge \varphi(b_n) & \\ \quad \mid x = b_1 b_2 \cdots b_n, & \\ \quad b_i \neq \epsilon, \forall i = 1, 2, \ldots, n, n = 1, 2, \ldots\} & \text{if } x \neq \epsilon.\end{cases} \tag{10}$$

Let X be a nonempty set and L a complete Heyting algebra. The map

$$R : X \times X \longrightarrow L \tag{11}$$

is called the binary lattice-valued relation on X. If $R(x, y) = \theta$, we denote it by $xR_\theta y$, which can be understood as the membership degree θ of x and y satisfying the relation R.

Let R, S be binary lattice-valued relations on X; the composition operation of R and S is defined by

$$(R \circ S)(x, z) = \vee_{y\in X}(R(x, y) \wedge S(y, z)). \tag{12}$$

Consider the following:

(a)

$$R^0(x) = \begin{cases}1 & x = y\\ 0 & x \neq y;\end{cases} \tag{13}$$

(b) $R \subseteq S \Leftrightarrow R(x, y) \leq S(x, y), \forall (x, y) \in X \times X$;

(c) $(R \cup S)(x, y) = R(x, y) \vee S(x, y), (R \cap S)(x, y) = R(x, y) \wedge S(x, y)$;

(d) $R^n = R^{n-1} \circ R$, $R^+ = \cup_{n=1}^{\infty} R^n$, and $R^* = \cup_{n=0}^{\infty} R^n$ are called n power, the positive closure, and the star-closure of R, respectively.

3. The Lattice-Valued Turing Machine and the Lattice-Valued Recursively Enumerable Language

A lattice-valued Turing machine $\mathscr{M}$ is a 7-tuple

$$\mathscr{M} = (Q, \Gamma, \Sigma, \delta, q_0, B, F), \tag{14}$$

where

Q is a finite set of states;

Γ is a finite set of tape symbol and B is a special symbol of Γ called the blank;

Σ is a subset of Γ not containing B and is called the input symbols set;

δ is a transition function, that is, a map from $Q\times\Gamma$ to $Q\times\Gamma\times\{-1,+1\}\times L$;

$\delta(q,X) = (p,Y,+1,\theta)$ means that $\mathscr{M}$ reads a symbol X in the state q, the next state is p, and the read-write head moves right one unit after the symbol Y in place of X on the tape. θ represents the membership degree of Y in place of X. Similarly, $\delta(q,X) = (p,Y,-1,\theta)$ means that the read-write head moves left one unit after the same process as above.

$F\subseteq Q$ is the set of final states.

The above description is the definition of a single tape lattice-valued Turing machine. For the n tapes lattice-valued Turing machine, the transition function is defined as the map from $Q\times\Gamma^n$ to $Q\times\Gamma^n\times\{-1,+1\}^n\times L$.

We use $\alpha_1q\alpha_2$ to denote an instantaneous description (ID for short) of the lattice-valued Turing machine $\mathscr{M}$, where $q_0\in Q$ is the current state of $\mathscr{M}$ and $\alpha_1\alpha_2$ is the string of Γ^*. When the read-write head of $\mathscr{M}$ directs symbols right which has nonblank character, $\alpha_1q\alpha_2$ is the string which consists of all nonblank symbols of the leftmost position to the rightmost position of the input tape of $\mathscr{M}$, otherwise, $\alpha_1q\alpha_2$ is the string which consists of all symbols of the leftmost position of the input tape of $\mathscr{M}$ to the tape location which is directed by the read head of $\mathscr{M}$, and $\mathscr{M}$ is directing the leftmost symbol of α_2.

Now, we define a binary lattice-valued relation on $\Gamma^*Q\Gamma^*$ as follows.

Let

$$\mathrm{ID}_1 = X_1X_2\cdots X_{i-1}qX_iX_{i+1}\cdots X_n \tag{15}$$

be a ID of $\mathscr{M}$; if $\delta(q,X_i) = (p,Y,+1,\theta)$, then the next ID of $\mathscr{M}$ is

$$\mathrm{ID}_2 = X_1X_2\cdots X_{i-1}YpX_{i+1}\cdots X_n. \tag{16}$$

The membership degree of Y replacing X_i is θ; that is,

$$\underset{\mathscr{M}}{\vdash}(\mathrm{ID}_1,\mathrm{ID}_2) = \theta\left(\text{or } \mathrm{ID}_1 \underset{\mathscr{M}\theta}{\vdash} \mathrm{ID}_2\right). \tag{17}$$

If $\delta(q,X_i) = (p,Y,+1,\theta)$, then, if $i\neq 1$, the next ID of $\mathscr{M}$ is

$$\mathrm{ID}_2 = X_1X_2\cdots pX_{i-1}YX_{i+1}\cdots X_n; \tag{18}$$

the membership degree of Y replacing X_i is θ; that is,

$$\vdash(\mathrm{ID}_1,\mathrm{ID}_2) = \theta\Bigg(\text{or } X_1X_2\cdots X_{i-1}pX_iX_{i+1}\cdots X_n \underset{\mathscr{M}\theta}{\vdash} X_1X_2\cdots pX_{i-1}YX_{i+1}\cdots X_n\Bigg). \tag{19}$$

If $i = 1$, before $\mathscr{M}$ moves on, the read-write head has been at the far left of the input tape, then the read-write head moves left, which would make the read-write head away from the input tape, which is not allowed. In order to avoid this phenomenon, in this case, we have defined that $\mathscr{M}$ has not the next ID.

Obviously, $\vdash_{\mathscr{M}}$ is a binary lattice-valued relation on $\Gamma^*Q\Gamma^*\times\Gamma^*Q\Gamma^*$.

$\vdash^n_{\mathscr{M}}$ denotes n power of $\vdash_{\mathscr{M}}$; that is, $\vdash^n_{\mathscr{M}} = (\vdash_{\mathscr{M}})^n$; $\vdash^+_{\mathscr{M}}$ denotes the positive closure of $\vdash_{\mathscr{M}}$; that is, $\vdash^+_{\mathscr{M}} = (\vdash_{\mathscr{M}})^+$; $\vdash^*_{\mathscr{M}}$ denotes the star-closure of $\vdash_{\mathscr{M}}$; that is, $\vdash^*_{\mathscr{M}} = (\vdash_{\mathscr{M}})^*$.

Let $\mathrm{ID}_1,\mathrm{ID}_2 \in \Gamma^*Q\Gamma^*\times\Gamma^*Q\Gamma^*$. By the composition definition of the binary lattice-valued relations, it is not difficult to see the following:

if $\vdash^n_{\mathscr{M}}(\mathrm{ID}_1,\mathrm{ID}_2) = \theta\neq 0$, which represents that the membership degree ID_1 turns to ID_2 after n steps is θ in $\mathscr{M}$; if $\vdash^+_{\mathscr{M}}(\mathrm{ID}_1,\mathrm{ID}_2) = \theta\neq 0$, which represents that the membership degree ID_1 changes into ID_2 after one step at least is θ in $\mathscr{M}$; if $\vdash^*_{\mathscr{M}}(\mathrm{ID}_1,\mathrm{ID}_2) = \theta\neq 0$, which represents that the membership degree ID_1 turns into ID_2 after several steps is θ in $\mathscr{M}$.

Definition 1. Let $\mathscr{M} = (Q,\Gamma,\Sigma,\delta,q_0,B,F)$ be a lattice-valued Turing machine. The acceptable lattice-valued language $\mathscr{M}$ is defined by

$$\varphi_{\mathscr{M}}(x) = \vee\left\{\overset{*}{\underset{\mathscr{M}}{\vdash}}(q_0x,\alpha_1q\alpha_2) \mid q\in F,\alpha_1,\alpha_2\in\Gamma^*\right\}. \tag{20}$$

The language which is accepted by the lattice-valued Turing machine is called a lattice-valued recursively enumerable language.

4. The Lattice-Valued Type 0 Grammar and Lattice-Valued Phrase Structure Language

The *lattice-valued grammar* $\mathscr{G}$ is a 4-tuple $\mathscr{G} = (V,T,P,S)$, where we have the following.

V is a nonempty finite set of variables and $\forall A\in V$, A is called a *syntactic variable* (variable for short) or a nonterminal symbol. It represents a syntactic category.

T is a nonempty finite set of the terminal symbols and $\forall a\in T$, a is called a terminal symbol. Since the variables in V represent the syntactic category and the characters in T are the characters that appear in the sentence of language, so we have $V\cap T=\emptyset$.

P is a set of lattice-valued of production; that is, P is a binary lattice-valued relationship on $(P\cup T)^*$. For $\alpha,\beta\in(V\cup T)^*$ and a fixed $\theta\in L$, if $P(\alpha,\beta)=\theta$ and $\theta\neq 0$, then it can be denoted by $\alpha\rightarrow_\theta\beta$ which is called the production on $\mathscr{G}$, and the production means that the membership degree that α is defined as β is θ. α is called the left part of $\alpha\rightarrow_\theta\beta$, β is the right part of $\alpha\rightarrow_\theta\beta$, and θ is the membership degree that α is defined as β. The productions are termed the definitions or the grammar rules.

$S\in V$ is the start symbol of the grammar $\mathscr{G}$.

According to Chomsky hierarchy of the grammar and the concept of the fuzzy grammar of Lee and Zadeh, the lattice-valued grammar can be divided into the following four types.

Definition 2. Let $\mathscr{G} = (V,T,P,S)$ be a lattice-valued grammar. Then one has the following.

(1) $\mathscr{G}$ is called the lattice-valued type 0 grammar or the lattice-valued phrase structure grammar, if the production in P is without any constraint conditions.

The corresponding language $\varphi_{\mathscr{G}}$ is called the lattice-valued type 0 language or the lattice-valued phrase structure language.

(2) $\mathscr{G}$ is called the lattice-valued type 1 grammar or the lattice-valued context sensitive grammar, if, for any production $\alpha \to_{\theta} \beta \in P$, one has $|\alpha| \leq |\beta|$. The corresponding language $\varphi_{\mathscr{G}}$ is called the lattice-valued type 1 language or the lattice-valued context sensitive language.

(3) $\mathscr{G}$ is called the lattice-valued type 2 grammar or the lattice-valued context-free grammar, if, for any production $\alpha \to_{\theta} \beta \in P$, one has $|\alpha| \leq |\beta|$ and $\alpha \in V$. The corresponding language $\varphi_{\mathscr{G}}$ is called the lattice-valued type 2 language or the lattice-valued context-free language.

(4) $\mathscr{G}$ is called the lattice-valued type 3 grammars or the lattice-valued regular grammar, if, for any production $\alpha \to_{\theta} \beta \in P$, one has

$$A \longrightarrow_{\theta} \omega, \quad A \longrightarrow_{\lambda} \omega B \quad \left(\text{or } A \longrightarrow_{\theta} \omega, A \longrightarrow_{\lambda} B\omega\right), \tag{21}$$

where $A, B \in V, \omega \in T^{*}$. The corresponding language $\varphi_{\mathscr{G}}$ is called the lattice-valued type 3 language or the lattice-valued regular language.

Definition 3. Let $\mathscr{G} = (V, T, P, S)$ be a lattice-valued grammar. Define a binary lattice-valued relationship $\underset{\mathscr{G}}{\Rightarrow}$ on $(V \cup T)^{*}$ as follows:

$$\underset{\mathscr{G}}{\Longrightarrow} \left(\gamma\alpha\delta, \gamma\beta\delta\right) = P\left(\alpha, \beta\right), \tag{22}$$

for all $\gamma, \delta \in (V \circ T)^{*}$ and $\alpha \to_{\theta} \beta \in P$, and it can be written as $\gamma\alpha\delta \underset{\mathscr{G}}{\Rightarrow}_{\theta} \gamma\beta\delta$.

If $\underset{\mathscr{G}}{\Rightarrow} (\gamma\alpha\delta, \gamma\beta\delta) = \theta \neq 0$, then we can say that the membership degree that $\gamma\alpha\delta$ can deduce $\gamma\beta\delta$ in the lattice-valued grammar $\mathscr{G}$ is θ, or the membership degree that $\gamma\beta\delta$ can be reduced into $\gamma\alpha\delta$ in the lattice-valued grammar $\mathscr{G}$ is θ.

$\underset{\mathscr{G}}{\overset{n}{\Rightarrow}}$ denotes the n power of $\underset{\mathscr{G}}{\Rightarrow}$; that is, $\underset{\mathscr{G}}{\overset{n}{\Rightarrow}} = (\Rightarrow_{\mathscr{G}})^{n}$; $\underset{\mathscr{G}}{\overset{+}{\Rightarrow}}$ denotes the positive closure of $\underset{\mathscr{G}}{\Rightarrow}$; that is, $\underset{\mathscr{G}}{\overset{+}{\Rightarrow}} = (\Rightarrow_{\mathscr{G}})^{+}$; $\underset{\mathscr{G}}{\overset{*}{\Rightarrow}}$ denotes the star-closure of $\underset{\mathscr{G}}{\Rightarrow}$; that is, $\underset{\mathscr{G}}{\overset{*}{\Rightarrow}} = (\Rightarrow_{\mathscr{G}})^{*}$.

Let $\alpha, \beta \in (N \cup T)^{*}$; according to the composition definition of the binary lattice-valued relationship, it is not difficult to see the following.

If $\underset{\mathscr{G}}{\overset{n}{\Rightarrow}} (\alpha, \beta) = \theta \neq 0$ represents the membership degree that α in $\mathscr{G}$ after n steps can deduce β is θ, or the membership degree that β in $\mathscr{G}$ after n steps can summarize α is θ, it also can be written as $\alpha \underset{\mathscr{G}}{\overset{n}{\Rightarrow}}_{\theta} \beta$.

If $\underset{\mathscr{G}}{\overset{+}{\Rightarrow}} (\alpha, \beta) = \theta \neq 0$ represents the membership degree that α in $\mathscr{G}$ after at least 1 step can deduce β is θ, or the membership degree that β in $\mathscr{G}$ after at least 1 step can summarize α is θ, it also can be written as $\alpha \underset{\mathscr{G}}{\overset{+}{\Rightarrow}}_{\theta} \beta$; if $\underset{\mathscr{G}}{\overset{*}{\Rightarrow}} (\alpha, \beta) = \theta \neq 0$ represents the membership degree that α in $\mathscr{G}$ after some steps can deduce β is θ, or the membership degree that β in $\mathscr{G}$ after some steps can summarize α is θ, it also can be written as $\alpha \underset{\mathscr{G}}{\overset{*}{\Rightarrow}}_{\theta} \beta$.

Definition 4. Let $\mathscr{G} = (V, T, P, S)$ be a lattice-valued type 0 grammar, $\omega \in T^{*}$. Define

$$\varphi_{\mathscr{G}}\left(\omega\right) = \left(\underset{\mathscr{G}}{\overset{*}{\Longrightarrow}}\right)\left(S, \omega\right). \tag{23}$$

$\varphi_{\mathscr{G}}$ is called the lattice-valued language generated by lattice-valued type 0 grammar which is termed the lattice-valued type 0 language or lattice-valued phrase structure language. One calls that two lattice-valued grammars $\mathscr{G}_1, \mathscr{G}_2$ are equivalent, if they can generate the same lattice-valued languages; that is, $\varphi_{\mathscr{G}_1} = \varphi_{\mathscr{G}_2}$.

5. The Equivalence between the Lattice-Valued Type 0 Grammar and the Lattice-Valued Turing Machine

Theorem 5. *If φ can be recognized by the lattice-valued Turing machine*

$$\mathscr{M} = \left(Q, \Gamma, \Sigma, \delta, q_0, B, F\right). \tag{24}$$

that is, $\varphi = \varphi_{\mathscr{G}}$, then there exists the lattice-valued type 0 grammar $\mathscr{G}$ such that $\varphi = \varphi_{\mathscr{G}}$.

Proof. Suppose that φ can be recognized by lattice-valued Turing machine

$$\mathscr{M} = \left(Q, \Gamma, \Sigma, \delta, q_0, B, F\right). \tag{25}$$

Formally, we construct a lattice-valued type 0 grammar $\mathscr{G} = (V, \Sigma, P, A_1)$, where

$$V = ((\Sigma \cup \epsilon) \times \Gamma) \cup \left\{A_1, A_2, A_3\right\} \cup Q, \tag{26}$$

and the production in P is as follows:

(1) $A_1 \to q_0 A_2$, its membership degree is 1, and the following productions which are not labeled with membership degree were regarded as 1;

(2) $A_2 \to [a, a] A_2, \forall a \in \Sigma$;

(3) $A_2 \to A_3$;

(4) $A_3 \to [\epsilon, B] A_2$;

(5) $A_3 \to \epsilon$;

(6) $q[a, X] \to_{\theta} [a, Y] p, \forall a \in \Sigma \cup \{\epsilon\}, \forall q \in Q, \forall X, Y \in \Gamma$, and $\delta(q, X) = (p, Y, +1, \theta)$;

(7) $[b, Z] q [a, X] \to_{\theta} p [a, Y], \forall X, Y, Z \in \Gamma, \forall a, b \in \Sigma \cup \{\epsilon\}$, $\forall q \in Q$, and $\delta(q, X) = (p, Y, -1, \theta)$;

(8) $[a, X] q \to qaq, q[a, X] \to qaq, q \to \varepsilon, \forall a \in \Sigma \cup \{\epsilon\}$, $\forall X \in \Gamma, \forall q \in F$.

By using rules (1) and (2), we can obtain

$$A_1 \overset{*}{\underset{\mathscr{G}}{\Longrightarrow}} q_0\left[a_1,a_1\right]\left[a_2,a_2\right]\cdots\left[a_n,a_n\right]A_2, \tag{27}$$

where every a_i belongs to Σ. Assume that $\mathscr{M}$ can accept a string $a_1, a_2, \ldots, a_n$, and then there exists m such that $\mathscr{M}$ can use at most m units of the string input on the right. Using rules (3), (4) (m times), and (5) in order, we can obtain

$$A_1 \overset{*}{\underset{\mathscr{G}}{\Longrightarrow}} q_0\left[a_1,a_1\right]\left[a_2,a_2\right]\cdots\left[a_n,a_n\right]\left[\epsilon,B\right]^m. \tag{28}$$

Then we only use rules (6) and (7), until an accepting state can be generated.

By the induction on movement numbers made by $\mathscr{M}$, we can prove that, if

$$q_0a_1a_2\cdots a_n \overset{*}{\underset{\mathscr{G}}{\Longrightarrow}} {}_\theta X_1, X_2\cdots X_{r-1}qX_r\cdots X_s, \tag{29}$$

then

$$\begin{aligned} &q_0\left[a_1,a_1\right]\left[a_2,a_2\right]\cdots\left[a_n,a_n\right]\left[\epsilon,B\right]^m \\ &\quad \overset{*}{\underset{\mathscr{G}}{\Rightarrow}}_{\theta} \left[a_1,X_1\right]\left[a_2,X_2\right]\cdots\left[a_{r-1},X_{r-1}\right]q \\ &\qquad \times\left[a_r,X_r\right]\cdots\left[a_{m+n},X_{m+n}\right], \end{aligned} \tag{30}$$

for all $a_1, a_2 \ldots, a_n \in \Sigma$, $a_{n-1} = a_{n+2} = a_{m+n} = \epsilon$, $X_1, X_2\cdots X_{m+n} \in \Gamma$, and $X_{s-1} = X_{s+2} = X_{m+n} = B$.

Obviously, for the 0 movement, the inductive hypothesis holds, since $r = 1$, $s = n$.

Assume that the inductive hypothesis holds for the $k-1$ movements. Let

$$\begin{aligned} &q_0a_1a_2\cdots a_n \overset{n}{\underset{\mathscr{G}}{\Rightarrow}}_{\theta_1} \\ &\quad X_1, X_2\cdots X_{r-1}qX_r\cdots X_s \underset{\mathscr{M}\theta_2}{\vdash} Y_1, Y_2\cdots Y_{t-1}pY_t\cdots Y_u. \end{aligned} \tag{31}$$

By the induction hypothesis,

$$\begin{aligned} &q_0\left[a_1,a_1\right]\left[a_2,a_2\right]\cdots\left[a_n,a_n\right]\left[\epsilon,B\right]^m \\ &\quad \overset{*}{\underset{\mathscr{G}}{\Rightarrow}}_{\theta_1} \left[a_1,X_1\right]\left[a_2,X_2\right]\cdots\left[a_{r-1},X_{r-1}\right]q \\ &\qquad \times\left[a_r,X_r\right]\cdots\left[a_{m+n},X_{m+n}\right], \end{aligned} \tag{32}$$

where each a_i and X_i satisfy the following conditions.

If $t = r+1$, then the kth movement of $\mathscr{M}$ goes to the right, so

$$\delta\left(q, X_r\right) = \left(p, Y_r, +1, \lambda\right), \tag{33}$$

and the membership degree $\theta_2 = \theta_1 \wedge \lambda$. By rule (6),

$$q\left[a_r,a_r\right] \longrightarrow_{\lambda} \left[a_r,Y_r\right]p \tag{34}$$

is a production of $\mathscr{G}$. Therefore,

$$\begin{aligned} &q_0\left[a_1,a_1\right]\left[a_2,a_2\right]\cdots\left[a_n,a_n\right]\left[\epsilon,B\right]^m \\ &\quad \overset{*}{\underset{\mathscr{G}}{\Rightarrow}}_{\theta_2} \left[a_1,Y_1\right]\left[a_2,Y_2\right]\cdots\left[a_{t-1},Y_{t-1}\right]p \\ &\qquad \times\left[a_t,Y_t\right]\cdots\left[a_{m+n},Y_{m+n}\right], \end{aligned} \tag{35}$$

for $i > u$, $Y_i = B$, and the membership degree $\theta_2 = \theta_1 \wedge \lambda$.

If $t = r-1$, then the kth movement of $\mathscr{M}$ goes to the left, we can apply rule (7) to prove the above equation, and we can obtain for the case of $r-1$

$$\delta\left(q, X_r\right) = \left(p, Y_r, -1, \lambda\right). \tag{36}$$

By rule (8), if p in F, then

$$\begin{aligned} &\left[a_1,Y_1\right]\cdots\left[a_{t-1},Y_{t-1}\right]p\left[a_t,Y_t\right]\cdots\left[a_{m+n},Y_{m+n}\right] \\ &\quad \overset{*}{\underset{\mathscr{G}}{\Rightarrow}}_{\theta} a_1a_2\cdots a_n. \end{aligned} \tag{37}$$

Thus

$$\begin{aligned} \varphi_{\mathscr{G}}(\omega) &= \overset{*}{\underset{\mathscr{G}}{\Longrightarrow}} \left(A_1, \omega\right) \\ &= \overset{*}{\underset{\mathscr{M}\theta}{\vdash}} \left(q_0a_1a_2\cdots a_n, X_1, X_2\cdots X_{r-1}qX_r\cdots X_s\right) \\ &= \overset{*}{\underset{\mathscr{M}\theta}{\vdash}} \left(q_0\omega, \alpha_1q\alpha_2\right) \\ &= \vee\left\{\overset{*}{\underset{\mathscr{M}}{\vdash}} \left(q_0\omega, \alpha_1q\alpha_2\right) \mid q \in F, \alpha_1, \alpha_2 \in \Gamma^*\right\} = \varphi_{\mathscr{M}}(\omega). \end{aligned} \tag{38}$$

□

Similarly, we have the following result.

Theorem 6. *Assume φ is a language $\varphi_{\mathscr{G}}$ which generated by the lattice-valued type 0 grammar $\mathscr{G} = (V, T, P, S)$, and then, φ can be recognized by the lattice-valued Turing machine $\mathscr{M} = (Q, \Gamma, \Sigma, \delta, q_0, B, F)$; that is, $\varphi = \varphi_{\mathscr{M}}$.*

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors thank reviewers for their patient and invaluable advice, all of which have been of inestimable worth to the completion of their paper. They also thank editor for being serious and responsible. The authors acknowledge the support of the National Natural Science Foundation of China (nos. 11161050, 31240020).

References

[1] W. G. Wee, *On generalizations of adaptive algorithm and application of the fuzzy sets concept to pattern classification [Ph.D. thesis]*, Purdue University, West Lafayette, Ind, USA, 1967.

[2] W. G. Wee and F. S. Fu, "A formulation of fuzzy automata and its application as a model of learning systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 5, no. 3, pp. 215–223, 1969.

[3] E. S. Santos, "Realizations of fuzzy languages by probabilistic, max-product, and maximin automata," *Information Sciences*, vol. 8, no. 1, pp. 39–53, 1975.

[4] E. Santos, "Fuzzy automata and languages," *Information Sciences*, vol. 10, no. 2, pp. 193–197, 1976.

[5] E. S. Santos, "Fuzzy and probabilistic programs," *Information Sciences*, vol. 10, no. 4, pp. 331–345, 1976.

[6] E. S. Santos, "Regular fuzzy expressions," in *Fuzzy Automata and Decision Processes*, M. M. Gupta, G. H. Saridis, and B. R. Gaines, Eds., pp. 169–175, Elsevier, North-Holland, The Netherlands, 1977.

[7] E. T. Lee and L. A. Zadeh, "Note on fuzzy languages," *Information Sciences*, vol. 1, no. 4, pp. 421–434, 1969.

[8] H. V. Kumbhojkar and S. R. Chaudhari, "On covering of products of fuzzy finite state machines," *Fuzzy Sets and Systems*, vol. 125, no. 2, pp. 215–222, 2002.

[9] H. V. Kumbhojkar and S. R. Chaudhari, "Fuzzy recognizers and recognizable sets," *Fuzzy Sets and Systems*, vol. 131, no. 3, pp. 381–392, 2002.

[10] D. S. Malik, J. N. Mordeson, and M. K. Sen, "Semigroups of fuzzy finite state machines," in *Advances in Fuzzy Technology*, P. P. Wang, Ed., Bookswrite, Durham, NC, USA, 1995.

[11] D. S. Malik, J. N. Mordeson, and M. K. Sen, "On subsystems of a fuzzy finite state machine," *Fuzzy Sets and Systems*, vol. 68, no. 1, pp. 83–92, 1994.

[12] D. S. Malik, J. N. Mordeson, and M. K. Sen, "Submachines of fuzzy finite state machines," *The Journal of Fuzzy Mathematics*, vol. 2, no. 4, pp. 781–792, 1994.

[13] D. S. Malik, J. N. Mordeson, and M. K. Sen, "Admissible partitions of fuzzy finite state machines," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 5, no. 6, pp. 723–732, 1997.

[14] D. S. Malik, J. N. Mordesotf, and M. K. Sen, "Products of fuzzy finite state machines," *Fuzzy Sets and Systems*, vol. 92, no. 1, pp. 95–102, 1997.

[15] J. N. Mordeson and D. S. Malik, *Fuzzy Automata and Languages: Theory and Applications*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2002.

[16] G. Birkhoff, *Lattice Theory*, Colloquium Publications—American Mathematical Society, New York, NY, USA, 1993.

[17] J. E. Hopcroft and J. D. Ullman, *Formal Languages and Their Relation to Automata*, Addison-Wesley, Boston, Mass, USA, 1969.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 874651, 6 pages
http://dx.doi.org/10.1155/2014/874651

Research Article

Derivative-Based Trapezoid Rule for the Riemann-Stieltjes Integral

Weijing Zhao and Zhaoning Zhang

College of Air Traffic Management, Civil Aviation University of China, Tianjin 300300, China

Correspondence should be addressed to Weijing Zhao; 693509394@qq.com

Received 23 January 2014; Revised 14 April 2014; Accepted 14 April 2014; Published 30 April 2014

Academic Editor: Ker-Wei Yu

The derivative-based trapezoid rule for the Riemann-Stieltjes integral is presented which uses 2 derivative values at the endpoints. This kind of quadrature rule obtains an increase of two orders of precision over the trapezoid rule for the Riemann-Stieltjes integral and the error term is investigated. At last, the rationality of the generalization of derivative-based trapezoid rule for Riemann-Stieltjes integral is demonstrated.

1. Introduction

In mathematics, the Riemann-Stieltjes integral is a kind of generalization of the Riemann integral, named after Bernhard Riemann and Thomas Stieltjes. It is Stieltjes that first gave the definition of this integral [1] in 1894. It serves as an instructive and useful precursor of the Lebesgue integral. It is known that the Riemann-Stieltjes integral has wide applications in the field of probability theory [2, 3], stochastic process [4], and functional analysis [5], especially in original formulation of F. Riesz's theorem and the spectral theorem for self-adjoint operators in a Hilbert space.

Definite integration is one of the most important and basic concepts in mathematics. And it has numerous applications in fields such as physics and engineering. In several practical problems, we need to calculate integrals. As is known to all, as for $I = \int_a^b f(x)dx$, once the primitive function $F(x)$ of integrand $f(x)$ is known, the definite integral of $f(x)$ over the interval $[a,b]$ is given by Newton-Leibniz formula, that is,

$$\int_a^b f(x)\,dx = F(b) - F(a). \tag{1}$$

However, the explicit primitive function $F(x)$ is not available or its primitive function is not easy to obtain, such as $e^{\pm x^2}$, $\sin x^2$, and $(\sin x)/x$. Moreover, the integrand $f(x)$ is only available at certain points x_i, $i = 1, 2, \dots, n$. How to get high-precision numerical integration formulas becomes one of the challenges in fields of mathematics [6].

The methods of quadrature are usually based on the interpolation polynomials and can be written in the following form:

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{n} w_i f(x_i), \tag{2}$$

where there are $n+1$ distinct integration points at $x_0, x_1, \dots, x_n$ within the interval $[a,b]$ and $n+1$ weights w_i. If the integration points are uniformly distributed over the interval, so $x_i = x_0 + ih$ in which $h = (b-a)/n$.

These w_i can be derived in several different ways [7–9]. One is interpolate $f(x)$ at the $n+1$ points $x_0, x_1, \dots, x_n$, using the Lagrange polynomials and then integrating the foresaid polynomials to obtain (2).

The other is based on the precision of a quadrature formula. Select the w_i, $i = 0, 1, \dots, n$, so that the error,

$$R_n(f) = \int_a^b f(x)\,dx - \sum_{i=0}^{n} w_i f(x_i), \tag{3}$$

is exactly zero for $f(x) = x^j$, $j = 0, 1, \dots, n$. Using the method of undetermined coefficients, this approach generates a system of $n+1$ linear equations for weights w_i.

Since the monomials $1, x, \ldots, x^n$ are linearly independent, the linear system of equations has a unique solution.

The trapezoidal rule is the most well-known numerical integration rule of this type. Trapezoidal rule for classical Riemann integral is

$$\int_a^b f(x)\,dx = \frac{b-a}{2}\left(f(a)+f(b)\right) - \frac{(b-a)^3}{12} f''(\xi), \quad (4)$$

where $\xi \in (a,b)$.

In spite of the many accurate and efficient methods for numerical integration being available in [7–9], recently Mercer has obtained trapezoid rule for Riemann-Stieltjes integral which engender a generalization of Hadamard's integral inequality [10]. Then he develops Midpoint and Simpson's rules for Riemann-Stieltjes integral [11] by using the concept of relative convexity. Burg has proposed derivative-based closed Newton-Cotes numerical quadrature [12] which uses both the function value and the derivative value on uniformly spaced intervals. Zhao and Li have proposed midpoint derivative-based closed Newton-Cotes quadrature [13] and numerical superiority has been shown. Recently, Simos and his partners have made a contribution to the Newton-Cotes formula for the Riemann integral and its applications [14–21], especially the connection between closed Newton-Cotes, trigonometrically fitted differential methods, symplectic integrators, and efficient solution of the Schrodinger equation [17–21].

Motivation for the research presented here lies in construction of derivative-based trapezoid rule for the Riemann-Stieltjes integral, which is generalization of the results in [10–12]. In this paper, the derivative-based trapezoid rule for the Riemann-Stieltjes integral is presented. This new scheme is investigated in Section 2. In Section 3, the error term is presented. Finally, conclusions are drawn in Section 4.

2. Derivative-Based Trapezoid Rule for the Riemann-Stieltjes Integral

In this section, by adding the derivatives at the endpoints, derivative-based trapezoid rule for the Riemann-Stieltjes integral is presented.

Theorem 1. *Suppose that f' and g are continuous on $[a,b]$ and g is increasing there. The derivative-based trapezoid rule for the Riemann-Stieltjes integral is*

$$\begin{aligned}
&\int_a^b f(t)\,dg \\
&\quad\approx T = \left(\frac{6}{(b-a)^2}\int_a^b\int_a^t g(x)\,dx\,dt \right.\\
&\qquad - \frac{12}{(b-a)^3}\int_a^b\int_a^t\int_a^y g(x)\,dx\,dy\,dt \\
&\qquad \left. -g(a)\right) f(a) \\
&\qquad + \left(g(b) - \frac{6}{(b-a)^2}\int_a^b\int_a^t g(x)\,dx\,dt \right.\\
&\qquad \left. + \frac{12}{(b-a)^3}\int_a^b\int_a^t\int_a^y g(x)\,dx\,dy\,dt\right) f(b) \\
&\qquad + \left(\frac{2}{b-a}\int_a^b\int_a^t g(x)\,dx\ dt \right.\\
&\qquad \left. - \frac{6}{(b-a)^2}\int_a^b\int_a^t\int_a^y g(x)\,dx\,dy\,dt\right) f'(a) \\
&\qquad + \left(\frac{4}{b-a}\int_a^b\int_a^t g(x)\,dx\,dt \right.\\
&\qquad - \frac{6}{(b-a)^2}\int_a^b\int_a^t\int_a^y g(x)\,dx\,dy\,dt \\
&\qquad \left. - \int_a^b g(t)\,dt\right) f'(b).
\end{aligned} \quad (5)$$

Proof. Looking for the derivative-based trapezoid rule for the Riemann-Stieltjes integral, we seek numbers a_0, a_1, b_0, b_1 such that

$$\int_a^b f(t)\,dg \approx a_0\ f(a) + a_1 f(b) + b_0 f'(a) + b_1\ f'(b) \quad (6)$$

is equality for $f(t) = 1, t, t^2, t^3$. That is,

$$\begin{aligned}
&\int_a^b 1\,dg = a_0 + a_1; \\
&\int_a^b t\,dg = a_0 a + a_1 b + b_0 + b_1; \\
&\int_a^b t^2 dg = a_0 a^2 + a_1 b^2 + 2b_0 a + 2b_1 b; \\
&\int_a^b t^3 dg = a_0 a^3 + a_1 b^3 + 3b_0 a^2 + 3b_1 b^2.
\end{aligned} \quad (7)$$

Therefore,

$$\begin{aligned}
&a_0 + a_1 = g(b) - g(a); \\
&a_0 a + a_1 b + b_0 + b_1 = bg(b) - ag(a) - \int_a^b g(t)\,dt; \\
&a_0 a^2 + a_1 b^2 + 2b_0 a + 2b_1 b \\
&\quad = b^2 g(b) - a^2 g(a) - 2b\int_a^b g(t)\,dt \\
&\qquad + 2\int_a^b\int_a^t g(x)\,dx\,dt;
\end{aligned}$$

$$\begin{aligned}
&a_0a^3 + a_1b^3 + 3b_0a^2 + 3b_1b^2 \\
&\quad = b^3 g(b) - a^3 g(a) - 3b^2 \int_a^b g(t)\,dt \\
&\qquad + 6b \int_a^b \int_a^t g(x)\,dx\,dt \\
&\qquad - 6 \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt.
\end{aligned} \tag{8}$$

Solving simultaneous (8) for a_0, a_1, b_0, b_1, we obtain

$$\begin{aligned}
a_0 &= \frac{6}{(b-a)^2} \int_a^b \int_a^t g(x)\,dx\,dt \\
&\quad - \frac{12}{(b-a)^2} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt - g(a); \\
a_1 &= g(b) - \frac{6}{(b-a)^2} \int_a^b \int_a^t g(x)\,dx\,dt \\
&\quad + \frac{12}{(b-a)^2} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt; \\
b_0 &= \frac{2}{b-a} \int_a^b \int_a^t g(x)\,dx\,dt \\
&\quad - \frac{6}{(b-a)^2} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt; \\
b_1 &= \frac{4}{b-a} \int_a^b \int_a^t g(x)\,dx\,dt \\
&\quad - \frac{6}{(b-a)^2} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt - \int_a^b g(t)\,dt.
\end{aligned} \tag{9}$$

So we have the derivative-based trapezoid rule for the Riemann-Stieltjes integral as desired. □

Corollary 2. *The precision of the derivative-based trapezoid rule for the Riemann-Stieltjes integral is 3.*

Proof. From the construction of a_0, a_1, b_0, b_1, we obtain that the derivative-based trapezoidal rule for the Riemann-Stieltjes integral has degree of precision not less than 3.

In Section 3 Theorem 3, we can easily see that the quadrature is not equality for $f(t) = t^4$. So the precision of this method is 3. □

3. The Error Term for Riemann-Stieltjes Derivative-Based Trapezoid Rule

In this section, the error term of the derivative-based trapezoid rule for the Riemann-Stieltjes is investigated. The error term can be found in mainly 3 different ways [8, 9].

Here, we use the concept of precision to calculate the error term, where the error term is related to the difference between the quadrature formula for the monomial $x^{p+1}/(p+1)!$ and the exact value $(1/(p+1)!) \int_a^b x^{p+1} dx$, where p is the precision of the quadrature formula.

Theorem 3. *Suppose that $f^{(4)}$ and g' are continuous on $[a, b]$ and g is increasing there. The derivative-based trapezoid rule for the Riemann-Stieltjes integral with the error term is*

$$\begin{aligned}
&\int_a^b f(t)\,dg \\
&\quad = \left(\frac{6}{(b-a)^2} \int_a^b \int_a^t g(x)\,dx\,dt \right. \\
&\qquad - \frac{12}{(b-a)^3} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \\
&\qquad \left. - g(a) \right) f(a) \\
&\quad + \left(g(b) - \frac{6}{(b-a)^2} \int_a^b \int_a^t g(x)\,dx\,dt \right. \\
&\qquad \left. + \frac{12}{(b-a)^3} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \right) f(b) \\
&\quad + \left(\frac{2}{b-a} \int_a^b \int_a^t g(x)\,dx\,dt \right. \\
&\qquad \left. - \frac{6}{(b-a)^2} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \right) f'(a) \\
&\quad + \left(\frac{4}{b-a} \int_a^b \int_a^t g(x)\,dx\,dt \right. \\
&\qquad - \frac{6}{(b-a)^2} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \\
&\qquad \left. - \int_a^b g(t)\,dt \right) f'(b) \\
&\quad + \left(\frac{(b-a)^2}{12} \int_a^b \int_a^t g(x)\,dx\,dt \right. \\
&\qquad - \frac{b-a}{12} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \\
&\qquad \left. + \int_a^b \int_a^t \int_a^z \int_a^y g(x)\,dx\,dy\,dz\,dt \right) \\
&\quad \times f^{(4)}(\xi)\, g'(\eta),
\end{aligned} \tag{10}$$

where $\xi, \eta \in (a, b)$. And the error term $R[f]$ of this method is

$$\begin{aligned}\Bigg(& \frac{(b-a)^2}{12} \int_a^b \int_a^t g(x)\,dx\,dt \\ & - \frac{b-a}{12} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \\ & + \int_a^b \int_a^t \int_a^z \int_a^y g(x)\,dx\,dy\,dz\,dt \Bigg) f^{(4)}(\xi)\, g'(\eta).\end{aligned} \tag{11}$$

Proof. Let $f(t) = t^4/4!$.

So

$$\begin{aligned}\frac{1}{4!} \int_a^b t^4 dg = {} & \frac{1}{24} \left(b^4 g(b) - a^4 g(a) \right) \\ & - \frac{b^3}{6} \int_a^b g(t)\,dt + \frac{b^2}{2} \int_a^b \int_a^t g(x)\,dx \\ & - b \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \\ & + \int_a^b \int_a^t \int_a^z \int_a^y g(x)\,dx\,dy\,dz\,dt.\end{aligned} \tag{12}$$

By Theorem 1, we have

$$\begin{aligned}T = {} & \Bigg(\frac{6}{(b-a)^2} \int_a^b \int_a^t g(x)\,dx\,dt \\ & - \frac{12}{(b-a)^3} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt - g(a) \Bigg) \frac{a^4}{24} \\ & + \Bigg(g(b) - \frac{6}{(b-a)^2} \int_a^b \int_a^t g(x)\,dx\,dt \\ & + \frac{12}{(b-a)^3} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \Bigg) \frac{b^4}{24} \\ & + \Bigg(\frac{2}{b-a} \int_a^b \int_a^t g(x)\,dx\,dt \\ & - \frac{6}{(b-a)^2} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \Bigg) \frac{a^3}{6} \\ & + \Bigg(\frac{4}{b-a} \int_a^b \int_a^t g(x)\,dx\,dt \\ & - \frac{6}{(b-a)^2} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \\ & - \int_a^b g(t)\,dt \Bigg) \frac{b^3}{6}.\end{aligned} \tag{13}$$

By Equations (12)-(13), we obtain

$$\begin{aligned}& \frac{1}{4!} \int_a^b t^4 dg - T \\ & \quad = \frac{(b-a)^2}{12} \int_a^b \int_a^t g(x)\,dx\,dt \\ & \quad - \frac{b-a}{12} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \\ & \quad + \int_a^b \int_a^t \int_a^z \int_a^y g(x)\,dx\,dy\,dz\,dt.\end{aligned} \tag{14}$$

This implies that

$$\begin{aligned}R[f] = {} & \Bigg(\frac{(b-a)^2}{12} \int_a^b \int_a^t g(x)\,dx\,dt \\ & - \frac{b-a}{12} \int_a^b \int_a^t \int_a^y g(x)\,dx\,dy\,dt \\ & + \int_a^b \int_a^t \int_a^z \int_a^y g(x)\,dx\,dy\,dz\,dt \Bigg) \\ & \times f^{(4)}(\xi)\, g'(\eta).\end{aligned} \tag{15}$$

□

Corollary 4. *Conditions are the same as for Theorem 3. When $g(t) = t$, (10) reduces to the derivative-based trapezoid rule (see [12]) for the classical Riemann integral.*

Proof. It is easy to obtain

$$\begin{aligned}& \int_a^b \int_a^t x\,dx\,dt = \frac{1}{6} b^3 - \frac{1}{2} a^2 b + \frac{1}{3} a^3, \\ & \int_a^b \int_a^t \int_a^y x\,dx\,dy\,dt \\ & \quad = \frac{1}{24} b^4 - \frac{1}{4} a^2 b^2 + \frac{1}{3} a^3 b - \frac{1}{8} a^4, \\ & \int_a^b \int_a^t \int_a^z \int_a^y x\,dx\,dy\,dz\,dt \\ & \quad = \frac{1}{120} b^5 - \frac{1}{12} a^2 b^3 + \frac{1}{6} a^3 b^2 - \frac{1}{8} a^3 b + \frac{1}{30} a^5.\end{aligned} \tag{16}$$

By Theorem 3,

$$\begin{aligned}\int_a^b f(t)\,dg = {} & \int_a^b f(t)\,dt \\ & = \Bigg(\frac{6}{(b-a)^2} \int_a^b \int_a^t x\,dx\,dt \\ & - \frac{12}{(b-a)^3} \int_a^b \int_a^t \int_a^y x\,dx\,dy\,dt - a \Bigg) f(a)\end{aligned}$$

$$
\begin{aligned}
&\quad+\left(b-\frac{6}{(b-a)^2}\int_a^b\int_a^t x\,dx\,dt\right.\\
&\qquad\left.+\frac{12}{(b-a)^3}\int_a^b\int_a^t\int_a^y x\,dx\,dy\,dt\right)f(b)\\
&\quad+\left(\frac{2}{b-a}\int_a^b\int_a^t x\,dx\,dt\right.\\
&\qquad\left.-\frac{6}{(b-a)^2}\int_a^b\int_a^t\int_a^y x\,dx\,dy\,dt\right)f'(a)\\
&\quad+\left(\frac{4}{b-a}\int_a^b\int_a^t x\,dx\,dt\right.\\
&\qquad-\frac{6}{(b-a)^2}\int_a^b\int_a^t\int_a^y x\,dx\,dy\,dt\\
&\qquad\left.-\int_a^b t\,dt\right)f'(b)\\
&\quad+\left(\frac{(b-a)^2}{12}\int_a^b\int_a^t x\,dx\,dt\right.\\
&\qquad-\frac{b-a}{12}\int_a^b\int_a^t\int_a^y x\,dx\,dy\,dt\\
&\qquad\left.+\int_a^b\int_a^t\int_a^z\int_a^y x\,dx\,dy\,dz\,dt\right)f^{(4)}(\xi)\\
&=\frac{b-a}{2}\left(f(a)+f(b)\right)\\
&\quad+\frac{(b-a)^2}{12}\left(f'(a)-f'(b)\right)\\
&\quad+\frac{(b-a)^5}{720}f^{(4)}(\xi).
\end{aligned}
\tag{17}
$$

□

Remark 5. From Corollary 4, we know that the results in Theorem 3 possess the reducibility. When $g(t) = t$, formula (10) reduces to the derivative-based trapezoid rule for the classical Riemann integral. So Theorem 3 is a reasonable generalization of the results in [12].

4. Conclusions

We briefly summarize our main conclusions in this paper as follows.

(1) The derivative-based trapezoid rule for the Riemann-Stieltjes integral is presented which uses 2 derivative values at the endpoints.

(2) This kind of quadrature rule obtains an increase of two orders of precision over the trapezoid rule for the Riemann-Stieltjes integral.

(3) The error term for Riemann-Stieltjes derivative-based trapezoid rule is investigated. And the rationality of the generalization of derivative-based trapezoid rule for Riemann-Stieltjes integral is demonstrated.

The derivative-based midpoint and Simpson's rules for the Riemann-Stieltjes integral will be achieved by further research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the Scientific Research Foundation of Civil Aviation University of China (no. 2013QD01X), the Fundamental Research Funds for the Central Universities (no. 3122014C023), and the National Natural Science Foundation of China (no. 71171190).

References

[1] R. A. Gordon, *The Integrals of Lebesgue, Denjoy, Perron, and Henstock*, vol. 4 of *Graduate Studies in Mathematics*, American Mathematical Society, Providence, RI, USA, 1994.

[2] P. Billingsley, *Probability and Measure*, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, New York, NY, USA, 3rd edition, 1995.

[3] L. Egghe, "Construction of concentration measures for general Lorenz curves using Riemann-Stieltjes integrals," *Mathematical and Computer Modelling*, vol. 35, no. 9-10, pp. 1149–1163, 2002.

[4] P. E. Kopp, *Martingales and Stochastic Integrals*, Cambridge University Press, Cambridge, UK, 1984.

[5] W. Rudin, *Functional Analysis*, International Series in Pure and Applied Mathematics, McGraw-Hill, New York, NY, USA, 2nd edition, 1991.

[6] D. H. Bailey and J. M. Borwein, "High-precision numerical integration: progress and challenges," *Journal of Symbolic Computation*, vol. 46, no. 7, pp. 741–754, 2011.

[7] K. E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1989.

[8] R. L. Burden and J. D. Faires, *Numerical Analysis*, Brooks/Cole, Boston, Mass, USA, 9th edition, 2011.

[9] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, John Wiley & Sons, New York, NY, USA, 1966.

[10] P. R. Mercer, "Hadamard's inequality and trapezoid rules for the Riemann-Stieltjes integral," *Journal of Mathematical Analysis and Applications*, vol. 344, no. 2, pp. 921–926, 2008.

[11] P. R. Mercer, "Relative convexity and quadrature rules for the Riemann-Stieltjes integral," *Journal of Mathematical Inequalities*, vol. 6, no. 1, pp. 65–68, 2012.

[12] C. O. E. Burg, "Derivative-based closed Newton-Cotes numerical quadrature," *Applied Mathematics and Computation*, vol. 218, no. 13, pp. 7052–7065, 2012.

[13] W. Zhao and H. Li, "Midpoint derivative-based closed Newton-Cotes quadrature," *Abstract and Applied Analysis*, vol. 2013, Article ID 492507, 10 pages, 2013.

[14] Z. Kalogiratou and T. E. Simos, "Newton-Cotes formulae for long-time integration," *Journal of Computational and Applied Mathematics*, vol. 158, no. 1, pp. 75–82, 2003.

[15] T. E. Simos, "New closed Newton-Cotes type formulae as multilayer symplectic integrators," *Journal of Chemical Physics*, vol. 133, no. 10, Article ID 104108, 2010.

[16] T. E. Simos, "New open modified Newton Cotes type formulae as multilayer symplectic integrators," *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 1983–1991, 2013.

[17] T. E. Simos, "Closed Newton-Cotes trigonometrically-fitted formulae for the solution of the Schrödinger equation," *MATCH. Communications in Mathematical and in Computer Chemistry*, vol. 60, no. 3, pp. 787–801, 2008.

[18] T. E. Simos, "Closed Newton-Cotes trigonometrically-fitted formulae of high order for the numerical integration of the Schrödinger equation," *Journal of Mathematical Chemistry*, vol. 44, no. 2, pp. 483–499, 2008.

[19] I. Alolyan and T. E. Simos, "New open modified trigonometrically-fitted Newton-Cotes type multilayer symplectic integrators for the numerical solution of the Schrödinger equation," *Journal of Mathematical Chemistry*, vol. 50, no. 4, pp. 782–804, 2012.

[20] T. E. Simos, "High order closed Newton-Cotes exponentially and trigonometrically fitted formulae as multilayer symplectic integrators and their application to the radial Schrödinger equation," *Journal of Mathematical Chemistry*, vol. 50, no. 5, pp. 1224–1261, 2012.

[21] T. E. Simos, "Accurately closed Newton-Cotes trigonometrically-fitted formulae for the numerical solution of the Schrödinger equation," *International Journal of Modern Physics C*, vol. 24, no. 3, Article ID 1350014, 20 pages, 2013.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 184540, 10 pages
http://dx.doi.org/10.1155/2014/184540

Research Article

Geometric Generalisation of Surrogate Model-Based Optimisation to Combinatorial and Program Spaces

Yong-Hyuk Kim,[1] Alberto Moraglio,[2] Ahmed Kattan,[3] and Yourim Yoon[4]

[1] *Department of Computer Science and Engineering, Kwangwoon University, Nowon-gu, Seoul 139-701, Republic of Korea*
[2] *Department of Computer Science, Streatham Campus, University of Exeter, Exeter EX4 4QF, UK*
[3] *Computer Science Department, Umm Al-Qura University, Makkah 21955, Saudi Arabia*
[4] *Department of Computer Engineering, Gachon University, Seongnam-si, Gyeonggi-do 461-701, Republic of Korea*

Correspondence should be addressed to Yourim Yoon; yryoon@gachon.ac.kr

Received 14 March 2014; Accepted 31 March 2014; Published 29 April 2014

Academic Editor: Ker-Wei Yu

Surrogate models (SMs) can profitably be employed, often in conjunction with evolutionary algorithms, in optimisation in which it is expensive to test candidate solutions. The spatial intuition behind SMs makes them naturally suited to continuous problems, and the only combinatorial problems that have been previously addressed are those with solutions that can be encoded as integer vectors. We show how radial basis functions can provide a generalised SM for combinatorial problems which have a geometric solution representation, through the conversion of that representation to a different metric space. This approach allows an SM to be cast in a natural way for the problem at hand, without ad hoc adaptation to a specific representation. We test this adaptation process on problems involving binary strings, permutations, and tree-based genetic programs.

1. Introduction

Some optimisation problems have objective functions which are prohibitively expensive to evaluate [1, 2]. Functions may be mathematically ill behaved (e.g., discontinuous, nonlinear, or nonconvex) or even a black box with largely unknown characteristics. Many engineering design problems have functions of this type [3, 4] and require experiments, lengthy simulations or both, to evaluate the extent to which the design objectives are met by a function of parameters controlling the design. In the jargon of evolutionary computation, these controlling parameters are the genotype that encodes the design (i.e., the phenotype) which has to be expressed by means of an expensive simulation (i.e., a fitness evaluation).

Optimisation methods based on surrogate models (SMs), also known as response surface models, can tackle this problem of expensive objective functions [5–7]. A survey of surrogate model-based optimisation (SMBO) methods can be found elsewhere [8]. An SM is an easily evaluated mathematical model that approximates an expensive objective function as precisely as possible. Inside knowledge of the objective function is not necessary to construct an SM, which is solely built from discrete evaluations of the expensive objective function. We refer to a pair of a candidate solution and its known objective function value as a data-point. Many simple problems have solutions which are real numbers, and perhaps the simplest example of an SM is piecewise-linear interpolation, which creates a function from data-points by linking them with straight-line segments. More useful SMs for solutions on the real line are polynomial interpolants, which have a continuous differential. These and other methods of building SMs naturally extend to spatial interpolation and regression.

The usual SMBO procedure [8] is given in Algorithm 1. An initial SM is constructed from a few solutions of the expensive objective function. Further evaluations are applied to candidate solutions which the SM predicts to be promising. Subsequently, the processes of searching the SM to obtain an optimum set of solutions, evaluation of the solutions using the expensive objective function, and update of the SM with the new data-points are repeated. An evolutionary algorithm can be used in the SMBO procedure to infer the location of

```
(1) Sample uniformly at random a small set of candidate solutions and evaluate them using the
    expensive objective function (initial set of data-points)
(2) while a limit on the number of expensive function evaluations has not been reached do
(3)     Construct a new surrogate model (SM) using all data-points available
(4)     Determine the optimum value of the SM by search, for example, using an evolutionary algorithm
        (this is feasible as the model is cheap to evaluate)
(5)     Evaluate the solution which optimises the SM using the expensive objective function
        (making an additional data-point available)
(6) end while
(7) Return the best solution found
```

Algorithm 1: Surrogate model-based optimisation (SMBO).

a promising set of solutions using the SM, rather than having to evaluate the expensive objective function. This is feasible because the computational cost of a complete run of the evolutionary algorithm on the SM is negligible (in the order of few seconds) with regard to the cost of evaluating a solution using the expensive objective function of the problem (in the order of minutes, hours, or even days depending on the problem).

Virtually all SMs are implicitly or explicitly spatial models, and the prediction process involves exploiting some assumed spatial relations (e.g., a smooth curve of surface) between the values of the objective function at a query point and those at the known data-points. This makes SMBOs naturally suited to continuous optimisation problems. However they are not obviously applicable to combinatorial optimisation problems, except those with solutions which are naturally represented as vectors of integers, when a discretized SM may be used. When each solution is a vector, an integer, or a real number, techniques for building SMs from data-points can be borrowed from statistics (e.g., multivariate regression [9]) or from machine learning (e.g., supervised learning by neural networks or support vector machines [10–12]).

There is increasing interest in optimisation problems with solutions with complicated representations which also have expensive objective functions. For example, permutations and related representations are natural representations of solutions to many scheduling problems. But a candidate schedule may have to be tested by simulating an entire production process, making the SMBO approach very attractive. However, although a permutation can be regarded as a special type of vector, permutations cannot be treated in the same way, because the information they encode is in the order of the elements, not their values. This makes the standard SMBO approach unsuitable.

Variable-length sequences occur in many bioinformatics problems [13], and an SMBO can be used to select biological sequences for detailed study or simulation at an atomic level: an example is the search for proteins with desired properties.

Genetic programming (GP) [14] normally operated on a tree representation of a problem, and a number of its well-known applications have expensive objective functions. For example, genetic programs can be used to encode a robot's behavioral controller, which may need to be tested repeatedly in a virtual or real environment to assess how good it is at controlling the robot in performing a task such as wall-following or obstacle avoidance [15].

Let us summarize current situation of SM with regard to solution representations. Evolutionary algorithms and other search algorithms have been widely used to optimise SMs for continuous spaces [16]. More recent work [17] has considered vector solutions. Other studies [18] have approached applications with expensive objective functions which are inherently combinatorial problems with structured solutions (e.g., graphs) by encoding solutions in vector form to allow the use of standard SMs. Evolutionary algorithms have also been used to train, rather than search, the SM using the known data-points [19]; in the approach, GP performs symbolic regression to obtain the vector-input function which best fits the data-points.

Apart from the recent initial work of the present authors [20, 21], SMs do not seem to have been defined *directly* on more complicated representations than vectors. In order to use SMs on search problems with structured representations, the state of the art is to shoe-horn the original representation into a vector form in a preprocessing phase, known as feature extraction in the machine learning literature [22]. There are a number of drawbacks to this approach. For a start, feature extraction is a very delicate task. Only a carefully chosen vector of features will be a good representation of the information relevant to a learning task. Secondly, the unnatural encoding of a solution in vector form introduces extra nonlinearity into an already expensive objective function, making it harder to learn and consequently requiring additional expensive function evaluations to approximate it well enough to locate the optimum solution. In addition, the extraction of features from structured representations such as GP trees is itself unnatural and hence ineffective. For example, a symbolic regression formula or a Boolean formula would appear to have no obvious mapping to a fixed-length vector.

The underlying difficulty is that of making a problem fit the format of current SMs. Surely is it better to modify the SM to accommodate the problem? Or is there some way to modify satisfactory SMs to accept more complicated solution representations?

We recently [20, 21] answered these questions by generalizing a well-known class of SMs—radial basis function networks [23]—using a geometric framework [24–27] which had previously been used to generalize search algorithms, such as particle swarm optimisation and differential evolution, from continuous spaces to combinatorial spaces. The generalization method is conceptually simple. Firstly, an algorithm which operated in a continuous space is rewritten in terms of Euclidean distances between points. Many spatial algorithms can be rewritten in this way. Then Euclidean distance is replaced with a generic distance metric, which yields a formally well-defined algorithm. This algorithm can be adapted to any solution representation by specifying an appropriate distance metric for that representation.

An algorithm generalised using this geometric methodology can readily be adapted to complicated representations because many types of structured object admit natural relations of distance or similarity. In particular *edit distances* are well suited to structured objects. The edit distance between two configurations is the minimum number of unit edit operations required to transform one of them into the other. For example, hamming distance is an edit distance between binary strings based on the unit edit of a bit flip. For permutations, another metric is swap distance, which is the minimum number of binary exchanges of elements required to transform one permutation into the other. For variable-length sequences, Levenshtein distance measures the minimum number of insertions, deletions, or changes of characters required to transform one sequence into the other. There are also edit distances defined on trees and graphs, based on modifications of edges and nodes.

In the remainder of this paper, we first review how radial basis function networks [23] can be generalised to a range of solution representations using this geometric methodology. We will show how the resulting generalised models can be linked to a target representation using an appropriate distance metric and then used within an SMBO to optimise problems on the target representation. We will illustrate the derivation of SMBOs for three target representations: binary strings, permutations, and GP trees. All our test problems are assumed to have costly objective functions. We use hamming distance as the metric for binary strings and test the resulting SMBO on the well-known NK-landscapes [28] problem. We use hamming distance and swap distance with permutations and test the SMBO on the quadratic assignment problem [29]. We use a form of tree edit distance with GP trees and address standard GP benchmarks of symbolic regression and parity. We should be clear that we are not aiming to show that a generalised SMBO can replace expensive objective functions with structured representations in solving practical problems, but to demonstrate that generalised SMBOs can be in principle applied to such problems, and that it provides meaningful results when applied to classic example problems in simple discrete spaces, which is itself a large conceptual leap.

2. Radial Basis Function Networks

The machine learning literature [22] contains a number of approaches to problems of finding a function in a certain class that best interpolates a set of the data-points which are naturally cast in terms of Euclidean distances, which could readily be generalised to other metric spaces, by replacing Euclidean distance with some metric. These methods include nearest-neighbor regression, inverse distance-weighted interpolation, radial basis function network interpolation, and Gaussian process regression (also known as kriging). The first two methods are relatively simple but they cannot be used as SMs because the global optimum of the functions created from the data-points coincides with a data-point used in the construction of these functions and these methods never provide better solutions than any of the data-points. Gaussian process regression [30] is a very powerful method with a solid theoretical foundation, which can not only extrapolate a global optimum but also give it an interval of confidence. Radial basis function network interpolation is similar to Gaussian process regression but conceptually simpler. We focus on radial basis function networks (RBFNs) and leave the generalization of Gaussian process regression for future work.

2.1. Classic RBFNs.

2.1. Classic RBFNs. A radial basis function (RBF) is a real-valued function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ whose value depends only on the distance from some point $\mathbf{c}$, called its *center*, so that $\phi(\mathbf{x}) = \phi(\|\mathbf{x}-\mathbf{c}\|)$. The point $\mathbf{c}$ is an argument of the function. The norm is usually Euclidean, so $\|\mathbf{x}-\mathbf{c}\|$ is Euclidean distance between $\mathbf{c}$ and $\mathbf{x}$, but other norms are possible and have been used. Commonly used RBFs include Gaussian functions, multiquadrics, poly-harmonic splines, and thin-plate splines. The most frequently used are Gaussian functions of the form:

$$\phi(\mathbf{x}) = \exp\left(-\beta\|\mathbf{x}-\mathbf{c}\|^2\right), \tag{1}$$

where $\beta > 0$ is the width parameter.

RBFs are typically used to build function approximations of the form:

$$y(\mathbf{x}) = w_0 + \sum_{i=1}^{N} w_i\ \phi\left(\|\mathbf{x}-\mathbf{c}_i\|\right). \tag{2}$$

The approximating function $y(\mathbf{x})$ is thus the sum of N RBFs, each associated with its own center $\mathbf{c}_i$, width β_i, and weighted by a coefficient w_i and there is a bias term w_0. Figure 1 shows an example of a function obtained in this way. Any continuous function can in principle be approximated with arbitrary accuracy by such a sum, if enough RBFs are used.

In an RBFN, there are three types of parameters that need to be determined to optimise the fit between $y(\mathbf{x})$ and the data: the weights w_i, the centers $\mathbf{c}_i$, and the width parameters β_i. The most common way to find these parameters has two phases. Firstly, unsupervised learning (i.e., clustering) is used to determine the position of the centers and the widths of the RBFs. Then, the weights w_i that optimise the fit are obtained by least-squares minimisation.

A simplified procedure for fitting an RBFN, which skips the unsupervised learning phase, is widely used. The centers $\mathbf{c}_i$ are first chosen to coincide with the known points $\mathbf{x}_i$. Then the widths β_i are determined by a heuristic based on the distance of each center $\mathbf{c}_i$ to the nearest neighbors (local model) or all widths are set to the same value, which is chosen in relation to the maximum distance between any two centers

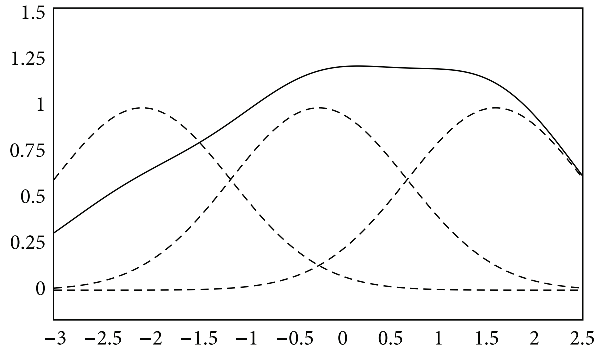


Figure 1: Example of a function (solid line) obtained as a weighted sum of three Gaussian functions (dashed lines) on the real line: the weighting factors w_1, w_2, and w_3 are 0.5, 1, and 1, respectively.

(global model). The bias w_0 can either be set to the mean of the function values b_i at the known data-points (i.e., training set), or to 0. The weights w_i are then determined by solving the system of N simultaneous linear equations in w_i which express the requirement that the function interpolates the data-points:

$$y(\mathbf{x}_i) = b_i, \quad i = 1 \cdots N. \tag{3}$$

Setting $g_{ij} = \phi(\|\mathbf{x}_j - \mathbf{x}_i\|)$, the system can be written in matrix form as $G\mathbf{w} = \mathbf{b}$. The matrix G is nonsingular if the points $\mathbf{x}_i$ are distinct and the family of functions ϕ is positive definite (which is the case for Gaussian functions), and thus the weights $\mathbf{w}$ can be obtained by simple linear algebra:

$$\mathbf{w} = G^{-1}\mathbf{b}. \tag{4}$$

2.2. Generalization of RBFNs to Arbitrary Representations. To generalize RBFNs, we need to generalize (i) the class of functions used to approximate the unknown function, (ii) the training procedure which finds the function within that class that best fits the data-points, and (iii) the model query procedure that predicts the value of the unknown function at a query point.

Following the geometric methodology of our generalization, we first need to rewrite each of the above three elements as a function of Euclidean distance alone and then substitute a distance metric which is chosen to suit the target representation. Finally we rewrite the algorithm in terms of that distance to obtain an instance of that algorithm specific to the target representation.

Let $\mathbf{M}$ be a metric space associated with a distance function d. An RBF $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ whose value depends only on the distance from some point $\mathbf{c} \in \mathbb{R}^n$, so that $\phi(\mathbf{x}) = \phi(\|\mathbf{x}-\mathbf{c}\|)$, can be generalised to a function $\phi : \mathbf{M} \rightarrow \mathbb{R}$ whose value depends only on the distance from some point $\mathbf{c} \in M$ *in the metric space*, so that $\phi(\mathbf{x}) = \phi(d(\mathbf{x}, \mathbf{c}))$. For example, generalised Gaussian functions can be obtained by replacing Euclidean distance with the generic metric d in the original definition, so that $\phi(\mathbf{x}) = \exp(-\beta d(\mathbf{x}, \mathbf{c})^2)$.

A set of configurations and an associated edit distance comprise a metric space, as all edit distances meet the metric axioms [27, 31, 32]. Consequently, a generalised RBF is well-defined on any set of configurations, making it a *representation-independent function*. For example, the set of binary strings $\mathbf{H}$ and hamming distance d_H form a metric space. If hamming distance d_H is used as the metric d, then generalised Gaussian functions become well-defined functions $\phi : \mathbf{H} \rightarrow \mathbb{R}$, which map binary strings to real numbers. Note that both $\mathbf{c}$ and $\mathbf{x}$ are binary strings. Alternatively, if the swap distance on permutations replaces the metric d, then these generalised Gaussian functions become well-defined functions mapping permutations to real numbers.

The SM $y(\mathbf{x})$, which is a linear combination of RBFs, can be generalised to a linear combination of generalised RBFs: $y(\mathbf{x}) = w_0 + \sum_{i=1}^{N} w_i \ \phi(d(\mathbf{x}, \mathbf{c}_i))$. Like its components, the generalised SM is representation independent and it can be applied to any solution representation by replacing the metric d with a metric appropriate to the target representation. An SM is generalized in this way of parameterizing many functions on general metric spaces economically in terms of $\mathbf{c}_i$, w_i, and β_i. This property is independent of the underlying representation. When the underlying metric space is finite as it is in combinatorial optimisation, any function can be approximated with arbitrary accuracy by a sufficiently large number of RBFs. In the limit, every point in space would be associated with an RBF, parameterised to fit the function value exactly.

The SM is fitted to the known data-points without reference to their underlying representation but solely in terms of the distances between data-points and the objective function values b_i. Therefore the fitting process is representation independent, like the model. In particular, a simplified model-fitting procedure can obtain the centers, widths, and weights by least-squares minimisation of the system $G\mathbf{w} = \mathbf{b}$. However, when the distance function d is not embeddable in Euclidean space, the RBFs are no longer necessarily positive definite, and neither is the matrix G, and hence the inverse matrix G^{-1} needed to determine the weights w_i, may not exist. This difficulty can be overcome by using the pseudoinverse of G, which always exists, is unique, and corresponds to G^{-1} when that exists. It can also be shown that the weights w_i determined by solving the system $G\mathbf{w} = \mathbf{b}$ using the pseudoinverse are the same as those obtained by least-squares minimisation. This way of generalizing RBFNs to structured representations is related to kernel methods in machine learning. However, in those methods, the types of distances to be used between objects can be difficult to design, because they must be implicitly embedded in a vector space (i.e., positive-definite kernels), which is not necessary for our approach.

3. Experiments on Binary Strings

Binary strings are of course a special type of vector. However, they can illustrate the application of generalised SMBOs to combinatorial spaces because their property of being vectors is not utilised. We experimented with the well-known NK-Landscape problem [28], which provides a tunable set of rugged, epistatic landscapes over a space of binary strings, and we consider it to have a costly objective function. We evaluated the SMBO algorithm with landscapes of size $n = 10, 15, 20, 25$, each for $k = 2, 3, 4, 5$.

We used a standard SMBO algorithm (Algorithm 1). The SM is an RBFN model fitted to the data-points using the simplified learning procedure presented in the previous section. The centers $\mathbf{c}_i$ of the RBFs are the data-points. The widths β_i of the RBFs are all set to $1/2D^2$, where D is the maximum distance between any two centers. Thus each RBF extends over all the centers, allowing the known function value at each center to contribute to the prediction of the function value at any point in the landscape near the given center. The value of the bias term w_0 is set to the average of the function values b_i at all the known data-points. Thus the SM returns this value at any point outside the influence of all centers. The coefficients w_i are determined by least-squares minimisation, as described in the previous section.

We set other parameters as a function of the problem size n. Our aim is to find the best solution to this problem with 2^n candidate solutions in quadratic time; that is, we set the number of allowable expensive function evaluations to n^2. Initially, 2 data-points are sampled, and $n^2 - 2$ sample points are suggested by the SM. To search the SM, we use a standard generational evolutionary algorithm with tournament selection with a tournament size of 2, uniform crossover at a rate of 0.5, and bitwise mutation at a rate of $1/n$. The population size and the number of generations are both set to $10n$. If the predicted value of the best solution found by the SM is better than the best value at any of the known data-points, then the model could extrapolate from the data, and that solution is evaluated using the expensive objective function. Otherwise, a point is chosen at random and evaluated with the expensive objective function in an attempt to gather more data about undersampled regions.

We compared SMBO with random search (RS), a standard (1 + 1) evolutionary algorithm ((1 + 1) EA), and a generational evolutionary algorithm (EA), all using the expensive objective function directly. We expect evolutionary algorithms to outperform random search, but we include the latter as it can do well with small samples. We allowed all the algorithms n^2 evaluations of the expensive objective function.

The (1 + 1) EA has a population of a single individual and uses bitwise mutation with a bit-flip probability of $1/n$. EA has a population of n individuals, runs for n generations, and uses tournament selection with tournament size 2, bitwise mutation with a bit-flip probability of $1/n$, and uniform crossover at a rate of 0.5. For each of the 16 combinations of n and k, we generated a single fitness landscape and ran all for algorithms 50 times each. We also estimated the global optimum using an evolutionary algorithm with 1,000 individuals and 1,000 generations.

Table 1 shows that, for each combination of n and k, SMBO consistently found the best solution and the best average solution. Furthermore, in 12 out of 16 cases, SMBO was able to find the estimated real optimum. As the problem size n increases, the differential in favor of SMBO increases. As expected, as the ruggedness k of the problem increases, search algorithms get less close to the estimated real optimum. As for the other algorithms in the comparison, the population-based EA generally did better than (1 + 1) EA and RS, especially on larger problems. Perhaps surprisingly, RS often did better than (1 + 1) EA. It seems that (1 + 1) EA can easily get trapped at local optima, especially when the sample and problem sizes are large.

4. Experiments on Permutations

This section greatly extends our previous work [21]. Experiments were carried out on six standard quadratic assignment problems (QAPs), kra30a, kra32, lipa30a, nug30, ste36a, and tho30 (where the number in the name indicates the problem size), and on two instances of a unimodal problem on permutations of size 30, in which the fitness of the permutation, to be minimised, is given by its distance to some fixed permutation. This unimodal problem can be seen as a generalization of the OneMax problem for binary strings [33], in which the fitness of a solution is the number of 1*s* in the string. This is in turn equivalent to a problem in which the fitness of a solution is given by hamming distance from the solution to the string with all bits set to 1. From the symmetry of hamming space, this problem is again equivalent to any problem in which a string with all bits set to one is to be replaced with some target string. The two instances of the unimodal problem are obtained by using two different distance functions on permutations, hamming distance (unih30), and swap distance [27] (unis30). We address this unimodal problem to test the SMBO on a fitness landscape with an explicit and visible topography. We will consider the problems in the test-bed as having costly objective functions and leave as future work testing the SMBO on real-world problems with expensive objective functions. Furthermore, using a larger test-bed and testing the scalability of SMBO with respect to instance size would be desirable. However, we found that it would take an excessive amount of time, as the SM is searched every time it is used to suggest a solution to test in the expensive objective function. We will also consider a larger test-bed and a scalability analysis in future work.

The algorithm that uses hamming distance is called SMBO_H and the algorithm using Swap distance is called SMBO_S. Clearly, the choice of a distance well suited to the problem at hand is crucial to obtain an SM able to make meaningful predictions and guide appropriately the search of the SMBO. In this paper, we limit ourselves to experiment with these two distances. In future work, we will investigate other distances and other problems in the attempt to find out a rule to select *a priori* a good distance for a given type of problem.

As in the previous section, that is, binary strings, we used a standard SMBO algorithm (Algorithm 1) with an RBFN model which is fitted to the available data-points using the simplified learning procedure. For SMBO_H, all the RBFs have the same widths $\beta = 1/2D^2$, where D is the maximum distance across all centers. However, this setting did not work well for SMBO_S, and we found that $\beta = 1/(D/5)$ produced better results. The value of β greatly affects the accuracy of the predictions of the SM. So, it needs to be tuned but might in

Table 1: Results for SMBO, an evolutionary algorithm (EA), a (1 + 1) evolutionary algorithm ((1 + 1) EA), and random search (RS) on the NK-landscape benchmark for all combinations of $k = 2, 3, 4, 5$ and $n = 10, 15, 20, 25$.

k (optimum)	SMBO		EA		(1 + 1) EA		RS	
	Best	Average	Best	Average	Best	Average	Best	Average
				$n = 10$				
2 (0.704)	**0.704**	0.702	**0.704**	0.686	*0.698*	0.675	**0.704**	0.649
3 (0.794)	**0.794**	0.775	**0.794**	0.724	*0.745*	0.705	**0.794**	0.724
4 (0.787)	**0.787**	0.755	**0.787**	0.725	**0.787**	0.714	**0.787**	0.727
5 (0.810)	**0.810**	0.762	*0.727*	0.706	**0.810**	0.729	**0.810**	0.718
				$n = 15$				
2 (0.743)	**0.743**	0.742	*0.714*	0.693	0.681	0.628	*0.714*	0.674
3 (0.738)	**0.738**	0.718	0.706	0.678	0.706	0.622	*0.717*	0.677
4 (0.747)	**0.747**	0.721	*0.711*	0.685	0.705	0.646	0.710	0.680
5 (0.760)	**0.758**	0.737	0.749	0.711	0.728	0.672	*0.757*	0.700
				$n = 20$				
2 (0.729)	**0.729**	0.726	*0.718*	0.689	0.668	0.613	0.711	0.673
3 (0.777)	**0.777**	0.767	*0.761*	0.718	0.639	0.606	**0.777**	0.706
4 (0.775)	**0.775**	0.747	*0.731*	0.708	0.676	0.640	0.707	0.684
5 (0.766)	**0.761**	0.744	*0.745*	0.710	0.709	0.637	0.721	0.684
				$n = 25$				
2 (0.753)	**0.753**	0.747	*0.727*	0.698	0.679	0.590	0.701	0.673
3 (0.798)	**0.798**	0.781	0.742	0.727	0.666	0.607	*0.749*	0.698
4 (0.775)	**0.762**	0.743	*0.750*	0.714	0.639	0.595	0.695	0.679
5 (0.774)	**0.756**	0.736	*0.751*	0.713	0.705	0.622	0.722	0.676

*The best (maximum) and average values of the best solution found by each algorithm are given for 50 runs.

†Bold numbers are the highest maxima and italic numbers are the second highest maxima.

future be "learnt" from the sampled data-point. The value of the bias term w_0 is set to the average $b_i s$ of the known data-points, and coefficients w_i of the RBFs are determined by least-squares minimisation, as in the previous section.

The other settings of the SMBO are as follows. For all problems, the allowance n of expensive function evaluations is set to 100. Initially, 10 data-points are sampled, and the number of sample points suggested by the SM is $n - 10 = 90$. To search the SM, we used a memetic algorithm on permutations with truncation selection, cycle crossover, swap mutation at a mutation rate of 0.01, and local search based on the 2-opt neighborhood. The population size and the number of generations were both set to 20, and 10 new offspring were generated in each generation, allowing an adequate solution to be obtained from the SM. The solution with the best predicted objective value is evaluated with the expensive objective function, provided it has not been sampled before. If it has, the second-best is sampled provided that it has not been sampled before. Otherwise the third best is sampled and so on.

We compared the SMBO algorithms with random search (RS) and a standard genetic algorithm (GA), both using the expensive objective function directly. We allowed all the algorithms the same number of evaluations of expensive objective function.

The GA had a population of 10 individuals ran for 18 generations, with 5 new individuals in each generation. It uses truncation selection, swap mutation with a probability of 0.01, and cycle crossover. We did 50 runs for each algorithm and problem.

The results, presented in Table 2, consistently rank SMBO_H as the most effective algorithm, followed by the GA and SMBO_S, and then random search (RS). Clearly, the SM based on hamming distance is effective and better than swap distance by a surprising margin, considering that hamming and swap distance are closely related. SMBO_H even outperforms SMBO_S on the unimodal landscape under Swap distance (unis30), which we expected to favor SMBO_S.

We performed further analyses to try to understand the mechanism of the SMBO algorithms more fully. Firstly, to make sure that the distance metrics used by the SMBOs are suitable, we did a static analysis of the predictive power of the SMs in isolation from the SMBOs. This analysis is presented in Table 3. Looking at the number of significantly positive correlations and the average correlation, presented in Table 3, it is evident that hamming metric always gives better predictions than swap. Neither metric is deceptive, as there are no negative correlations. Fitness-distance correlations [34] of both distance metrics, given in Table 4, show that swap distance has higher correlations for the QAP, even though this is not reflected in its performance. This suggests that static prediction power gives a better indication whether a distance metric is appropriate for an SMBO.

Other aspects of the SM may affect the performance of an SMBO. For instance, we would obviously prefer to locate the real optimum of the SM before sampling with the expensive

Table 2: Results for random search (RS), a genetic algorithm (GA), SMBO_H, and SMBO_S on QAP instances (kra30a, kra32, lipa30a, nug30, ste36a, and tho30) and unimodal instances (unih30 and unis30) of permutation problems.

Instance	RS			GA			SMBO_H			SMBO_S		
	Best	Average	SD	Best	Average	SD	Best	Average	SD	Best	Average	SD
kra30a	118730	122777.00	2034.86	115840	123445.80	2642.21	**110850**	**119649.60**	3389.70	117270	122027.00	2205.65
kra32	24156	25008.04	434.75	23440	24625.03	586.60	**22590**	**24094.32**	616.71	23848	24833.92	452.22
lipa30a	13664	13710.08	16.38	13646	13704.42	22.53	**13633**	13700.52	21.89	13638	**13696.32**	22.03
nug30	7350	7618.00	84.60	7296	7558.36	126.65	**7276**	**7500.24**	97.36	7328	7563.88	94.38
ste36a	16736	18335.56	602.04	16516	18287.96	841.59	**15364**	**17311.00**	922.20	15654	17840.68	889.74
tho30	190256	196662.82	3211.20	**180274**	194389.91	4414.22	180860	**193415.44**	4629.03	186172	195231.92	3534.21
unih30	24	25.80	0.78	22	25.04	1.26	**17**	**20.84**	1.55	21	25.18	0.95
unis30	19	21.66	0.87	17	20.91	1.29	**15**	**18.48**	1.83	19	21.04	0.94

*The best, average, and standard deviation of the best fitness found by each algorithm are reported for 50 runs.

Table 3: Correlation between predicted and real fitness on a test set of randomly sampled solutions after the SMs have been trained on 50 randomly sampled data-points. The columns contain counts of significantly positive (larger than 0.15) and significantly negative correlations (less than −0.15), together with average correlation coefficients.

Instance	Hamming model		Swap model	
	Pos. (Ave.)	Neg.	Pos. (Ave)	Neg.
kra30a	50 (0.26)	0	31 (0.20)	0
kra32	46 (0.22)	0	34 (0.20)	0
lipa30a	0 (N/A)	0	0 (N/A)	0
nug30	32 (0.18)	0	21 (0.19)	0
ste36a	50 (0.23)	0	27 (0.18)	0
tho30	12 (0.18)	0	7 (0.16)	0
unih30	50 (0.77)	0	50 (0.43)	0
unis30	47 (0.24)	0	39 (0.19)	0

*Each test was repeated 50 times.

Table 4: Fitness-distance correlation for permutation problems using hamming and swap distance.

Instance	Hamming	Swap
kra30a	−0.02	0.31
kra32	−0.11	0.11
lipa30a	0.32	0.41
nug30	0.38	0.44
ste36a	0.07	0.12
tho30	0.46	0.52
unih30	1.00	1.00
unis30	1.00	1.00

Table 5: Fitness-distance correlation for SMs based on hamming and swap distance, after training them with 100 randomly sampled data-points.

Instance	Hamming SM	Swap SM
kra30a	0.85	0.57
kra32	0.88	0.80
lipa30a	0.93	0.39
nug30	0.88	0.71
ste36a	0.81	0.59
tho30	0.82	0.83
unih30	0.86	0.93
unis30	0.87	0.79

Table 6: Number of solutions of permutation problems obtained by optimizing the SM by the SMBO algorithm (90 sequential optimisations) with predicted fitness which is better than or equal to the fitness of the best previous solution.

Instance	SMBO_H		SMBO_S	
	Better	Equal	Better	Equal
kra30a	22.78	0	0.00	0
kra32	22.28	0	0.00	0
lipa30a	33.80	0	0.04	0
nug30	26.70	0	0.02	0
ste36a	29.66	0	0.04	0
tho30	28.16	0	0.04	0
unih30	4.28	0	0.28	0
unis30	19.64	0	0.04	0

*The results are averaged over 50 runs.

objective function and the GA which we actually use to search the SM provides no guarantee of finding of the optimum or even a good solution. How good are the solutions that it finds? Table 5 shows fitness-distance correlations for the SMs, after training with 100 randomly sampled data-points. All these values are extremely high, suggesting that the GA usually locates very good solutions.

Another attribute of the SM that may affect the performance of an SMBO is the effect of the distance metric and the parameter β on the topography of the model. These choices affect the extrapolative property of the model, which allows an optimum value to be found which is higher than that of any data-points. Table 6 shows that SMBO_H can extrapolate much more often than SMBO_S. This may well provide any reason why SMBO_H outperforms SMBO_S. However, the precise merit of hamming distance in this regard remains a subject for future work.

5. Experiments on Genetic Programming

Experiments were carried out on standard GP problems, symbolic regression and parity problems, and a unimodal problem, in which the fitness of a tree (to be minimised) is given by its distance to a given tree. This last problem can again be seen as a generalization of the OneMax problem for binary strings [33].

We have used structural hamming distance [35] as the metric of distance between two GP trees: this is a parameterless variant of the well-known structural distance for GP trees [36].

As in previous sections, we used a standard SMBO with an RBFN model fitted using the simplified learning procedure. The RBFs have the same widths $\beta = 1/2D^2$, where D is the maximum distance across all centers. The value of the bias term w_0 is set to the average function value of the known data-points. The coefficients w_i of the RBFs in the linear model are determined by least-squares minimisation.

We set other parameters as a function of the maximum depth md of the trees in the initial population, which is likely to determine the proportion of the search space that will actually be visited. The maximum number of nodes in a binary tree with a maximum depth md is $2^{\mathrm{md}} - 1$. The number of expensive function evaluations allowed was $n = 2^{\mathrm{md}}$. Thus our aim was to get each algorithm to produce the best solution in a time linearly proportional to the maximum size of the trees in the initial population. We set the initial sample size to 2 data-points and the number of points suggested by the SM to $n - 2$. To search the SM, we use a standard GP with tournament selection using a tournament size of 2, subtree crossover at a rate of 0.8, subtree mutation at a rate of 0.17, and reproduction operator at a rate of 0.03. The population size and the number of generations were both set to n, which we expected to provide GP with enough trials to locate a good solution of the SM. If the predicted value of the best solution found by the SM is better than the best value at any of the known data-points, then the model could extrapolate from the data, and that solution is evaluated using the expensive objective function. Otherwise, a point is chosen at random and evaluated with the expensive objective function in an attempt to gather more data about undersampled regions.

We compare the SMBO algorithm with random search (RS) and a standard GP, both using the expensive objective function directly. We allowed all the algorithms n evaluations of the expensive objective function. The GP used has a population of approximately $\sqrt{n}$ individuals and it runs for approximately $\sqrt{n}$ generations. For fairness, the exact values of these two parameters are assigned in a way that their product is exactly n. It uses tournament selection with a tournament of size 2, subtree mutation with a probability of 0.17, subtree crossover at a rate of 0.8, and reproduction operator at a rate of 0.03. For each problem, we varied the maximum depth md between 3 and 7 and did 50 runs.

The results given in Table 7 make it immediately apparent that all algorithms get better results as md is increased, as we would expect. On the unimodal problem, looking at the average results, SMBO is consistently the best, followed by RS and finally by GP. The unimodal problem has the best fitness distance correlation with structural Hamming distance, suggesting that this metric is well suited for applying SMBO to this problem. This suggests that a good distance metric for SMBO in general should have good fitness-distance correlation for the problem at hand.

Surprisingly, RS does better than GP, which appears not to have had enough fitness evaluations available to get the evolution process properly started, especially when the sample and problem sizes were large. On the parity problem, SMBO wins again but with a smaller margin. Again, GP is worse than RS; however, if it is allowed a larger budget of expensive evaluations (i.e., md = 7), its performance matches RS. But more evaluations improve the performance of SMBO even more. On the symbolic regression problem, RS performs the best and GP the worst, although more evaluations allow SMBO and GP to outperform RS. This suggests that structural hamming distance is not particularly suitable for applying the SMBO to this problem.

There are many possible distances for parse trees we could use as basis for the SMBO. In future work, we should select distances suitable for the problem at hand, that is, that give rise to smoother/more unimodal landscape. In recent, Moraglio et al. [37] introduced a distance for GP, the semantic distance, that turns any GP problems into a unimodal problem. So for future work it could be interesting to use this distance as a base for SMBO.

6. Conclusions and Future Work

New applications are opened up by extending surrogate model-based optimisation (SMBO) to more complicated representations which cannot be naturally mapped to vectors of features. We have put forward a conceptually simple, formal, general, and systematic approach to adapting SMBO using radial basis function (RBF) networks to *any* target representation. Any algorithm that can be written in terms of Euclidean distances between candidate solutions can be generalised by replacing Euclidean distance function with a generic metric appropriate to the target representation (e.g., edit distance). RBF networks can be naturally generalised to encompass any representations because both the approximating model and the learning of the model parameter can be cast completely in a representation-independent way and rely only on distance relations between training instances and query instances.

We have validated experimentally the framework on three representations. First, we have considered the binary strings representation endowed with the hamming distance and tested the SMBO on the NK-landscapes, obtaining consistently that with the same budget of expensive function evaluations, the SMBO performs the best in comparison with other search algorithms. The second representation we have considered is the permutation representation endowed with hamming distance and with swap distance and tested the SMBO on the quadratic assignment problem and on unimodal problems, obtaining consistently that with the same budget of expensive function evaluations, the SMBO with hamming distance performs the best in comparison with other search algorithms. Surprisingly, the SMBO based on swap distance does not work as well as the SMBO based

Table 7: Results for unimodal, parity, and symbolic regression problems obtained by SMBO, random search (RS), and genetic programming (GP).

(a) ⟨Unimodal⟩

md		SMBO			RS			GP	
	Best	Average	SD	Best	Average	SD	Best	Average	SD
3	**0.11**	0.47	0.19	**0.11**	**0.43**	0.16	0.22	0.50	0.18
4	**0.07**	**0.14**	0.05	0.13	0.37	0.13	0.11	0.49	0.25
5	**0.04**	**0.07**	0.03	0.08	0.24	0.08	0.05	0.48	0.21
6	**0.01**	**0.04**	0.04	0.08	0.14	0.08	0.14	0.46	0.21
7	**0.01**	**0.02**	0.04	0.04	0.18	0.04	0.06	0.32	0.20

(b) ⟨4-Odd Parity⟩

md		SMBO			RS			GP	
	Best	Average	SD	Best	Average	SD	Best	Average	SD
3	37.50	**45.00**	6.45	37.50	**45.00**	6.45	37.50	48.75	3.95
4	37.50	**40.00**	5.27	37.50	41.25	6.04	37.50	42.50	6.45
5	37.50	**37.50**	0.00	37.50	**37.50**	0.00	37.50	47.50	5.27
6	37.50	**37.50**	0.00	37.50	**37.50**	0.00	37.50	41.25	6.04
7	**25.00**	**33.75**	6.04	37.50	37.50	0.00	37.50	37.50	0.00

(c) ⟨Symbolic Regression⟩

md		SMBO			RS			GP	
	Best	Average	SD	Best	Average	SD	Best	Average	SD
3	3.44	**4.88**	0.82	3.44	**4.88**	0.78	**2.64**	5.17	1.35
4	4.46	6.35	1.17	**4.27**	**5.78**	1.41	4.46	6.39	1.58
5	3.84	5.58	1.21	**3.51**	**5.18**	1.21	4.05	5.39	1.27
6	**2.95**	3.74	0.73	2.99	**3.52**	3.48	3.48	4.39	0.57
7	**3.45**	**4.50**	0.77	3.81	4.96	0.54	3.71	4.62	0.67

*The best (minimum) and average fitness values for the best solution found by each algorithm, for md = 3, 4, 5, 6, 7, over 50 runs.

on hamming distance. We have presented an analysis in the attempt to elucidate the causes of the different performance. Further investigation is required to pinpoint the structural difference between Hamming distance and Swap distance that gives rise to the performance difference. Lastly, as an experimental validation of the framework on a nontrivial discrete space and structured representation, we have considered the genetic programming (GP) trees endowed with the structural hamming distance and tested the SMBO on a test-bed of standard GP problems, obtaining that with the same budget of expensive function evaluations, the SMBO performs well in a comparison with other search algorithms. These results suggest that our approach has the potential to solve real-world combinatorial optimisation problems with complicated solution representations and nontrivial discrete search spaces.

Much work remains to be done. Firstly, we plan to look at further well-known permutation and GP problems and consider different distance metrics. For instance, the traveling salesman problem may be cast in terms of a distance based on the 2-opt move. Then we intend to consider problems with other complicated nonvectorial representations, such as variable-length sequences. Our eventual aim is to address some challenging real-world problems in a new way. We will also experiment with different types of RBF and more complex learning processes (i.e., learning the centers and the widths of the RBFs). Lastly, we will attempt the generalization of more sophisticated interpolation and regression methods, including Gaussian process regression, which is a state-of-the-art method in machine learning.

Disclosure

A preliminary version of this paper appeared in *Proceedings of the Eleventh European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 142–154, 2011, and in *Proceedings of the Genetic and Evolutionary Computation Conference (Companion Material)*, pp. 133-134, ACM SIGEVO, 2011.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] P. M. Pardalos and M. G. C. Resende, *Handbook of Applied Optimization*, Oxford University Press, 2002.

[2] P. M. Pardalos and E. Romeijn, Eds., *Handbook of Global Optimization. Volume 2*, vol. 62, Kluwer Academic Publishers, 2002.

[3] J.-H. Seo, Y.-H. Kim, H.-B. Ryou, S.-H. Cha, and M. Jo, "Optimal sensor deployment for wireless surveillance sensor networks by a hybrid steady-state genetic algorithm," *IEICE Transactions on Communications*, vol. E91-B, no. 11, pp. 3534–3543, 2008.

[4] S. S. Tong and B. A. Gregory, "Turbine preliminary design using artificial intelligence and numerical optimization techniques," *Journal of Turbomachinery*, vol. 114, no. 1, pp. 1–10, 1992.

[5] H.-M. Gutmann, "A radial basis function method for global optimization," *Journal of Global Optimization*, vol. 19, no. 3, pp. 201–227, 2001.

[6] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[7] R. G. Regis and C. A. Shoemaker, "Constrained global optimization of expensive black box functions using radial basis functions," *Journal of Global Optimization*, vol. 31, no. 1, pp. 153–171, 2005.

[8] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, 2001.

[9] N. A. C. Cressie, *Statistics for Spatial Data*, John Wiley & Sons, New York, NY, USA, 1993.

[10] T. Mitchell, *Machine Learning*, McGraw Hill, 1997.

[11] J.-H. Seo, Y.-H. Lee, and Y.-H. Kim, "Feature selection for very short-term heavy rainfall prediction using evolutionary computation," *Advances in Meteorology*, vol. 2014, Article ID 203545, 15 pages, 2014.

[12] G.-M. Yoon, J. Kim, Y.-H. Kim, and B.-R. Moon, "Performance improvement by genetic feature selection and adjusting ratings'mid-point value in the neural network-based recommendation models," *Advances in Information Sciences and Service Sciences*, vol. 4, no. 11, pp. 37–43, 2012.

[13] A. Moraglio, R. Poli, and R. Seehuus, "Geometric crossover for biological sequences," in *Proceedings of the European Conference on Genetic Programming*, pp. 121–132, 2006.

[14] R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming*, Lulu Enterprises, 2008.

[15] K. Seo, S. Hyun, and E. D. Goodman, "Genetic programming-based automatic gait generation in joint space for a quadruped robot," *Advanced Robotics*, vol. 24, no. 15, pp. 2199–2214, 2010.

[16] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing Journal*, vol. 9, no. 1, pp. 3–12, 2005.

[17] L. Bajer and M. Holena, "Surrogate model for continuous and discrete genetic optimization based on RBF networks," in *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*, pp. 251–258, 2010.

[18] I. Voutchkov, A. J. Keane, A. Bhaskar, and T. M. Olsen, "Weld sequence optimization: the use of surrogate models for solving sequential combinatorial problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 30–33, pp. 3535–3551, 2005.

[19] T. L. Lew, A. B. Spencer, F. Scarpa, K. Worden, A. Rutherford, and F. Hemez, "Identification of response surface models using genetic programming," *Mechanical Systems and Signal Processing*, vol. 20, no. 8, pp. 1819–1831, 2006.

[20] A. Moraglio and A. Kattan, "Geometric generalisation of surrogate model based optimisation to combinatorial spaces," in *Proceedings of the 11th European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 142–154, 2011.

[21] A. Moraglio, Y.-H. Kim, and Y. Yoon, "Geometric surrogate-based optimisation for permutation-based problems," in *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO '11)*, pp. 133–134, July 2011.

[22] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2007.

[23] L. C. Jain, *Radial Basis Function Networks*, Springer, 2001.

[24] A. Moraglio, *Towards a Geometric Unification of Evolutionary Algorithms [Ph.D. thesis]*, University of Essex, 2007.

[25] A. Moraglio, Y.-H. Kim, Y. Yoon, and B.-R. Moon, "Geometric crossovers for multiway graph partitioning," *Evolutionary Computation*, vol. 15, no. 4, pp. 445–474, 2007.

[26] Y. Yoon and Y.-H. Kim, "Geometricity of genetic operators for real-coded representation," *Applied Mathematics and Computation*, vol. 219, no. 23, pp. 10915–10927, 2013.

[27] Y. Yoon, Y.-H. Kim, A. Moraglio, and B.-R. Moon, "Quotient geometric crossovers and redundant encodings," *Theoretical Computer Science*, vol. 425, pp. 4–16, 2012.

[28] S. Kauffman, *Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, 1993.

[29] Y.-H. Kim and Y. Yoon, "A new Kernighan-Lin-type local search for the quadratic assignment problem," in *Proceedings of the International Conference on Scientific Computing*, pp. 185–189, 2009.

[30] C. E. Rasmussen, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[31] Y.-H. Kim and B.-R. Moon, "New topologies for genetic search space," in *Genetic and Evolutionary Computation Conference (GECCO '05)*, pp. 1393–1399, June 2005.

[32] Y. Yoon and Y.-H. Kim, "A Mathematical Design of Genetic Operators on $GL_n(Z_2)$," *Mathematical Problems in Engineering*, vol. 2014, Article ID 540936, 8 pages, 2014.

[33] J. D. Schaffer and L. J. Eshelman, "On crossover as an evolutionary viable strategy," in *Proceedings of the 4th International Conference on Genetic Algorithms*, pp. 61–68, 1991.

[34] T. Jones and S. Forrest, "Fitness distance correlation as a measure of problem difficulty for genetic algorithms," in *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 184–192, 1995.

[35] A. Moraglio and R. Poli, "Geometric landscape of homologous crossover for syntactic trees," in *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC '05)*, pp. 427–434, September 2005.

[36] A. Ekart and S. Z. Nemeth, "A metric for genetic programs and fitness sharing," in *Proceedings of the European Conference on Genetic Programming*, pp. 259–270, 2000.

[37] A. Moraglio, K. Krawiec, and C. G. Johnson, "Geometric semantic genetic programming," in *Proceedings of the 12th International Conference on Parallel Problem Solving from Nature*, pp. 21–31, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 257947, 11 pages
http://dx.doi.org/10.1155/2014/257947

Research Article

Hybrid Soft Computing Schemes for the Prediction of Import Demand of Crude Oil in Taiwan

Yuehjen E. Shao,[1] Chi-Jie Lu,[2] and Chia-Ding Hou[1]

[1] *Department of Statistics and Information Science, Fu Jen Catholic University, Xinzhuang, New Taipei 24205, Taiwan*
[2] *Department of Industrial Management, Chien Hsin University of Science and Technology, Zhongli, Taoyuan 32097, Taiwan*

Correspondence should be addressed to Chia-Ding Hou; stat0002@mail.fju.edu.tw

Received 18 February 2014; Accepted 7 April 2014; Published 28 April 2014

Academic Editor: Ker-Wei Yu

Crude oil is the most important nonrenewable energy resource and the most key element for the world. In contrast to typical forecasts of oil price, this study aims at forecasting the demand of imported crude oil (ICO). This study proposes different single stage and two-stage hybrid stages of forecasting models for prediction of ICO in Taiwan. The single stage forecasting modeling includes multiple linear regression (MLR), support vector regression (SVR), artificial neural networks (ANN), and extreme learning machine (ELM) approaches. While the first step of the two-stage modeling is to select the fewer but more significant explanatory variables, the second step is to generate predictions by using these significant explanatory variables. The proposed two-stage hybrid models consist of integration of different modeling components. Mean absolute percentage error, root mean square error, and mean absolute difference are utilized as the performance measures. Real data set of crude oil in Taiwan for the period of 1993–2010 and twenty-three associated explanatory variables are sampled and investigated. The forecasting results reveal that the proposed two-stage hybrid modeling is able to accurately predict the demand of crude oil in Taiwan.

1. Introduction

Natural resources are often classified into two groups: renewable and non-renewable resources. Renewable energy is the one which comes from natural resources such as sunlight, wind, and rain, and it is naturally replenished. It is reported that about 16% of global final energy consumption comes from renewable energy. The share of renewable energy in global electricity generation is around 19% [1]. In the United States, renewables provided 12.7% of total domestic electricity in 2011, up from 10.2% in 2010, and 9.3% in 2009. In China, wind power generation increased by more than 48.2% in 2011. In European Union, renewables accounted for more than 71% of total electric capacity additions in 2011 [2]. Non-renewable resources form very slowly or do not naturally form in the environment. A good example of the nonrenewable is fossil fuels. Although many studies have reported that the renewable sources of energy are currently receiving considerable attention, the fossil fuels are still the most needed element of the world energies [3]. In particular, crude oil is the most important nonrenewable energy resource and the most key element for the world [4, 5].

The relationship between economic growth and oil consumption was addressed [6], and the study determined that the minimum statistical (lower-bound) annual oil consumption for developed countries was 11 barrels per capita. By using autoregressive distributed lag (ARDL) bounds testing approach of cointegration, the study discussed a long-run relationship among quantity of crude oil import, income, and price of the imported crude in India [7]. The results showed that the long-term income elasticity of imported crude in India is 1.97 and there existed a unidirectional long-run causality running from economic growth to crude oil import. The demand for ICO in South Africa as a function of real income and the price of crude oil were studied [8]. The estimated long-run price and income elasticities revealed that import demand for crude oil is price and income inelastic.

The study estimated the short-run and long-run elasticities of demand for crude oil in Turkey for the period of 1980 to 2005 [9]. The comparative study was performed in the study [10]. A decision support system for forecasting fossil fuel production in Turkey, using a regression ARIMA and SARIMA method, was developed for the period of 1950–2003. Also, the study applied the ARIMA and SARIMA methods to predict primary energy demand in Turkey for the period from 2005 to 2020 [11].

In addition to forecasting the oil demand or consumption, several forecasting methods were applied to predict different types of energies. For example, regression modeling was employed to forecast the coal, oil, and gas electricity requirement [12]. The MR models were also applied to predictions for electricity consumption in Taiwan, Brazil, Delhi, and Hong-Kong, respectively [13–15]. The study [16] used the multiple linear regression techniques to develop simple empirical relations for the estimation of daily and monthly evaporation in Kuwait. While the ARIMA approaches were commonly used for predictions of energy demand [10, 11], the ANN models were also widely used for predictions of energy demand [11, 17–20]. Another promising forecasting technique, SVR, was used to forecast the energy demand [21, 22]. Additionally, the various hybrid modeling approaches were reported in forecasting energy demand in several studies [23–26], and the hybrid modeling schemes appear to be a promising technique. The study provided literature review in detail about the forecasting modeling in energy issue [27].

Extreme learning machine (ELM) proposed by Huang et al. [28, 29] is a novel learning algorithm for single-hidden-layer feedforward neural networks (SLFN). It provides much better generalization performance with much faster learning speed and avoids many issues faced in the traditional algorithms such as stopping criterion, learning rate, number of epochs and local minima, and the overtuned problems [29]. Moreover, the universal approximation capability of ELM has been analyzed and proven by [30–32] to show the effectiveness of ELM. Thus, ELM has attracted a lot of attentions in recent years and been used for various forecasting issues, such as sales forecasting [33–35], stock price forecasting [36, 37], and electricity price forecasting [38].

In June 1946, Chinese Petroleum Corp. (CPC) was funded, and the headquarters was set up in Taipei under the direction of the Ministry of Economic Affairs. With service facilities covering the whole nation, its operations today include the import, exploration, development, refining, transport, marketing, and sale of petroleum and natural gas. CPC's total capital stands at NT$130.1 billion, and its total revenues in 2011 amounted to NT$1.03 trillion.

Since crude oil is extremely important for development of Taiwan's economy, the predictions of the demand of imported crude oil are a must. Accordingly, this study is aimed at proposing single and two-stage forecasting techniques to predict the demand of imported crude oil in Taiwan. The single stage forecasting modeling includes the support vector regression (SVR), artificial neural networks (ANN), extreme learning machine (ELM), and multiple linear regression (MLR) approaches. The two-stage models combine the two modeling components. The first component of the model uses its own feature to capture the significant explanatory variables. Then, the second component generates the predictions based on these explanatory variables. In this study, the combinations of MLR and SVR (i.e., refer to MLR-SVR), MLR and ANN (i.e., refer to MLR-ANN), and MLE and ELM (i.e., refer to MLR-ELM) are used as the two-stage models.

Real data are sampled for the period of 1993–2010 for the ICO in Taiwan. According to the suggestion [39], twenty-three associated variables are collected for serving as the explanatory variables. The predictions for ICO in Taiwan can be made based on the single stage and two-stage forecasting models. This study uses the first 14 years (1993–2006) of data for model building, and the last four years' data are used for the purpose of confirmation. The mean absolute percentage error (MAPE), the root mean square error (RMSE), and the mean absolute difference (MAD) are used as the forecasting accuracy measures.

The contents of this study are organized as follows. The following section introduces the proposed forecasting techniques. Section 3 presents the real data of ICO and the forecasting results. The performances for all of the forecasting models are demonstrated and discussed. The final section, Section 4, concludes this study.

2. Research Methodologies

This study considers MLR, SVR, ANN, ELM, and their hybrid modeling schemes as possible forecasting models for import demand of crude oil in Taiwan. These forecasting techniques are introduced in the subsequent sections.

2.1. Multiple Linear Regression. The multiple linear regression analysis is the procedure by which an algebraic equation is formulated to estimate the value of a dependent variable Y_i, given the values of some other independent variables $X_{1i}, X_{2i}, \ldots, X_{pi}$, $i = 1, 2, \ldots, n$. The relationship between the dependent and independent variables is expressed by a linear algebraic model as follows:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_p X_{pi} + \varepsilon_i, \quad i = 1, 2, \ldots, n. \tag{1}$$

Based on the least squares or maximum likelihood criterion, the so-called normal equations and the point estimators of $\beta_0, \beta_1, \ldots, \beta_p$ can be obtained. Under the normal and some mild assumptions, the sampling distributions and hence the statistical inference for the estimators of $\beta_0, \beta_1, \ldots, \beta_p$ can be derived.

To identify significant independent variables, the backward elimination, forward selection, or stepwise regression procedures can be applied. The backward elimination procedure begins with the model which includes all of the available explanatory variables, and successively deletes one variable at a time to the model in such a way that at each step, the variable deleted is the variable contributing the least to the prediction of dependent variable at that step. On the contrary, the forward selection procedure begins with the constant model that includes no explanatory variable, and successively adds one variable at a time to the model in such a way that, at each step, the variable added is the variable contributing the most to the prediction at that step. The stepwise regression procedure is an admixture of the backward elimination procedure and the forward selection procedure. This selection procedure builds a sequence of models and at each step deletes or adds an explanatory variable according to some selection criterions such as coefficient of partial correlation or error sum of squares reduction.

2.2. Support Vector Regression. While support vector machine (SVM) is one of the most powerful techniques in machine learning areas [40–44], support vector regression (SVR) is the most common application form of SVMs. Due to its effectiveness, SVR has been used for predictions in many fields [45–49].

Statistical learning theory and structural risk minimization principle have provided a very effective framework for development of support vector regression [48–51]. Based on the computation of a linear regression function in a high dimensional feature space, the inputs for SVR are mapped via a nonlinear function. The modeling of SVR can be described as follows. Suppose that

$$f(x) = (w \cdot \Phi(x)) + b, \tag{2}$$

where w is the weight vector, x represents the model inputs, b is a bias, and $\Phi(x)$ stands for a kernel function which uses a nonlinear function to transform the nonlinear input to be linear mode in a high dimension feature space.

Typical regression modeling obtains the coefficients through minimizing the square error, which can be considered as empirical risk based on loss function. The ε-insensitivity loss function was introduced [51], and it can be described as follows:

$$L_\varepsilon(f(x), y) = \begin{cases} |f(x) - y| - \varepsilon & \text{if } |f(x) - y| \geq \varepsilon \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where y is the target outputs and ε defines the region of ε-insensitivity; when the predicted value falls into the band area, the loss is zero. However, when the predicted value falls outside the band area, the loss is defined as the difference between the predicted value and the margin.

When empirical risk and structure risk are both considered, the SVR can be setup to minimize the following quadratic programming problem:

$$\begin{aligned} \text{Minimize:} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & (y_i - w \cdot \Phi(x_i)) - b \leq \varepsilon + \xi_i \\ & (w \cdot \Phi(x_i)) + b - y_i \leq \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0, \\ & \quad \text{for } i = 1, \dots, n, \end{aligned} \tag{4}$$

where $i = 1, \dots, n$ is the number of training data, $(\xi_i + \xi_i^*)$ represents the empirical risk, $(1/2)\|w\|^2$ stands for the structure risk preventing over-learning and lack of applied universality, and C is a modifying coefficient representing the trade-off between empirical risk and structure risk. With an appropriate modifying coefficient C, band area width ε, and kernel function, the optimum value of each parameter can be solved by Lagrange procedure.

The general form of the SVR-based regression function is described as follows [51]:

$$f(x, w) = f(x, r, r^*) = \sum_{i=1}^{N}(r_i - r_i^*)\,\Phi(x, x_i) + b, \tag{5}$$

where r_i and r_i^* are Lagrangian multipliers and satisfy the equality $r_i r_i{}^* = 0$. Additionally, since the radial basis function (RBF) is the most widely used kernel function [49], this study uses it for our experimental study. The RBF can be defined as follows:

$$\phi(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right), \tag{6}$$

where σ denotes the width of the RBF.

2.3. Artificial Neural Networks. Artificial neural networks, originally derived from neurobiological models, are massively parallel, computer-intensive, and data-driven algorithmic systems composed of a multitude of highly interconnected nodes, known as neurons as well. Mimicking human neurobiological information-processing activities, each elementary node of a neural network is able to receive an input single from external sources or other nodes and the algorithmic procedure equipped in each node is sequentially activated to locally transforming the corresponding input single into an output single to other nodes or environments.

It was indicated that knowledge is not stored within individual processing units, but is represented by the strength between units [52]. They also stated that neural networks can be classified into two categories: the feedforward networks and the feedback networks. The feedback networks contain neurons that are connected to each other, enabling a neuron to influence other neurons. The feedback networks contain neurons that are connected to themselves, enabling a neuron to influence other neurons and itself. Also, in recent years, there has been a great deal of attention toward the field of ANN [41, 43, 53, 54].

For ANN modeling, the relationship between output (y) and inputs $(x_1, x_2, \dots, x_m)$ can be formed as

$$y = \lambda_0 + \sum_{j=1}^{n}\lambda_j g\left(\delta_{0_j} + \sum_{i=1}^{m}\delta_{ij}x_i\right) + \varepsilon, \tag{7}$$

where λ_j $(j = 0, 1, 2, \dots, n)$ and δ_{ij} $(i = 0, 1, 2, \dots, m; j = 0, 1, 2, \dots, n)$ are model connection weights, m is the number of input nodes, n is the number of hidden nodes, and ε is the error term. The transfer function in the hidden layer is often represented by a logistic function; that is,

$$g(z) = \frac{1}{1 + \exp(-z)}. \tag{8}$$

Accordingly, the ANN model in (7) accomplishes a nonlinear functional mapping from the inputs $(x_1, x_2, \ldots, x_a)$ to the output y; that is,

$$y = f\left(x_1, x_2, \ldots, x_m, w\right) + \varepsilon, \tag{9}$$

where w is a vector of all model parameters and f is a function determined by the ANN structure and connection weights.

2.4. Extreme Learning Machine. ELM randomly selected the input weights and analytically determined the output weights of SLFNs. One may randomly choose and fix the hidden node parameters which are the key principle of the ELM. After randomly choosing the hidden nodes parameters, SLFN becomes a linear system where the output weights of the network can be analytically determined using simple generalized inverse operation of the hidden layer output matrices [28, 29].

In general, the concept of ELM is similar to that of the random vector functional-link (RVFL) network where the hidden neurons are randomly selected. However, the main difference between ELM and RVFL is the characteristics of hidden neuron parameters. In ELM, all the hidden node parameters are randomly generated independently of the target functions and the training patterns [55]. In RVFL, the selection of hidden neurons is based on partial randomness and the randomly generated hidden node parameters are not completely independent of the training data [56]. That is, the universal approximation capability of ELM can be linearly extended to RVFL [55, 56].

Consider N arbitrary distinct samples (x_i, t_i) where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in R^n$ and $t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in R^m$. SLFNs with $\widetilde{N}$ hidden neurons and activation function $g(x)$ can approximate N samples with zero error. This means that

$$H\beta = T, \tag{10}$$

where

$$\begin{gathered} H\left(w_1, \ldots, w_{\widetilde{N}}, b_1, \ldots, b_{\widetilde{N}}, x_1, \ldots, x_N\right) \\ = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_{\widetilde{N}} \cdot x_1 + b_{\widetilde{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_{\widetilde{N}} \cdot x_N + b_{\widetilde{N}}) \end{bmatrix}_{N\times\widetilde{N}}, \\ \beta_{\widetilde{N}\times m} = \left(\beta_1^T, \ldots, \beta_{\widetilde{N}}^T\right)^t, \\ T_{N\times m} = \left(T_1^T, \ldots, T_N^T\right)^t, \end{gathered} \tag{11}$$

and $w_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$, $i = 1, 2, \ldots, \widetilde{N}$, is the weight vector connecting the ith hidden node and the input nodes, $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ is the weight vector connecting the ith hidden node and the output nodes, and b_i is the threshold of the ith hidden node. $w_i \cdot x_j$ denotes the inner product of w_i and x_j. H is called the hidden layer output matrix of the neural network; the ith column of H is the ith hidden node output with respect to inputs $x_1, x_2, \ldots x_N$. Therefore, the determination of the output weights is as simple as finding the least-square solution to the given linear system. The minimum norm least-square (LS) solution to the linear system is

$$\widehat{\beta} = H^{\Psi} T, \tag{12}$$

where H^{Ψ} is the Moore-Penrose generalized inverse of matrix H. The minimum norm LS solution is unique and has the smallest norm among all the LS solutions.

The first step of ELM algorithm is to randomly assign input weight w_i and bias b_i. Then, the hidden layer output matrix H is calculated. Finally, one can calculate the output weight β, $\widehat{\beta} = H^{\Psi}T$, where $T = (t_1, \ldots, t_N)^t$.

2.5. Hybrid Models. Recent research indicates that hybrid systems which are integrated with several standard ones can help to achieve a better performance for some applications. For example, the hybrid modeling applications have been reported in forecasting [57–62], credit risk [61], and manufacturing process [41–44]. As a consequence, this study proposes a two-step hybrid data mining mechanisms to predict the demand of ICO in Taiwan. This study proposes three hybrid data mining models, namely, the integration of MLR and ANN (i.e., MLR-ANN), MLR and SVR (i.e., MLR-SVR), and MLR and ELM (i.e., MLR-ELM). The concept of those hybrid modeling is as follows.

In the first stage of hybrid modeling, more significant variables are selected using MLR with forward selection, backward elimination or stepwise regression, say, $X_{i1}^*, X_{i2}^*, \ldots, X_{ir}^*$. In the second stage, the selected significant variables $X_{i1}^*, X_{i2}^*, \ldots, X_{ir}^*$ obtained in the first stage are served as the inputs of ANN, SVR, and ELM in order to establish the two-stage hybrid forecasting models. By providing the ANN, SVR, and ELM with good starting points, it is hoped that more effective models can be developed on the strength of their learning ability. Such two-stage hybrid models then are compared with the single-stage of MLR, ANN, SVR, and ELM models.

3. Results and Analysis

To show the effectiveness of the proposed hybrid modeling, the real data, from the years 1993 to 2010, were sampled for the ICO from Bureau of Energy in Taiwan [62]. Following the suggestion [39] and having discussed with the ICO practitioners, we sample 23 associated influential variables. Table 1 lists these 23 variables. The yearly data were collected for the period of 1993–2010 from the web sites of Bureau of Energy in Taiwan [62]. The first 14 years' data are used for model building, and the last four years' data are used for model confirmation.

3.1. Single-Stage Modeling. Figure 1 displays the yearly data of ICO in Taiwan for the periods of 1993–2006. For MLR modeling, we first compute the variance inflation factors (VIFs) to examine the existence of collinearity.

Our numerical results reveal that all the values of VIFs of independent variables are greater than 10 except the variables of X_4 and X_5, indicating that there may exists

Table 1: Meaning of the influential variables for ICO model building.

Variable	Meaning
Y	Imported crude oil
X_1	Gross domestic product
X_2	Consumer price index
X_3	Personal disposable income
X_4	Average temperature
X_5	Average sunshine per day (hours)
X_6	Average electricity households
X_7	Average electricity price
X_8	National income
X_9	Population
X_{10}	Foreign trade total
X_{11}	Wholesale prices
X_{12}	Consumer index
X_{13}	GNP deflators
X_{14}	Foreign exchange reserves
X_{15}	Total primary energy supply
X_{16}	Total final consumption
X_{17}	Total domestic consumption
X_{18}	Energy productivity
X_{19}	Energy intensity
X_{20}	Energy consumption of energy intensive industries
X_{21}	Value-added of energy intensive industries
X_{22}	Dependence on imported energy
X_{23}	Electricity average load

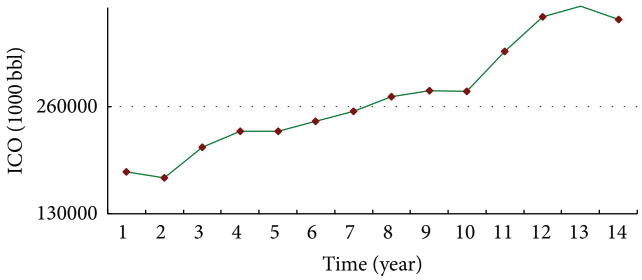


Figure 1: Historical yearly data of ICO in Taiwan.

high collinearity among the independent variables. To eliminate variables with high collinearity, we apply the Pearson correlation coefficient. If the correlation coefficient between two independent variables is greater than 0.7, we eliminate the variable which has a lower relationship with dependent variable Y. Table 2 lists the analysis results. As shown in Table 2, after excluding the variables with high collinearity, there remained six independent variables, including X_4, X_5, X_7, X_{13}, X_{19}, and X_{22}. In Table 3, it can be found that all the values of VIFs of the remaining variables are smaller than 10. As a consequence, there is no high collinearity among these independent variables.In addition, we apply the backward elimination, forward selection, and stepwise regression proceduresto identify significant independent variables, and use a 0.05 significance level to perform the MLR analysis.All the three procedures have the same selection results. As listed in Table 4, significant independent variables related to imported crude oil in Taiwan include average electricity price (X_7) and dependence on imported energy (X_{22}). Accordingly, the MLR model is derived as follows:

$$\widehat{Y} = -11862349.62 - 75695.53X_7 + 124776.53X_{22}. \quad (13)$$

The regression coefficients in the MLR model indicate that the higher the average electricity price is, the lower the imported crude oil is. On the contrary, the higher the dependence on imported energy is, the lower the imported crude oil is.

For ANN modeling, since the backpropagation neural network (BPNN) structure has been widely used [43, 54], this study employs BPNN as ANN modeling structure. In BPNN structure, we have 23 input nodes and one output node. The hidden nodes range from $i+2$ to $i-2$, where i is the number of input variables. Thus, the hidden nodes were set up as 21, 22, 23, 24, and 25, respectively. The training and testing processes include 14 and 4 data vectors for possible parameter setting. The learning rates are 0.01, 0.005, and 0.001, respectively, according to the suggestions [43].

After applying ANN to ICO data, we have obtained the {23-22-1} topology with a learning rate of 0.01 which provides the best result. The $\{N_i\text{-}N_h\text{-}N_o\}$ represents the number of nodes in the input layer, hidden layer, and output layer, respectively. For SVR modeling, same as ANN modeling, we have 23 input variables. The two parameters, C and gamma, were estimated as 2^{-15} and 2^{-15}, respectively.

As mentioned earlier, the most important ELM parameter is the number of hidden nodes and that ELM tends to be unstable in single run forecasting [29, 33]. Therefore, the ELM models with different numbers of hidden nodes varying from 1 to 15 are constructed. For each number of nodes, an ELM model is repeated 30 times and the average RMSE of each node is calculated. The number of hidden nodes that gives the smallest average RMSE value is selected as the best parameter of ELM model. In modeling ELM, the forecasting model with eight hidden nodes has the smallest average RMSE values and is therefore the best ELM model.

3.2. Proposed Hybrid Modeling. A rational strategy for a hybrid modeling is to use the fewer but more informative variables, which were selected by the first stage of modeling approaches, as the inputs for the second stage of classifier approaches. Accordingly, in this study, the significant variables selected, that is, average electricity price (X_7) and dependence on imported energy (X_{22}), are used as the input variables of the ANN and SVR for hybrid modeling.

After completing the first stage of hybrid modeling, the ANN topology settings can be established. This study has found that the {2-3-1} topology with a learning rate of 0.01 provides the best result for the hybrid model. The network topology with the minimum testing RMSE is also considered as the optimal network topology. For the MR/SVR hybrid modeling, the parameters of C and gamma were still estimated as 2^{-15} and 2^{-15}, respectively. In the construction of

Table 2: Pearson correlations for pairs of variables.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}	X_{20}	X_{21}	X_{22}	X_{23}	Y
X_1	1.00																							
X_2	**0.97**	1.00																						
X_3	**0.99**	**0.96**	1.00																					
X_4	0.31	0.27	0.32	1.00																				
X_5	0.29	0.26	0.30	0.00	1.00																			
X_6	**0.99**	**0.98**	**0.99**	0.31	0.35	1.00																		
X_7	0.33	0.45	0.31	−0.02	0.13	0.37	1.00																	
X_8	**0.99**	**0.96**	**0.99**	0.32	0.30	**0.99**	0.30	1.00																
X_9	**0.99**	**0.95**	**0.99**	0.34	0.38	**0.99**	0.30	**0.99**	1.00															
X_{10}	**0.92**	**0.90**	**0.91**	0.09	0.16	**0.89**	0.43	**0.91**	**0.87**	1.00														
X_{11}	**0.81**	**0.85**	**0.79**	0.04	0.16	**0.80**	0.55	**0.79**	**0.75**	**0.94**	1.00													
X_{12}	**0.97**	**0.99**	**0.96**	0.27	0.26	**0.98**	0.45	**0.96**	**0.95**	**0.90**	**0.85**	1.00												
X_{13}	0.19	0.18	0.21	0.45	−0.02	0.20	−0.49	0.22	0.23	−0.13	−0.26	0.18	1.00											
X_{14}	**0.90**	**0.86**	**0.89**	0.13	0.40	**0.89**	0.57	**0.89**	**0.88**	**0.92**	**0.86**	**0.86**	−0.22	1.00										
X_{15}	**0.99**	**0.92**	**0.99**	0.31	0.38	**0.97**	0.25	**0.99**	**0.99**	**0.89**	**0.76**	**0.92**	0.17	**0.89**	1.00									
X_{16}	**0.98**	**0.93**	**0.98**	0.30	0.39	**0.98**	0.27	**0.98**	**0.99**	**0.90**	**0.77**	**0.93**	0.15	**0.91**	**1.00**	1.00								
X_{17}	**0.98**	**0.93**	**0.98**	0.30	0.38	**0.98**	0.25	**0.99**	**0.99**	**0.89**	**0.77**	**0.92**	0.16	**0.90**	**1.00**	**1.00**	1.00							
X_{18}	0.11	0.24	0.09	−0.24	−0.31	0.11	0.75	0.08	0.00	0.37	0.54	0.24	−0.53	0.30	−0.02	−0.01	−0.02	1.00						
X_{19}	−0.09	−0.22	−0.07	0.25	0.34	−0.08	−0.72	−0.06	0.02	−0.35	−0.52	−0.22	0.53	−0.27	0.05	0.03	0.05	**−1.00**	1.00					
X_{20}	**0.96**	**0.91**	**0.96**	0.25	0.44	**0.95**	0.38	**0.96**	**0.96**	**0.91**	**0.83**	**0.91**	−0.01	**0.95**	**0.98**	**0.98**	**0.98**	0.07	−0.05	1.00				
X_{21}	**0.98**	**0.92**	**0.98**	0.27	0.33	**0.96**	0.34	**0.98**	**0.96**	**0.94**	**0.82**	**0.92**	0.07	**0.93**	**0.98**	**0.98**	**0.98**	0.12	−0.10	**0.98**	1.00			
X_{22}	**0.98**	**0.93**	**0.98**	0.27	0.38	**0.98**	0.25	**0.98**	**0.99**	**0.88**	**0.75**	**0.93**	0.20	**0.88**	**0.99**	**0.99**	**0.99**	−0.01	0.04	**0.96**	**0.97**	1.00		
X_{23}	**0.99**	**0.94**	**0.99**	0.35	0.35	**0.98**	0.30	**0.99**	**0.99**	**0.91**	**0.80**	**0.94**	0.16	**0.90**	**0.99**	**0.99**	**0.99**	0.06	−0.03	**0.97**	**0.98**	**0.98**	1.00	
Y	0.91	0.84	0.92	0.19	0.41	0.90	0.06	0.92	0.92	0.81	0.68	0.84	0.23	0.82	0.94	0.94	0.94	−0.13	0.15	0.90	0.90	0.95	0.93	1.00

Table 3: Collinearity diagnosis for MLR modeling.

Variable X_i	X_4	X_5	X_7	X_{13}	X_{19}	X_{22}
VIF	1.52	2.04	3.91	1.99	4.36	1.49

Table 4: MLR model for ICO in Taiwan.

Variables	Estimated coefficient	Standard error	P value
Constant	−11862349.62	853356.39	<0.01**
Average electricity price (X_7)	−75695.53	28564.49	0.02**
Dependence on imported energy (X_{22})	124776.53	8782.77	<0.01**

**Denotes significance at 5% level.

Table 5: Various forecasting models' accuracy measures for ICO.

	MAPE	RMSE	MAD
Single stage models			
ANN	15.681	56190.676	52155.891
SVR	22.304	77973.147	76318.645
MR_{SEL}	11.753	48139.849	39431.691
ELM	10.174	43189.042	33271.068
Proposed hybrid models			
MR_{SEL}-ANN	7.302	33875.635	24572.516
MR_{SEL}-SVR	9.385	35523.521	32465.307
MR_{SEL}-ELM	7.094	28902.330	23271.608

Table 6: Improvement of the proposed models in comparison with the single stage models.

Models	MAPE (%)	RMSE (%)	MAD (%)
Proposed hybrid MR_{SEL}-ANN model			
ANN	53.43	39.71	52.89
SVR	67.26	56.55	67.80
MR_{SEL}	37.87	29.63	37.68
ELM	28.23	21.56	26.14
Proposed hybrid MR_{SEL}-SVR model			
ANN	40.15	36.78	37.75
SVR	57.92	54.44	57.46
MR_{SEL}	20.15	26.21	17.67
ELM	7.76	17.75	2.42
Proposed hybrid MR_{SEL}-ELM model			
ANN	54.76	48.56	55.38
SVR	68.19	62.93	69.51
MR_{SEL}	39.64	39.96	40.98
ELM	30.27	33.08	30.05

MR/ELM hybrid model, the model with eight hidden nodes has the smallest average RMSE values is the best MR/ELM hybrid model.

3.3. Experimental Results. In this study, we consider the forecasting accuracy measures of MAPE, MSE, and MAD to address the forecasting performance for the five different approaches, ANN, SVR, ELM, MR_{SEL}, MR_{SEL}-ANN, MR_{SEL}-SVR, and MR_{SEL}-ELM, respectively. These prediction measurements are defined as follows:

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|e_t|}{Y_t} \times 100,$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(e_t)^2}, \tag{14}$$

$$\text{MAD} = \frac{1}{n}\sum_{t=1}^{n}|e_t|,$$

where e_t stands for the value of the residual at time t. A low MAPE, MSE or MAD is associated with better forecasting accuracy. The results are listed in Table 5. In Table 5, by considering single stage modeling approaches, we note that ELM model has the best performance than the MR, ANN, and SVR models.

In comparison to the single stage and our proposed hybrid models in Table 5, one is able to apparently observe that our proposed hybrid models provide more accurate results than the single stage models. In terms of MAPE, MSE, or MAD, the two hybrid models are all lower than the three single stage models. For example, the MAPE percentage improvements of the proposed MR_{SEL}-ANN model over the four-single-stage models, ANN, SVR, MR_{SEL}, and ELM are 53.43%, 67.26%, 37.87%, and 28.23%, respectively. Table 6 lists a comparison with respect to the overall improvement percentage in the single stage models. In addition, we use independent sample t-test to test whether the hybrid models are superior to the single ones. Let μ_s and μ_h be the means of the absolute value of residuals for the single-stage models and hybrid models, respectively. The independent sample t-test is applied to test the following:

$$H_0 : \mu_s - \mu_h \leq 0 \quad \text{versus} \quad H_1 : \mu_s - \mu_h > 0. \tag{15}$$

The P value is 0.0127. Obviously, the hybrid models outperform the single stage models.

Figure 2 shows the actual ICO values for the last four years and the corresponding forecasts by using four-single-stage models. It can be seen that the ELM model provides the best predictions of ICO. Figure 3 displays the actual ICO values and the corresponding forecasts by using three hybrid models. It shows that the hybrid MR-ANN model has the best forecasting capability. Figure 4 plots the actual ICO values and the corresponding forecasts by using the single MR and ANN models and their hybrid technique, MR-ANN. It apparently can be observed that the proposed hybrid model outperforms the single models. The same conclusions can be drawn by observing Figures 5 and 6. That is, the performance of the hybrid models is better than the single models.

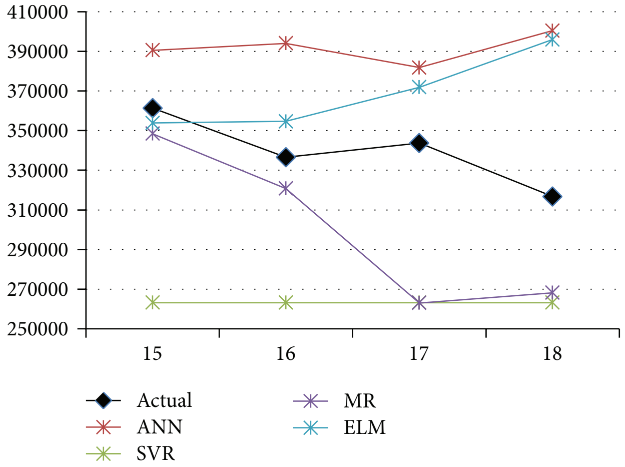


Figure 2: Plot of actual ICO values for the last four years and the forecasts by using four single-stage models.

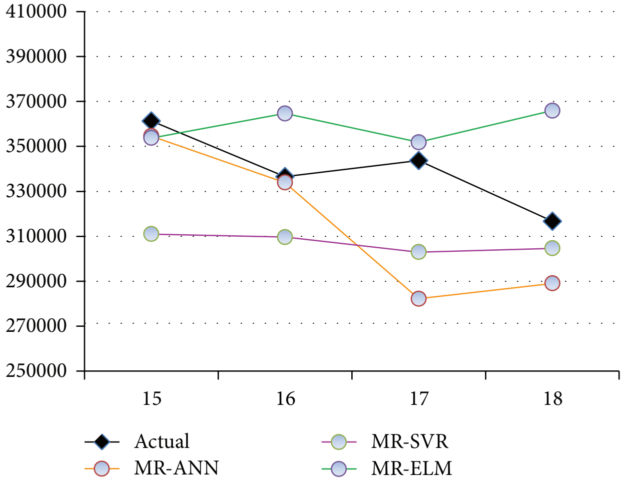


Figure 3: Plot of actual ICO values for the last four years and the forecasts by using three hybrid models.

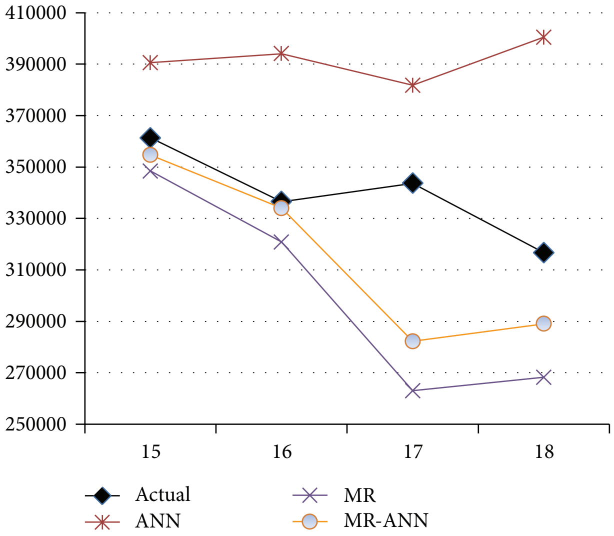


Figure 4: Plot of actual ICO values for the last years and the forecasts by using MR, ANN, and the hybrid MR-ANN model.

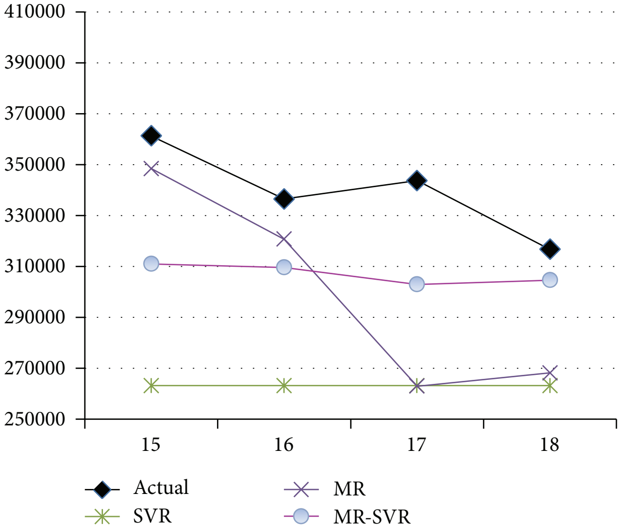


Figure 5: Plot of actual ICO values for the last years and the forecasts by using MR, SVR, and the hybrid MR-SVR model.

As shown in Figures 4, 5, and 6 and Tables 5 and 6, the proposed hybrid models outperform all three-single-stage models. As a consequence, the proposed two-stage hybrid approaches are more efficient for forecasting ICO in Taiwan than the typical single stage methods.

4. Conclusions

Oil is not only used to make gas for cars, but for heating homes, producing electricity, making plastics, and other commodities. Oil and its byproducts are ingrained into almost every culture in the world. Therefore, the accurate prediction of the demand of ICO is very important for the economic development of a country.

Because it is difficult to fully capture the characteristics of the real ICO data, the two hybrid prediction models are then proposed to forecast the demand of ICO in Taiwan. Based on our numerical results, it is found that the proposed hybrid approaches are more accurate than the established single-stage ones. The modeling procedures and results of this work may provide a guidance to develop forecasting models for other energies.

Besides, there are many other two-stage hybrid forecasting models that have been proposed and applied in various fields [63–70]. The proposed hybrid model in this study is not the only hybrid forecasting scheme for the prediction of demand of crude oil, as one can combine other artificial intelligence techniques or traditional multivariate models, like decision tree, multivariate adaptive regression splines, logistic regression, rough set, or independent component analysis, with ANN, SVR, or ELM for further improving the prediction accuracies. Based on our work, further research may be

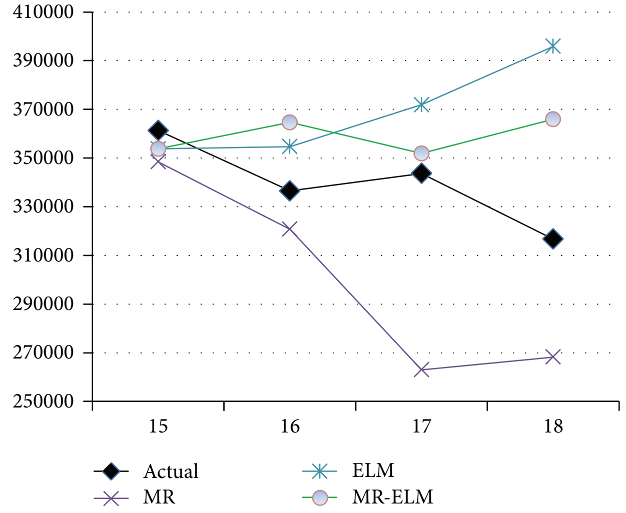


Figure 6: Plot of actual ICO values for the last years and the forecasts by using MR, ELM, and the hybrid MR-ELM model.

expanded. For example, extensions of the proposed hybrid prediction methods to other machine learning techniques or statistical prediction methods or to multistage prediction procedures are possible. Such works deserve further research and are our future concern.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported in part by the National Science Council of Taiwan, Grants NSC102-2221-E-030-019 and NSC 102-2118-M-030-001.

References

[1] REN21 Renewables 2011: Global Status Report, 2011, http://www.ren21.net/Portals/0/documents/activities/gsr/2011 GSR Press Release-English.pdf.

[2] REN21, Renewables 2012: Global Status Report, 2012, http://new.ren21.net/Portals/0/documents/activities/gsr/GSR2012_low res_FINAL.pdf.

[3] International Energy Agency (IEA). World Energy Outlook 2004; IEA, Paris, France, 2004, http://www.worldenergyoutlook.org/media/weowebsite/2008-1994/WEO2004.pdf.

[4] International Energy Agency (IEA), World Energy Outlook 2007; IEA, Paris, France, 2007, http://www.worldenergyoutlook.org/media/weowebsite/2008-1994/weo_2007.pdf.

[5] A. Mirchi, S. Hadian, K. Madani, O. M. Rouhani, and A. M. Rouhani, "World energy balance outlook and OPEC production capacity: implications for global oil security," *Energies*, vol. 5, pp. 2626–2651, 2012.

[6] W. P. Nel and C. J. Cooper, "A critical review of IEA's oil demand forecast for China," *Energy Policy*, vol. 36, no. 3, pp. 1096–1106, 2008.

[7] S. Ghosh, "Import demand of crude oil and economic growth: evidence from India," *Energy Policy*, vol. 37, no. 2, pp. 699–702, 2009.

[8] E. Ziramba, "Price and income elasticities of crude oil import demand in South Africa: a cointegration analysis," *Energy Policy*, vol. 38, no. 12, pp. 7844–7849, 2010.

[9] G. Altinay, "Short-run and long-run elasticities of import demand for crude oil in Turkey," *Energy Policy*, vol. 35, no. 11, pp. 5829–5835, 2007.

[10] V. Ş. Ediger, S. Akar, and B. Uğurlu, "Forecasting production of fossil fuel sources in Turkey using a comparative regression and ARIMA model," *Energy Policy*, vol. 34, no. 18, pp. 3836–3846, 2006.

[11] V. Ş. Ediger and S. Akar, "ARIMA forecasting of primary energy demand by fuel in Turkey," *Energy Policy*, vol. 35, no. 3, pp. 1701–1708, 2007.

[12] H. Farahbakhsh, V. I. Ugursal, and A. S. Fung, "A residential end-use energy consumption model for Canada," *International Journal of Energy Research*, vol. 22, no. 13, pp. 1133–1143, 1998.

[13] R. Sadownik and E. P. Barbosa, "Short-term Forecasting of industrial electricity consumption in Brazil," *Journal of Forecasting*, vol. 18, no. 3, pp. 215–224, 1999.

[14] M. Ranjan and V. K. Jain, "Modelling of electrical energy consumption in Delhi," *Energy*, vol. 24, no. 4, pp. 351–361, 1999.

[15] Y. Y. Yan, "Climate and residential electricity consumption in Hong Kong," *Energy*, vol. 23, no. 1, pp. 17–20, 1998.

[16] J. Almedeij, "Modeling pan evaporation for Kuwait by multiple linear regression," *The Scientific World Journal*, vol. 2012, Article ID 574742, 9 pages, 2012.

[17] K. Ermis, A. Midilli, I. Dincer, and M. A. Rosen, "Artificial neural network analysis of world green energy use," *Energy Policy*, vol. 35, no. 3, pp. 1731–1743, 2007.

[18] Z. W. Geem and W. E. Roper, "Energy demand estimation of South Korea using artificial neural network," *Energy Policy*, vol. 37, no. 10, pp. 4049–4054, 2009.

[19] L. Ekonomou, "Greek long-term energy consumption prediction using artificial neural networks," *Energy*, vol. 35, no. 2, pp. 512–517, 2010.

[20] Y. S. Murat and H. Ceylan, "Use of artificial neural networks for transport energy demand modeling," *Energy Policy*, vol. 34, no. 17, pp. 3165–3172, 2006.

[21] S. Fan, L. N. Chen, and W.-J. Lee, "Machine learning based switching model for electricity load forecasting," *Energy Conversion and Management*, vol. 49, no. 6, pp. 1331–1344, 2008.

[22] J. Wang, W. Zhu, W. Zhang, and D. Sun, "A trend fixed on firstly and seasonal adjustment model combined with the ε-SVR for short-term forecasting of electricity demand," *Energy Policy*, vol. 37, no. 11, pp. 4901–4909, 2009.

[23] A. Azadeh, S. F. Ghaderi, S. Tarverdian, and M. Saberi, "Integration of artificial neural networks and genetic algorithm to predict electrical energy consumption," *Applied Mathematics and Computation*, vol. 186, no. 2, pp. 1731–1741, 2007.

[24] M. Amina, V. S. Kodogiannis, I. Petrounias, and D. Tomtsis, "A hybrid intelligent approach for the prediction of electricity consumption," *International Journal of Electrical Power & Energy Systems*, vol. 43, pp. 99–108, 2012.

[25] X. Wang and M. Meng, "A hybrid neural network and ARIMA model for energy consumption forecasting," *Journal of Computers*, vol. 7, no. 5, pp. 1184–1190, 2012.

[26] H. T. Pao, "Forecasting energy consumption in Taiwan using hybrid nonlinear models," *Energy*, vol. 34, no. 10, pp. 1438–1446, 2009.

[27] L. Suganthi and A. A. Samuel, "Energy models for demand forecasting—a review," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 2, pp. 1223–1240, 2012.

[28] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 985–990, Budapest, Hungary, July 2004.

[29] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[30] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[31] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16–18, pp. 3056–3062, 2007.

[32] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, no. 16–18, pp. 3460–3468, 2008.

[33] Z.-L. Sun, T.-M. Choi, K.-F. Au, and Y. Yu, "Sales forecasting using extreme learning machine with applications in fashion retailing," *Decision Support Systems*, vol. 46, no. 1, pp. 411–419, 2008.

[34] C. J. Lu, T. S. Lee, and C. M. Lian, "Sales forecasting for computer wholesalers: a comparison of multivariate adaptive regression splines and artificial neural networks," *Decision Support Systems*, vol. 54, no. 1, pp. 584–596, 2012.

[35] C. J. Lu and Y. E. Shao, "Forecasting computer products sales by integrating ensemble empirical mode decomposition and extreme learning machine," *Mathematical Problems in Engineering*, vol. 2012, Article ID 831201, 15 pages, 2012.

[36] Z. Fang, J. Zhao, F. Fei, Q. Wang, and X. He, "An approach based on multi-feature wavelet and ELM algorithm for forecasting outlier occurrence in Chinese stock market," *Journal of Theoretical and Applied Information Technology*, vol. 49, no. 1, pp. 369–377, 2013.

[37] J. Zhao, Z. Wang, and D. S. Park, "Online sequential extreme learning machine with forgetting mechanism," *Neurocomputing*, vol. 87, pp. 79–89, 2012.

[38] X. Chen, Z. Y. Dong, K. Meng, Y. Xu, K. P. Wong, and H. W. Ngan, "Electricity price forecasting with extreme learning machine and bootstrapping," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 2055–2062, 2012.

[39] Y. E. Shao, C. D. Hou, and C. S. Lin, "Applying artificial neural networks, multiple regression and their hybrid models for predictions of electricity sales in Taiwan," *ICIC Express Letters*. In press.

[40] J. Park, I.-H. Kwon, S.-S. Kim, and J.-G. Baek, "Spline regression based feature extraction for semiconductor process fault detection using support vector machine," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5711–5718, 2011.

[41] Y. E. Shao and B.-S. Hsu, "Determining the contributors for a multivariate SPC chart signal using artificial neural networks and support vector machine," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, pp. 4899–4906, 2009.

[42] C.-J. Lu, Y. E. Shao, and P.-H. Li, "Mixture control chart patterns recognition using independent component analysis and support vector machine," *Neurocomputing*, vol. 74, no. 11, pp. 1908–1914, 2011.

[43] Y. E. Shao, C.-J. Lu, and C.-C. Chiu, "A fault detection system for an autocorrelated process using SPC/EPC/ANN AND SPC/EPC/SVM schemes," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 9, pp. 5417–5428, 2011.

[44] Y. E. Shao, C. J. Lu, and Y. C. Wang, "A hybrid ICA-SVM approach for determining the fault quality variables in a multivariate process," *Mathematical Problems in Engineering*, vol. 2012, Article ID 284910, 12 pages, 2012.

[45] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6164–6173, 2009.

[46] C.-J. Lu and Y.-W. Wang, "Combining independent component analysis and growing hierarchical self-organizing maps with support vector regression in product demand forecasting," *International Journal of Production Economics*, vol. 128, no. 2, pp. 603–613, 2010.

[47] S. Pang, L. Song, and N. Kasabov, "Correlation-aided support vector regression for forex time series prediction," *Neural Computing and Applications*, vol. 20, no. 8, pp. 1193–1203, 2011.

[48] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing-Letters and Reviews*, vol. 11, no. 10, pp. 203–224, 2007.

[49] W. Dai, Y. E. Shao, and C. J. Lu, "Incorporating feature selection method into support vector regression for stock index forecasting," *Neural Computing and Applications*, vol. 23, no. 6, pp. 1551–1561, 2013.

[50] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[51] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 2000.

[52] M. Titterington, "Neural networks," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 1, pp. 1–8, 2010.

[53] L. Mi and F. Takeda, "Analysis on the robustness of the pressure-based individual identification system based on neural networks," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 1, pp. 97–110, 2007.

[54] C.-C. Chiu, Y. E. Shao, T.-S. Lee, and K.-M. Lee, "Identification of process disturbance using SPC/EPC and neural networks," *Journal of Intelligent Manufacturing*, vol. 14, no. 3-4, pp. 379–388, 2003.

[55] G.-B. Huang, M.-B. Li, L. Chen, and C.-K. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, no. 4–6, pp. 576–583, 2008.

[56] G. B. Huang, "An insight into extreme learning machines: random neurons, random features and kernels," *Cognitive Computation*, 2014.

[57] P. G. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[58] H. F. Zou, G. P. Xia, F. T. Yang, and H. Y. Wang, "An investigation and comparison of artificial neural network and time series models for Chinese food grain price forecasting," *Neurocomputing*, vol. 70, no. 16–18, pp. 2913–2923, 2007.

[59] Y. E. Shao, "Body fat percentage prediction using intelligent hybrid approaches," *The Scientific World Journal*, vol. 2014, Article ID 383910, 8 pages, 2014.

[60] H. C. Co and R. Boosarawongse, "Forecasting Thailand's rice export: Statistical techniques vs. Artificial neural networks,"

Computers and Industrial Engineering, vol. 53, no. 4, pp. 610–627, 2007.

[61] S. L. Lin, "A new two-stage hybrid approach of credit risk in banking industry," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8333–8341, 2009.

[62] Bureau of Energy, http://web3.moeaboe.gov.tw/ECW/populace/content/SubMenu.aspx?menu_id=141.

[63] C. J. Lu, Y. E. Shao, and C. C. Li, "Recognition of concurrent control chart patterns by integrating ICA and SVM," *Applied Mathematics & Information Sciences*, vol. 8, no. 2, pp. 681–689, 2014.

[64] Y. E. Shao, C. D. Hou, and C. C. Chiu, "Hybrid intelligent modeling schemes for heart disease classification," *Applied Soft Computing*, vol. 14, pp. 47–52, 2014.

[65] Y. E. Shao and C. D. Hou, "Hybrid artificial neural networks modeling for faults identification of a stochastic multivariate process," *Abstract and Applied Analysis*, vol. 2013, Article ID 386757, 10 pages, 2013.

[66] Y. E. Shao and C.-D. Hou, "Change point determination for a multivariate process using a two-stage hybrid scheme," *Applied Soft Computing Journal*, vol. 13, no. 3, pp. 1520–1527, 2013.

[67] M. Arun Kumar and M. Gopal, "A hybrid SVM based decision tree," *Pattern Recognition*, vol. 43, no. 12, pp. 3977–3987, 2010.

[68] P. Rajendran and M. Madheswaran, "Hybrid medical image classification using association rule mining with decision tree algorithm," *Journal of Computing*, vol. 2, pp. 127–136, 2010.

[69] B. B. Nair, V. P. Mohandas, and N. R. Sakthivel, "A genetic algorithm optimized decision tree-SVM based stock market trend prediction system," *International Journal on Computer Science and Engineering*, vol. 2, pp. 2981–2988, 2010.

[70] B. W. Heumann, "An object-based classification of mangroves using a hybrid decision tree-support vector machine approach," *Remote Sensing*, vol. 3, no. 11, pp. 2440–2460, 2011.