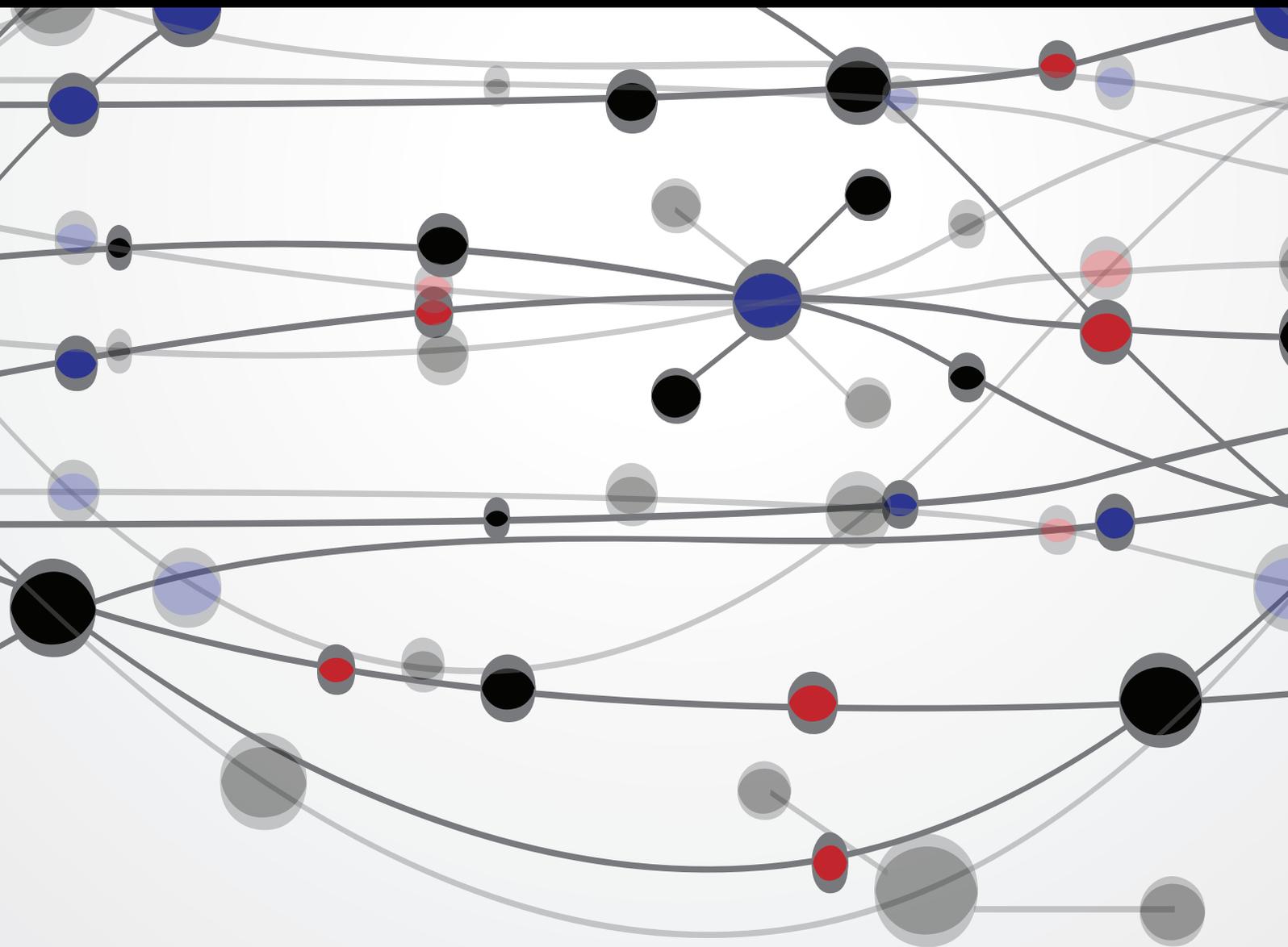


Research and Development of Advanced Computing Technologies

Guest Editors: Shifei Ding, Zhongzhi Shi, and Ahmad Taher Azar





Research and Development of Advanced Computing Technologies

The Scientific World Journal

Research and Development of Advanced Computing Technologies

Guest Editors: Shifei Ding, Zhongzhi Shi,
and Ahmad Taher Azar



Copyright © 2015 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in “The Scientific World Journal.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Contents

Research and Development of Advanced Computing Technologies, Shifei Ding, Zhongzhi Shi, and Ahmad Taher Azar
Volume 2015, Article ID 239723, 2 pages

Unbiased Feature Selection in Learning Random Forests for High-Dimensional Data, Thanh-Tung Nguyen, Joshua Zhexue Huang, and Thuy Thi Nguyen
Volume 2015, Article ID 471371, 18 pages

A Heuristic Ranking Approach on Capacity Benefit Margin Determination Using Pareto-Based Evolutionary Programming Technique, Muhammad Murtadha Othman, Nurulazmi Abd Rahman, Ismail Musirin, Mahmud Fotuhi-Firuzabad, and Abbas Rajabi-Ghahnavieh
Volume 2015, Article ID 731013, 15 pages

New Enhanced Artificial Bee Colony (JA-ABC5) Algorithm with Application for Reactive Power Optimization, Noorazliza Sulaiman, Junita Mohamad-Saleh, and Abdul Ghani Abro
Volume 2015, Article ID 396189, 11 pages

Kernel Method Based Human Model for Enhancing Interactive Evolutionary Optimization, Yan Pei, Qiangfu Zhao, and Yong Liu
Volume 2015, Article ID 185860, 12 pages

Unsupervised Spectral-Spatial Feature Selection-Based Camouflaged Object Detection Using VNIR Hyperspectral Camera, Sungho Kim
Volume 2015, Article ID 834635, 8 pages

Fast Adapting Ensemble: A New Algorithm for Mining Data Streams with Concept Drift, Agustín Ortíz Díaz, José del Campo-Ávila, Gonzalo Ramos-Jiménez, Isvani Frías Blanco, Yailé Caballero Mota, Antonio Mustelier Hechavarría, and Rafael Morales-Bueno
Volume 2015, Article ID 235810, 14 pages

Negative Correlation Learning for Customer Churn Prediction: A Comparison Study, Ali Rodan, Ayham Fayyouni, Hossam Faris, Jamal Alsakran, and Omar Al-Kadi
Volume 2015, Article ID 473283, 7 pages

An Approach to Model Based Testing of Multiagent Systems, Shafiq Ur Rehman and Aamer Nadeem
Volume 2015, Article ID 925206, 12 pages

A Novel Clustering Algorithm Inspired by Membrane Computing, Hong Peng, Xiaohui Luo, Zhisheng Gao, Jun Wang, and Zheng Pei
Volume 2015, Article ID 929471, 8 pages

Primary Path Reservation Using Enhanced Slot Assignment in TDMA for Session Admission, Suresh Koneri Chandrasekaran, Prakash Savarimuthu, Priya Andi Elumalai, and Kathirvel Ayyaswamy
Volume 2015, Article ID 405974, 11 pages

A Novel Multiobjective Evolutionary Algorithm Based on Regression Analysis, Zhiming Song, Maocai Wang, Guangming Dai, and Massimiliano Vasile
Volume 2015, Article ID 439307, 10 pages

A Novel Psychovisual Threshold on Large DCT for Image Compression,

Nur Azman Abu and Ferda Ernawan

Volume 2015, Article ID 821497, 11 pages

Chaos Time Series Prediction Based on Membrane Optimization Algorithms, Meng Li, Liangzhong Yi,

Zheng Pei, Zhisheng Gao, and Hong Peng

Volume 2015, Article ID 589093, 14 pages

Composition of Web Services Using Markov Decision Processes and Dynamic Programming,

Víctor Uc-Cetina, Francisco Moo-Mena, and Rafael Hernandez-Ucan

Volume 2015, Article ID 545308, 9 pages

Integrating Reconfigurable Hardware-Based Grid for High Performance Computing,

Julio Dondo Gazzano, Francisco Sanchez Molina, Fernando Rincon, and Juan Carlos López

Volume 2015, Article ID 272536, 19 pages

Performance Evaluation of the Machine Learning Algorithms Used in Inference Mechanism of a Medical Decision Support System, Mert Bal, M. Fatih Amasyali, Hayri Sever, Guven Kose,

and Ayse Demirhan

Volume 2014, Article ID 137896, 15 pages

A Hybrid Approach of Stepwise Regression, Logistic Regression, Support Vector Machine, and Decision Tree for Forecasting Fraudulent Financial Statements, Suduan Chen, Yeong-Jia James Goo,

and Zone-De Shen

Volume 2014, Article ID 968712, 9 pages

SVM-RFE Based Feature Selection and Taguchi Parameters Optimization for Multiclass SVM Classifier,

Mei-Ling Huang, Yung-Hsiang Hung, W. M. Lee, R. K. Li, and Bo-Ru Jiang

Volume 2014, Article ID 795624, 10 pages

Comparative Study on Interaction of Form and Motion Processing Streams by Applying Two Different Classifiers in Mechanism for Recognition of Biological Movement, Bardia Yousefi and Chu Kiong Loo

Volume 2014, Article ID 723213, 12 pages

Efficiently Hiding Sensitive Itemsets with Transaction Deletion Based on Genetic Algorithms,

Chun-Wei Lin, Binbin Zhang, Kuo-Tung Yang, and Tzung-Pei Hong

Volume 2014, Article ID 398269, 13 pages

Color Image Segmentation Based on Different Color Space Models Using Automatic GrabCut,

Dina Khattab, Hala Mousher Ebied, Ashraf Saad Hussein, and Mohamed Fahmy Tolba

Volume 2014, Article ID 126025, 10 pages

Ephedrine QoS: An Antidote to Slow, Congested, Bufferless NoCs, Juan Fang, Zhicheng Yao, Xiufeng Sui, and Yungang Bao

Volume 2014, Article ID 691865, 11 pages

Discrete Bat Algorithm for Optimal Problem of Permutation Flow Shop Scheduling, Qifang Luo,

Yongquan Zhou, Jian Xie, Mingzhi Ma, and Liangliang Li

Volume 2014, Article ID 630280, 15 pages

Contents

Feature Selection and Classifier Parameters Estimation for EEG Signals Peak Detection Using Particle Swarm Optimization, Asrul Adam, Mohd Ibrahim Shapiai, Mohd Zaidi Mohd Tumari, Mohd Saberi Mohamad, and Marizan Mubin
Volume 2014, Article ID 973063, 13 pages

A Community Detection Algorithm Based on Topology Potential and Spectral Clustering, Zhixiao Wang, Zhaotong Chen, Ya Zhao, and Shaoda Chen
Volume 2014, Article ID 329325, 9 pages

Editorial

Research and Development of Advanced Computing Technologies

Shifei Ding,^{1,2} Zhongzhi Shi,² and Ahmad Taher Azar³

¹*School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China*

²*Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*

³*Faculty of Computers and Information, Benha University, Benha 13511, Egypt*

Correspondence should be addressed to Shifei Ding; dingsf@cumt.edu.cn

Received 21 December 2014; Accepted 21 December 2014

Copyright © 2015 Shifei Ding et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Welcome to this special issue. Advanced computing technologies are one of the hot research fields in artificial intelligence. This special issue aims to promote the research, development, and applications of advanced computing technologies by providing a high-level international forum for researchers and practitioners to exchange research results and share development experiences. The papers in this edition were selected among the highest rated papers in submitted manuscripts. The selection of papers featured here covers the topics of the main advanced computing technologies and experimental studies of some application systems.

C.-W. Lin et al. proposed a privacy-preserving data mining method by using a compact prelarge GA-based algorithm to delete transactions for hiding sensitive itemsets. This method solves the limitations of the evolutionary process by adopting both the compact GA-based mechanism and the prelarge concept. A. Adam et al. build a framework for peak detection on EEG signals in time domain analysis using feature selection and classifier parameters estimation based on particle swarm optimization. The proposed framework tries to find the best combination for all the available features that offers good peak detection and a high classification rate from the results in the conducted experiments. Y. Pei et al. propose a method to establish a human model in high dimensional search space by kernel classification for enhancing interactive evolutionary computation search. N. Sulaiman et al. propose a modified artificial bee colony algorithm to enhance the convergence speed and improve

the ability of the standard artificial bee colony algorithm to reach the global optimum by balancing the exploration and exploitation processes. This method was tested on the reactive power optimization problem and has outperformed other compared algorithms. Aiming at solving the biased feature selection in random forests, T. T. Nguyen et al. propose a modified random forests algorithm to select good features in learning random forests for high dimensional data. L. Li et al. put forward a prediction model based on membrane computing optimization algorithm for chaos time series; the model optimizes simultaneously the parameters of phase space reconstruction and least squares support vector machine by using membrane computing optimization algorithm. M. R. Al-Othman et al. propose a heuristic ranking approach on capacity benefit margin (CBM) determination using Pareto-based evolutionary programming technique. This paper introduces a novel multiobjective approach for CBM assessment taking into account tie-line reliability of interconnected systems and presents a new Pareto-based evolutionary programming (EP) technique used to perform a simultaneous determination of CBM for all areas of the interconnected system.

In order to build a QoS-aware bufferless network-on-chip scheme for datacenters, J. Fang et al. propose QBLESS, a QoS-aware bufferless NoC scheme for datacenters. QBLESS consists of two components: a routing mechanism (QBLESS-R) that can substantially reduce flit deflection for high-priority applications and a congestion-control mechanism

(QBLESS-CC) that guarantees performance for high-priority applications and improves overall system throughput. S. K. Chandrasekaran et al. propose a framework of primary path reservation admission control protocol, which achieves improved QoS by making use of backup route combined with resource reservation. In this paper, a network topology has been simulated and the present approach proves to be a mechanism that admits the session effectively. V. Uc-Cetina et al. propose a Markov decision process model for solving the web service composition problem. Iterative policy evaluation, value iteration, and policy iteration algorithms are used to experimentally validate the present approach. S. U. Rehman and A. Nadeem propose a novel approach to the testing of multiagent systems based on Prometheus design artifacts. In the proposed approach, different interactions between the agent and actors are considered to test the multiagent system. M. Bal et al. study 11 machine learning methods (SVM, MLP, C4.5, etc.) for the inference mechanism of medical decision support system based on ALARM network. The performances of 11 machine learning algorithms are tested on 44 synthetic data sets (11 different dependent variables and 4 different dataset sizes). The comparison of algorithms applied two different tests (statistically difference and average rank tests). C4.5 decision tree is the best algorithm according to both of the tests for our 44 datasets. The datasets having more samples can be better predicted than having fewer samples.

A comparative study on interaction of form and motion processing streams by applying two different classifiers in mechanism for recognition of biological movement is proposed by B. Yousefi and C. K. Loo. The presented approach has addressed a very substantial interrelevant comparison of the interaction of two processing streams of mammalian brain visual system. For mining data streams, A. O. Diaz et al. present a fast adapting ensemble method which adapts very quickly to both abrupt and gradual concept drifts, and the method has been specifically designed to deal with recurring concepts. After initializing color image by utilizing the unsupervised Orchard and Bouman clustering technique, D. Khattab et al. present a comparative study using different color spaces to evaluate the performance of color image segmentation using the automatic GrabCut technique. M.-L. Huang et al. combine feature selection and SVM recursive feature elimination to investigate the classification accuracy of multiclass problems for Dermatology and Zoo databases. Meanwhile, Taguchi's method was jointly combined with SVM classifier in order to optimize parameters to increase classification accuracy for multiclass classification.

For image compression, N. A. Abu and F. Ernawan propose a psychovisual threshold on the large discrete cosine transform (DCT) image block which will be used to automatically generate the much needed quantization tables. G. C. Kim proposes a fully autonomous feature selection and camouflaged object detection method based on the online analysis of spectral and spatial features.

Z. Wang et al. point out that spectral clustering algorithms applied in community detection have two inadequacies and present a novel community detection algorithm based on topology potential and spectral clustering that contains rich structural information of the network. A. Rodan et al. utilize

an ensemble of multilayer perceptrons whose training is obtained using negative correlation learning for predicting customer churn in a telecommunication company. S. Chen et al. describe a hybrid approach for forecasting fraudulent financial statements. The authors firstly screen the important variables using the stepwise regression and then use logistic regression, support vector machine, and decision tree to construct the classification models to make a comparison.

J. D. Gazzano et al. propose a complete grid infrastructure for distributed high-performance computing based on dynamically reconfigurable FPGAs. This infrastructure was tested and significant performance gains have been achieved. P systems are a class of distributed parallel computing models, H. Peng et al. present a novel clustering algorithm, which is inspired from mechanism of a tissue-like P system with a loop structure of cells, called membrane clustering algorithm. Q. Luo et al. propose a discrete bat algorithm (DBA) for optimal problem of permutation flow shop scheduling. The authors construct a direct relationship between the job sequence and the vector of individuals in DBA.

Acknowledgments

As guest editors of this issue, we would like to thank all the authors for submitting their work and the all the reviewers for their indispensable contribution in finalizing this issue. We are very pleased to have edited this special issue and we sincerely hope that the readers will find interesting ideas in it and enjoy reading these papers. This special issue is supported by the National Natural Science Foundation of China under Grant no. 61379101 and the National Basic Research Program of China under Grant no. 2013CB329502.

*Shifei Ding
Zhongzhi Shi
Ahmad Taher Azar*

Research Article

Unbiased Feature Selection in Learning Random Forests for High-Dimensional Data

Thanh-Tung Nguyen,^{1,2,3} Joshua Zhexue Huang,^{1,4} and Thuy Thi Nguyen⁵

¹ Shenzhen Key Laboratory of High Performance Data Mining, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

² University of Chinese Academy of Sciences, Beijing 100049, China

³ School of Computer Science and Engineering, Water Resources University, Hanoi 10000, Vietnam

⁴ College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

⁵ Faculty of Information Technology, Vietnam National University of Agriculture, Hanoi 10000, Vietnam

Correspondence should be addressed to Thanh-Tung Nguyen; tungnt@wru.vn

Received 20 June 2014; Accepted 20 August 2014

Academic Editor: Shifei Ding

Copyright © 2015 Thanh-Tung Nguyen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Random forests (RFs) have been widely used as a powerful classification method. However, with the randomization in both bagging samples and feature selection, the trees in the forest tend to select uninformative features for node splitting. This makes RFs have poor accuracy when working with high-dimensional data. Besides that, RFs have bias in the feature selection process where multivalued features are favored. Aiming at debiasing feature selection in RFs, we propose a new RF algorithm, called xRF, to select good features in learning RFs for high-dimensional data. We first remove the uninformative features using p -value assessment, and the subset of unbiased features is then selected based on some statistical measures. This feature subset is then partitioned into two subsets. A feature weighting sampling technique is used to sample features from these two subsets for building trees. This approach enables one to generate more accurate trees, while allowing one to reduce dimensionality and the amount of data needed for learning RFs. An extensive set of experiments has been conducted on 47 high-dimensional real-world datasets including image datasets. The experimental results have shown that RFs with the proposed approach outperformed the existing random forests in increasing the accuracy and the AUC measures.

1. Introduction

Random forests (RFs) [1] are a nonparametric method that builds an ensemble model of decision trees from random subsets of features and bagged samples of the training data.

RFs have shown excellent performance for both classification and regression problems. RF model works well even when predictive features contain irrelevant features (or noise); it can be used when the number of features is much larger than the number of samples. However, with randomizing mechanism in both bagging samples and feature selection, RFs could give poor accuracy when applied to high dimensional data. The main cause is that, in the process of growing a tree from the bagged sample data, the subspace of features randomly sampled from thousands of features to

split a node of the tree is often dominated by uninformative features (or noise), and the tree grown from such bagged subspace of features will have a low accuracy in prediction which affects the final prediction of the RFs. Furthermore, Breiman et al. noted that feature selection is biased in the classification and regression tree (CART) model because it is based on an information criteria, called multivalued problem [2]. It tends in favor of features containing more values, even if these features have lower importance than other ones or have no relationship with the response feature (i.e., containing less missing values, many categorical or distinct numerical values) [3, 4].

In this paper, we propose a new random forests algorithm using an unbiased feature sampling method to build a good subspace of unbiased features for growing trees.

We first use random forests to measure the importance of features and produce raw feature importance scores. Then, we apply a statistical Wilcoxon rank-sum test to separate informative features from the uninformative ones. This is done by neglecting all uninformative features by defining threshold θ ; for instance, $\theta = 0.05$. Second, we use the Chi-square statistic test (χ^2) to compute the related scores of each feature to the response feature. We then partition the set of the remaining informative features into two subsets, one containing highly informative features and the other one containing weak informative features. We independently sample features from the two subsets and merge them together to get a new subspace of features, which is used for splitting the data at nodes. Since the subspace always contains highly informative features which can guarantee a better split at a node, this feature sampling method enables avoiding selecting biased features and generates trees from bagged sample data with higher accuracy. This sampling method also is used for dimensionality reduction, the amount of data needed for training the random forests model. Our experimental results have shown that random forests with this weighting feature selection technique outperformed recently the proposed random forests in increasing of the prediction accuracy; we also applied the new approach on microarray and image data and achieved outstanding results.

The structure of this paper is organized as follows. In Section 2, we give a brief summary of related works. In Section 3 we give a brief summary of random forests and measurement of feature importance score. Section 4 describes our new proposed algorithm using unbiased feature selection. Section 5 provides the experimental results, evaluations, and comparisons. Section 6 gives our conclusions.

2. Related Works

Random forests are an ensemble approach to make classification decisions by voting the results of individual decision trees. An ensemble learner with excellent generalization accuracy has two properties, high accuracy of each component learner and high diversity in component learners [5]. Unlike other ensemble methods such as bagging [1] and boosting [6, 7], which create basic classifiers from random samples of the training data, the random forest approach creates the basic classifiers from randomly selected subspaces of data [8, 9]. The randomly selected subspaces increase the diversity of basic classifiers learnt by a decision tree algorithm.

Feature importance is the importance measure of features in the feature selection process [1, 10–14]. In RF frameworks, the most commonly used score of importance of a given feature is the mean error of a tree in the forest when the observed values of this feature are randomly permuted in the *out-of-bag* samples. Feature selection is an important step to obtain good performance for an RF model, especially in dealing with high dimensional data problems.

For feature weighting techniques, recently Xu et al. [13] proposed an improved RF method which uses a novel feature weighting method for subspace selection and therefore

enhances classification performance on high dimensional data. The weights of feature were calculated by information gain ratio or χ^2 -test; Ye et al. [14] then used these weights to propose a stratified sampling method to select feature subspaces for RF in classification problems. Chen et al. [15] used a stratified idea to propose a new clustering method. However, implementation of the random forest model suggested by Ye et al. is based on a binary classification setting, and it uses linear discriminant analysis as the splitting criteria. This stratified RF model is not efficient on high dimensional datasets with multiple classes. With the same way for solving two-class problem, Amaratunga et al. [16] presented a feature weighting method for subspace sampling to deal with microarray data, the *t*-test of variance analysis is used to compute weights for the features. Genuer et al. [12] proposed a strategy involving a ranking of explanatory features using the RFs score weights of importance and a stepwise ascending feature introduction strategy. Deng and Runger [17] proposed a guided regularized RF (GRRF), in which weights of importance scores from an ordinary random forest (RF) are used to guide the feature selection process. They found that the regularized least subset selected by their GRRF with minimal regularization ensures better accuracy than the complete feature set. However, regular RF was used as a classifier due to the fact that regularized RF may have higher variance than RF because the trees are correlated.

Several methods have been proposed to correct bias of importance measures in the feature selection process in RFs to improve the prediction accuracy [18–21]. These methods intend to avoid selecting an uninformative feature for node splitting in decision trees. Although the methods of this kind were well investigated and can be used to address the high dimensional problem, there are still some unsolved problems, such as the need to specify in advance the probability distributions, as well as the fact that they struggle when applied to large high dimensional data.

In summary, in the reviewed approaches, the gain at higher levels of the tree is weighted differently than the gain at lower levels of the tree. In fact, at lower levels of the tree, the gain is reduced because of the effect of splits on different features at higher levels of the tree. That affects the final prediction performance of RFs model. To remedy this, in this paper we propose a new method for unbiased feature subsets selection in high dimensional space to build RFs. Our approach differs from previous approaches in the techniques used to partition a subset of features. All uninformative features (considered as noise) are removed from the system and the best feature set, which is highly related to the response feature, is found using a statistical method. The proposed sampling method always provides enough highly informative features for the subspace feature at any levels of the decision trees. For the case of growing an RF model on data without noise, we used *in-bag* measures. This is a different importance score of features, which requires less computational time compared to the measures used by others. Our experimental results showed that our approach outperformed recently the proposed RF methods.

input: $\mathbb{L} = \{(X_i, Y_i)_{i=1}^N \mid X \in \mathbb{R}^M, Y \in \{1, 2, \dots, c\}\}$: the training data set,
 K : the number of trees,
 $mtry$: the size of the subspaces.

output: A random forest RF

- (1) **for** $k \leftarrow 1$ **to** K **do**
- (2) Draw a bagged subset of samples \mathbb{L}_k from \mathbb{L} .
- (4) **while** (*stopping criteria is not met*) **do**
- (5) Select randomly $mtry$ features.
- (6) **for** $m \leftarrow 1$ **to** $\|mtry\|$ **do**
- (7) Compute the decrease in the node impurity.
- (8) Choose the feature which decreases the impurity the most and the node is divided into two children nodes.
- (9) Combine the K trees to form a random forest.

ALGORITHM 1: Random forest algorithm.

3. Background

3.1. Random Forest Algorithm. Given a training dataset $\mathbb{L} = \{(X_i, Y_i)_{i=1}^N \mid X_i \in \mathbb{R}^M, Y \in \{1, 2, \dots, c\}\}$, where X_i are features (also called predictor variables), Y is a class response feature, N is the number of training samples, and M is the number of features and a random forest model RF described in Algorithm 1, let \hat{Y}^k be the prediction of tree T_k given input X . The prediction of random forest with K trees is

$$\hat{Y} = \text{majority vote } \{\hat{Y}^k\}_1^K. \quad (1)$$

Since each tree is grown from a bagged sample set, it is grown with only two-thirds of the samples in \mathbb{L} , called *in-bag* samples. About one-third of the samples is left out and these samples are called *out-of-bag* (OOB) samples which are used to estimate the prediction error.

The OOB predicted value is $\hat{Y}^{\text{OOB}} = (1/\|\mathcal{O}_{i'}\|) \sum_{k \in \mathcal{O}_{i'}} \hat{Y}^k$ where $\mathcal{O}_{i'} = \mathbb{L} \setminus \mathcal{O}_i$, i and i' are in-bag and out-of-bag sampled indices, $\|\mathcal{O}_{i'}\|$ is the size of OOB subdataset, and the OOB prediction error is

$$\widehat{\text{Err}}^{\text{OOB}} = \frac{1}{N_{\text{OOB}}} \sum_{i=1}^{N_{\text{OOB}}} \mathcal{E}(Y, \hat{Y}^{\text{OOB}}), \quad (2)$$

where $\mathcal{E}(\cdot)$ is an error function and N_{OOB} is OOB samples' size.

3.2. Measurement of Feature Importance Score from an RF. Breiman presented a permutation technique to measure the importance of features in the prediction [1], called an *out-of-bag* importance score. The basic idea for measuring this kind of importance score of features is to compute the difference between the original mean error and the randomly permuted mean error in OOB samples. The method rearranges stochastically all values of the j th feature in OOB for each tree and uses the RF model to predict this permuted feature and get the mean error. The aim of this permutation is to eliminate the existing association between the j th feature and Y values

and then to test the effect of this on the RF model. A feature is considered to be in a strong association if the mean error decreases dramatically.

The other kind of feature importance measure can be obtained when the random forest is growing. This is described as follows. At each node t in a decision tree, the split is determined by the decrease in node impurity $\Delta R(t)$. The node impurity $R(t)$ is the gini index. If a subdataset in node t contains samples from c classes, $\text{gini}(t)$ is defined as

$$R(t) = 1 - \sum_{j=1}^c \hat{p}_j^2, \quad (3)$$

where \hat{p}_j^2 is the relative frequency of class j in t . $\text{Gini}(t)$ is minimized if the classes in t are skewed. After splitting t into two child nodes t_1 and t_2 with sample sizes $N_1(t)$ and $N_2(t)$, the gini index of the split data is defined as

$$\text{Gini}_{\text{split}}(t) = \frac{N_1(t)}{N(t)} \text{Gini}(t_1) + \frac{N_2(t)}{N(t)} \text{Gini}(t_2). \quad (4)$$

The feature providing smallest $\text{Gini}_{\text{split}}(t)$ is chosen to split the node. The importance score of feature X_j in a single decision tree T_k is

$$\text{IS}_k(X_j) = \sum_{t \in T_k} \Delta R(t), \quad (5)$$

and it is computed over all K trees in a random forest, defined as

$$\text{IS}(X_j) = \frac{1}{K} \sum_{k=1}^K \text{IS}_k(X_j). \quad (6)$$

It is worth noting that a random forest uses *in-bag* samples to produce a kind of importance measure, called an *in-bag* importance score. This is the main difference between the *in-bag* importance score and an *out-of-bag* measure, which is produced with the decrease of the prediction error using RF in OOB samples. In other words, the *in-bag* importance score requires less computation time than the *out-of-bag* measure.

4. Our Approach

4.1. *Issues in Feature Selection on High Dimensional Data.* When Breiman et al. suggested the classification and regression tree (CART) model, they noted that feature selection is biased because it is based on an information gain criteria, called multivalued problem [2]. Random forest methods are based on CART trees [1]; hence this bias is carried to random forest RF model. In particular, the importance scores can be biased when very high dimensional data contains multiple data types. Several methods have been proposed to correct bias of feature importance measures [18–21]. The conditional inference framework (referred to as cRF [22]) could be successfully applied for both the null and power cases [19, 20, 22]. The typical characteristic of the power case is that only one predictor feature is important, while the rest of the features are redundant with different cardinality. In contrast, in the null case all features used for prediction are redundant with different cardinality. Although the methods of this kind were well investigated and can be used to address the multivalued problem, there are still some unsolved problems, such as the need to specify in advance the probability distributions, as well as the fact that they struggle when applied to high dimensional data.

Another issue is that, in high dimensional data, when the number of features is large, the fraction of importance features remains so small. In this case the original RF model which uses simple random sampling is likely to perform poorly with small m , and the trees are likely to select an uninformative feature as a split too frequently (m denotes a subspace size of features). At each node t of a tree, the probability of uninformative feature selection is too high.

To illustrate this issue, let G be the number of noisy features, denote by M the total number of predictor features, and let the features $M - G$ be important ones which have a high correlation with Y values. Then, if we use simple random sampling when growing trees to select a subset of m features ($m \ll M$), the total number of possible uninformative a \mathcal{C}_{M-G}^m and the total number of all subset features is \mathcal{C}_M^m . The probability distribution of selecting a subset of m ($m > 1$) important features is given by

$$\begin{aligned} \frac{\mathcal{C}_{M-G}^m}{\mathcal{C}_M^m} &= \frac{(M - G)(M - G - 1) \cdots (M - G - m + 1)}{M(M - 1) \cdots (M - m + 1)} \\ &= \frac{(1 - G/M) \cdots (1 - G/M - m/M + 1/M)}{(1 - 1/M) \cdots (1 - m/M + 1/M)} \quad (7) \\ &\approx \left(1 - \frac{G}{M}\right)^m. \end{aligned}$$

Because the fraction of important features is too small, the probability in (7) tends to 0, which means that the important features are rarely selected by the simple sampling method in RF [1]. For example, with 5 informative and 5000 noise or uninformative features, assuming $m = \sqrt{(5 + 5000)} \approx 70$, the probability of an informative feature to be selected at any split is 0.068.

4.2. *Bias Correction for Feature Selection and Feature Weighting.* The bias correction in feature selection is intended to make the RF model to avoid selecting an uninformative feature. To correct this kind of bias in the feature selection stage, we generate shadow features to add to the original dataset. The shadow features set contains the same values, possible cut-points, and distribution with the original features but have no association with Y values. To create each shadow feature, we rearrange the values of the feature in the original dataset R times to create the corresponding shadow. This disturbance of features eliminates the correlations of the features with the response value but keeps its attributes. The shadow feature participates only in the competition for the best split and makes a decrease in the probability of selecting this kind of uninformative feature. For the feature weight computation, we first need to distinguish the important features from the less important ones. To do so, we run a defined number of random forests to obtain raw importance scores, each of which is obtained using (6). Then, we use Wilcoxon rank-sum test [23] that compares the importance score of a feature with the maximum importance scores of generated noisy features called shadows. The shadow features are added to the original dataset and they do not have prediction power to the response feature. Therefore, any feature whose importance score is smaller than the maximum importance score of noisy features is considered less important. Otherwise, it is considered important. Having computed the Wilcoxon rank-sum test, we can compute the p -value for the feature. The p -value of a feature in Wilcoxon rank-sum test is assigned a weight with a feature X_j , p -value $\in [0, 1]$, and this weight indicates the importance of the feature in the prediction. The smaller the p -value of a feature, the more correlated the predictor feature to the response feature, and therefore the more powerful the feature in prediction. The feature weight computation is described as follows.

Let M be the number of features in the original dataset, and denote the feature set as $\mathbb{S}_X = \{X_j, j = 1, 2, \dots, M\}$. In each replicate r ($r = 1, 2, \dots, R$), shadow features are generated from features X_j in \mathbb{S}_X , and we randomly permute all values of X_j R times to get a corresponding shadow feature A_j ; denote the shadow feature set as $\mathbb{S}_A = \{A_j\}_1^M$. The extended feature set is denoted by $\mathbb{S}_{X,A} = \{\mathbb{S}_X, \mathbb{S}_A\}$.

Let the importance score of $\mathbb{S}_{X,A}$ at replicate r be $IS_{X,A}^r = \{IS_X^r, IS_A^r\}$ where $IS_{X_j}^r$ and $IS_{A_j}^r$ are the importance scores of X_j and A_j at the r th replicate, respectively. We built a random forest model RF from the $\mathbb{S}_{X,A}$ dataset to compute $2M$ importance scores for $2M$ features. We repeated the same process R times to compute R replicates getting $IS_{X_j} = \{IS_{X_j}^r\}_1^R$ and $IS_{A_j} = \{IS_{A_j}^r\}_1^R$. From the replicates of shadow features, we extracted the maximum value from r th row of IS_{A_j} and put it into the comparison sample denoted by IS_A^{\max} . For each data feature X_j , we computed Wilcoxon test and performed hypothesis test on $\overline{IS}_{X_j} > \overline{IS}_A^{\max}$ to calculate the p -value for the feature. Given a statistical significance level, we can identify important features from less important ones. This test confirms that if a feature is important, it consistently

scores higher than the shadow over multiple permutations. This method has been presented in [24, 25].

In each node of trees, each shadow A_j shares approximately the same properties of the corresponding X_j , but it is independent on Y and consequently has approximately the same probability of being selected as a splitting candidate. This feature permutation method can reduce bias due to different measurement levels of X_j according to p -value and can yield correct ranking of features according to their importance.

4.3. Unbiased Feature Weighting for Subspace Selection. Given all p -values for all features, we first set a significance level as the threshold θ , for instance $\theta = 0.05$. Any feature whose p -value is greater than θ is considered a uninformative feature and is removed from the system; otherwise, the relationship with Y is assessed. We now consider the set of features \bar{X} obtained from \mathbb{L} after neglecting all uninformative features.

Second, we find the best subset of features which is highly related to the response feature; a measure correlation function $\chi^2(\bar{X}, Y)$ is used to test the association between the categorical response feature and each feature X_j . Each observation is allocated to one cell of a two-dimensional array of cells (called a contingency table) according to the values of (\bar{X}, Y) . If there are r rows and c columns in the table and N is the number of total samples, the value of the test statistic is

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}. \quad (8)$$

For the test of independence, a chi-squared probability of less than or equal to 0.05 is commonly interpreted for rejecting the hypothesis that the row variable is independent of the column feature.

Let \mathbf{X}_s be the best subset of features, we collect all feature X_j whose p -value is smaller or equal to 0.05 as a result from the χ^2 statistical test according to (8). The remaining features $\{\bar{X} \setminus \mathbf{X}_s\}$ are added to \mathbf{X}_w , and this approach is described in Algorithm 2. We independently sample features from the two subsets and put them together as the subspace features for splitting the data at any node, recursively. The two subsets partition the set of informative features in data without irrelevant features. Given \mathbf{X}_s and \mathbf{X}_w , at each node, we randomly select $mtry$ ($mtry > 1$) features from each group of features. For a given subspace size, we can choose proportions between highly informative features and weak informative features that depend on the size of the two groups. That is $mtry_s = \lceil mtry \times (\|\mathbf{X}_s\| / \|\bar{X}\|) \rceil$ and $mtry_w = \lfloor mtry \times (\|\mathbf{X}_w\| / \|\bar{X}\|) \rfloor$, where $\|\mathbf{X}_s\|$ and $\|\mathbf{X}_w\|$ are the number of features in the groups of highly informative features \mathbf{X}_s and weak informative features \mathbf{X}_w , respectively. $\|\bar{X}\|$ is the number of informative features in the input dataset. These are merged to form the feature subspace for splitting the node.

4.4. Our Proposed RF Algorithm. In this section, we present our new random forest algorithm called xRF, which uses the new unbiased feature sampling method to generate splits

at the nodes of CART trees [2]. The proposed algorithm includes the following main steps: (i) weighting the features using the feature permutation method, (ii) identifying all unbiased features and partitioning them into two groups \mathbf{X}_s and \mathbf{X}_w , (iii) building RF using the subspaces containing features which are taken randomly and separately from \mathbf{X}_s , \mathbf{X}_w , and (iv) classifying a new data. The new algorithm is summarized as follows.

- (1) Generate the extended dataset $\mathbb{S}_{\mathbf{X},A}$ of $2M$ dimensions by permuting the corresponding predictor feature values for shadow features.
- (2) Build a random forest model RF from $\{\mathbb{S}_{\mathbf{X},A}, Y\}$ and compute R replicates of raw importance scores of all predictor features and shadows with RF. Extract the maximum importance score of each replicate to form the comparison sample IS_A^{\max} of R elements.
- (3) For each predictor feature, take R importance scores and compute Wilcoxon test to get p -value, that is, the weight of each feature.
- (4) Given a significance level threshold θ , neglect all uninformative features.
- (5) Partition the remaining features into two subsets \mathbf{X}_s and \mathbf{X}_w described in Algorithm 2.
- (6) Sample the training set \mathbb{L} with replacement to generate bagged samples $\mathbb{L}_1, \mathbb{L}_2, \dots, \mathbb{L}_K$.
- (7) For each L_k , grow a CART tree T_k as follows.
 - (a) At each node, select a subspace of $mtry$ ($mtry > 1$) features randomly and separately from \mathbf{X}_s and \mathbf{X}_w and use the subspace features as candidates for splitting the node.
 - (b) Each tree is grown nondeterministically, without pruning until the minimum node size n_{\min} is reached.
- (8) Given a $X = x_{\text{new}}$, use (1) to predict the response value.

5. Experiments

5.1. Datasets. Real-world datasets including image datasets and microarray datasets were used in our experiments. Image classification and object recognition are important problems in computer vision. We conducted experiments on four benchmark image datasets, including the *Caltech* categories (<http://www.vision.caltech.edu/html-files/archive.html>) dataset, the *Horse* (<http://pascal.inrialpes.fr/data/horses/>) dataset, the extended *YaleB* database [26], and the *AT&T ORL* dataset [27].

For the *Caltech* dataset, we use a subset of 100 images from the *Caltech* face dataset and 100 images from the *Caltech* background dataset following the setting in ICCV (<http://people.csail.mit.edu/torralba/shortCourseRLOC/>). The extended *YaleB* database consists of 2414 face images of 38 individuals captured under various lighting conditions. Each image has been cropped to a size of 192×168 pixels

```

input: The training data set  $\mathbb{L}$  and a random forest RF
           $R, \theta$ : The number of replicates and the threshold.
output:  $\mathbf{X}_s$  and  $\mathbf{X}_w$ .
(1) Let  $\mathbb{S}_X = \{\mathbb{L} \setminus Y\}$ ,  $M = \|\mathbb{S}_X\|$ .
(2) for  $r \leftarrow 1$  to  $R$  do
(3)    $\mathbb{S}_A \leftarrow \text{permute}(\mathbb{S}_X)$ .
(4)    $\mathbb{S}_{X,A} = \mathbb{S}_X \cup \mathbb{S}_A$ .
(5)   Build RF model from  $\mathbb{S}_{X,A}$  to produce  $\{\text{IS}_{X_j}^r\}$ ,
(6)    $\{\text{IS}_{A_j}^r\}$  and  $\text{IS}_A^{\max}$ , ( $j = 1, \dots, M$ ).
(7) Set  $\bar{\mathbf{X}} = \emptyset$ .
(8) for  $j \leftarrow 1$  to  $M$  do
(9)   Compute Wilcoxon rank-sum test with  $\text{IS}_{X_j}$  and  $\text{IS}_A^{\max}$ .
(10)  Compute  $p_j$  values for each feature  $X_j$ .
(11)  if  $p_j \leq \theta$  then
(12)    $\bar{\mathbf{X}} = \bar{\mathbf{X}} \cup X_j$  ( $X_j \in \mathbb{S}_X$ )
(13) Set  $\mathbf{X}_s = \emptyset$ ,  $\mathbf{X}_w = \emptyset$ .
(14) Compute  $\chi^2(\bar{\mathbf{X}}, Y)$  statistic to get  $p_j$  value
(15) for  $j \leftarrow 1$  to  $\|\bar{\mathbf{X}}\|$  do
(16)  if ( $p_j < 0.05$ ) then
(17)    $\mathbf{X}_s = \mathbf{X}_s \cup X_j$  ( $X_j \in \bar{\mathbf{X}}$ )
(18)  $\mathbf{X}_w = \{\bar{\mathbf{X}} \setminus \mathbf{X}_s\}$ 
(19) return  $\mathbf{X}_s, \mathbf{X}_w$ 

```

ALGORITHM 2: Feature subspace selection.

and normalized. The *Horse* dataset consists of 170 images containing horses for the positive class and 170 images of the background for the negative class. The *AT&T ORL* dataset includes of 400 face images of 40 persons.

In the experiments, we use a bag of words for image features representation for the *Caltech* and the *Horse* datasets. To obtain feature vectors using bag-of-words method, image patches (subwindows) are sampled from the training images at the detected interest points or on a dense grid. A visual descriptor is then applied to these patches to extract the local visual features. A clustering technique is then used to cluster these, and the cluster centers are used as visual code words to form visual codebook. An image is then represented as a histogram of these visual words. A classifier is then learned from this feature set for classification.

In our experiments, traditional k -means quantization is used to produce the visual codebook. The number of cluster centers can be adjusted to produce the different vocabularies, that is, dimensions of the feature vectors. For the *Caltech* and *Horse* datasets, nine codebook sizes were used in the experiments to create 18 datasets as follows: $\{\text{CaltechM300}, \text{CaltechM500}, \text{CaltechM1000}, \text{CaltechM3000}, \text{CaltechM5000}, \text{CaltechM7000}, \text{CaltechM1000}, \text{CaltechM12000}, \text{CaltechM15000}\}$, and $\{\text{HorseM300}, \text{HorseM500}, \text{HorseM1000}, \text{HorseM3000}, \text{HorseM5000}, \text{HorseM7000}, \text{HorseM1000}, \text{HorseM12000}, \text{HorseM15000}\}$, where M denotes the number of codebook sizes.

For the face datasets, we use two type of features: eigenface [28] and the random features (randomly sample pixels from the images). We used four groups of datasets with four different numbers of dimensions $\{M30, M56, M120, \text{ and } M504\}$. Totally, we created 16 subdatasets as

TABLE 1: Description of the real-world datasets sorted by the number of features and grouped into two groups, microarray data and real-world datasets, accordingly.

Dataset	No. of features	No. of training	No. of tests	No. of classes
Colon	2,000	62	—	2
Srbct	2,308	63	—	4
Leukemia	3,051	38	—	2
Lymphoma	4,026	62	—	3
breast.2.class	4,869	78	—	2
breast.3.class	4,869	96	—	3
nci	5,244	61	—	8
Brain	5,597	42	—	5
Prostate	6,033	102	—	2
adenocarcinoma	9,868	76	—	2
Fbis	2,000	1,711	752	17
La2s	12,432	1,855	845	6
La1s	13,195	1,963	887	6

$\{\text{YaleB.EigenfaceM30}, \text{YaleB.EigenfaceM56}, \text{YaleB.EigenfaceM120}, \text{YaleB.EigenfaceM504}\}$, $\{\text{YaleB.RandomfaceM30}, \text{YaleB.RandomfaceM56}, \text{YaleB.RandomfaceM120}, \text{YaleB.RandomfaceM504}\}$, $\{\text{ORL.EigenfaceM30}, \text{ORL.EigenM56}, \text{ORL.EigenM120}, \text{ORL.EigenM504}\}$, and $\{\text{ORL.RandomfaceM30}, \text{ORL.RandomM56}, \text{ORL.RandomM120}, \text{ORL.RandomM504}\}$.

The properties of the remaining datasets are summarized in Table 1. The Fbis dataset was compiled from the archive of the Foreign Broadcast Information Service and the *La1s*, *La2s*

datasets were taken from the archive of the Los Angeles Times for TREC-5 (<http://trec.nist.gov/>). The ten gene datasets are used and described in [11, 17]; they are always high dimensional and fall within a category of classification problems which deal with large number of features and small samples. Regarding the characteristics of the datasets given in Table 1, the proportion of the subdatasets, namely, *Fbis*, *La1s*, *La2s*, was used individually for a training and testing dataset.

5.2. Evaluation Methods. We calculated some measures such as error bound ($c/s2$), strength (s), and correlation ($\bar{\rho}$) according to the formulas given in Breiman's method [1]. The correlation measures indicate the independence of trees in a forest, whereas the average strength corresponds to the accuracy of individual trees. Lower correlation and higher strength result in a reduction of general error bound measured by ($c/s2$) which indicates a high accuracy RF model.

The two measures are also used to evaluate the accuracy of prediction on the test datasets: one is the area under the curve (AUC) and the other one is the test accuracy (Acc), defined as

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^N I \left(Q(d_i, y_i) - \max_{j \neq y_i} Q(d_i, j) > 0 \right), \quad (9)$$

where $I(\cdot)$ is the indicator function and $Q(d_i, j) = \sum_{k=1}^K I(h_k(d_i) = j)$ is the number of votes for $d_i \in \mathbb{D}_t$ on class j , h_k is the k th tree classifier, N is the number of samples in test data \mathbb{D}_t , and y_i indicates the true class of d_i .

5.3. Experimental Settings. The latest R -packages random Forest and RRF [29, 30] were used in R environment to conduct these experiments. The GRRF model was available in the RRF R -package. The wsRF model, which used weighted sampling method [13] was intended to solve classification problems. For the image datasets, the 10-fold cross-validation was used to evaluate the prediction performance of the models. From each fold, we built the models with 500 trees and the feature partition for subspace selection in Algorithm 2 was recalculated on each training fold dataset. The $mtry$ and n_{\min} parameters were set to \sqrt{M} and 1, respectively. The experimental results were evaluated in two measures AUC and the test accuracy according to (9).

We compared across a wide range the performances of the 10 gene datasets, used in [11]. The results from the application of GRRE, varSelRF, and LASSO logistic regression on the ten gene datasets are presented in [17]. These three gene selection methods used RF R -package [30] as the classifier. For the comparison of the methods, we used the same settings which are presented in [17], for the coefficient γ we used value of 0.1, because GR-RF(0.1) has shown competitive accuracy [17] when applied to the 10 gene datasets. The 100 models were generated with different seeds from each training dataset and each model contained 1000 trees. The $mtry$ and n_{\min} parameters were of the same settings on the image dataset. From each of the datasets two-thirds of the data were randomly selected for training. The other one-third of the dataset was used to validate the models. For

comparison, Breiman's RF method, the weighted sampling random forest wsRF model, and the xRF model were used in the experiments. The guided regularized random forest GRRF [17] and the two well-known feature selection methods using RF as a classifier, namely, *varSelRF* [31] and *LASSO logistic regression* [32], are also used to evaluate the accuracy of prediction on high-dimensional datasets.

In the remaining datasets, the prediction performances of the ten random forest models were evaluated, each one was built with 500 trees. The number of features candidates to split a node was $mtry = \lceil \log_2(M) + 1 \rceil$. The minimal node size n_{\min} was 1. The xRF model with the new unbiased feature sampling method is a new implementation. We implemented the xRF model as multithread processes, while other models were run as single-thread processes. We used R to call the corresponding C/C++ functions. All experiments were conducted on the six 64-bit Linux machines, with each one being equipped with Intel R Xeon R CPU E5620 2.40 GHz, 16 cores, 4 MB cache, and 32 GB main memory.

5.4. Results on Image Datasets. Figures 1 and 2 show the average accuracy plots of recognition rates of the models on different subdatasets of the datasets *YaleB* and *ORL*. The GRRF model produced slightly better results on the subdataset *ORL.RandomM120* and *ORL* dataset using eigenface and showed competitive accuracy performance with the xRF model on some cases in both *YaleB* and *ORL* datasets, for example, *YaleB.EigenM120*, *ORL.RandomM56*, and *ORL.RandomM120*. The reason could be that truly informative features in this kind of datasets were many. Therefore, when the informative feature set was large, the chance of selecting informative features in the subspace increased, which in turn increased the average recognition rates of the GRRF model. However, the xRF model produced the best results in the remaining cases. The effect of the new approach for feature subspace selection is clearly demonstrated in these results, although these datasets are not high dimensional.

Figures 3 and 5 present the box plots of the test accuracy (mean \pm std-dev%); Figures 4 and 6 show the box plots of the AUC measures of the models on the 18 image subdatasets of the *Caltech* and *Horse*, respectively. From these figures, we can observe that the accuracy and the AUC measures of the models GRRF, wsRF, and xRF were increased on all high-dimensional subdatasets when the selected subspace $mtry$ was not so large. This implies that when the number of features in the subspace is small, the proportion of the informative features in the feature subspace is comparatively large in the three models. There will be a high chance that highly informative features are selected in the trees so the overall performance of individual trees is increased. In Breiman's method, many randomly selected subspaces may not contain informative features, which affect the performance of trees grown from these subspaces. It can be seen that the xRF model outperformed other random forests models on these subdatasets in increasing the test accuracy and the AUC measures. This was because the new unbiased feature sampling was used in generating trees in the xRF model; the feature subspace provided enough highly informative

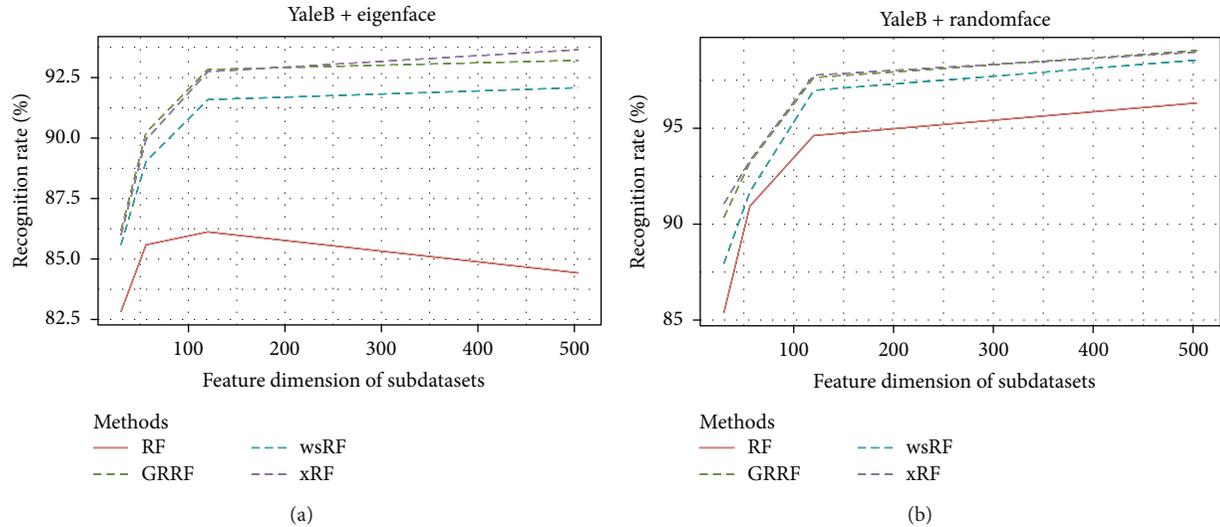


FIGURE 1: Recognition rates of the models on the YaleB subdatasets, namely, YaleB.EigenfaceM30, YaleB.EigenfaceM56, YaleB.EigenfaceM120, YaleB.EigenfaceM504, and YaleB.RandomfaceM30, YaleB.RandomfaceM56, YaleB.RandomfaceM120, and YaleB.RandomfaceM504.

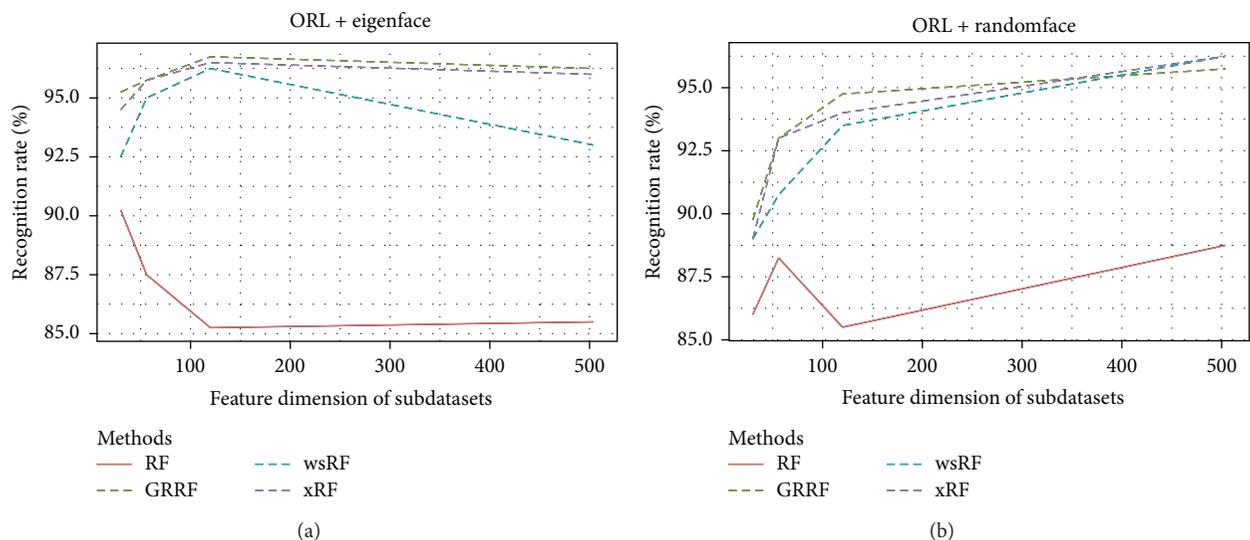


FIGURE 2: Recognition rates of the models on the ORL subdatasets, namely, ORL.EigenfaceM30, ORL.EigenM56, ORL.EigenM120, ORL.EigenM504, and ORL.RandomfaceM30, ORL.RandomM56, ORL.RandomM120, and ORL.RandomM504.

features at any levels of the decision trees. The effect of the unbiased feature selection method is clearly demonstrated in these results.

Table 2 shows the results of $c/s2$ against the number of codebook sizes on the *Caltech* and *Horse* datasets. In a random forest, the tree was grown from a bagging training data. Out-of-bag estimates were used to evaluate the strength, correlation, and $c/s2$. The GRRF model was not considered in this experiment because this method aims to find a small subset of features, and the same RF model in *R*-package [30] is used as a classifier. We compared the xRF model with two kinds of random forest models RF and wsRF. From this table, we can observe that the lowest $c/s2$ values occurred when the wsRF model was applied to the *Caltech* dataset.

However, the xRF model produced the lowest error bound on the *Horse* dataset. These results demonstrate the reason that the new unbiased feature sampling method can reduce the upper bound of the generalization error in random forests.

Table 3 presents the prediction accuracies (mean \pm std-dev%) of the models on subdatasets *CaltechM3000*, *HorseM3000*, *YaleB.EigenfaceM504*, *YaleB.randomfaceM504*, *ORL.EigenfaceM504*, and *ORL.randomfaceM504*. In these experiments, we used the four models to generate random forests with different sizes from 20 trees to 200 trees. For the same size, we used each model to generate 10 random forests for the 10-fold cross-validation and computed the average accuracy of the 10 results. The GRRF model showed slightly better results on *YaleB.EigenfaceM504* with

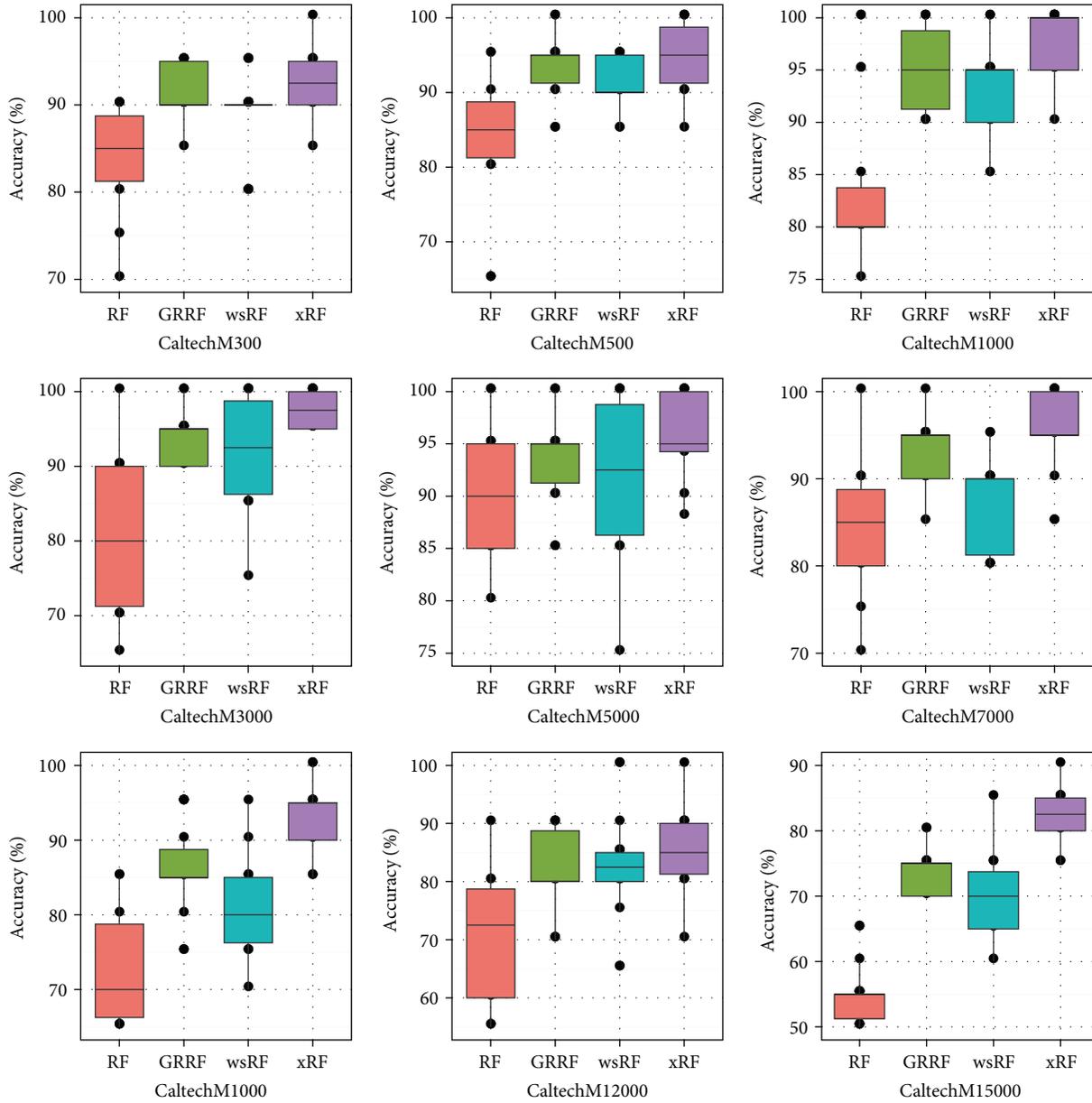


FIGURE 3: Box plots: the test accuracy of the nine Caltech subdatasets.

different tree sizes. The wsRF model produced the best prediction performance on some cases when applied to small subdatasets *YaleB.EigenfaceM504*, *ORL.EigenfaceM504*, and *ORL.randomfaceM504*. However, the xRF model produced, respectively, the highest test accuracy on the remaining subdatasets and AUC measures on high-dimensional subdatasets *CaltechM3000* and *HorseM3000*, as shown in Tables 3 and 4. We can clearly see that the xRF model also outperformed other random forests models in classification accuracy on most cases in all image datasets. Another observation is that the new method is more stable in classification performance because the mean and variance of the test accuracy measures were minor changed when varying the number of trees.

5.5. Results on Microarray Datasets. Table 5 shows the average test results in terms of accuracy of the 100 random forest models computed according to (9) on the gene datasets. The average number of genes selected by the xRF model, from 100 repetitions for each dataset, is shown on the right of Table 5, divided into two groups X_s (strong) and X_w (weak). These genes are used by the unbiased feature sampling method for growing trees in the xRF model. LASSO logistic regression, which uses the RF model as a classifier, showed fairly good accuracy on the two gene datasets *srbct* and *leukemia*. The GRRF model produced slightly better result on the *prostate* gene dataset. However, the xRF model produced the best accuracy on most cases of the remaining gene datasets.

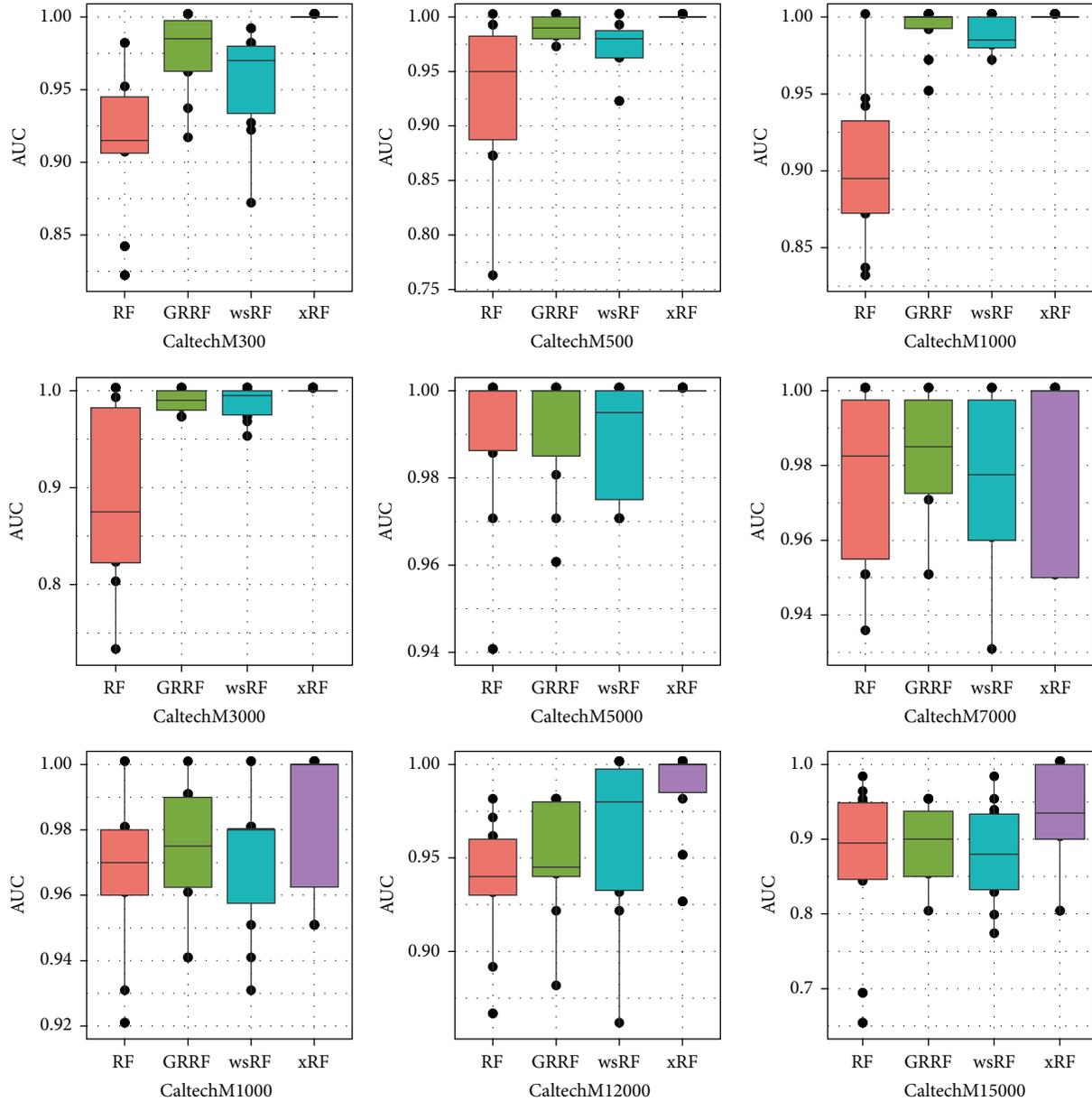


FIGURE 4: Box plots of the AUC measures of the nine Caltech subdatasets.

The detailed results containing the median and the variance values are presented in Figure 7 with box plots. Only the GRRF model was used for this comparison; the LASSO logistic regression and varSelRF method for feature selection were not considered in this experiment because their accuracies are lower than that of the GRRF model, as shown in [17]. We can see that the xRF model achieved the highest average accuracy of prediction on nine datasets out of ten. Its result was significantly different on the *prostate* gene dataset and the variance was also smaller than those of the other models.

Figure 8 shows the box plots of the (c/s) error bound of the RF, wsRF, and xRF models on the ten gene datasets from 100 repetitions. The wsRF model obtained lower error bound

rate on five gene datasets out of 10. The xRF model produced a significantly different error bound rate on two gene datasets and obtained the lowest error rate on three datasets. This implies that when the optimal parameters such as $mtry = \lceil \sqrt{M} \rceil$ and $n_{\min} = 1$ were used in growing trees, the number of genes in the subspace was not small and out-of-bag data was used in prediction, and the results were comparatively favored to the xRF model.

5.6. Comparison of Prediction Performance for Various Numbers of Features and Trees. Table 6 shows the average c/s error bound and accuracy test results of 10 repetitions of random forest models on the three large datasets. The xRF model produced the lowest error c/s on the dataset *Lals*,

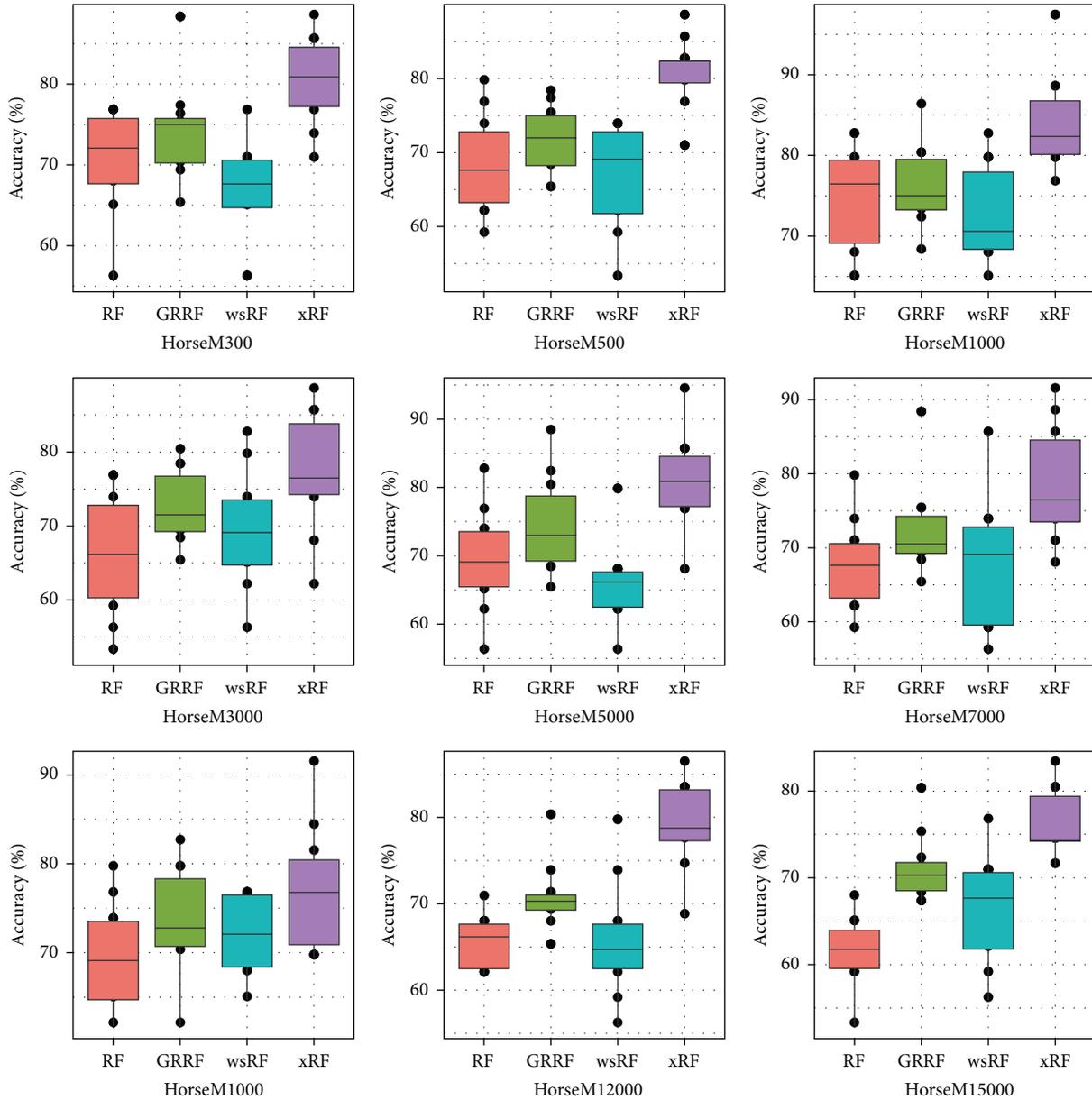


FIGURE 5: Box plots of the test accuracy of the nine Horse subdatasets.

while the wsRF model showed the lower error bound on other two datasets *Fbis* and *La2s*. The RF model demonstrated the worst accuracy of prediction compared to the other models; this model also produced a large $c/s2$ error when the small subspace size $mtry = \lceil \log_2(M) + 1 \rceil$ was used to build trees on the *Lals* and *La2s* datasets. The number of features in the X_s and X_w columns on the right of Table 6 was used in the xRF model. We can see that the xRF model achieved the highest accuracy of prediction on all three large datasets.

Figure 9 shows the plots of the performance curves of the RF models when the number of trees and features increases. The number of trees was increased stepwise by 20 trees from 20 to 200 when the models were applied to the *Lals*

dataset. For the remaining data sets, the number of trees increased stepwise by 50 trees from 50 to 500. The number of random features in a subspace was set to $mtry = \lceil \sqrt{M} \rceil$. The number of features, each consisting of a random sum of five inputs, varied from 5 to 100, and for each, 200 trees were combined. The vertical line in each plot indicates the size of a subspace of features $mtry = \lceil \log_2(M) + 1 \rceil$. This subspace was suggested by Breiman [1] for the case of low-dimensional datasets. Three feature selection methods, namely, GRRF, varSelRF, and LASSO, were not considered in this experiment. The main reason is that, when the $mtry$ value is large, the computational time of the GRRF and varSelRF models required to deal with large high datasets was too long [17].

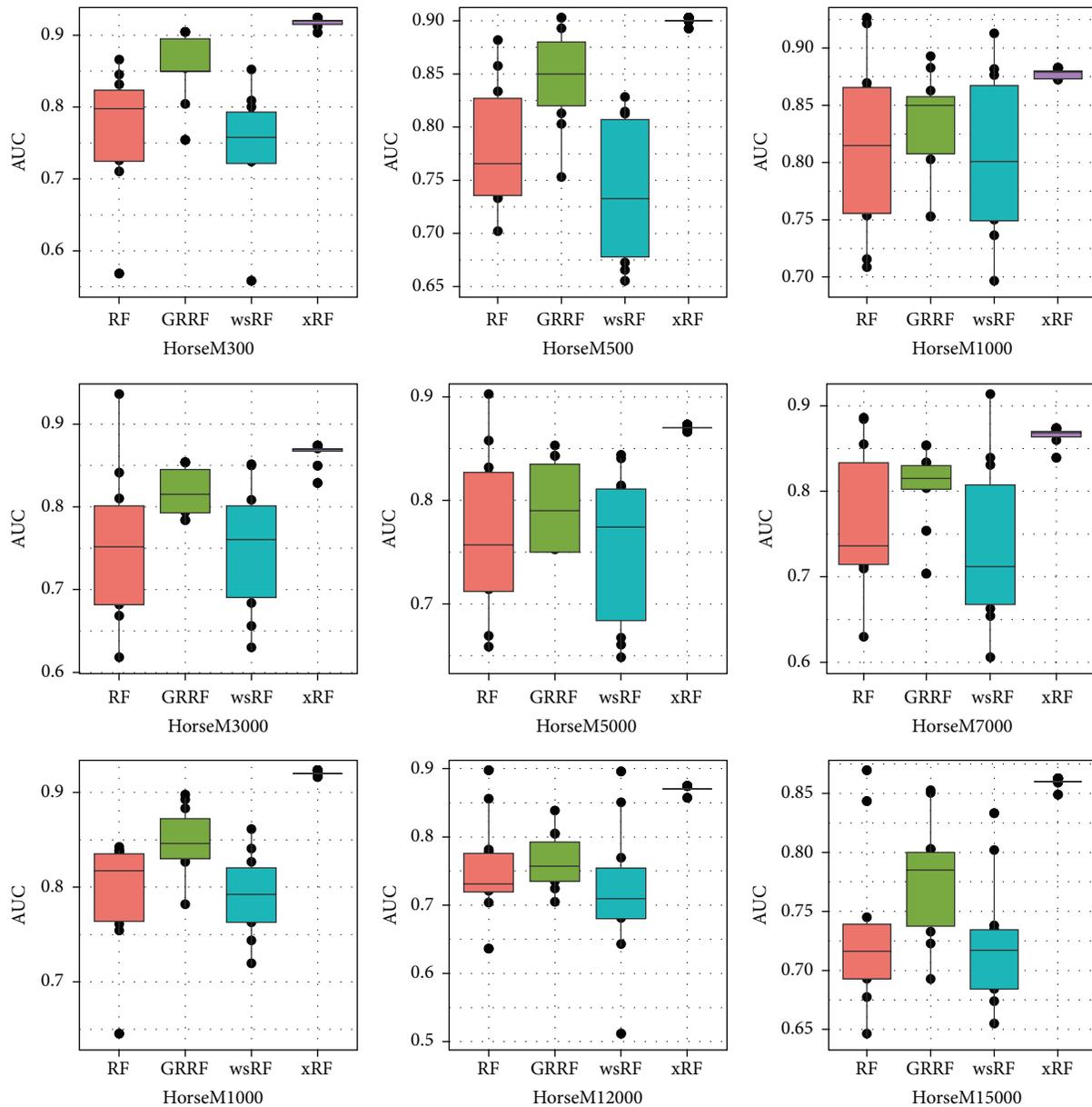


FIGURE 6: Box plots of the AUC measures of the nine Horse subdatasets.

It can be seen that the xRF and wsRF models always provided good results and achieved higher prediction accuracies when the subspace $mtry = \lceil \log_2(M) + 1 \rceil$ was used. However, the xRF model is better than the wsRF model in increasing the prediction accuracy on the three classification datasets. The RF model requires the larger number of features to achieve the higher accuracy of prediction, as shown in the right of Figures 9(a) and 9(b). When the number of trees in a forests was varied, the xRF model produced the best results on the *Fbis* and *La2s* datasets. In the *Lals* dataset where the xRF model did not obtain the best results, as shown in Figure 9(c) (left), the differences from the best results were minor. From the right of Figures 9(a), 9(b), and 9(c), we can observe that the xRF model does not need

many features in the selected subspace to achieve the best prediction performance. These empirical results indicate that, for application on high-dimensional data, when the xRF model uses the small subspace, the achieved results can be satisfactory.

However, the RF model using the simple sampling method for feature selection [1] could achieve good prediction performance only if it is provided with a much larger subspace, as shown in the right part of Figures 9(a) and 9(b). Breiman suggested to use a subspace of size $mtry = \sqrt{M}$ in classification problem. With this size, the computational time for building a random forest is still too high, especially for large high datasets. In general, when the xRF model is used with a feature subspace of the same size as the one suggested

TABLE 2: The $(c/s2)$ error bound results of random forest models against the number of codebook size on the Caltech and Horse datasets. The bold value in each row indicates the best result.

Dataset	Model	300	500	1000	3000	5000	7000	10000	12000	15000
Caltech	xRF	.0312	.0271	.0280	.0287	.0357	.0440	.0650	.0742	.0789
	RF	.0369	.0288	.0294	.0327	.0435	.0592	.0908	.1114	.3611
	wsRF	.0413	.0297	.0268	.0221	.0265	.0333	.0461	.0456	.0789
Horse	xRF	.0266	.0262	.0246	.0277	.0259	.0298	.0275	.0288	.0382
	RF	.0331	.0342	.0354	.0374	.0417	.0463	.0519	.0537	.0695
	wsRF	.0429	.0414	.0391	.0295	.0288	.0333	.0295	.0339	.0455

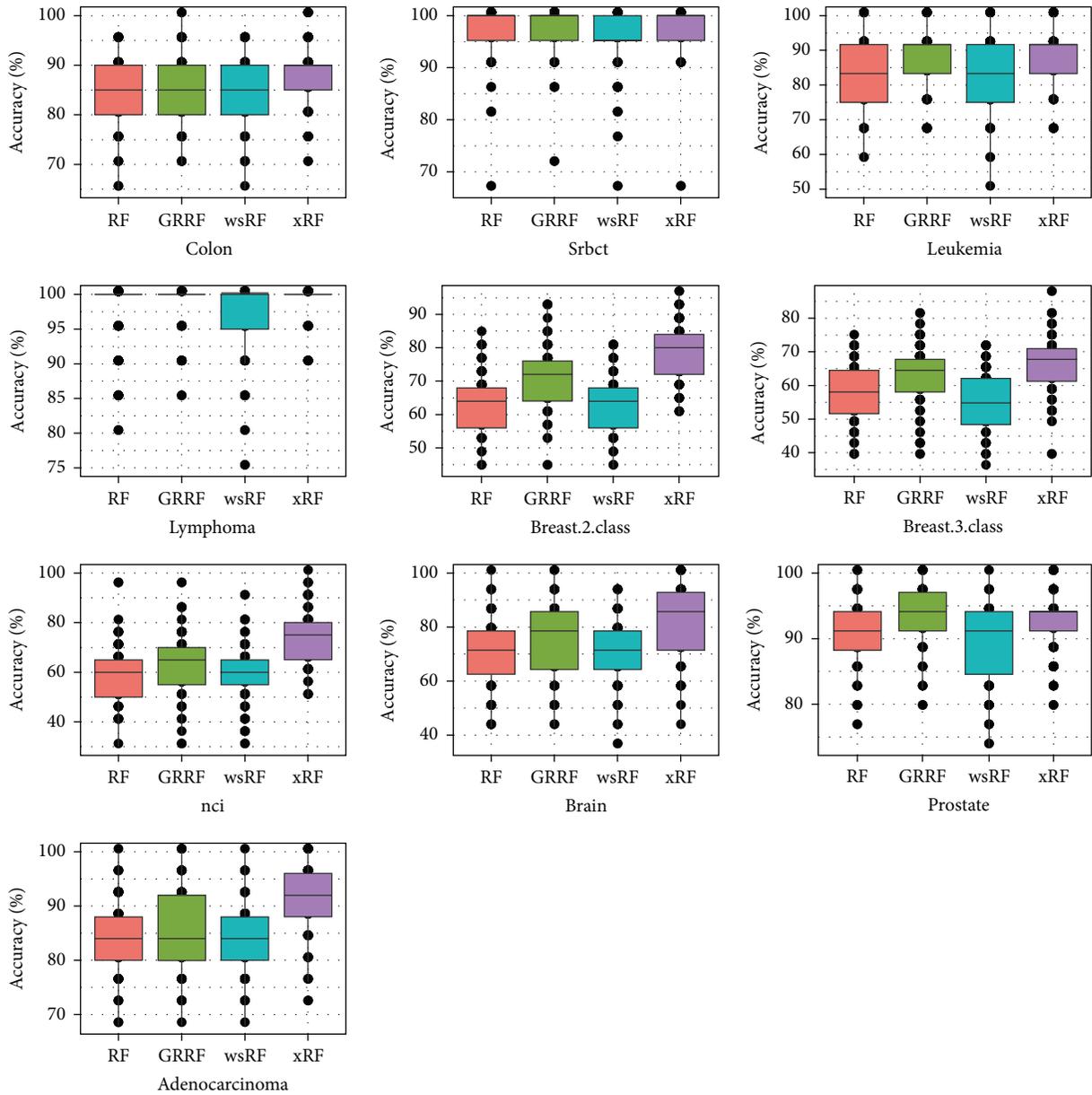


FIGURE 7: Box plots of test accuracy of the models on the ten gene datasets.

TABLE 3: The prediction test accuracy (mean% \pm std-dev%) of the models on the image datasets against the number of trees K . The number of feature dimensions in each subdataset is fixed. Numbers in bold are the best results.

Dataset	Model	$K = 20$	$K = 50$	$K = 80$	$K = 100$	$K = 200$
CaltechM3000	xRF	95.50 \pm .2	96.50 \pm .1	96.50 \pm .2	97.00 \pm .1	97.50 \pm .2
	RF	70.00 \pm .7	76.00 \pm .9	77.50 \pm 1.2	82.50 \pm 1.6	81.50 \pm .2
	wsRF	91.50 \pm .4	91.00 \pm .3	93.00 \pm .2	94.50 \pm .4	92.00 \pm .9
	GRRF	93.00 \pm .2	96.00 \pm .2	94.50 \pm .2	95.00 \pm .3	94.00 \pm .2
HorseM3000	xRF	80.59 \pm .4	81.76 \pm .2	79.71 \pm .6	80.29 \pm .1	77.65 \pm .5
	RF	50.59 \pm 1.0	52.94 \pm .8	56.18 \pm .4	58.24 \pm .5	57.35 \pm .9
	wsRF	62.06 \pm .4	68.82 \pm .3	67.65 \pm .3	67.65 \pm .5	65.88 \pm .7
	GRRF	65.00 \pm .9	63.53 \pm .3	68.53 \pm .3	63.53 \pm .9	71.18 \pm .4
YaleB.EigenfaceM504	xRF	75.68 \pm .1	85.65 \pm .1	88.08 \pm .1	88.94 \pm .0	91.22 \pm .0
	RF	71.93 \pm .1	79.48 \pm .1	80.69 \pm .1	81.67 \pm .1	82.89 \pm .1
	wsRF	77.60 \pm .1	85.61 \pm .0	88.11 \pm .0	89.31 \pm .0	90.68 \pm .0
	GRRF	74.73 \pm .0	84.70 \pm .1	87.25 \pm .0	89.61 \pm .0	91.89 \pm .0
YaleB.randomfaceM504	xRF	94.71 \pm .0	97.64 \pm .0	98.01 \pm .0	98.22 \pm .0	98.59 \pm .0
	RF	88.00 \pm .0	92.59 \pm .0	94.13 \pm .0	94.86 \pm .0	96.06 \pm .0
	wsRF	95.40 \pm .0	97.90 \pm .0	98.17 \pm .0	98.14 \pm .0	98.38 \pm .0
	GRRF	95.66 \pm .0	98.10 \pm .0	98.42 \pm .0	98.92 \pm .0	98.84 \pm .0
ORL.EigenfaceM504	xRF	76.25 \pm .6	87.25 \pm .3	91.75 \pm .2	93.25 \pm .2	94.75 \pm .2
	RF	71.75 \pm .2	78.75 \pm .4	82.00 \pm .3	82.75 \pm .3	85.50 \pm .5
	wsRF	78.25 \pm .4	88.75 \pm .3	90.00 \pm .1	91.25 \pm .2	92.50 \pm .2
	GRRF	73.50 \pm .6	85.00 \pm .2	90.00 \pm .1	90.75 \pm .3	94.75 \pm .1
ORL.randomfaceM504	xRF	87.75 \pm .3	92.50 \pm .2	95.50 \pm .1	94.25 \pm .1	96.00 \pm .1
	RF	77.50 \pm .3	82.00 \pm .7	84.50 \pm .2	87.50 \pm .2	86.00 \pm .2
	wsRF	87.00 \pm .5	93.75 \pm .2	93.75 \pm .0	95.00 \pm .1	95.50 \pm .1
	GRRF	87.25 \pm .1	93.25 \pm .1	94.50 \pm .1	94.25 \pm .1	95.50 \pm .1

TABLE 4: AUC results (mean \pm std-dev%) of random forest models against the number of trees K on the CaltechM3000 and HorseM3000 subsets. The bold value in each row indicates the best result.

Dataset	Model	$K = 20$	$K = 50$	$K = 80$	$K = 100$	$K = 200$
CaltechM3000	xRF	.995 \pm .0	.999 \pm .5	1.00 \pm .2	1.00 \pm .1	1.00 \pm .1
	RF	.851 \pm .7	.817 \pm .4	.826 \pm 1.2	.865 \pm .6	.864 \pm 1
	wsRF	.841 \pm 1	.845 \pm .8	.834 \pm .7	.850 \pm .8	.870 \pm .9
	GRRF	.846 \pm .1	.860 \pm .2	.862 \pm .1	.908 \pm .1	.923 \pm .1
HorseM3000	xRF	.849 \pm .1	.887 \pm .0	.895 \pm .0	.898 \pm .0	.897 \pm .0
	RF	.637 \pm .4	.664 \pm .7	.692 \pm 1.5	.696 \pm .3	.733 \pm .9
	wsRF	.635 \pm .8	.687 \pm .4	.679 \pm .6	.671 \pm .4	.718 \pm .9
	GRRF	.786 \pm .3	.778 \pm .3	.785 \pm .8	.699 \pm .1	.806 \pm .4

TABLE 5: Test accuracy results (%) of random forest models, GRRF(0.1), varSelRF, and LASSO logistic regression, applied to gene datasets. The average results of 100 repetitions were computed; higher values are better. The number of genes in the strong group X_s and the weak group X_w is used in xRF.

Dataset	xRF	RF	wsRF	GRRF	varSelRF	LASSO	X_s	X_w
colon	87.65	84.35	84.50	86.45	76.80	82.00	245	317
srbct	97.71	95.90	96.76	97.57	96.50	99.30	606	546
Leukemia	89.25	82.58	84.83	87.25	89.30	92.40	502	200
Lymphoma	99.30	97.15	98.10	99.10	97.80	99.10	1404	275
breast.2.class	78.84	62.72	63.40	71.32	61.40	63.40	194	631
breast.3.class	65.42	56.00	57.19	63.55	58.20	60.00	724	533
nci	74.15	58.85	59.40	63.05	58.20	60.40	247	1345
Brain	81.93	70.79	70.79	74.79	76.90	74.10	1270	1219
Prostate	92.56	88.71	90.79	92.85	91.50	91.20	601	323
Adenocarcinoma	90.88	84.04	84.12	85.52	78.80	81.10	108	669

TABLE 6: The accuracy of prediction and error bound $c/s2$ of the models using a small subspace $mtry = \lfloor \log_2(M) + 1 \rfloor$; better values are bold.

Dataset	$c/s2$ Error bound			Test accuracy (%)					
	RF	wsRF	xRF	RF	GRRF	wsRF	xRF	X_s	X_w
Fbis	.2149	.1179	.1209	76.42	76.51	84.14	84.69	201	555
La2s	152.6	.0904	.0780	66.77	67.99	87.26	88.61	353	1136
La1s	40.8	.0892	.1499	77.76	80.49	86.03	87.21	220	1532

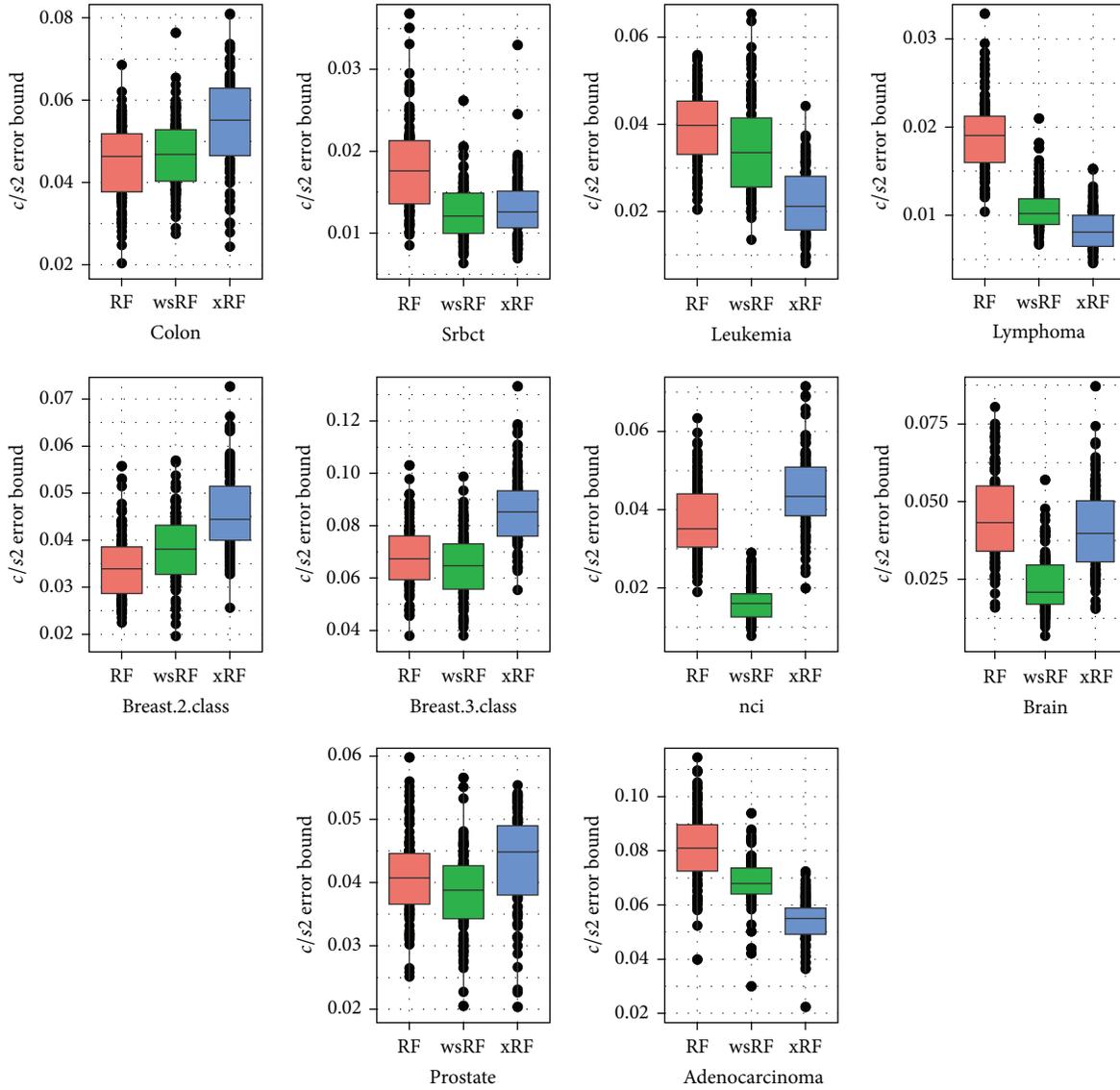


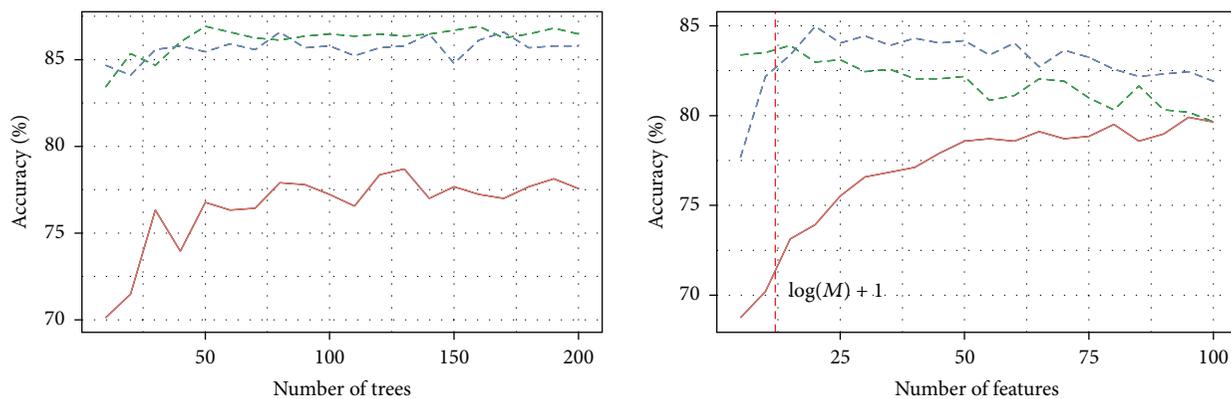
FIGURE 8: Box plots of $(c/s2)$ error bound for the models applied to the 10 gene datasets.

by Breiman, it demonstrates higher prediction accuracy and shorter computational time than those reported by Breiman. This achievement is considered to be one of the contributions in our work.

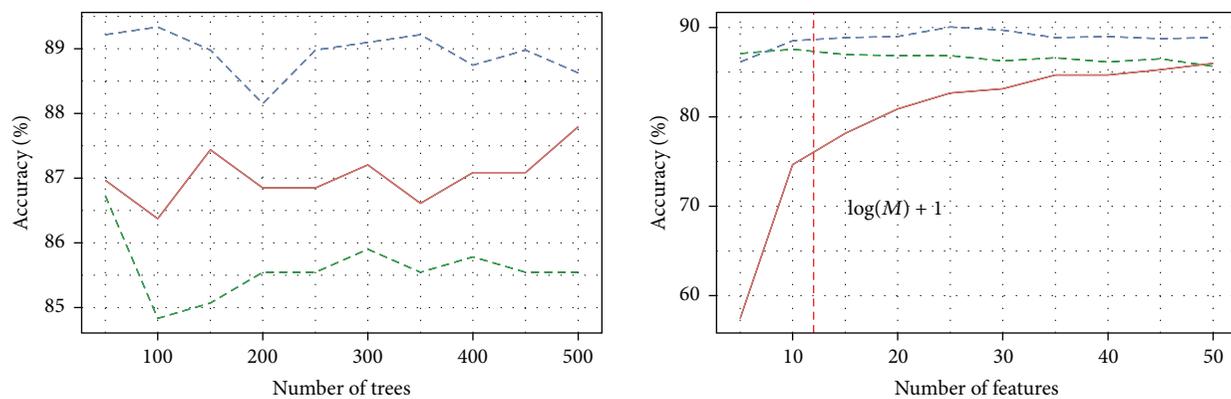
6. Conclusions

We have presented a new method for feature subspace selection for building efficient random forest xRF model for

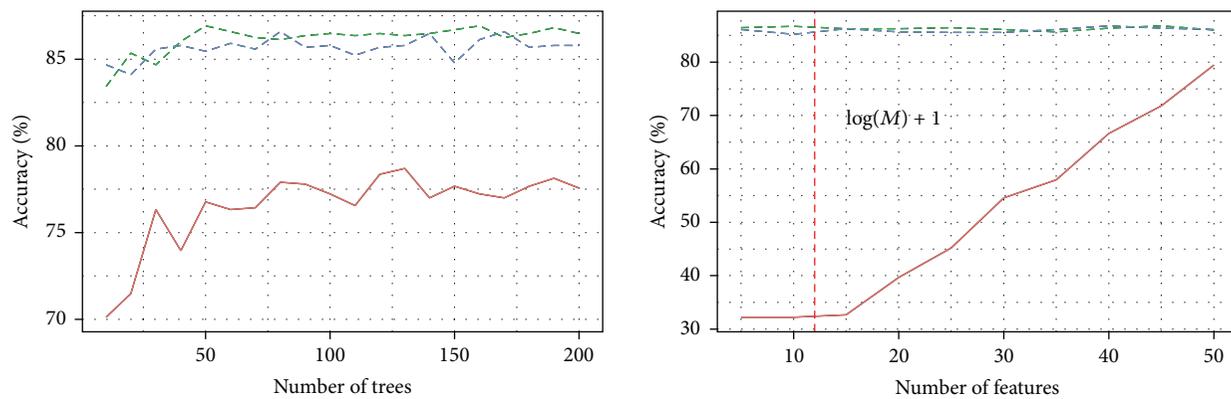
classification high-dimensional data. Our main contribution is to make a new approach for unbiased feature sampling, which selects the set of unbiased features for splitting a node when growing trees in the forests. Furthermore, this new unbiased feature selection method also reduces dimensionality using a defined threshold to remove uninformative features (or noise) from the dataset. Experimental results have demonstrated the improvements in increasing of the test accuracy and the AUC measures for classification problems,



(a) Fbis



(b) La2s



Methods
 — RF
 - - wsRF
 - - xRF

Methods
 — RF
 - - wsRF
 - - xRF

(c) Lals

FIGURE 9: The accuracy of prediction of the three random forests models against the number of trees and features on the three datasets.

especially for image and microarray datasets, in comparison with recent proposed random forests models, including RF, GRRF, and wsRF.

For future work, we think it would be desirable to increase the scalability of the proposed random forests algorithm by parallelizing them on the cloud platform to deal with big data, that is, hundreds of millions of samples and features.

Conflict of Interests

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

Acknowledgments

This research is supported in part by NSFC under Grant no. 61203294 and Hanoi-DOST under the Grant no. 01C-07/01-2012-2. The author Thuy Thi Nguyen is supported by the project “Some Advanced Statistical Learning Techniques for Computer Vision” funded by the National Foundation of Science and Technology Development, Vietnam, under the Grant no. 102.01-2011.17.

References

- [1] L. Breiman, “Random forests,” *Machine Learning*, vol. 450, no. 1, pp. 5–32, 2001.
- [2] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*, CRC Press, Boca Raton, Fla, USA, 1984.
- [3] H. Kim and W.-Y. Loh, “Classification trees with unbiased multiway splits,” *Journal of the American Statistical Association*, vol. 96, no. 454, pp. 589–604, 2001.
- [4] A. P. White and W. Z. Liu, “Technical note: bias in information-based measures in decision tree induction,” *Machine Learning*, vol. 15, no. 3, pp. 321–329, 1994.
- [5] T. G. Dietterich, “Experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization,” *Machine Learning*, vol. 40, no. 2, pp. 139–157, 2000.
- [6] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in *Computational Learning Theory*, pp. 23–37, Springer, 1995.
- [7] T.-T. Nguyen and T. T. Nguyen, “A real time license plate detection system based on boosting learning algorithm,” in *Proceedings of the 5th International Congress on Image and Signal Processing (CISP '12)*, pp. 819–823, IEEE, October 2012.
- [8] T. K. Ho, “Random decision forests,” in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282, 1995.
- [9] T. K. Ho, “The random subspace method for constructing decision forests,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.
- [10] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [11] R. Díaz-Uriarte and S. Alvarez de Andrés, “Gene selection and classification of microarray data using random forest,” *BMC Bioinformatics*, vol. 7, article 3, 2006.
- [12] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, “Variable selection using random forests,” *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.
- [13] B. Xu, J. Z. Huang, G. Williams, Q. Wang, and Y. Ye, “Classifying very high-dimensional data with random forests built from small subspaces,” *International Journal of Data Warehousing and Mining*, vol. 8, no. 2, pp. 44–63, 2012.
- [14] Y. Ye, Q. Wu, J. Zhixue Huang, M. K. Ng, and X. Li, “Stratified sampling for feature subspace selection in random forests for high dimensional data,” *Pattern Recognition*, vol. 46, no. 3, pp. 769–787, 2013.
- [15] X. Chen, Y. Ye, X. Xu, and J. Z. Huang, “A feature group weighting method for subspace clustering of high-dimensional data,” *Pattern Recognition*, vol. 45, no. 1, pp. 434–446, 2012.
- [16] D. Amaratunga, J. Cabrera, and Y.-S. Lee, “Enriched random forests,” *Bioinformatics*, vol. 240, no. 18, pp. 2010–2014, 2008.
- [17] H. Deng and G. Runger, “Gene selection with guided regularized random forest,” *Pattern Recognition*, vol. 46, no. 12, pp. 3483–3489, 2013.
- [18] C. Strobl, “Statistical sources of variable selection bias in classification trees based on the gini index,” Tech. Rep. SFB 386, 2005, http://epub.ub.uni-muenchen.de/archive/00001789/01/paper_420.pdf.
- [19] C. Strobl, A.-L. Boulesteix, and T. Augustin, “Unbiased split selection for classification trees based on the gini index,” *Computational Statistics & Data Analysis*, vol. 520, no. 1, pp. 483–501, 2007.
- [20] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, “Bias in random forest variable importance measures: illustrations, sources and a solution,” *BMC Bioinformatics*, vol. 8, article 25, 2007.
- [21] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, “Conditional variable importance for random forests,” *BMC Bioinformatics*, vol. 9, no. 1, article 307, 2008.
- [22] T. Hothorn, K. Hornik, and A. Zeileis, Party: a laboratory for recursive partytioning, r package version 0.9-9999, 2011, <http://cran.r-project.org/package=party>.
- [23] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics*, vol. 10, no. 6, pp. 80–83, 1954.
- [24] T.-T. Nguyen, J. Z. Huang, and T. T. Nguyen, “Two-level quantile regression forests for bias correction in range prediction,” *Machine Learning*, 2014.
- [25] T.-T. Nguyen, J. Z. Huang, K. Imran, M. J. Li, and G. Williams, “Extensions to quantile regression forests for very high-dimensional data,” in *Advances in Knowledge Discovery and Data Mining*, vol. 8444 of *Lecture Notes in Computer Science*, pp. 247–258, Springer, Berlin, Germany, 2014.
- [26] A. S. Georghiadis, P. N. Belhumeur, and D. J. Kriegman, “From few to many: illumination cone models for face recognition under variable lighting and pose,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.
- [27] F. S. Samaria and A. C. Harter, “Parameterisation of a stochastic model for human face identification,” in *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*, pp. 138–142, IEEE, December 1994.
- [28] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [29] H. Deng, “Guided random forest in the RRF package,” <http://arxiv.org/abs/1306.0237>.

- [30] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 20, no. 3, pp. 18–22, 2002.
- [31] R. Diaz-Uriarte, "varselrf: variable selection using random forests," R package version 0.7-1, 2009, <http://ligarto.org/rdiaz/Software/Software.html>.
- [32] J. H. Friedman, T. J. Hastie, and R. J. Tibshirani, "glmnet: Lasso and elastic-net regularized generalized linear models," R package version , pages 1-1, 2010, <http://CRAN.R-project.org/package=glmnet>.

Research Article

A Heuristic Ranking Approach on Capacity Benefit Margin Determination Using Pareto-Based Evolutionary Programming Technique

Muhammad Murtadha Othman,^{1,2} Nurulazmi Abd Rahman,³ Ismail Musirin,^{1,2}
Mahmud Fotuhi-Firuzabad,⁴ and Abbas Rajabi-Ghahnavieh⁵

¹Committee of Research (CORE), Advanced Computing & Communication (ACC), Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia

²Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia

³Engineering Centre, University Malaysia Perlis, Kampus Kubang Gajah, 02600 Arau, Perlis, Malaysia

⁴Centre of Excellence in Power System Management and Control, Electrical Engineering Department, Sharif University of Technology, Tehran 11365-11155, Iran

⁵Department of Energy Engineering, Sharif University of Technology, Tehran 11365-11155, Iran

Correspondence should be addressed to Muhammad Murtadha Othman; mamat505my@yahoo.com

Received 6 May 2014; Accepted 15 September 2014

Academic Editor: Shifei Ding

Copyright © 2015 Muhammad Murtadha Othman et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper introduces a novel multiobjective approach for capacity benefit margin (CBM) assessment taking into account tie-line reliability of interconnected systems. CBM is the imperative information utilized as a reference by the load-serving entities (LSE) to estimate a certain margin of transfer capability so that a reliable access to generation through interconnected system could be attained. A new Pareto-based evolutionary programming (EP) technique is used to perform a simultaneous determination of CBM for all areas of the interconnected system. The selection of CBM at the Pareto optimal front is proposed to be performed by referring to a heuristic ranking index that takes into account system loss of load expectation (LOLE) in various conditions. Eventually, the power transfer based available transfer capability (ATC) is determined by considering the firm and nonfirm transfers of CBM. A comprehensive set of numerical studies are conducted on the modified IEEE-RTS79 and the performance of the proposed method is numerically investigated in detail. The main advantage of the proposed technique is in terms of flexibility offered to an independent system operator in selecting an appropriate solution of CBM simultaneously for all areas.

1. Introduction

In a deregulated power system environment, electricity is considered as a commodity that can be traded in a free market where the generators and loads participated. The transition to a new structure of electricity market is to ensure the quality and efficient production of electrical energy that can be offered at a lower electricity price as well as maximizing the utilization of generation and transmission facilities [1, 2]. Hence, it is important for the independent system operator (ISO) to calculate and provide the information of available transfer capability (ATC) associated with the transfer paths

to the open access same-time information system (OASIS) so that electricity market could be conducted in an effective manner [3, 4]. ATC is defined as the maximum amount of power that can be transferred from a selling area to a buying area without jeopardizing a system security [5]. ATC can be calculated as the total transfer capability (TTC) reduced by the transmission reliability margin (TRM), capacity benefit margin (CBM), and existing transmission commitment (ETC). CBM is one of the main components considered in the ATC calculation and is defined as the amount of transfer capability reserved by load-serving entities, which is anticipated to be used in cases of generation deficiency [5–9].

Inaccurate determination of CBM may result in either underestimation or overestimation of the ATC. Underestimating the ATC value possibility will cause an ineffective use in the transmission facility, while overestimating the ATC value will threaten a power system security [3, 7].

So far, several methods have been proposed to determine CBM [10–19]. The basic method used to compute the CBM for each area of an interconnected system is based on trial and error [10], by prescribing 5% of the maximum transfer capability [11] or the CBM value is specified as zero [12, 13]. Reference [14] has proposed an analytic model used for multi-area generation reliability assessment and then applied into the sequential quadratic programming (SQP) for determining the CBM values considering the loss of load expectation (LOLE) as the system reliability criterion. Rajathy et al. [15] use the differential evolution and Monte Carlo techniques to determine the CBM. A method that has been proposed in [16] is used to determine the CBM for each area of an interconnected system using the evolutionary programming (EP) as an accelerated search technique. Furthermore, CBM determination is formulated as an optimization problem which is solved by using the particle swarm optimization (PSO) technique [17, 18]. In order to provide a set of choices for different cases, three methods have been proposed in [17, 18] which will provide different values of CBM. It is observed that the existing CBM calculations do not provide adequate flexibility for the ISO to select a CBM value in accordance with system requirements [10–19]. In addition, tie-line availability is an influential factor which has an effect on the reliability of an interconnected system followed by the value of CBM. This imperative factor has been taken into account for CBM calculation in [19].

A novel multiobjective based optimization approach is presented in this paper to determine several optimum values of CBM using the Pareto-based EP technique that takes into account the tie-line reliability of an interconnected system. The proposed Pareto-based EP technique has several advantages compared to the methodology previously presented in [16] and it provides the ISO with several choices of optimum CBM values. The multiobjective function of EP technique is referred to as the transfer capability margin of CBM for all areas with LOLE less than a specified value at initial condition. Moreover, the CBMs of all areas are obtained simultaneously at every execution of the proposed technique. The first order sensitivity with modified Gaussian formulation is used as a new mutation technique to enhance the EP performance in searching for a new population at global maximum domain with less computational time. Then, the Pareto optimal front approach is used to select several optimal solutions of CBM values using the ranking index of total LOLE and total difference of LOLE. A modified IEEE-RTS79 is used as the numerical test bed to verify effectiveness of the proposed method in providing the solutions of CBMs [17]. The robustness of the proposed method in CBM determination is compared with that of the basic methodology used for the CBM calculation [17]. Performance comparison has also been performed which investigates the effect of tie-line reliability included in the CBM determination. Finally, the significance of CBM considered as firm and nonfirm

transfers can be observed through its impact on the ATC determination.

2. Multiobjective Functions of Capacity Benefit Margins Determination

A process involved in the Pareto-based EP technique used for determining the multiobjective function of CBMs is described as follows.

Step (a). Establish a solved base case power flow solution.

Step (b). Determine the LOLE for each area of the interconnected system at the base case condition.

Step (c). Identify the assisting areas with LOLE less than the specified value, ξ (e.g., 2.4 hrs/yr). It signifies that these areas conserve a certain amount of reserve generating capacity that could be used to compensate for the generation deficiency which may occur in the assisted area. LOLE associated with the assisted area is usually greater than ξ . It is important to mention that the assisting and assisted areas are the terms used to signify the direction of power transfer based CBM (CBM_{asg}^{Pareto}) and this is different from the selling and buying areas which are the terms used to signify the direction of power transfer based ATC.

Step (d). Identify the assisted area with the largest LOLE above ξ .

Step (e). Determine the parent or initial population for each assisting area with LOLE below ξ . Equation (1) is used to generate the individuals $xpar_{m,asg}$, for parent or initial population using uniform random distribution. The determination of $xpar_{m,asg}$ is based on either total rating of all tie-lines connecting between the assisting and assisted areas, $PLIt_{asg}$, or the total reserve generating capacity of the assisting area, $DPGt_{asg}$. The $xpar_{m,asg}$ is determined based on the former condition when $DPGt_{asg}$ exceeds the $PLIt_{asg}$. This means that tie-lines are the constraining factors for power transfer based CBM and, thus, $xpar_{m,asg}$ are generated randomly based on $PLIt_{asg}$. The latter condition is used to determine $xpar_{m,asg}$ when $DPGt_{asg}$ is less than $PLIt_{asg}$. Each individual, $xpar$, is considered as an external generating capacity, PG_{Ext} , or CBM, which is provided by the assisting area to support generating capacity deficiency in the assisted area having the highest LOLE:

$$xpar_{m,asg} = \begin{cases} \text{rand}_m(DPGt_{asg}), & \text{if } DPGt_{asg} < PLIt_{asg}, \\ \text{rand}_m(PLIt_{asg}), & \text{if } DPGt_{asg} > PLIt_{asg}, \end{cases} \quad (1)$$

where

$$DPGt_{asg} = PGt_{asg} - PLt_{asg}, \quad (2)$$

$$PLIt_{asg} = \sum_{l=1}^L PLI_l^{asg}.$$

$CBM_{m,asg}$ or $xpar_{m,asg}$ is the CBM in the case of transfer from assisting area to assisted area; PGt is the total generating

capacity; PLt is the total peak load; PLI is the tie-line rating; L is the total number of tie-lines; m is 1, 2, 3, ..., pop; asg is 1, 2, 3, ..., N_{asg}; pop is the population size; and N_{asg} is the total number of assisting areas.

Step (f). Calculate a new total generation capacity, new PG _{m,asg} , for each assisting area according to CBM or xpar _{m,asg} as given in (3) and (4). The generating capacity of the assisting area is reduced as it is partially assigned to the assisted area. The new generating capacity for each bus g of the assisting area new PG _{m,asg} is obtained based on the ratio of generating capacity as

$$\text{new PG}_{m,asg} = \sum_{g=1}^{NG} \text{new PG}_g^{m,asg}, \quad (3)$$

where

$$\text{new PG}_g^{m,asg} = \text{PG}_g^{\text{asg}} - \frac{\text{PG}_g^{\text{asg}}}{\sum_{g=1}^{NG} \text{PG}_g^{\text{asg}}} \times \text{xpar}_{m,asg}. \quad (4)$$

PG is the generating capacity and NG is the total number of generator buses.

Step (g). Determine the LOLE for each assisting area (LOLE _{m,asg}) considering the new PG _{m,asg} , hourly peak load, and cumulative probability of generation capacity outage (PC(C _{s})) as discussed in [19].

Step (h). Determine a new total generation capacity, new PG _{$m,asd=1$} , for an assisted area with the largest LOLE above ξ using (5) and (6). In (6), apportionment of the total xpar _{m,asg} or total CBM _{m,asg} to each generator is performed based on the ratio of generating capacity and total generating capacity of an assisted area. For an assisted area, there are pop number of individuals for the size of new total generating capacity, new PG _{$m,asd=1$} ,

$$\text{new PG}_{m,asd=1} = \sum_{g=1}^{NG} \text{new PG}_g^{m,asd=1}, \quad (5)$$

where

$$\text{new PG}_g^{m,asd=1} = \text{PG}_g^{\text{asd=1}} + \frac{\text{PG}_g^{\text{asd=1}}}{\sum_{g=1}^{NG} \text{PG}_g^{\text{asd=1}}} \sum_{asg=1}^{N_{asg}} \text{xpar}_{m,asg}, \quad (6)$$

where asd is the number of assisted areas, 1.

Step (i). Calculate the fitness value (f_m), that is, LOLE _{$m,asd=1$} as discussed in [19]. f_m is an important parameter used to assist the determination of a new xpar _{m,asg} and the convergence criteria for the optimization process. This will be explained thoroughly in the following steps. f_m or LOLE _{$m,asd=1$} is calculated by taking into account the increased amount of new PG _{$m,asd=1$} obtained in Step (h).

Step (j). Perform the mutation to obtain an offspring for each assisting area with LOLE less than ξ . In the proposed

mutation approach, the modified Gaussian technique is used to improve the capability of global maximum search of a new population with less computational time [16]. This technique is suitable in solving the optimization problems in which considerable discrepancy does exist among the individual values. Each offspring comprising new individuals, xoff _{m,asg} , is originated from xpar _{m,asg} . The new individuals, xoff _{m,asg} , are obtained using a new mutation technique that incorporates the first order sensitivity, $\partial \text{xpar}_{asg} / \partial N(f, \xi, \sigma)$, and the modified Gaussian formulation, $N(f_m, \xi, \sigma)$, as expressed in (7). The value of xoff _{m,asg} is varied in accordance with the changes in f_m to the estimated LOLE limit, ξ . Consider

$$\text{xoff}_{m,asg} = \text{xpar}_{m,asg} + \left(\left| \frac{\partial \text{xpar}_{asg}}{\partial N(f, \xi, \sigma)} \right| (1 - N(f_m, \xi, \sigma)) \right), \quad (7)$$

where

$$\frac{\partial \text{xpar}_{asg}}{\partial N(f, \xi, \sigma)} = \frac{\max \text{xpar}_{asg} - \min \text{xpar}_{asg}}{\max N(f, \xi, \sigma) - \min N(f, \xi, \sigma)}, \quad (8)$$

$$N(f_m, \xi, \sigma) = e^{-(f_m - \xi)^2 / 2\sigma^2},$$

where max xpar _{asg} and min xpar _{asg} are the maximum and minimum values of xpar _{m,asg} for every assisting area, respectively; max $N(f, \xi, \sigma)$ and min $N(f, \xi, \sigma)$ are the maximum and minimum values of $N(f_m, \xi, \sigma)$, respectively; and σ or f_{\max} is the maximum value of fitness, f_m or LOLE _{$m,asd=1$} .

The first order sensitivity is used to overcome the impediment of local maxima or minima which normally occurs in the case of large f_m . Hence, robustness in searching for the global maxima or minima can easily be guaranteed by using the new mutation technique.

Step (k). Perform Steps (h) and (i) to determine f_m or LOLE _{$m,asd=1$} in relation to a new value of new PG _{$m,asd=1$} obtained according to (5) considering xoff _{m,asg} . This implies that the xpar _{m,asg} in (6) has been replaced by xoff _{m,asg} , yielding to a new value of new PG _{$m,asd=1$} . Apart from the new PG _{$m,asd=1$} obtained based on xoff _{m,asg} , determination of LOLE _{$m,asd=1$} also requires several other parameters such as the hourly peak load and new cumulative probability of the generation capacity outage (PC(C _{s})) as discussed in [19].

Step (l). Perform pairwise comparison to determine the next generation of population comprising the best individuals selected from xoff _{m,asg} and xpar _{m,asg} . For each assisting area, f_m or LOLE _{$m,asd=1$} has been used as a reference for selecting the best individuals as the next generation of xpar _{m,asg} . In this case, f_m for xpar _{m,asg} and xoff _{m,asg} are obtained from Steps (h) and (j), respectively. The concept of selection is elucidated in terms of the formulation given in (9). Otherwise, when the total number of chosen individuals is not adequate for

population size, pop , then the offspring, $xoff_{m,asg}$, is selected as the next generation of $xpar_{m,asg}$ as illustrated in

$$xsel_{m,asg} = \left[\begin{array}{c} xoff_{m=1,asg} |_{f_{m=1}(xoff_{m=1,asg}) < \xi} \\ \vdots \\ xoff_{m,asg} |_{f_m(xoff_{m,asg}) < \xi} \\ \dots\dots\dots \\ xpar_{m=1,asg} |_{f_{m=1}(xpar_{m=1,asg}) < \xi} \\ \vdots \\ xpar_{m,asg} |_{f_m(xpar_{m,asg}) < \xi} \end{array} \right], \quad (9)$$

$$xpar_{m,asg} = \begin{cases} xsel_{m,asg}, & \text{if size}(xsel_{m,asg}) \geq pop, \\ xoff_{m,asg}, & \text{if size}(xsel_{m,asg}) < pop, \end{cases} \quad (10)$$

where $xsel_{m,asg}$ is the best individuals selected from $xoff_{m,asg}$ and $xpar_{m,asg}$ having $f_m < \xi$; $f_m(xoff_{m,asg})$ is the f_m corresponding to the m th value of individual $xoff_{m,asg}$; $f_m(xpar_{m,asg})$ is the f_m corresponding to the m th value of individual $xpar_{m,asg}$; and $size(xsel_{m,asg})$ is the size of $xsel_{m,asg}$.

Step (m). The convergence criteria for the EP optimization process is achieved when the mismatch between maximum fitness, f_{max} , and minimum fitness, f_{min} , is within a specified range, ϵ . f_{max} and f_{min} are the maximum and minimum values of f_m , respectively, obtained based on the $xpar_{m,asg}$ in Step (l):

$$f_{max} - f_{min} \leq \epsilon, \quad (11)$$

where f_{min} is the minimum value of f_m or $LOLE_{m,asd=1}$ and ϵ is the desired accuracy, 0.1 for an example [16].

Go to Step (f) for the next generation of EP optimization process when the mismatch does not reach to the desired level and the new value of $xpar_{m,asg}$ obtained in Step (l) will be used to calculate a new $PGT_{m,asg}$ in Step (f). Otherwise, proceed to Step (n) once the mismatch has reached the predetermined limit ϵ .

Step (n). Record the optimized multiobjective function of CBM_{asg} for the transfer case from assisting areas to an assisted area. The optimized multiobjective CBM_{asg} will be recorded at the last iteration of the optimization process. The CBM_{asg} is obtained as the average value of $xpar_{m,asg}$ or $CBM_{m,asg}$ associated with the assisting area previously calculated in Step (l). This implies that the CBM_{asg} is calculated through (12). Hence, the multiobjective function (M.O.F) comprising several optimized CBM_{asg} for the case of power transferred from the assisting areas can be expressed by (13). Then,

$LOLE_{asg}$ is computed based on the CBM allocated for each assisting area, $CBM_{m,asg}$, as discussed in [19]. Consider

$$CBM_{asg} = \mu(xpar_{m,asg}) = \mu(CBM_{m,asg}), \quad (12)$$

$$M.O.F = [CBM_{asg=1}, CBM_{asg=2}, \dots, CBM_{asg=Nasg}]. \quad (13)$$

Therefore, CBM_{asd} for an assisted area is calculated by summing the optimum amount of CBM_{asg} transferred from all the assisting areas as given in

$$CBM_{asd=1} = \sum_{asg=1}^{Nasg} CBM_{asg}. \quad (14)$$

Step (o). Repeat Steps (a)–(n) several times in order to obtain numerous optimal solutions of multiobjective CBM_{asg} . These results will be applied into the Pareto optimal concept in such a way to find several superior multiobjective CBM_{asg} . Figure 1 presents the flowchart of the proposed EP optimization technique used to determine several multiobjective functions of $CBMs$.

3. Ranking Index in the Pareto Optimality Concept for the Best Selection of Optimal Multiobjective Capacity Benefit Margins

Pareto optimality is a concept that has been commonly used to select several optimal solutions of the multiobjective CBM_{asg} designated as multiobjective CBM_{asg}^{Pareto} . This implies that the concept of Pareto does not provide a single solution that can be considered as the global optima for a problem related to the multiobjective CBM_{asg} . This is important to the ISO since it will provide flexibility to select the optimal as well as the most inexpensive result of multiobjective CBM_{asg}^{Pareto} . These inexpensive results usually fall under the cluster of the Pareto optimal front. However, it is not worthy to select an expensive optimal result of multiobjective CBM_{asg} and this type of solution is usually categorized under the cluster of non-Pareto optimal. Figure 2 shows an example elucidating two clusters of the Pareto optimal concept. In Figure 2, $F1$ represents the axis plane of $CBM_{asg=1}$ solution for the transfer case from assisting area 3 to assisted area 1. $F2$ is the axis plane of $CBM_{asg=2}$ solution for the transfer case from assisting area 2 to assisted area 1.

The EP optimization technique is performed several times in order to provide numerous optimal solutions of CBM_{asg} . In addition, solution x is the intersection point for the two CBM_{asg} results. The solutions x marked with a circle represent the cluster of Pareto optimal front. Usually, the best optimal solution of CBM_{asg} , so-called CBM_{asg}^{Pareto} , is selected from the cluster of Pareto optimal front. Solutions x marked with \times represent the cluster of non-Pareto optimal front which do not have the best optimal solution of CBM_{asg} due to their expensive multiobjective function. For instance, this can be observed through the comparison between x_1 and x_3 , which have the same $CBM_{asg=1}$ value for the $F1$ axis, that is, the transfer case from assisting area 3 to assisted area 1.

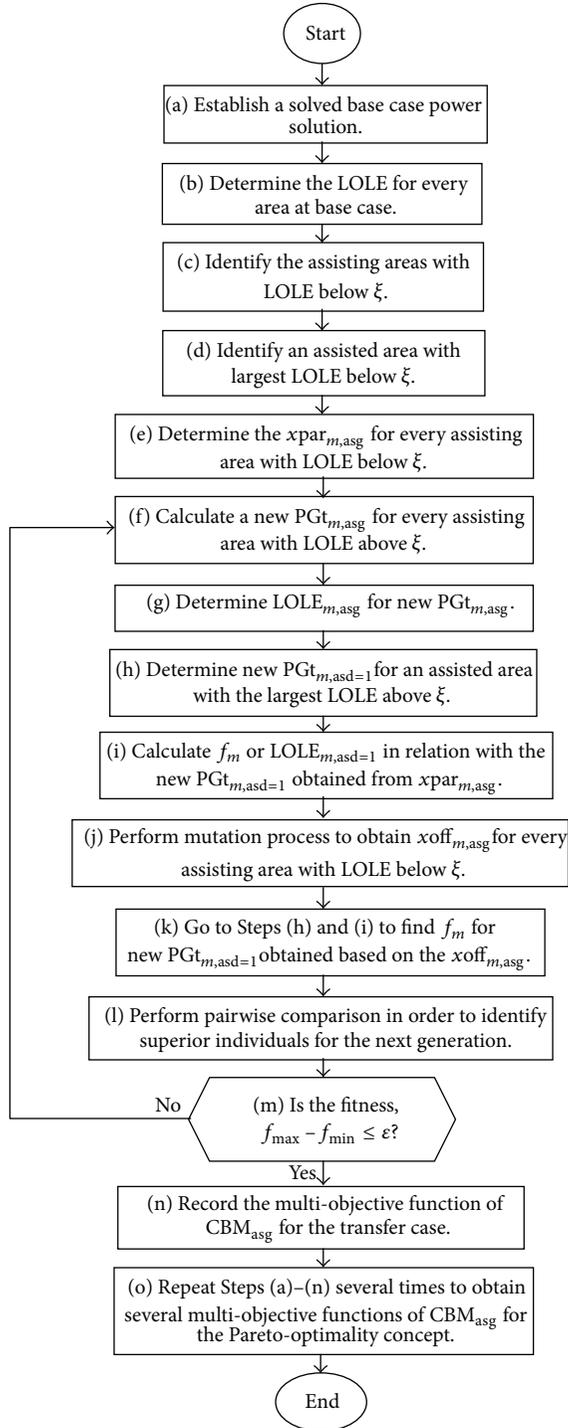


FIGURE 1: Proposed EP technique to determine several multiobjective functions of CBMs.

However, by referring to the $F2$ axis, that is, the transfer case from assisting area 2 to assisted area 1, x_3 yields to an expensive $CBM_{asg=2}$ value compared to x_1 . Thus, x_3 and x_1 are optimal solutions of multiobjective CBM_{asg} which can be categorized under the non-Pareto and Pareto optimal fronts, respectively.

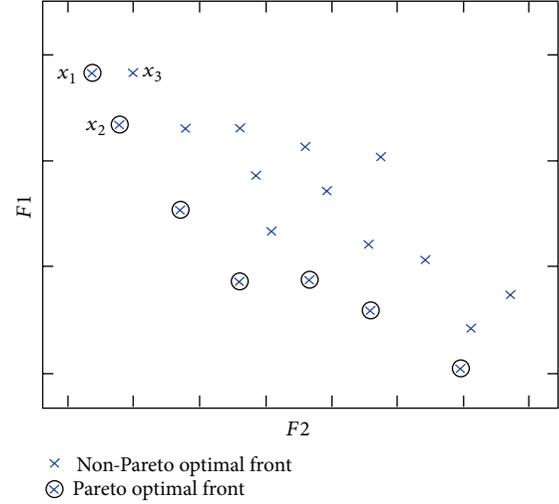


FIGURE 2: Pareto and non-Pareto optimal fronts for the multiobjective function CBM_{asg} .

Theoretically, the Pareto optimal front can be defined as the solution x that is not dominated by any other feasible solutions x [20]. If the domination operator is labeled “ $>$,” the Pareto optimal concept can be described through the following criteria and this is referring to Figure 2.

- (a) $x_1 > x_3$ and $x_2 > x_3$. Hence, the x_3 solution is said to be dominated or a non-Pareto optimal front solution.
- (b) $x_1 > x_2$ and $x_2 > x_1$. Hence, the x_1 and x_2 solutions are said to be nondominated or Pareto optimal front solution.

The aforementioned criteria can also be used to determine the Pareto optimal front for a multiobjective function which has more than two transfer case solutions of CBM_{asg} .

Furthermore, the selection of CBM_{asg}^{Pareto} will be performed by the ISO according to the ranking index of either total LOLE or total LOLE difference. The proposed method has the advantage of introducing CBM_{asg}^{Pareto} which will also provide the optimum results of LOLE and LOLE difference located at the Pareto optimal front cluster. In the initial selection based on the ranking index of total LOLE, CBM_{asg}^{Pareto} is arranged according to the ranking index of total LOLE sorted in an ascending order. Then, the CBM_{asg}^{Pareto} is selected in accordance with the ranking index of total LOLE as shown in

$$CBM_{asg}^{Pareto} \in \text{Rank}(\text{total LOLE}), \quad (15)$$

where

$$\text{total LOLE} = \sum_{asg=1}^{Nasg} \text{LOLE}_{asg}. \quad (16)$$

Equation (15) shows that CBM_{asg}^{Pareto} is selected based on the ranking index of reliability or total LOLE in the assisting areas.

In the subsequent selection based on the ranking index of total LOLE difference, CBM_{asg}^{Pareto} is arranged according to the ranking index of total LOLE difference sorted in an ascending order. Then, the ranking index of total LOLE difference is used to select CBM_{asg}^{Pareto} . This is illustrated in

$$CBM_{asg}^{Pareto} \in \text{Rank}(\text{total } \Delta \text{LOLE}), \quad (17)$$

where

$$\text{total } \Delta \text{LOLE} = \sum_{asg=1}^{N_{asg}} (\text{LOLE}_{asg} - \text{LOLE}_{asg}^o), \quad (18)$$

where LOLE_{asg}^o is the LOLE at the base case condition of each assisting area.

Finally, the selected CBM_{asg}^{Pareto} will be taken into account as firm and nonfirm transfer margins in the ATC determination.

4. Firm and Nonfirm Available Transfer Capability Determination

This section discusses the ATC determination that takes into account each optimum CBM_{asg}^{Pareto} value selected by referring to the ranking index of total LOLE and total LOLE difference. The proposed method uses the iterative power flow solutions to determine ATC by taking into account CBM_{asg}^{Pareto} for the transfer case from an assisting area to an assisted area [21]. Basically, the determination of ATC considering CBM_{asg}^{Pareto} requires an iterative power flow solution to be performed at every increase of generation capacity and load at the respective selling and buying areas until one of the system constraints is met. This method is used to determine ATC considering CBM_{asg}^{Pareto} for the next case of power transfer. It is important to note that two approaches are available to calculate ATC taking into account CBM_{asg}^{Pareto} as firm or nonfirm transfer. In the former approach, the assisting and assisted areas are experiencing changes in total generation capacity according to the firm transfer of CBM_{asg}^{Pareto} , whereas, in the latter approach, ATC is determined as the total transfer capability, TTC, reduced by CBM_{asg}^{Pareto} . The procedure for both approaches discussed in this paper are implemented as follows.

Step (a). Establish a solved base power flow solution.

Step (b). Specify the selling and buying areas for a power transfer.

Step (c). Proceed to Step (e) if CBM_{asg}^{Pareto} is considered to be a nonfirm transfer. Otherwise, adjust the generation outputs according to CBM_{asg}^{Pareto} for all areas. The modification of

generation outputs in assisted area and assisting area is done by using (19) and (20), respectively,

$$\text{new } PG_g^{\text{asd}=1} = PG_g^{\text{asd}=1} - \frac{PG_g^{\text{asd}=1}}{\sum_{g=1}^{NG} PG_g^{\text{asd}=1}} \sum_{asg=1}^{N_{asg}} CBM_{asg}^{Pareto}, \quad (19)$$

$$\text{new } PG_g^{\text{asg}} = PG_g^{\text{asg}} + \frac{PG_g^{\text{asg}}}{\sum_{g=1}^{NG} PG_g^{\text{asg}}} CBM_{asg}^{Pareto}. \quad (20)$$

Notice that (19) and (20) may cause the assisting area to transfer its reverse generation capacity (CBM_{asg}^{Pareto}) for compensating the generation deficiency which may occur in the assisted area. This is different from what has been dealt previously with, with (4) and (6) whereby the generating capacity of an assisting area and assisted area is decreased and increased, respectively, in order to identify the amount of generation capacity reserved for the CBM so that LOLE will be less than ξ .

Step (d). Perform the power flow solution to allow an assisting area to transfer power based CBM_{asg}^{Pareto} required for compensating the generation deficiency occurring in the assisted area.

Step (e). Simultaneously, increase the power injection and extraction at the selling and buying areas, respectively, until either one of the line flows or voltage constraints is met through the load flow solution. The lower and upper voltage limits are considered to be 0.90 and 1.10 p.u., respectively. The injected power is referring to the increase of generation capacity in a selling area resulting in a power transfer which will be extracted by the load increased in a buying area. The maximum power transfer so-called TTC is acquired once the increased power flow solution has met one of the system constraints as mentioned previously.

Step (f). Calculate the ATC at three different cases of TTC determined in Step (e). In conjunction with the TTC^o for the first case, the ATC at base case condition is obtained by employing (21) which does not require the execution of Steps (c) and (d):

$$\text{ATC}^o = \text{TTC}^o - \text{ETC}, \quad (21)$$

where TTC^o is the total transfer capability or the maximum power transfer at base case condition obtained and ETC is the existing transmission commitment or base case load flow solution considering system components variations.

With regard to the TTC^o and CBM_{asg}^{Pareto} for the second case, (22) is used to calculate ATC taking into account nonfirm transfer of CBM:

$$\text{ATC}_{\text{nonfirm}} = \text{TTC}^o - CBM_{asg}^{Pareto} - \text{ETC}. \quad (22)$$

By referring to $\text{TTC}|_{CBM_{asg}^{Pareto}}$ given for the third case, the CBM is taken as a firm transfer for ATC determination and the associated formulation is introduced through

$$\text{ATC}_{\text{firm}} = \text{TTC}|_{CBM_{asg}^{Pareto}} - \text{ETC}. \quad (23)$$

By referring to (23), the modification of generation capacity is performed in Step (c) consecutively with the load flow solution performed in Step (d) so that the ATC is determined by considering the firm transfer of CBM.

Step (g). Repeat Steps (a)–(f) to determine ATC for the next transfer case between the selling and buying areas. The determination of ATC for the next transfer case will also consider the same CBMs determined for the assisting and assisted areas.

The flowchart of ATC determination that takes into account the firm and nonfirm transfer margins of CBM_{asg}^{Pareto} is illustrated in Figure 3.

5. Results and Discussion

A modified IEEE-RTS79 is used to demonstrate the effectiveness of the proposed method in determining the CBM for each area [19, 22]. The generating units and transmission line information are given in [19, 22]. In this paper, the specified value of LOLE limit, ξ , is assumed to be 2.4 hrs/yr.

5.1. Capacity Benefit Margin Considering Interconnected System Reliability. In the base case condition of a modified IEEE-RTS79, the total generation, total load, and LOLE associated with each area is presented in Table 1. Based on the predetermined LOLE, areas 2 and 3 are considered the assisting areas and area 1 is referred to as the assisted area.

Table 2 presents the results of CBM considering tie-line reliability and is determined using the basic methodology discussed in [19]. It is observed that 88 MW and 33 MW are the amount of CBM reserved for the transfer from assisting areas 2 and 3 to area 1, respectively, resulting in the LOLE value being below 2.4 hrs/yr. Hence, new generation capacities of 2156 MW, 1660 MW, and 751 MW are obtained for areas 1, 2, and 3, respectively.

5.2. Multiobjective Capacity Benefit Margins Result Determined by the Ranking Index in Pareto-Based Evolutionary Programming Technique. It is noteworthy that Table 1 has presented the total generation capacity and total load for every area at base case condition of IEEE-RTS79. In conjunction with this matter, the LOLE less than 2.4 hrs/yr implies that the assisting areas 2 and 3 have sufficient amount of total reserve generation capacity that can be used as a reference to estimate the amount of CBM for accommodating the generation deficiency which may occur in the assisted area 1 with LOLE above 2.4 hrs/yr. Hence, the EP optimization technique is used to perform simultaneous determination of CBM that can be transferred from the assisting areas 2 and 3 towards the assisted area 1.

In the EP optimization technique, there are 10 individuals in a population representing the $xpar_{m,asg=1}$ or CBMs for assisting area 2. The same situation goes to the next population representing the $xpar_{m,asg=2}$ or CBMs for assisting area 3. The initial process of EP optimization technique will randomly generate a uniform distribution of $xpar_{m,asg}$ using (1) based on the reserve generating capacity available

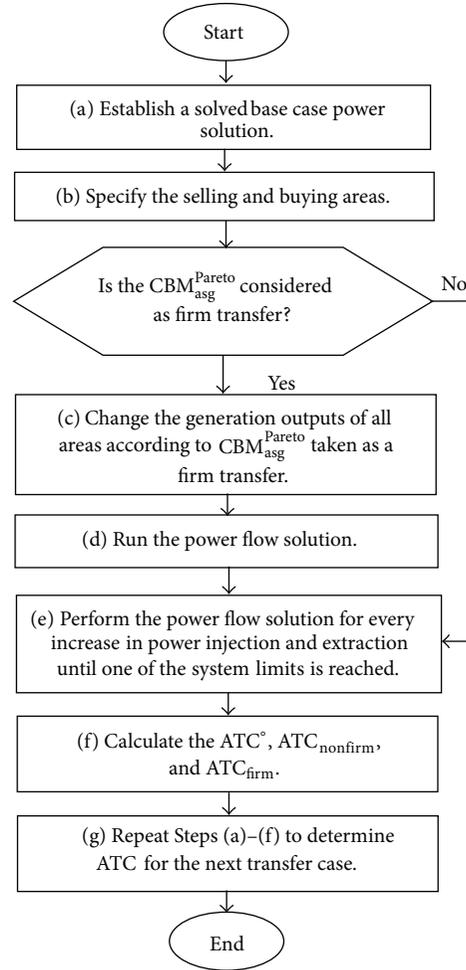


FIGURE 3: Flowchart of firm and nonfirm ATC determination technique.

TABLE 1: Generation, load, and LOLE for the three areas.

Area	Generation [MW]	Load [MW]	LOLE [hrs/yr]
1	2035	1125	4.7756
2	1748	1141	0.6380
3	784	584	0.6917

TABLE 2: CBM results considering interconnected system reliability using the method introduced in [19].

Area	Generation [MW]	CBM [MW]	LOLE [hrs/yr]
Assisted area 1	2156	121	2.3972
Assisting area 2	1660	88	1.3943
Assisting area 3	751	33	1.3569

in the assisting area. In particular, the initial population, $xpar_{m,asg=1}$, for assisting area 2 is obtained through the randomly generated variables that are in the range of 1 MW and 607 MW. This signifies that $1748 \text{ MW} - 1141 \text{ MW} = 607 \text{ MW}$ is the reserved generating capacity available in the assisting area 2. In the overleaf case, that is, referring to

the assisting area 3, the initial population, $xpar_{m,asg=2}$, is obtained via the randomly generated variables which are within the range of 1 MW and 200 MW. Both of the $xpar_{m,asg}$ representing the initial population for assisting area 2 and area 3 are tabulated in Table 3. Simultaneously, both of the initial populations are applied into the mutation in (7) and pairwise comparison process (10) to obtain $xoff_{m,asg}$ and a new $xpar_{m,asg}$, respectively, for the assisting areas 2 and 3. All of the optimization process embedded in the EP optimization technique is repeated until the difference between maximum fitness, f_{max} , and minimum fitness, f_{min} , for the assisted area 1 is equal or less than the specified $\epsilon = 0.1$. In the last iteration of EP optimization process, the average value of $xpar_{m,asg}$ for both populations represents the optimum value of CBM for assisting areas 2 and 3. The $xpar_{m,asg}$ obtained at the final iteration of EP optimization process are shown in Table 4. In relation to each population of $xpar_{m,asg}$, it is obvious that a relatively similar value is obtained for all of the individuals, and the average value of $xpar_{m,asg}$ in (12) may yield to CBM specified for the assisting areas 2 and 3. This result is obtained only for one optimization run of EP technique. The EP optimization technique is executed for several times so that the Pareto optimal fronts of CBMs (CBM_{asg}^{Pareto}) are obtained which provides flexibility to the transmission provider in selecting optimum CBMs in tandem with the changes of economic, load-serving entity requirement or resource planner. The analysis of CBM_{asg}^{Pareto} will be elucidated in the following discussion.

Figure 4 shows different optimized values of CBM obtained at every execution of the EP optimization process. The x -axis represents the CBM transferred from the assisting area 2 to assisted area 1, whereas the y -axis represents the CBM transferred from assisting area 3 to assisted area 1.

It is observed that, with an increase in CBM associated with a particular assisting area, CBM at the other assisting area would decrease and vice versa. The best optimum values for the multiobjective function of CBMs are obtained based on the Pareto optimal front and the cluster for this case is illustrated in Figure 4. The other cluster represents the non-Pareto optimal front of CBMs with excessive value which may yield to an invidious violation of power system security and ineffective utilization of the existing network resources. Figure 5 represents the cluster of Pareto optimal front of CBMs extracted from Figure 4. In the Pareto optimal front, the results of CBM have less potential in violating system security compared with the excessive amount of CBMs obtained based on the non-Pareto optimal front.

Furthermore, the Pareto optimal front approach used in the EP technique gives sufficient flexibility to the ISO in selecting the optimum value of CBM for every transfer case depending on the system requirements. This is obviously contradictory with CBM results tabulated in Table 2 which are obtained using a basic approach [19]. Based on the CBM results shown in Table 2, ISO does not have the flexibility to select other choices with suitable set of CBMs for compensating any generation deficiency at different system operating states. In relation to Figure 5, CBM results for each area yielding to the Pareto optimal front are also tabulated in

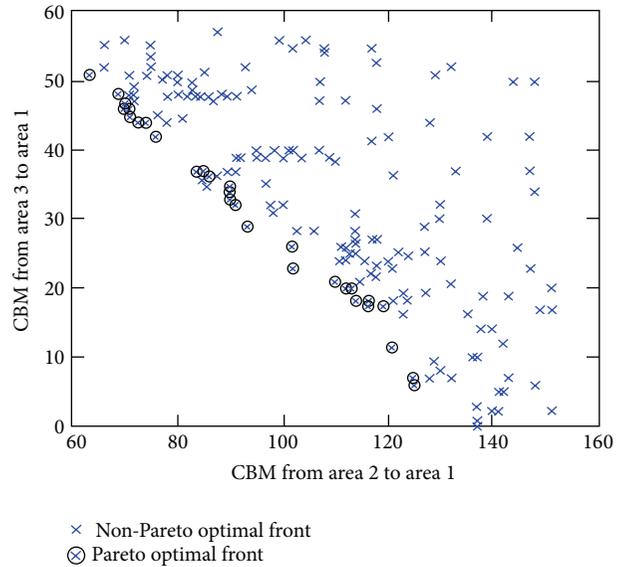


FIGURE 4: Pareto and non-Pareto optimal fronts of CBM for the two transfer cases.

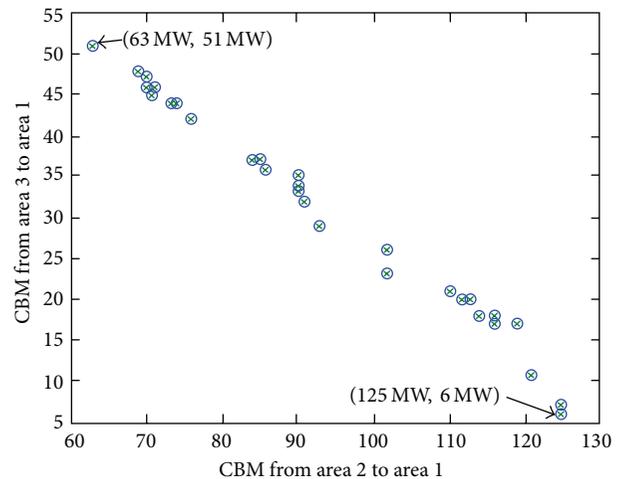


FIGURE 5: Pareto optimal fronts of CBMs for the two transfer cases.

Table 5. Every result of Pareto optimal front CBM will be used as a reference to estimate the power transferred from assisting areas 2 and 3 to accommodate possible generation deficiency in the assisted area 1.

It is observed that the Pareto optimal front of CBM values was obtained while fulfilling the LOLE criterion of less than 2.4 hrs/yr. The other advantage of the proposed method is that CBM_{asg}^{Pareto} results also yield Pareto optimal front clusters of LOLE and difference in LOLE values. This can be verified in Figure 6 where LOLE located at the Pareto optimal front cluster refers to the CBM_{asg}^{Pareto} results obtained for each case of power transfer depicted in Figure 4. Consequently, the results of total LOLE obtained through (15) are arranged in ascending order and the ranking index is assigned to every result to distinguish the reliability of the assisting areas shown

TABLE 3: Initial population of EP technique for the assisting areas 2 and 3.

Number of individuals	Assisting area 2		Assisting area 3	
	$xpar_{m,asg=1}$ or CBM (MW)	LOLE (hrs/yr)	$xpar_{m,asg=2}$ or CBM (MW)	LOLE (hrs/yr)
1	474.61	52.67	47.96	1.89
2	237.57	5.22	71.63	3.44
3	147.71	2.29	165.24	37.63
4	246.18	5.46	4.08	0.70
5	59.55	1.03	9.60	0.80
6	81.11	1.25	34.80	1.43
7	572.83	137.47	130.82	15.57
8	581.37	148.06	147.35	23.20
9	350.15	15.24	130.55	15.57
10	37.29	0.85	91.19	5.62

TABLE 4: Final population for the assisting areas 2 and 3 based on one run of EP optimization process.

Number of individuals	Assisting area 2		Assisting area 3	
	$xpar_{m,asg=1}$ or CBM (MW)	LOLE (hrs/yr)	$xpar_{m,asg=2}$ or CBM (MW)	LOLE (hrs/yr)
1	86	1.32	35	1.43
2	86	1.32	35	1.43
3	86	1.32	35	1.43
4	86	1.32	35	1.43
5	86	1.32	35	1.43
6	86	1.32	35	1.43
7	86	1.32	35	1.43
8	86	1.32	35	1.43
9	85	1.29	35	1.43
10	85	1.29	35	1.43

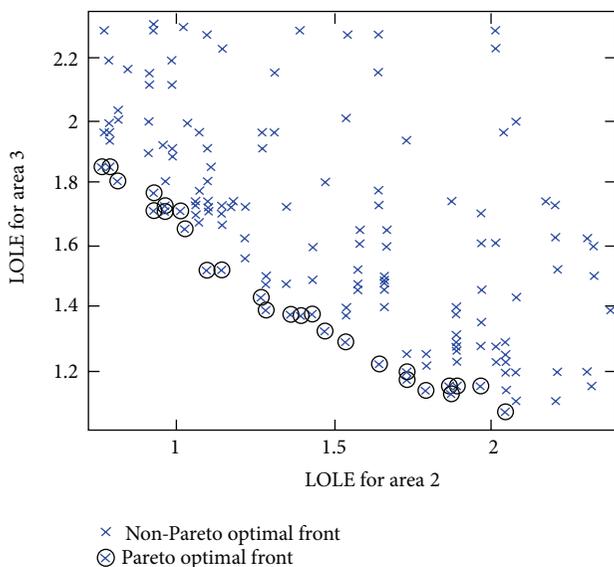


FIGURE 6: Pareto optimal fronts of LOLE for areas 2 and 3.

in Table 6. With respect to each value of total LOLE, total CBM_{asg}^{Pareto} was obtained based on the two transfer cases also

shown in Table 6. The total CBM_{asg}^{Pareto} is equivalent to CBM for an assisted area, $CBM_{asd=1}$.

Figure 7 represents the difference between LOLE values located at the Pareto optimal front cluster. The results are referring to the CBM_{asg}^{Pareto} obtained based on the power transfer cases shown in Figure 4. Then, the results of total LOLE difference calculated using (17) were arranged in ascending order and the ranking index is assigned to each result indicating level of reliability available for the assisting areas as shown in Table 6. Table 6 reveals that the total CBM_{asg}^{Pareto} or CBM_{asd}^{Pareto} values were arranged according to the total LOLE and total LOLE difference possessing the same ranking index. The results divulge that the Pareto-based EP method has the advantage of providing simultaneous optimum results of CBM_{asg}^{Pareto} , LOLE, and difference of LOLE in which all are located at the Pareto optimal front cluster.

As noted earlier and clearly presented in Table 6, the proposed method has the advantage of providing several choices of CBM that can be selected by ISO based on the ranking index of total LOLE and/or total LOLE difference. For instance, the ISO shall set the CBM_{asg}^{Pareto} , respectively, to 125 MW and 6 MW for the transfer case from assisting areas 2 and 3, respectively, to the assisted area 1 so that the assisting areas will operate in a highly reliable condition because

TABLE 5: Pareto optimal front of CBM values for each area.

EP run	Assisted area 1		Assisting area 2		Assisting area 3	
	CBM [MW]	LOLE [hrs/yr]	CBM [MW]	LOLE [hrs/yr]	CBM [MW]	LOLE [hrs/yr]
1	114	2.3997	63	1.0646	51	2.0485
2	117	2.3627	69	1.1475	48	1.8874
3	117	2.3794	73	1.1739	44	1.731
4	121	2.3967	85	1.2913	36	1.4705
5	117	2.3656	70	1.1458	47	1.9693
6	117	2.3723	71	1.1237	46	1.8714
7	116	2.3867	70	1.1458	46	1.8714
8	116	2.3941	71	1.1237	45	1.7887
9	118	2.3641	74	1.1847	44	1.731
10	118	2.3667	76	1.2076	42	1.6394
11	121	2.3996	86	1.3183	35	1.431
12	122	2.3785	86	1.3183	36	1.4705
13	125	2.3411	90	1.3775	35	1.431
14	121	2.3954	84	1.2902	37	1.539
15	122	2.3766	85	1.2913	37	1.539
16	124	2.3534	90	1.3775	34	1.3913
17	123	2.3695	90	1.3775	33	1.3569
18	123	2.3711	91	1.3914	32	1.2769
19	122	2.3931	93	1.428	29	1.2597
20	128	2.3248	102	1.5166	26	1.1351
21	125	2.3697	102	1.5166	23	1.0929
22	131	2.3091	110	1.6479	21	1.0247
23	132	2.3999	112	1.7002	20	1.0107
24	133	2.3934	113	1.7095	20	1.0107
25	132	2.3985	114	1.7198	18	0.9511
26	134	2.373	116	1.7116	18	0.9511
27	133	2.3841	116	1.7116	17	0.9235
28	136	2.3455	119	1.7702	17	0.9235
29	132	2.3932	121	1.8074	11	0.8044

the aforementioned power transfers are obtained based on the lowest total LOLE of 2.608 hrs/yr at the 1st ranking index. The combination of CBM_{asg}^{Pareto} for both transfer cases will provide a total CBM_{asg}^{Pareto} of 131 MW which results in the lowest total LOLE difference of 1.278 hrs/yr at the 1st ranking index as shown in Table 6. Due to a relatively large total CBM_{asg}^{Pareto} of 131 MW, the tie-line capacity will not be fully utilized as a medium power transfer based ATC for electricity transfer. The total CBM_{asg}^{Pareto} of 131 MW can also be obtained at the 10th ranking index as shown in Table 6. However, a total CBM_{asg}^{Pareto} of 131 MW at the 10th ranking index will not be the best choice for the ISO since the assisting areas will operate in a less reliable condition due to the total LOLE of 2.673 hrs/yr and total LOLE difference of 1.343 hrs/yr. Furthermore, the 10th ranking index yields to a result that is close with the largest total CBM_{asg}^{Pareto} of 136 MW located at the 12th ranking

index. However, total LOLE of 2.694 hrs/yr and total LOLE difference of 1.364 hrs/yr signify a reasonable or moderately reliable operation of the assisting areas in conjunction with the largest total CBM_{asg}^{Pareto} of 136 MW at the 12th ranking index.

In another situation whereby the ISO is not interested in a highly reliable condition of a power system, the CBM_{asg}^{Pareto} of 70 MW can be selected for the transfer case from assisting area 2 to area 1 and the CBM_{asg}^{Pareto} of 47 MW can be chosen for the transfer case from assisting area 3 to assisted area 1. This would be a less reliable choice prior to the largest value of total LOLE which is 3.115 hrs/yr at the 29th ranking index as tabulated in Table 6. Consequently, the total CBM_{asg}^{Pareto} of 117 MW is obtained contributing to the largest total LOLE difference of 1.785 hrs/yr located at the 29th ranking index. For this case, a highly reliable condition incurred from

TABLE 6: CBM results with ranking index of total LOLE and total LOLE difference.

CBM ^{Pareto} _{asd} received by area 1 [MW]	CBM ^{Pareto} _{asg} from area 2 to area 1 [MW]	CBM ^{Pareto} _{asg} from area 3 to area 1 [MW]	Total LOLE [hrs/yr]	Total difference of LOLE [hrs/yr]	Rank index
131	125	6	2.608	1.278	1
125	102	23	2.610	1.280	2
132	121	11	2.612	1.282	3
132	125	7	2.629	1.300	4
133	116	17	2.635	1.305	5
128	102	26	2.652	1.322	6
134	116	18	2.663	1.333	7
123	91	32	2.668	1.339	8
132	114	18	2.671	1.341	9
131	110	21	2.673	1.343	10
122	93	29	2.688	1.358	11
136	119	17	2.694	1.364	12
132	112	20	2.711	1.381	13
133	113	20	2.720	1.391	14
123	90	33	2.734	1.405	15
124	90	34	2.769	1.439	16
122	86	36	2.789	1.459	17
125	90	35	2.809	1.479	18
121	84	37	2.829	1.500	19
122	85	37	2.830	1.501	20
118	76	42	2.847	1.517	21
117	73	44	2.905	1.575	22
116	71	45	2.912	1.583	23
118	74	44	2.916	1.586	24
117	71	46	2.995	1.665	25
116	70	46	3.017	1.688	26
117	69	48	3.035	1.705	27
114	63	51	3.113	1.783	28
117	70	47	3.115	1.785	29

a specific amount of CBM reserved through tie-line capacity is not the main intention for the ISO. Besides, ISO is more interested in the utilization of tie-line capacity for ATC in order to enhance and perform as an important role in the electricity market. Similar to the 29th ranking index, the CBM^{Pareto}_{asg} of 117 MW at the 22nd ranking index can also be used in this case study. It has the advantage in providing total LOLE of 2.905 hrs/yr and total LOLE difference of 1.575 hrs/yr which is much better than the results obtained at the 29th ranking index. By comparing with the total CBM of 117 MW at the 22nd ranking index, ISO may choose the lowest value of total CBM, that is, 114 MW at the 28th ranking index, only when the objective is not solely on the reliability improvement of the assisting areas.

In a detailed analysis, ISO may select the CBM^{Pareto}_{asg} of 90 MW and 33 MW for the transfer case from the assisting areas 2 and 3 to the assisted area 1, respectively, so that

the assisting areas are operating at the mid ranking level (index 15) having the total LOLE of 2.734 hrs/yr and total LOLE difference of 1.405 hrs/yr. This indicates that ISO has chosen the value of CBM^{Pareto}_{asg} for both transfer cases resulting in 50% priority on the reliability of assisting areas and 50% priority on the power transfer based ATC reserved for electricity market activities. The aforementioned discussion shows that the optimal value of CBM specified for each case of power transfer is actually dependent on similar ranking indices of total LOLE and total LOLE difference.

The previous results have well demonstrated that CBM^{Pareto}_{asg}, LOLEs, and difference of LOLEs clustered in the Pareto optimal front are the criteria to be satisfied by the ISO before conducting the finest selection of CBM^{Pareto}_{asg}. The performance of the Pareto optimal front embedded in the proposed optimization technique is not limited only to the CBM^{Pareto}_{asg} value that provides the highest reliability

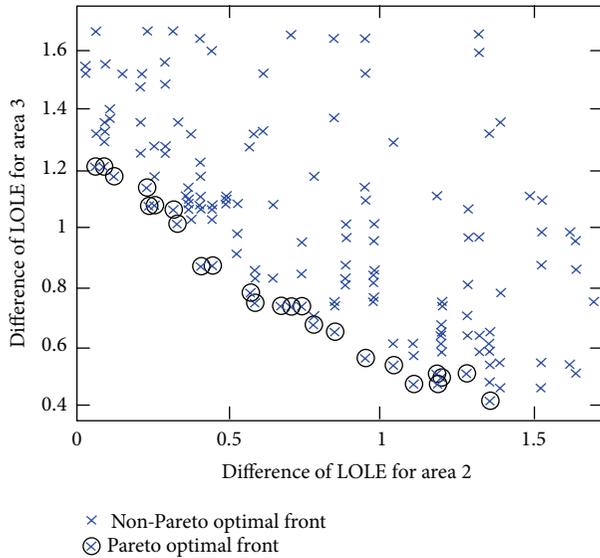


FIGURE 7: Pareto optimal fronts of LOLE difference for areas 2 and 3.

of assisting areas due to the lowest total LOLE and total LOLE difference associated with the 1st ranking index. Nevertheless, it also is not confined to the CBM_{asg}^{Pareto} value with large amount of ATC yielding to the largest total LOLE and total LOLE difference selected at the 29th ranking index. This implies that ISO has several choices for CBM for each case of power transfer depending on the ranking index selected based on the Pareto optimal front of total LOLE and total LOLE difference.

5.3. Performance Comparison with Existing Capacity Benefit Margin Calculation Methods. It is worthwhile to mention that the proposed method is robust in providing simultaneous optimum results of the CBM_{asg}^{Pareto} , LOLE, and difference in LOLE, all of which are located at the Pareto optimal front cluster. In the proposed method, the ranking index has the advantage of providing a clearer depiction on the relationship between the three optimal results which will be a great help to the ISO in making the finest decision for selecting optimum CBM values. This is contradictory to other methods in [17], whereby the optimization process is performed separately to find the minimum total LOLE, minimum total LOLE difference, or minimum CBM considering weight of the tie-lines. As shown in Figure 8, the lowest total LOLE of 2.608 hrs/yr and lowest total LOLE difference of 1.278 hrs/yr, computed using the method presented in [17], will give a total CBM result of 131 MW which is quite large according to the Pareto optimal front tabulated in Table 6. Using both methods discussed in [17], ISO does not have a choice other than to utilize a large total CBM value of 131 MW to ensure a highly reliable operating condition of the assisting areas in accordance with the lowest total LOLE of 2.608 hrs/yr and lowest total LOLE difference of 1.278 hrs/yr. Thus, the proposed method of Pareto optimal front provides a solution to the abovementioned problem by

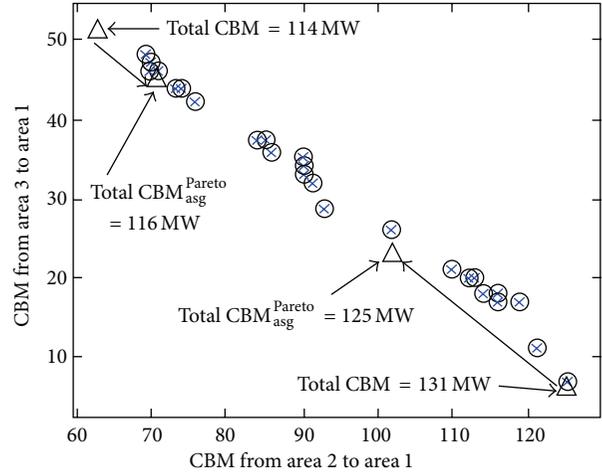


FIGURE 8: Total CBM_{asg}^{Pareto} selection based on the ranking index in Pareto optimal front concept.

providing the total CBM_{asg}^{Pareto} of 125 MW at the 2nd ranking index considered as the other option that is smaller than the total CBM of 131 MW at the 1st ranking index in Table 6 and Figure 8. The result of total CBM_{asg}^{Pareto} , that is, 125 MW at the 2nd ranking index, also provides a highly reliable operating condition for the assisting areas that are nearly identical with the lowest total LOLE and lowest total LOLE difference at the 1st ranking index. Consequently, the total CBM_{asg}^{Pareto} of 125 MW at the 2nd ranking index provides a more conservative space to transfer the power based ATC compared with the total CBM of 131 MW at the 1st ranking index.

The method discussed in [17] provides some limited choices of CBM results as it considers the weight specified on each tie-line. However, the proposed method provides several CBM_{asg}^{Pareto} values without considering the weight for the tie-lines. Once the tie-line weight is not considered in [17], the minimum total CBM of 114 MW is obtained. It is depicted in Figure 8 and Table 6 that the total CBM of 114 MW is obtained at the assisted area when the CBMs of 63 MW and 51 MW are transferred from the assisting areas 2 and 3, respectively. However, the total LOLE of 3.113 hrs/yr and total LOLE difference of 1.783 hrs/yr are relatively large at the 28th ranking index although the minimum total CBM of 114 MW is obtained using the abovementioned equation given in [17]. Therefore, the proposed Pareto optimal front concept is used to provide several choices of solution that are relatively similar to the minimum total CBM of 114 MW. For this case, the total CBM_{asg}^{Pareto} of 116 MW is chosen from the 23rd ranking index of Pareto optimal front and it is the nearest value to the minimum total CBM of 114 MW. By referring to Figure 8 and Table 6, it can be observed that the total CBM_{asg}^{Pareto} of 116 MW will improve the reliability of the assisting areas due to the total LOLE of 2.912 hrs/yr and total LOLE difference of 1.583 hrs/yr which are smaller than the LOLE results obtained from the minimum total CBM of 114 MW. For other cases of

TABLE 7: Results of ATC from area 1 to area 2.

EP run	ATC _{base} [MW]	ATC _{firm} [MW]	ATC _{nonfirm} [MW]
1	586	542	523
2	586	538	517
3	586	535	513
4	586	527	501
5	586	537	516
6	586	537	515
7	586	537	516
8	586	537	515
9	586	535	512
10	586	533	510
11	586	526	500
12	586	526	500
13	586	524	496
14	586	528	502
15	586	527	501
16	586	524	496
17	586	524	496
18	586	523	495
19	586	522	493
20	586	515	484
21	586	515	484
22	586	510	476
23	586	508	474
24	586	508	473
25	586	507	472
26	586	506	470
27	586	506	470
28	586	504	467
29	586	502	465

TABLE 8: Results of ATC from area 1 to area 3.

EP run	ATC _{base} [MW]	ATC _{firm} [MW]	ATC _{nonfirm} [MW]
1	271	258	220
2	271	259	223
3	271	260	227
4	271	262	235
5	271	259	224
6	271	259	225
7	271	259	225
8	271	260	226
9	271	260	227
10	271	260	229
11	271	262	236
12	271	262	235
13	271	262	236
14	271	262	234
15	271	262	234
16	271	263	237
17	271	263	238
18	271	263	239
19	271	264	242
20	271	265	245
21	271	265	248
22	271	266	250
23	271	266	251
24	271	266	251
25	271	267	253
26	271	267	253
27	271	267	254
28	271	267	254
29	271	268	260

different weights assigned to each tie-line, the selection of CBM result is performed similarly by referring to the abovementioned explanation of Pareto optimal front concept.

5.4. *Results of Available Transfer Capability Incorporating Capacity Benefit Margin.* In this section, the ATC results are obtained based on the four cases of power transfer as shown in Tables 7, 8, 9, and 10. The results of ATCs are obtained by considering the firm and nonfirm transfers of CBM_{asg}^{Pareto} located at the Pareto optimal front cluster as depicted in Table 6. Hence, for every case of power transfer, there are 29 results of ATC that give the flexibility to the ISO in choosing a suitable power transfer. It is noted that CBM_{asg}^{Pareto} taken as a firm transfer contributes to slightly larger ATC values as compared to CBM_{asg}^{Pareto} which is taken as a nonfirm transfer. It can be concluded that, by incorporating the nonfirm transfer of CBM_{asg}^{Pareto} into ATC, there will be loss for certain amount of ATC in the power transfer contracts.

6. Conclusion

This paper has presented a new approach for calculating CBM taking into account tie-line reliability in the interconnected system. The proposed approach employs the ranking index in a Pareto-based EP technique that provides several choices of optimum CBM values. The effectiveness of the proposed method in determining the CBM has been tested on the modified IEEE-RTS79. The results presented have shown that the Pareto optimal front of CBMs is an inexpensive solution compared to the CBMs located at the non-Pareto optimal front. The other advantage associated with the proposed method is due to its ability in providing simultaneous optimal results of CBM, LOLE, and LOLE difference whereby all are located at the Pareto optimal front cluster. Hence, selection of the result does not rely solely on the value of CBM, but it is also concurrently based on the impact of total LOLE and total LOLE difference included under the ranking of Pareto optimal front. In short, ISO has the flexibility to select the CBM at the Pareto optimal front referring to the ranking index of total LOLE and total difference of LOLE. Finally,

TABLE 9: Results of ATC from area 2 to area 1.

EP run	ATC _{base} [MW]	ATC _{firm} [MW]	ATC _{nonfirm} [MW]
1	1171	1115	1108
2	1171	1110	1102
3	1171	1106	1098
4	1171	1096	1086
5	1171	1109	1101
6	1171	1108	1100
7	1171	1109	1101
8	1171	1108	1100
9	1171	1106	1097
10	1171	1104	1095
11	1171	1095	1085
12	1171	1095	1085
13	1171	1091	1081
14	1171	1097	1087
15	1171	1096	1086
16	1171	1091	1081
17	1171	1091	1081
18	1171	1090	1080
19	1171	1088	1078
20	1171	1080	1069
21	1171	1080	1069
22	1171	1073	1061
23	1171	1071	1059
24	1171	1070	1058
25	1171	1069	1057
26	1171	1067	1055
27	1171	1067	1055
28	1171	1065	1052
29	1171	1063	1050

TABLE 10: Results of ATC from area 3 to area 1.

EP run	ATC _{base} [MW]	ATC _{firm} [MW]	ATC _{nonfirm} [MW]
1	71	21.5	20
2	71	24.5	23
3	71	28.5	27
4	71	36.5	35
5	71	25.5	24
6	71	26.5	25
7	71	26.5	25
8	71	27.5	26
9	71	28.5	27
10	71	30.5	29
11	71	37.5	36
12	71	36.5	35
13	71	37.5	36
14	71	35.5	34
15	71	35.5	34
16	71	38.5	37
17	71	39.5	38
18	71	40.5	39
19	71	43.5	42
20	71	46.5	45
21	71	49.5	48
22	71	51.5	50
23	71	52.5	51
24	71	52.5	51
25	71	54.5	53
26	71	54.5	53
27	71	55.5	54
28	71	55.5	54
29	71	21.5	20

CBM taken as a firm transfer yields to a relatively large value of ATC compared to CBM considered as nonfirm transfer.

Conflict of Interests

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

Acknowledgments

This work was supported in part by the Research Management Institute (RMI), Universiti Teknologi MARA, Malaysia, under Grants 600-RMI/DANA 5/3/PSI (186/2013) and 600-RMI/DANA 5/3/CFI (56/2013); the Ministry of Higher Education (MOHE), Malaysia, under Grant 600-RMI/ERGS5/3 (18/2012); and the Ministry of Science, Technology and Innovation (MOSTI), Malaysia, under Grant 03-01-01-SF0476.

References

- [1] J. Zhang and A. Yokoyama, "Application of interline power flow controller to ATC enhancement by optimal power flow control," in *Proceedings of the IEEE Lausanne POWERTECH*, pp. 1226–1231, Lausanne, Switzerland, July 2007.
- [2] M. M. Othman, A. Mohamed, and A. Hussain, "Determination of transmission reliability margin using parametric bootstrap technique," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1689–1700, 2008.
- [3] A. V. Gheorghe, M. Masera, M. Weijnen, and L. de Vries, *Critical Infrastructures at Risk: Securing the European Electric Power System*, Springer, 2006.
- [4] *Available Transfer Capability Definitions and Determination*, NERC Report, North American Electric Reliability Council, Atlanta, Ga, USA, 1996.
- [5] P. W. Sauer, "Technical challenges of computing available transfer capability (ATC) in electric power systems," in *Proceedings*

- of the 30th Annual Hawaii International Conference on System Sciences, pp. 589–593, January 1997.
- [6] K. Thai and T. Tran, “Use of capacity benefit margin,” NERC Standard MOD-006-0.1 and MOD-007-0, Tacoma Public Utilities, 2009.
- [7] S. C. Savulescu, “A metric for quantifying the risk of blackout,” in *Proceedings of the IEEE PES Power Systems Conference and Exposition*, pp. 1661–1664, October 2004.
- [8] P. Kundur, J. Paserba, V. Ajjarapu et al., “Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions,” *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1387–1401, 2004.
- [9] I. Dobson, S. Greene, R. Rajaraman et al., “Electric power transfer capability: concepts, applications, sensitivity and uncertainty,” PSERC Publication 01-34, Power Systems Engineering Research Centre, Cornell University, New York, NY, USA, 2001.
- [10] Y. Ou and C. Singh, “Assessment of available transfer capability and margins,” *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 463–468, 2002.
- [11] Y. Liu, J. Wang, L. Zhang, and D. Zou, “Research on effect of renewable energy power generation on available transfer capability,” *Journal of Software*, vol. 8, no. 4, pp. 802–808, 2013.
- [12] H. Farahmand, M. Rashidinejad, A. Mousavi, A. A. Gharaveisi, M. R. Irving, and G. A. Taylor, “Hybrid mutation particle swarm optimization method for available transfer capability enhancement,” *International Journal of Electrical Power and Energy Systems*, vol. 42, no. 1, pp. 240–249, 2012.
- [13] T. Akbari, A. Rahimikian, and A. Kazemi, “A multi-stage stochastic transmission expansion planning method,” *Energy Conversion and Management*, vol. 52, no. 8-9, pp. 2844–2853, 2011.
- [14] R.-F. Sun, Y.-H. Song, and Y.-Z. Sun, “Capacity benefit margin assessment based on multi-area generation reliability exponential analytic model,” *IET Generation, Transmission and Distribution*, vol. 2, no. 4, pp. 610–620, 2008.
- [15] R. Rajathy, R. Gnanadass, K. Manivannan, and H. Kumar, “Computation of capacity benefit margin using differential evolution,” *International Journal of Computing Science and Mathematics*, vol. 3, no. 3, pp. 275–287, 2010.
- [16] M. M. Othman, A. Mohamed, and A. Hussain, “Available transfer capability assessment using evolutionary programming based capacity benefit margin,” *International Journal of Electrical Power and Energy Systems*, vol. 28, no. 3, pp. 166–176, 2006.
- [17] M. Ramezani, M. R. Haghifam, C. Singh, H. Seifi, and M. P. Moghaddam, “Determination of capacity benefit margin in multiarea power systems using particle swarm optimization,” *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 631–641, 2009.
- [18] M. Ramezani, H. Falaghi, and C. Singh, “Capacity benefit margin evaluation in multi-area power systems including wind power generation using particle swarm optimization,” in *Wind Power Systems*, Green Energy and Technology, pp. 105–123, Springer, Berlin, Germany, 2010.
- [19] N. B. A. Rahman, M. M. Othman, I. Musirin, A. Mohamed, and A. Hussain, “Capacity Benefit Margin (CBM) assessment incorporating tie-line reliability,” in *Proceedings of the 4th International Power Engineering and Optimization Conference (PEOCO '10)*, pp. 337–344, Shah Alam Selangor, Malaysia, June 2010.
- [20] D. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley, New York, NY, USA, 1989.
- [21] N. A. Salim, M. M. Othman, M. S. Serwan, M. Fotuhi-Firuzabad, A. Safdarian, and I. Musirin, “Determination of available transfer capability with implication of cascading collapse uncertainty,” *IET Generation, Transmission and Distribution*, vol. 8, no. 4, pp. 705–715, 2014.
- [22] P. M. Subcommittee, “IEEE reliability test system,” *IEEE Transactions on Power Apparatus and Systems*, vol. 98, no. 6, pp. 2047–2054, 1979.

Research Article

New Enhanced Artificial Bee Colony (JA-ABC5) Algorithm with Application for Reactive Power Optimization

Noorazliza Sulaiman,¹ Junita Mohamad-Saleh,¹ and Abdul Ghani Abro²

¹ School of Electrical & Electronic Engineering, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia

² College of Engineering, King Saud University, Muzahmyiah Campus, Riyadh 11451, Saudi Arabia

Correspondence should be addressed to Junita Mohamad-Saleh; jms@usm.my

Received 24 June 2014; Accepted 6 October 2014

Academic Editor: Ahmad T. Azar

Copyright © 2015 Noorazliza Sulaiman et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The standard artificial bee colony (ABC) algorithm involves exploration and exploitation processes which need to be balanced for enhanced performance. This paper proposes a new modified ABC algorithm named JA-ABC5 to enhance convergence speed and improve the ability to reach the global optimum by balancing exploration and exploitation processes. New stages have been proposed at the earlier stages of the algorithm to increase the exploitation process. Besides that, modified mutation equations have also been introduced in the employed and onlooker-bees phases to balance the two processes. The performance of JA-ABC5 has been analyzed on 27 commonly used benchmark functions and tested to optimize the reactive power optimization problem. The performance results have clearly shown that the newly proposed algorithm has outperformed other compared algorithms in terms of convergence speed and global optimum achievement.

1. Introduction

Bioinspired algorithms (BIAs) are metaheuristics method that imitates the biological phenomenon of nature [1, 2]. Various BIAs have been developed to solve complex optimization problems. For example, Davidović et al. (2011) have implemented Bee Colony Optimization (BCO) algorithm to solve p -center problem [3] and, in 2012, Badar et al. have used particle swarm optimization (PSO) algorithm to handle a reactive power control problem [4]. Karaboga and Latifoglu then applied artificial bee colony (ABC) algorithm as a tool to solve adaptive filtering noisy transcranial Doppler signal [5]. Bacanin and Tuba (2014) have recently employed firefly algorithm to encounter cardinality constrained mean-variance portfolio optimization problem [6]. A few new BIAs have also been developed such as in the work of Obagbuwa and Adewumi that introduced Improved Cockroach Swarm Optimization (CSO) algorithm. The algorithm includes the insertion of hunger element to the existing CSO to enhance the exploration capabilities and the diversity of cockroach population [7]. Meanwhile, Zhou et al. (2014) have proposed Cloud

Model Bat algorithm which is based on the ideas of bat echolocation together with the attribute of cloud model in order to depict good performance in optimization [8].

BIAs consist of several classes such as evolutionary algorithms (EA), swarm-intelligence-based (SI) algorithms, and many more. Among them, SI is the most prominent BIAs. SI algorithms imitate the social behavior of nature, such as bird flocking, fish schooling, and bees' swarming. SI has basically been a technique which is based on the interaction of organisms in a population, such as the flocks of bird and a swarm of bees. The optimization algorithms have been developed by observing the interaction among the swarm members [7]. Various optimization algorithms which are based on this technique have been successfully used in various optimization applications such as in real power loss minimization [4], estimation of induction motor's parameter [9], multi-level image thresholding [10], and many more. Among the techniques, optimization algorithms based on honeybees' behaviors have become the most commonly investigated and explored phenomenon by optimization researchers. Abbas

(2001) has investigated marriage in honeybees [11]. Later on, Karaboga (2005) has proposed the artificial bee colony (ABC) algorithm based on the foraging behavior of honeybees [12]. Next, the concept of honeybees mating has been studied by Marinakis et al. [13] and Niknam et al. [14] in 2011. Besides that, the idea of the waggle dances of honeybees has been investigated by Duangphakdee et al. in 2011 who found out that the honeybees have complexity in waggle dances as soon as the sun comes close to its zenith. Thus, they have studied the relation of foraging and absconding to the azimuth [15].

ABC was proposed by Karaboga in 2005 [12]. It mimics the intelligent foraging behavior of honeybees that shows how organized the honeybees interact among them to search for food. ABC has fewer tuned parameters compared to other optimization algorithms such as genetic algorithm (GA) and differential evolution (DE). Thus, it is a simple and efficient optimization algorithm [16]. Moreover, ABC has been proven to show superior performance in comparison to other prominent optimization algorithms such as genetic algorithm (GA), differential evolution (DE), evolutionary strategies (ES), and particle swarm optimization (PSO) algorithms [16–18]. Nevertheless, ABC has been found to suffer from few limitations such as slow convergence speed [19, 20] and premature convergence [21, 22]. Due to that, researchers have tried to solve them by developing various ABC variants, for example, Gbest-guided ABC (GABC) by Zhu and Kwong in 2010 [23], Best-so-far ABC (BsfABC) by Banharnsakun et al. [24], and Improved ABC (IABC) by Gao and Liu [25] in 2011 as well as modified ABC (MABC) by Gao and Liu [20], Global-best ABC (BABC) by Gao et al. [19], and enhanced ABC by Abro and Mohamad-Saleh [26] in 2012. However, some of these variants are still incapable of efficiently solving the problems, whilst a number of the variants could still be improved. For instance, the idea of IABC using the best solution is very convincing because it enhances the convergence speed [25]. Furthermore, its incorporation of random search equation into the algorithm is rather promising as the equation is known for its randomness and able to generate diverse population [25]. However, IABC is unable to solve Rosenbrock function as it is actually poor in exploitation [25]. Meanwhile, one of the BABC variants, BABC1, has also incorporated the idea of using the previous best solution as the guidance for the search [19]. With some adjustment to the solution search equation, BABC1 has shown the best performance among other variants at that time. Nevertheless, BABC1 is actually prone to premature convergence when dealing with complex multimodal problems [27]. With the motivation from one of the BABC variants which is BABC2, enhanced ABC (EABC) has been proposed with the idea to balance the exploration and exploitation abilities of the algorithm. Nonetheless, EABC has a tendency to suffer from slow convergence speed (i.e., lack of exploitation process) as shown in [28]. With the motivation from the existing ABC variants and their limitations, a new modified ABC is proposed in this paper. This new enhanced ABC is expected to give excellent performance in terms of convergence speed and robust global minimum search.

2. Artificial Bee Colony (ABC) Algorithm Model

The standard ABC algorithm is a population-based optimization algorithm. The working principle of ABC is as illustrated in Figure 1. Based on the figure, the working principle of ABC can be categorized into five main phases which are initialization, employed-bees, onlooker-bees, scout-bee, and termination phases which consist of a total of twelve stages or processes.

In ABC, three phases are performance-deciding phases which are employed-bees, onlooker-bees, and scout-bee phases while the other two are supporting phases. The exploration process of the algorithm takes place in employed-bees and onlooker-bees phases where the bees need to explore the neighborhood of the food sources allocated to them. Meanwhile, the exploitation process happens in the onlooker-bees phase when onlooker-bees apply fitness-proportion selection scheme in order to select the selected-fitter food sources. The details of the phases are discussed in the following subsections and more details of ABC can also be found in [18].

2.1. Initialization. In ABC algorithm, food sources represent the possible solution among the population of a problem. They are randomly initialized. The initialization of the population is based on user predetermined values of the population size. These food sources are then assigned to the employed-bees. Next, the nectar amounts which represent the fitness value of each food source are calculated using equation found in [18, 29, 30]:

$$\text{fit}_i = \begin{cases} \frac{1}{1 + f_i}, & f_i \geq 0, \\ 1 + \text{abs}(f_i), & f_i < 0, \end{cases} \quad (1)$$

where f_i is objective function value of i th food source.

2.2. Employed-Bees Phase. In this phase, employed-bees explore the neighborhood of the food sources assigned to them and update the food sources using the mutation equation given by

$$z_{ij} = y_{ij} + \phi_{ij} (y_{ij} - y_{kj}), \quad (2)$$

where z_{ij} is the candidate solution of food sources, y_{ij} is the j th dimension of the i th food sources, and y_{kj} is the k th food sources that are randomly chosen from a neighborhood of i th food sources for $k \in [1, 2, \dots, SN]$ and SN is the number of food sources. Subscripts k and i are mutually exclusive food sources. For the equation, k and j are chosen randomly and $j \in [1, 2, \dots, D]$ where D represents the dimension of the search space and ϕ_{ij} is the control parameter that represents random number from $[-1, 1]$, inclusively.

The explorations by employed-bees generate new food sources (i.e., candidate solutions of food sources). A selection between the candidate solution and the old food sources is based on which of them exhibits the best fitness value. This selection is done using greedy-selection scheme. The chosen food sources are potentially fitter food sources and are shared with onlooker-bees in onlooker-bees phase.

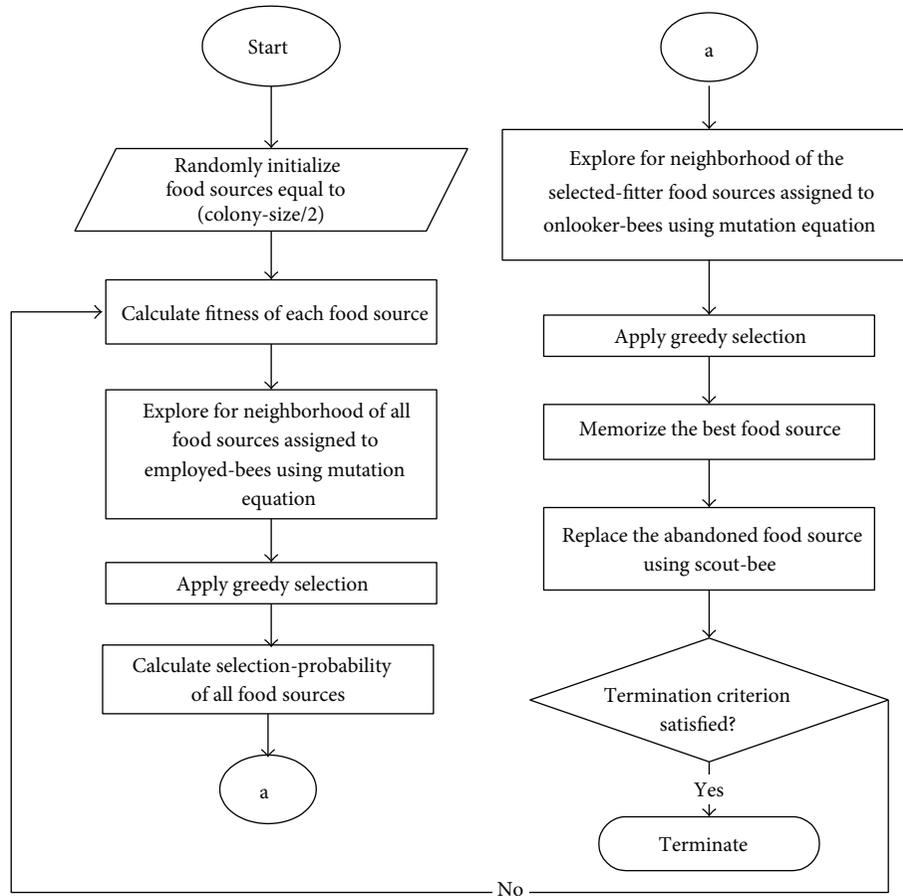


FIGURE 1: The flowchart of standard ABC algorithm.

2.3. *Onlooker-Bees Phase.* During this phase, the onlooker-bees do not update all potentially fitter food sources shared with them by employed-bees. They apply fitness-proportion selection scheme to choose few selected-fitter food sources among all the food sources shared with them. The exploitation of the food sources by onlooker-bees has actually made the algorithm converge fast. The fitness-proportion selection scheme is dependent on the probability value, P_i given by

$$P_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j}, \quad (3)$$

where P_i is the probability of i th food source, fit_i is the fitness value of i th food source, and SN represents the number of available food sources.

Onlooker-bees then explored the neighborhood of the selected-fitter food sources and update the food sources using the equation given in (2). The new candidate solution is then compared with the old food source using the greedy-selection scheme. Next, the best food source so far for that generation is memorized before entering the scout-bee phase.

2.4. *Scout-Bee Phase.* In scout-bee phase, a food source which has become exhausted and does not show improvement over a *limit* is abandoned [23]. *Limit* is a control parameter used

to signify exhausted food source [19]. Employed-bee whose food source has reached *limit* will become scout-bee. The scout-bee will take consequent flights and search the search space randomly to find new food source using

$$y_i^j = y_{min}^j + \text{rand}(0, 1) (y_{max}^j - y_{min}^j), \quad (4)$$

where y_{min}^j and y_{max}^j are the lower and upper limit of the search space, respectively. $\text{rand}(0, 1)$ is a function which randomly generates numbers within $[0, 1]$. This action is necessary for the scout-bee to replace the abandoned food source with new food source and thus balance the number of populations again.

2.5. *Termination.* The termination criterion of the algorithm is based on the maximum number of generations or maximum cycle number (MCN) [18]. This number is preset by user prior to the simulation of ABC algorithm.

3. New Enhanced ABC (JA-ABC5) Algorithm

The limitations of ABC are due to (2) that is known to be good in exploration but poor in exploitation. This imbalances of exploration and exploitation capabilities of the standard ABC algorithm contribute to its lack in performance. Thus,

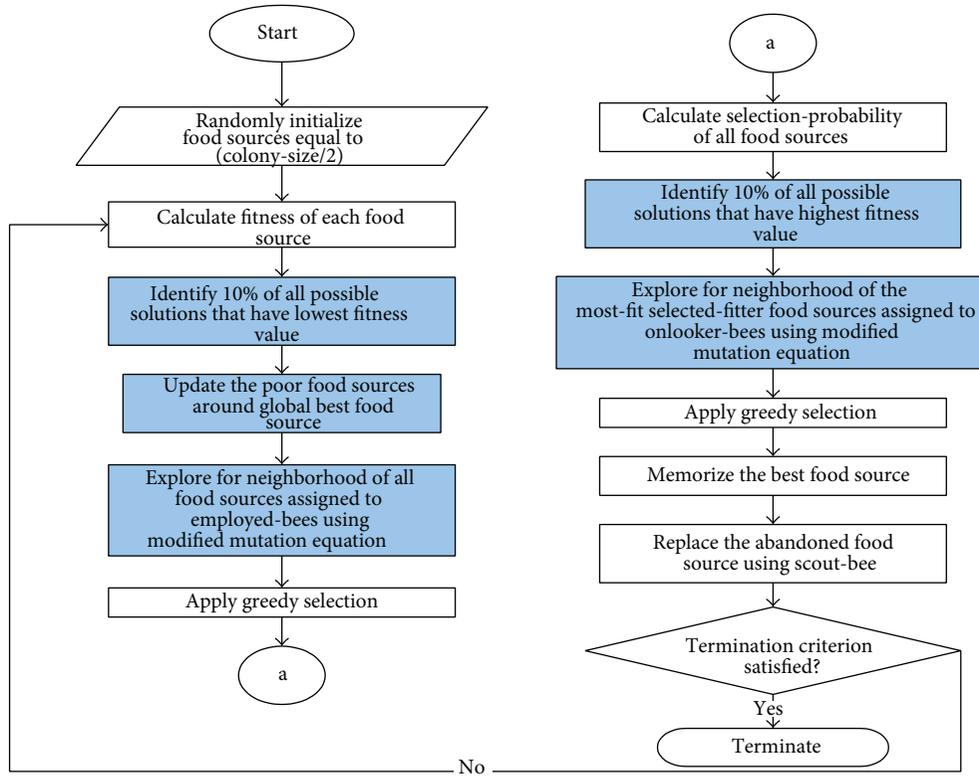


FIGURE 2: The flowchart of new enhanced ABC (JA-ABC5) algorithm.

few modifications have been introduced to the standard ABC algorithm for the purpose of balancing the exploration and exploitation capabilities of the algorithm. The proposed algorithm introduces four modifications to the standard ABC algorithm as highlighted in Figure 2.

The first modification is the insertion of new phase between initialization and employed-bees phases. This phase consists of two stages illustrated by stages 4 and 5 in Figure 2. The first stage aims to identify few food sources that have the lowest fitness values, referred to as poor food sources. Next, these poor food sources are updated around global best (g -best) food source using the mutation equation inspired from [19] given by

$$z_{ij} = y_{\text{best},j} + \phi_{ij} (y_{pj} - y_{kj}), \quad (5)$$

where z_{ij} represents the candidate solution of i th food source with j th dimension. $y_{\text{best},j}$ is the best food source, y_{pj} is j th dimension of p th food source and is randomly chosen. Subscripts i , k , and p are mutually exclusive food sources and the rest of the parameters are the same as in (2).

The generated food sources would now be fitter since they are being directed towards the global best food source based on (5). This has increased the exploitation process of the algorithm and makes the current population consist of fitter food sources. The random selection of food sources has also made the population not only fitter, but diverse as well.

Then, in employed-bees phases, the fitter populations are updated. Here comes the second modification which is represented by stage 6 in Figure 2. Since the population is now

fitter, there is a possibility for the algorithm to be trapped in local optima. Thus, to overcome this, the exploration process should be enhanced. The enhancement of the exploration process has been done by adapting new mutation equation in employed-bees phase. This new mutation equation is obtained by adapting modified mutation equation inspired from [25] which is well known for its randomness. The modification produces a modified equation given by

$$z_{ij} = y_{r1j} + \phi_{ij} (y_{r2j} - y_{r3j}), \quad (6)$$

where z_{ij} represents the candidate solution of i th food source with j th dimension. y_{r1j} , y_{r2j} , and y_{r3j} are the $r1$ th, $r2$ th, and $r3$ th food sources that are randomly chosen from neighborhood of i th food sources. Subscripts $r1$, $r2$, and $r3$ are mutually exclusive food sources and the rest of the parameters are the same as in (2). Equation (6) updates the food sources by directing the interaction among randomly chosen food sources. This increases the diversity of the exploration process that enhances the capability of the algorithm to avoid local optima trapping.

The next modification is aimed at increasing the convergence speed of the algorithm since random searching has a tendency to slow down the execution of the algorithm. The enhancement of the exploitation capability in onlooker-bees phase has been formulated to overcome this problem. The onlooker-bees have been directed to update only few most-fit-selected-fitter food sources. As already mentioned, onlooker-bees basically do not update all food sources but update only selected-fitter food sources. Hence, in this

proposed algorithm, onlooker-bees will update only few most-fit food sources among the selected-fitter food sources. Thus, with only few fitter food sources to be updated, the convergence speed of the algorithm has been increased. This modification is shown by stage 9 in Figure 2.

The fourth modification is to replace the mutation of onlooker-bees from (2) to the equation adapted from the work of [25]

$$z_{ij} = y_{best,j} + \phi_{ij} (y_{ij} - y_{mj}), \quad (7)$$

where z_{ij} represents the candidate solution of i th food source with j th dimension. $y_{best,j}$ is the best food source and y_{mj} represents j th dimension of m th food source and is randomly chosen. Subscripts i and m are mutually exclusive food sources and the rest of the parameters are the same as (2).

Equation (7) is able to enhance the convergence speed since the fitter food sources in onlooker-bees phase have been updated towards the g -best food sources. This modification is presented by stage 10 in Figure 2. Thus, in the end, the proposed algorithm, JA-ABC5, has enhanced and balanced exploration and exploitation processes. With this, it is expected to converge faster and to be able to reach global optimum efficiently. Its ability is assessed by comparing its performance with existing variants on 27 benchmark functions and at solving the reactive power optimization problem.

4. Simulations on Benchmark Functions

In order to justify the robustness of the proposed JA-ABC5 algorithm, it has been simulated on 27 commonly used benchmark functions as listed in Table 1. These benchmark functions vary from different types of functions such as random shifted, unimodal, multimodal, and rotated functions prior to testing the capabilities of the algorithm to solve a wide range of problems.

The performance of JA-ABC5 has been compared with the standard ABC algorithm and three other sophisticated existing ABC variants: Improved ABC (IABC) [25], Global best ABC (BABC1) [19], and enhanced ABC (EABC) [26, 29] to show the effectiveness of JA-ABC5 in solving those functions.

For all algorithms, the dimensionality of the benchmark functions has been set to 30, the population size has been set to 50, number of generations has been limited to 1000, and the parameter *limit* has been set as $D \times SN$, where D represents the dimension of the search space and SN is the number of food sources. The P value of IABC has been set to 0.25 [25]. As for global solution validation, each of the compared algorithms including JA-ABC5 has been set to be simulated for 30 times on each benchmark function [26]. All these values follow those used and recommended in the literature [18–20, 23, 25, 26, 30].

The simulation and testing process have been carried out using Matlab R2010a on an Intel Core i7 with 2.80 GHz speed computer.

4.1. Results of Benchmark Functions Simulation. Figures 3, 4, 5, 6, and 7 show the graphical results of the proposed

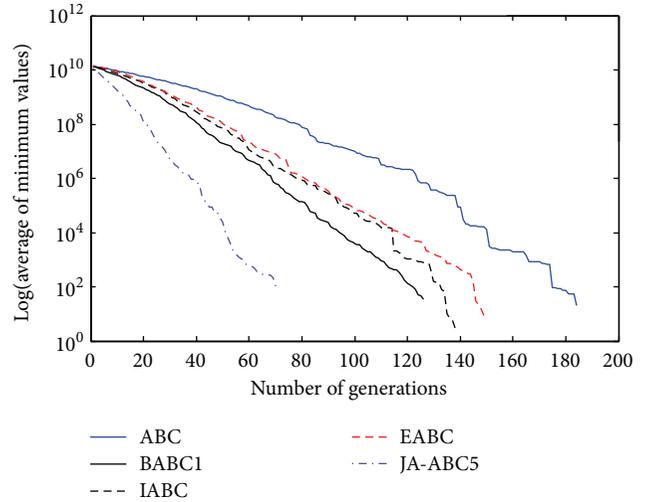


FIGURE 3: The convergence rates of optimization algorithms on Himmelblau function.

TABLE 1: Benchmark functions.

Function	Function name	Initialization range
f_1	Griewank	± 600
f_2	Rastrigin	± 15
f_3	Rosenbrock	± 15
f_4	RS Ackley	± 32
f_5	Schwefel	± 500
f_6	Himmelblau	± 600
f_7	RS Sphere	± 600
f_8	Step	± 600
f_9	Bohachevsky 2	± 100
f_{10}	RS Schwefel 2.22	± 100
f_{11}	RS Schwefel Ridges	± 100
f_{12}	RS Schwefel Ridges with Noise	± 15
f_{13}	RS Elliptic	± 100
f_{14}	Zekhelip	± 15
f_{15}	Non-continuous Rastrigin	± 15
f_{16}	Michalewicz	0–180
f_{17}	First Expanded Function	± 15
f_{18}	Second Expanded Function	± 15
f_{19}	Third Expanded Function	± 15
f_{20}	Fourth Expanded Function	± 500
f_{21}	Fifth Expanded Function	± 100
f_{22}	Sixth Expanded Function	± 100
f_{23}	Seventh Expanded Function	± 15
f_{24}	Eighth Expanded Function	± 100
f_{25}	Rotated Griewank Function	0–600
f_{26}	Rotated Ackley Function	± 32
f_{27}	Rotated Rastrigin Function	± 5

algorithm, JA-ABC5 algorithm. The figures have shown that the proposed algorithm has outperformed other algorithms in terms of convergence speed. It exhibits faster convergence

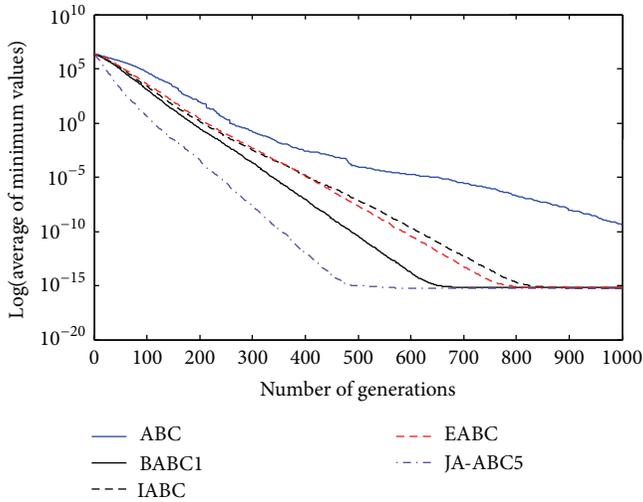


FIGURE 4: The convergence rates of optimization algorithms on Random Shifted Sphere function.

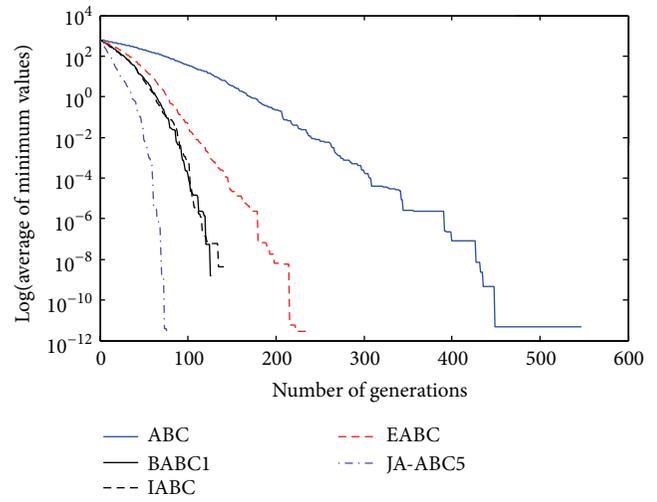


FIGURE 6: The convergence rates of the optimization algorithms on Rotated Griewank function.

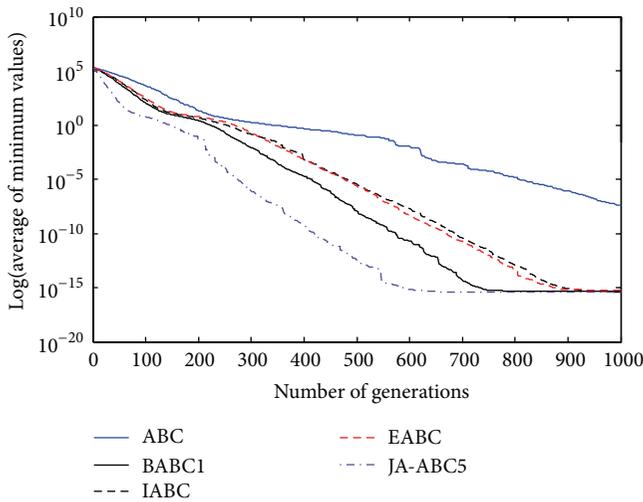


FIGURE 5: The convergence rates of optimization algorithms on Bohachevsky 2 function.

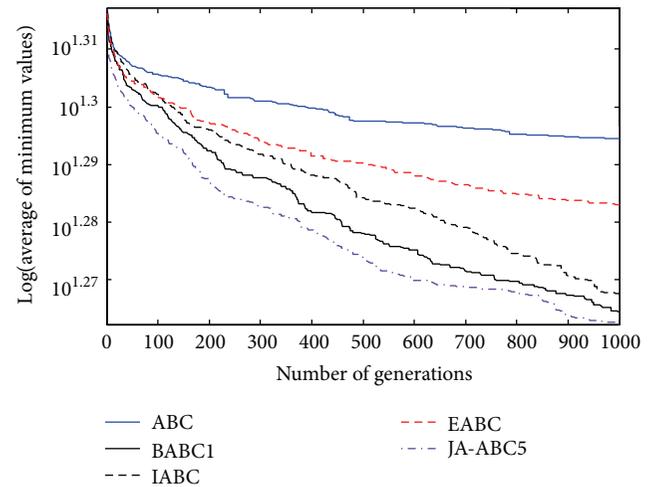


FIGURE 7: The convergence rates of the optimization algorithms on Rotated Ackley function.

as compared to others. Moreover, the considerable difference of the proposed algorithm in comparison with other compared variants has clearly justified that the proposed algorithm is a robust ABC variant that has potential to solve optimization problems. The standard ABC exhibits the worst performance among all since it has suffered from few limitations as mentioned earlier.

Meanwhile, the statistical data in Table 2 reveal the numerical performance results of various ABC variants illustrating the values of minimum, mean, and standard deviation of the compared optimization algorithms. The results have shown that JA-ABC5 exhibits the least value of minimum, mean, and standard deviation on most of the benchmark functions. Thus, this vividly demonstrates that JA-ABC5 has the best performance in comparison with other compared ABC variants.

5. Reactive Power Optimization Application

Reactive power optimization (RPO) is known to be a large-scale nonlinear combinatorial constrained problem [31]. RPO basically serves to determine the optimal setting of the power system network to satisfy few constraints such as the power flow equation system security and equipment operating limits [32]. This problem has been discovered by Carpentier in 1962 [33] and, since then, many have tried to solve it. Researchers and engineers have tried to solve it by developing various search strategies since this kind of problem is very essential to be solved. This is because this problem is the important tool in the power system's operation and planning [34] since it actually has close contact with the security and economic dispatch of a power system [35]. For example, they have attempted to solve RPO problem using various classical methods such

TABLE 2: Statistical Results of Optimization Algorithms.

<i>f1</i>	MIN	Average	STD DEV	<i>f15</i>	MIN	Average	STD DEV
ABC	1.22E - 14	1.47E - 13	1.65E - 13	ABC	3.56E - 07	9.44E - 01	8.38E - 01
BABC1	4.44E - 16	6.22E - 16	1.50E - 16	BABC1	0.00E + 00	2.00E - 01	4.84E - 01
IABC	4.44E - 16	6.40E - 16	4.82E - 16	IABC	0.00E + 00	7.01E - 14	6.95E - 14
EABC	4.44E - 16	6.31E - 16	1.58E - 16	EABC	0.00E + 00	6.72E - 12	2.10E - 11
JA-ABC5	3.33E - 16	5.48E - 16	8.71E - 17	JA-ABC5	0.00E + 00	1.00E - 01	3.05E - 01
<i>f2</i>	MIN	Average	STD DEV	<i>f16</i>	MIN	Average	STD DEV
ABC	2.29E - 08	3.77E - 01	5.87E - 01	ABC	-2.92E + 01	-2.90E + 01	1.40E - 01
BABC1	0.00E + 00	2.98E - 01	4.64E - 01	BABC1	-2.96E + 01	-2.94E + 01	1.08E - 01
IABC	0.00E + 00	0.00E + 00	0.00E + 00	IABC	-2.96E + 01	-2.96E + 01	1.58E - 02
EABC	0.00E + 00	8.02E - 14	3.02E - 13	EABC	-2.96E + 01	-2.95E + 01	9.68E - 02
JA-ABC5	0.00E + 00	2.32E - 01	5.01E - 01	JA-ABC5	-2.96E + 01	-2.90E + 01	1.29E - 02
<i>f3</i>	MIN	Average	STD DEV	<i>f17</i>	MIN	Average	STD DEV
ABC	2.23E - 02	9.03E + 00	2.11E + 00	ABC	7.85E - 03	4.09E - 02	1.81E - 01
BABC1	2.32E - 02	3.04E + 01	3.30E + 01	BABC1	7.85E - 03	5.69E - 01	5.62E - 01
IABC	9.66E - 02	5.63E + 00	5.85E + 00	IABC	7.85E - 03	4.08E - 02	1.81E - 01
EABC	1.48E - 02	4.42E + 00	7.16E + 00	EABC	7.85E - 03	7.85E - 03	1.75E - 16
JA-ABC5	1.31E - 02	4.97E + 00	1.44E + 01	JA-ABC5	3.03E - 03	6.55E - 03	1.37E - 01
<i>f4</i>	MIN	Average	STD DEV	<i>f18</i>	MIN	Average	STD DEV
ABC	2.48E - 06	1.82E - 05	1.12E - 05	ABC	1.59E - 14	3.29E - 13	2.53E - 13
BABC1	3.46E - 14	4.86E - 14	7.52E - 15	BABC1	4.42E - 16	6.29E - 16	1.05E - 16
IABC	8.98E - 12	1.97E - 11	6.75E - 12	IABC	4.44E - 16	5.72E - 16	8.91E - 17
EABC	1.66E - 12	4.15E - 12	1.56E - 12	EABC	4.92E - 16	6.62E - 16	1.23E - 16
JA-ABC5	2.75E - 14	3.12E - 14	3.16E - 15	JA-ABC5	4.19E - 16	5.32E - 16	7.03E - 17
<i>f5</i>	MIN	Average	STD DEV	<i>f19</i>	MIN	Average	STD DEV
ABC	1.07E - 01	2.96E + 02	1.20E + 02	ABC	2.80E + 01	3.13E + 01	6.05E + 00
BABC1	3.82E - 04	1.54E + 02	1.33E + 02	BABC1	2.78E + 01	6.29E + 01	3.13E + 01
IABC	3.82E - 04	6.71E + 01	8.62E + 01	IABC	2.80E + 01	4.20E + 01	1.87E + 01
EABC	3.82E - 04	9.28E + 01	1.06E + 02	EABC	2.76E + 01	4.76E + 01	2.68E + 01
JA-ABC5	3.82E - 04	1.03E + 02	1.23E + 02	JA-ABC5	2.74E + 01	4.18E + 01	1.30E + 01
<i>f6</i>	MIN	Average	STD DEV	<i>f20</i>	MIN	Average	STD DEV
ABC	-7.83E + 01	-7.83E + 01	1.55E - 07	ABC	3.20E - 08	6.65E - 02	3.64E - 01
BABC1	-7.83E + 01	-7.82E + 01	4.09E - 01	BABC1	0.00E + 00	5.98E - 01	9.30E - 01
IABC	-7.83E + 01	-7.83E + 01	1.21E - 14	IABC	0.00E + 00	0.00E + 00	0.00E + 00
EABC	-7.83E + 01	-7.83E + 01	2.28E - 01	EABC	0.00E + 00	1.67E - 16	5.18E - 16
JA-ABC5	-7.83E + 01	-7.83E + 01	2.39E - 01	JA-ABC5	0.00E + 00	1.33E - 01	5.06E - 01
<i>f7</i>	MIN	Average	STD DEV	<i>f21</i>	MIN	Average	STD DEV
ABC	4.60E - 11	4.53E - 10	3.99E - 10	ABC	1.66E - 10	2.30E - 09	3.28E - 09
BABC1	4.55E - 16	6.66E - 16	1.10E - 16	BABC1	1.11E - 16	3.89E - 16	1.66E - 16
IABC	4.09E - 16	5.89E - 16	9.03E - 16	IABC	5.55E - 17	3.11E - 16	1.85E - 16
EABC	4.72E - 16	6.57E - 16	1.24E - 16	EABC	1.11E - 16	4.86E - 16	1.32E - 16
JA-ABC5	3.29E - 16	5.62E - 16	1.08E - 16	JA-ABC5	5.55E - 17	3.05E - 16	1.21E - 16
<i>f8</i>	MIN	Average	STD DEV	<i>f22</i>	MIN	Average	STD DEV
ABC	4.27E - 11	4.82E - 10	5.12E - 10	ABC	1.80E - 10	2.89E - 09	3.60E - 09
BABC1	4.54E - 16	6.37E - 16	1.02E - 16	BABC1	0.00E + 00	3.94E - 16	1.45E - 16
IABC	2.87E - 16	5.38E - 16	9.62E - 17	IABC	0.00E + 00	2.55E - 16	1.30E - 16
EABC	4.69E - 16	6.65E - 16	8.62E - 17	EABC	2.22E - 16	4.37E - 16	1.33E - 16
JA-ABC5	4.21E - 16	4.66E - 16	7.66E - 17	JA-ABC5	0.00E + 00	2.46E - 16	1.28E - 16

TABLE 2: Continued.

<i>f</i> 9	MIN	Average	STD DEV	<i>f</i> 23	MIN	Average	STD DEV
ABC	9.72E - 10	3.85E - 08	4.32E - 08	ABC	7.05E - 12	1.60E - 10	1.70E - 10
BABC1	1.11E - 16	4.57E - 16	1.36E - 16	BABC1	4.32E - 16	6.20E - 16	1.07E - 16
IABC	1.11E - 16	4.09E - 16	1.42E - 16	IABC	3.23E - 16	5.57E - 16	8.87E - 17
EABC	2.22E - 16	5.41E - 16	1.76E - 16	EABC	4.80E - 16	6.78E - 16	9.50E - 17
JA-ABC5	1.11E - 16	3.81E - 16	1.27E - 16	JA-ABC5	3.22E - 16	5.57E - 16	8.72E - 17
<i>f</i> 10	MIN	Average	STD DEV	<i>f</i> 24	MIN	Average	STD DEV
ABC	1.96E - 06	1.07E - 05	5.44E - 06	ABC	7.50E - 01	7.50E - 01	2.20E - 04
BABC1	5.55E - 17	2.44E - 15	1.79E - 15	BABC1	7.50E - 01	7.50E - 01	3.23E - 16
IABC	6.93E - 12	2.64E - 11	9.63E - 12	IABC	7.50E - 01	7.50E - 01	3.44E - 16
EABC	2.95E - 12	8.20E - 12	4.41E - 12	EABC	7.50E - 01	7.50E - 01	1.78E - 16
JA-ABC5	0.00E + 00	1.85E - 18	1.01E - 17	JA-ABC5	7.50E - 01	7.50E - 01	1.36E - 16
<i>f</i> 11	MIN	Average	STD DEV	<i>f</i> 25	MIN	Average	STD DEV
ABC	1.85E - 10	2.62E - 09	2.40E - 09	ABC	0.00E + 00	0.00E + 00	0.00E + 00
BABC1	3.93E - 16	6.42E - 16	1.11E - 16	BABC1	0.00E + 00	0.00E + 00	0.00E + 00
IABC	3.95E - 16	5.77E - 16	9.15E - 17	IABC	0.00E + 00	0.00E + 00	0.00E + 00
EABC	4.39E - 16	6.54E - 16	1.04E - 16	EABC	0.00E + 00	0.00E + 00	0.00E + 00
JA-ABC5	3.31E - 16	5.76E - 16	1.06E - 17	JA-ABC5	0.00E + 00	0.00E + 00	0.00E + 00
<i>f</i> 12	MIN	Average	STD DEV	<i>f</i> 26	MIN	Average	STD DEV
ABC	3.33E - 03	8.34E - 03	3.92E - 03	ABC	1.79E + 01	1.97E + 01	4.65E - 01
BABC1	4.60E - 13	1.77E - 12	1.30E - 12	BABC1	1.70E + 01	1.82E + 01	6.89E - 01
IABC	5.65E - 09	1.82E - 08	1.15E - 08	IABC	1.66E + 01	1.85E + 01	7.64E - 01
EABC	2.50E - 10	8.02E - 10	5.34E - 10	EABC	1.80E + 01	1.92E + 01	5.92E - 01
JA-ABC5	5.24E - 16	8.08E - 16	1.71E - 16	JA-ABC5	1.60E + 01	1.73E + 01	3.86E - 01
<i>f</i> 13	MIN	Average	STD DEV	<i>f</i> 27	MIN	Average	STD DEV
ABC	1.63E - 10	7.50E - 09	1.60E - 08	ABC	2.98E + 02	3.25E + 02	1.36E + 01
BABC1	4.49E - 16	6.38E - 16	1.06E - 16	BABC1	2.12E + 02	2.88E + 02	2.01E + 01
IABC	4.21E - 16	5.41E - 16	9.08E - 17	IABC	2.59E + 02	2.99E + 02	1.97E + 01
EABC	4.64E - 16	6.34E - 16	1.06E - 16	EABC	2.50E + 02	2.93E + 02	2.32E + 01
JA-ABC5	2.67E - 16	5.30E - 16	8.66E - 17	JA-ABC5	2.08E + 02	2.85E + 02	1.26E + 01
<i>f</i> 14	MIN	Average	STD DEV				
ABC	3.32E - 08	4.03E - 04	1.31E - 03				
BABC1	4.08E - 16	7.40E - 03	2.82E - 02				
IABC	2.72E - 16	1.06E - 15	2.41E - 15				
EABC	3.03E - 16	1.43E - 11	4.70E - 11				
JA-ABC5	2.57E - 16	3.70E - 03	2.03E - 02				

as linear programming, Newton method, interior point, and many more. Nonetheless, the methods have shown some inefficiency in solving it [31]. Recently, researchers have tried to implement stochastic and heuristics techniques to solve this problem [31]. Thus, this has shown that RPO basically can be a perfect tool in order to validate the robustness of the proposed algorithm.

RPO problem is a combinatorial nonlinear constrained problem. The general mathematical formulation for that kind of problem is given by

$$\min (f(x)) \tag{8}$$

such that

$$\begin{aligned} g(x) &= 0, \\ h(x) &\leq 0, \end{aligned} \tag{9}$$

where $f(x)$ is the objective function to be minimized, $g(x)$ is the equality constraints, and $h(x)$ is the inequality constraints. Hence, the mathematical formulation of RPO problem with equality and inequality constraints is discussed in next subsections.

5.1. Objective Function-Active Power Loss. The objective function for RPO problem can be either the active power loss, total cost of compensation, total energy generation cost, and many

more [31]. In this paper, only active power loss is considered as the objective function to be solved by the proposed algorithm, JA-ABC5. The mathematical formulation of active power loss is given by

$$P_{\text{loss}} = \sum_{k=1}^{NL} P_{\text{loss},k} \tag{10}$$

$$= \sum_{i=1}^N \sum_{j=1}^N g_{ij} [V_i^2 + V_j^2 - 2V_i V_j \cos \theta_{ij}],$$

where P_{loss} is an active power loss, g_{ij} is the conductance between bus i and bus j , V_i is the voltage magnitude of bus i , V_j is the voltage magnitude of bus j , θ_{ij} is the angle difference of ij th transmission line, N is the total number of system's buses, and NL is the total number of transmission lines.

5.2. Equality Constraints. The equality constraints of the problem has been set to the power flow equations given by [36]

$$P_{Gi} - P_{Di} - \sum_{j=1}^N |V_i| |V_j| |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j) = 0, \tag{11}$$

$$Q_{Gi} - Q_{Di} + \sum_{j=1}^N |V_i| |V_j| |Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j) = 0,$$

where P_{Gi} is the active power generation at bus i , P_{Di} is the active power demand at bus i , Q_{Gi} is the reactive power generation at bus i , Q_{Di} is the reactive power demand at bus i , Y_{ij} is the admittance between bus i and bus j , δ_i and δ_j are the voltage angle at bus i and bus j , respectively, and the rest of the parameters are the same as in (10).

5.3. Inequality Constraints. The inequality constraints of the problem are the control variables that are to be optimized within their ranges. These control variables are the food sources or possible solutions that need to be optimized by JA-ABC5. The range of the possible solutions follows the following limits:

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max},$$

$$V_i^{\min} \leq V_i \leq V_i^{\max}, \tag{12}$$

$$Q_{Ci}^{\min} \leq Q_{Ci} \leq Q_{Ci}^{\max},$$

$$T_i^{\min} \leq T_i \leq T_i^{\max},$$

where P_{Gi} is the active power generation at bus i , V_i is the voltage magnitude at bus i , Q_{Ci} is the shunt compensation at bus i , and T_i is the transformer tap setting at bus i . Moreover, P_{Gi}^{\min} and P_{Gi}^{\max} are lower and upper limits of active power generation, V_i^{\min} and V_i^{\max} are lower and upper limits of voltage magnitude, Q_{Ci}^{\min} and Q_{Ci}^{\max} are lower and upper limits of shunt compensation, and T_i^{\min} and T_i^{\max} are lower and upper limits of tap setting.

5.4. Penalty Function. Penalty function is derived in order to convert constrained problem to unconstrained problem by adding penalty terms. Since RPO problem consists of several constraints as mentioned in the previous subsection, penalty terms have been added to (10) and the equation for the objective function of the problem now becomes

$$P(x) = P_{\text{loss}} + \Omega_P + \Omega_Q + \Omega_V + \Omega_G + \Omega_C + \Omega_T, \tag{13}$$

where $P(x)$ is the penalty function and Ω_P , Ω_Q , Ω_V , Ω_G , Ω_C , and Ω_T are the penalty terms of the listed equality and inequality constraints, respectively. Thus, the penalty terms are given by

$$\Omega_P = \rho \sum_{i=1}^N \left\{ P_{Gi} - P_{Di} - V_i \sum_{j=1}^N V_j (g_{ij} \sin \theta_{ij} + B_{ij} \cos \theta_{ij}) \right\}^2,$$

$$\Omega_Q = \rho \sum_{i=1}^N \left\{ Q_{Gi} + Q_{Ci} - Q_{Di} - V_i \sum_{j=1}^N V_j (g_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \right\}^2,$$

$$\Omega_V = \rho \sum_{i=1}^N \{ \max(0, V_i - V_i^{\max}) \}^2 + \rho \sum_{i=1}^N \{ \max(0, V_i^{\min} - V_i) \}^2,$$

$$\Omega_G = \rho \sum_{i=1}^{NG} \{ \max(0, P_{Gi} - P_{Gi}^{\max}) \}^2 + \rho \sum_{i=1}^{NG} \{ \max(0, P_{Gi}^{\min} - P_{Gi}) \}^2,$$

$$\Omega_C = \rho \sum_{i=1}^{NC} \{ \max(0, Q_{Ci} - Q_{Ci}^{\max}) \}^2 + \rho \sum_{i=1}^{NC} \{ \max(0, Q_{Ci}^{\min} - Q_{Ci}) \}^2,$$

$$\Omega_T = \rho \sum_{i=1}^{NT} \{ \max(0, T_i - T_i^{\max}) \}^2 + \rho \sum_{i=1}^{NT} \{ \max(0, T_i^{\min} - T_i) \}^2, \tag{14}$$

where P_{Gi} is the active power generation at bus i , P_{Di} is the active power demand at bus i , Q_{Gi} is the reactive power generation at bus i , Q_{Di} is the reactive power demand at bus i , Q_{Ci} is the shunt compensation at bus i , T_i is the transformer tap settings of transformer i , B_{ij} is the susceptance between bus i and bus j , NG is the total number of generators, NC is the total number of shunt compensator, NT is the total number

TABLE 3: Performance of optimization algorithms in solving RPO.

Algorithms	Ploss (MW)
SARGA	4.5740
PSO	4.6282
CLPSO	4.5615
EGA-DQLF	3.2008
ABC	1.5522
IABC	1.5185
BABC1	1.5215
EABC	1.5180
JA-ABC5	1.4985

of transformers, and the rest of the parameters are the same as in (10) [31, 36].

The proposed algorithm, JA-ABC5, is implemented to solve RPO problem by finding the optimal possible solutions to solve the objective function which is the penalty function obtained from (13). The possible solutions that need to be optimized which basically act as the food sources of JA-ABC5 are given by the previous subsection. They are active power generation, P_G , voltage magnitude, V , shunt compensation, Q_C , and transformer tap setting, T , at the required bus. JA-ABC5 is expected to produce less value of power loss which is affected by the above mentioned control variables' values. Thus, it is important to find the optimal values or settings of the control variables so that less amount of power loss has been generated.

5.5. Results of RPO. For the purpose of solving the RPO problem, IEEE 30-bus power system data has been obtained from [31]. To validate the performance of JA-ABC5 in solving the RPO problem, it has been compared with three existing ABC variants: IABC [25], BABC1 [19], and EABC [26, 29] as well as with other optimization algorithms available in the work of [31] which are self-adaptive real coded genetic algorithm (SARGA) [37], particle swarm optimization (PSO) [38], comprehensive learning PSO (CLPSO) [38], and enhanced genetic algorithm with decoupled quadratic load flow (EGA-DQLF) [39]. The performance of JA-ABC5 in solving the RPO problem in comparison with other optimization algorithms is tabulated in Table 3.

From Table 3, it is clear that variants of ABC algorithm have outperformed the other optimization algorithms. Most importantly, the results have shown that the proposed algorithm, JA-ABC5, has produced the minimum power loss of 1.4985 MW when compared to other optimization algorithms. Thus, this vividly shows that JA-ABC5 is able to solve complex optimization problem and hence can be applied to solve other optimization problems.

6. Conclusion

This work presents a new variant of the ABC algorithm referred to as JA-ABC5 by modifying the standard ABC algorithm to balance out the effects of exploration and exploitation processes into the performance of the algorithm.

The balanced exploration and exploitation capabilities are able to enhance the performance of the algorithm in terms of convergence speed and global optimum achievement. The performance results have clearly exhibited the best performance of JA-ABC5 in comparison to the compared ABC variants on 27 benchmark functions. Moreover, the efficiency of the algorithm in solving a complex real-world problem, the reactive power optimization (RPO), has vividly depicted that the algorithm is robust, effective, and reliable in solving optimization problems.

Conflict of Interests

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

Acknowledgment

The authors acknowledge the Ministry of Higher Education (MOHE), Malaysia, FRGS Grant no. 203/PELECT/6071247 for the financial support.

References

- [1] S. Binitha and S. S. Sathya, "A survey of bio inspired optimization algorithms," *International Journal of Soft Computing and Engineering*, vol. 2, no. 2, pp. 2231–2307, 2012.
- [2] Z. Cui, R. Alex, R. Akerkar, and X.-S. Yang, "Recent advances on bioinspired computation," *The Scientific World Journal*, vol. 2014, Article ID 934890, 3 pages, 2014.
- [3] T. Davidović, D. Ramljak, M. Šelmić, and D. Teodorović, "Bee colony optimization for the p-center problem," *Computers & Operations Research*, vol. 38, no. 10, pp. 1367–1376, 2011.
- [4] A. Q. H. Badar, B. S. Umre, and A. S. Junghare, "Reactive power control using dynamic Particle Swarm Optimization for real power loss minimization," *International Journal of Electrical Power & Energy Systems*, vol. 41, no. 1, pp. 133–136, 2012.
- [5] N. Karaboga and F. Latifoglu, "Adaptive filtering noisy transcranial Doppler signal by using artificial bee colony algorithm," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 2, pp. 677–684, 2013.
- [6] N. Bacanin and M. Tuba, "Firefly algorithm for cardinality constrained mean-variance portfolio optimization problem with entropy diversity constraint," *The Scientific World Journal*, vol. 2014, Article ID 721521, 16 pages, 2014.
- [7] I. C. Obagbuwa and A. O. Adewumi, "An improved cockroach swarm optimization," *The Scientific World Journal*, vol. 2014, Article ID 375358, 13 pages, 2014.
- [8] Y. Zhou, J. Xie, L. Li, and M. Ma, "Cloud model bat algorithm," *The Scientific World Journal*, vol. 2014, Article ID 237102, 11 pages, 2014.
- [9] A. G. Abro and J. Mohamad-Saleh, "Multiple-global-best guided artificial bee colony algorithm for induction motor parameter estimation," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 22, no. 3, pp. 620–636, 2014.
- [10] K. Charansiriphaisan, S. Chiewchanwattana, and K. Sunat, "A comparative study of improved artificial bee colony algorithms applied to multilevel image thresholding," *Mathematical Problems in Engineering*, vol. 2013, Article ID 927591, 17 pages, 2013.

- [11] H. A. Abbass, "MBO: marriage in honey bees optimization a haplometrosis polygynous swarming approach," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 207–214, Seoul, Korea, May 2001.
- [12] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, 2005.
- [13] Y. Marinakis, M. Marinaki, and G. Dounias, "Honey bees mating optimization algorithm for the Euclidean traveling salesman problem," *Information Sciences*, vol. 181, no. 20, pp. 4684–4698, 2011.
- [14] T. Niknam, H. D. Mojarrad, H. Z. Meymand, and B. B. Firouzi, "A new honey bee mating optimization algorithm for non-smooth economic dispatch," *Energy*, vol. 36, no. 2, pp. 896–908, 2011.
- [15] O. Duangphakdee, S. E. Radloff, C. W. W. Pirk, and H. R. Hepburn, "Waggle dances and azimuthal windows," *Psyche*, vol. 2011, Article ID 318985, 7 pages, 2011.
- [16] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.
- [17] M. El-Abd, "Performance assessment of foraging algorithms vs. evolutionary algorithms," *Information Sciences*, vol. 182, pp. 243–263, 2012.
- [18] D. Karaboga and B. Akay, "A comparative study of artificial Bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [19] W. Gao, S. Liu, and L. Huang, "A global best artificial bee colony algorithm for global optimization," *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2741–2753, 2012.
- [20] W.-F. Gao and S.-Y. Liu, "A modified artificial bee colony algorithm," *Computers and Operations Research*, vol. 39, no. 3, pp. 687–697, 2012.
- [21] A. G. Abro and J. Mohamad-Saleh, "Enhanced probability-selection artificial bee colony algorithm for economic load dispatch: a comprehensive analysis," *Engineering Optimization*, vol. 46, no. 10, pp. 1315–1330, 2014.
- [22] G. Li, P. Niu, and X. Xiao, "Development and investigation of efficient artificial bee colony algorithm for numerical function optimization," *Applied Soft Computing Journal*, vol. 12, no. 1, pp. 320–332, 2012.
- [23] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166–3173, 2010.
- [24] A. Banharnsakun, T. Achalakul, and B. Sirinaovakul, "The best-so-far selection in Artificial Bee Colony algorithm," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2888–2901, 2011.
- [25] W. Gao and S. Liu, "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters*, vol. 111, no. 17, pp. 871–882, 2011.
- [26] A. G. Abro and J. Mohamad-Saleh, "Enhanced global-best artificial bee colony optimization algorithm," in *Proceedings of the 6th UKSim-AMSS European Modelling Symposium (EMS '12)*, pp. 95–100, Valetta, Malta, November 2012.
- [27] A. G. Abro and J. Mohamad-Saleh, "An enhanced artificial bee colony optimization algorithm," in *Recent Advances in Systems Science and Mathematical Modelling*, WSEAS Press, 2012.
- [28] N. Sulaiman, J. Mohamad-Saleh, and A. G. Abro, "A modified artificial bee colony (JA-ABC) optimization algorithm," in *Proceedings of the International Conference on Applied Mathematics and Computational Methods in Engineering*, pp. 74–79, Rhodes Island, Greece, July 2013.
- [29] A. G. Abro, *Performance enhancement of artificial bee colony optimization algorithm [Ph.D. thesis]*, 2013.
- [30] 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.
- [31] K. Ayan and U. Kılıç, "Artificial bee colony algorithm solution for optimal reactive power flow," *Applied Soft Computing*, vol. 12, no. 5, pp. 1477–1482, 2012.
- [32] J. Zhu, *Optimization of Power System Operation*, John Wiley & Sons, New York, NY, USA, 2009.
- [33] J. Rahul, Y. Sharma, and D. Birla, "A new attempt to optimize optimal power flow based transmission losses using genetic algorithm," in *Proceedings of the 4th International Conference on Computational Intelligence and Communication Networks (CICN '12)*, pp. 566–570, Mathura, India, November 2012.
- [34] C. Sumpavakup, I. Srikun, and S. Chusanapiputt, "A solution to the optimal power flow using artificial bee colony algorithm," in *Proceedings of the International Conference on Power System Technology (POWERCON '10)*, pp. 1–5, Hangzhou, China, October 2010.
- [35] A. M. Abusorrah, "Optimal power flow using adaptive fuzzy logic controllers," *Mathematical Problems in Engineering*, vol. 2013, Article ID 975170, 7 pages, 2013.
- [36] U. Leeton, D. Uthitsunthorn, U. Kwannetr, N. Sinsuphun, and T. Kulworawanichpong, "Power loss minimization using optimal power flow based on particle swarm optimization," in *Proceedings of the 7th Annual International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON '10)*, pp. 440–444, Chiang Mai, Thailand, May 2010.
- [37] P. Subbaraj and P. N. Rajnarayanan, "Optimal reactive power dispatch using self-adaptive real coded genetic algorithm," *Electric Power Systems Research*, vol. 79, no. 2, pp. 374–381, 2009.
- [38] K. Mahadevan and P. S. Kannan, "Comprehensive learning particle swarm optimization for reactive power dispatch," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 641–652, 2010.
- [39] M. S. Kumari and S. Maheswarapu, "Enhanced genetic algorithm based computation technique for multi-objective optimal power flow solution," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 6, pp. 736–742, 2010.

Research Article

Kernel Method Based Human Model for Enhancing Interactive Evolutionary Optimization

Yan Pei, Qiangfu Zhao, and Yong Liu

The University of Aizu, Tsuruga, Ikki-machi, Aizuwakamatsu, Fukushima 965-8580, Japan

Correspondence should be addressed to Yan Pei; peiyan@u-aizu.ac.jp

Received 16 April 2014; Revised 2 September 2014; Accepted 20 October 2014

Academic Editor: Ahmad T. Azar

Copyright © 2015 Yan Pei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A fitness landscape presents the relationship between individual and its reproductive success in evolutionary computation (EC). However, discrete and approximate landscape in an original search space may not support enough and accurate information for EC search, especially in interactive EC (IEC). The fitness landscape of human subjective evaluation in IEC is very difficult and impossible to model, even with a hypothesis of what its definition might be. In this paper, we propose a method to establish a human model in projected high dimensional search space by kernel classification for enhancing IEC search. Because bivalent logic is a simplest perceptual paradigm, the human model is established by considering this paradigm principle. In feature space, we design a linear classifier as a human model to obtain user preference knowledge, which cannot be supported linearly in original discrete search space. The human model is established by this method for predicting potential perceptual knowledge of human. With the human model, we design an evolution control method to enhance IEC search. From experimental evaluation results with a pseudo-IEC user, our proposed model and method can enhance IEC search significantly.

1. Introduction

Interactive evolutionary computation (IEC) is an optimization method that can incorporate human knowledge into an optimization process. It converges to a solution accordingly with certain human preference. From a framework viewpoint, IEC can be implemented with any evolutionary computation (EC) algorithm by replacing the fitness function with a human user. General category of IEC methods includes interactive genetic algorithm (IGA) [1], interactive genetic programming [2], interactive evolution strategy [3], and human-based genetic algorithm [4]. There are many challenges in IEC researches and its applications. Reference [5] presented a review of research on IEC challenges. These research areas include discrete fitness value input method, prediction of fitness values, user interface for dynamic tasks, acceleration of IEC convergence, combination of IEC and non-IEC, active intervention, and IEC theoretical research. Utilization of IEC allows fusing human and computer for problem solving. However, taking the evaluation process into the hands of an user sets up a different scenario compared to normal optimization methods, and it leads to serious problems when

putting IEC into practice. One of the problems is user fatigue in an evaluation process of the IEC.

It is necessary to relieve user fatigue for many IEC applications to improve performance of target systems. References [6, 7] presented to use semisupervised learning technique in IGA to enhance IEC search. Reference [8] embedded decision-maker's preferences in IEC in multiobjective optimization problem. Another solution to solve this problem is to accelerate IEC search by using fitness landscape directly [9]. Fourier transform is applied to obtain frequency information to analyze fitness landscape [10, 11]. A landscape approximation method with simpler shape was proposed, but the computational cost of approximation in an original high dimensional search space is costly [12]. Reference [13] presented an approximation method of projecting an original fitness landscape into each lower dimension. From comparison evaluation results, it can save computational cost significantly [14]. Dimensionality reduction method can obtain a fitness landscape in lower dimensional space to support useful information for search. This method has been applied to the travelling salesman problem in a real world application [15, 16]. On the other hand, if we project an original search space

into a higher dimensional space by kernel method, we can also obtain useful information for finding optimum region. For an IEC application, kernel method is a tool to establish a human model.

When a human conducts an IEC experiment, fitness landscape of IEC is an approximate model of human evaluation landscape. It is different to establish an exact mathematical model to express IEC fitness landscape, that is, IEC user model, which is usually nonlinear, discrete, constraint, multimodal, noisy, and high dimension. The great difference between an IEC user model (the terms, “user model” and “human model,” have the same meaning in this paper. However, user model refers to a concept of individuality in physical level usually, and human model refers to a concept of abstraction in logic layer) and an ordinary fitness functions is in the implementations of (a) relative and (b) discrete fitness evaluations that are produced by a human user. Unlike an ordinary fitness function, a human IEC user compares given objectives in relative terms and never produces an absolute fitness value. He or she also cannot give precise fitness values, but rather can only rank according to discrete levels (e.g., 1 to 5 or 1 to 7 levels) every generation, while ordinary fitness functions give continuous values. When a difference between individuals is less than a minimum discrete fitness range, that is, an evaluation threshold, a human IEC user cannot distinguish the difference. Such difference becomes fitness noise that IEC user model should implement. Kernel method is a powerful tool that can project an IEC search space from its original discrete search space into a new higher dimensional space (*feature space*) by conducting a non-linear transformation with suitable kernel function. After then, we can use a linear model as a human model in the feature space to analyze human perceptual knowledge easily. The linear model in feature space corresponds to an original complex nonlinear model in an original IEC search space.

This paper proposes a method to obtain and analyze IEC human model in high dimensional search space by kernel classification method, which is beneficial to a discrete search space problem, such as IEC. First, we separate some individuals into two groups as training sample data. One group is near optimum with related better fitness, and the other group is beyond optimum with a worse fitness. Second, we project these individuals into high dimensional feature space by some kernel functions. Utilization of different kernel function is to map an original search space into different topological feature space. Third, in feature space, we establish a human model by a linear classifier to support correct classified fitness landscape that linear classifier in original search space cannot support. It is a novel method to establish a human model in IEC research and can be extended into IEC application in many perspective interdisciplinary research. The method for establishing a human model presents an originality of this paper. With the obtained human model in a high dimensional search space, we propose an evolution control method to enhance IEC search and use four Gaussian mixture models as pseudo-IEC user to evaluate our proposed methods. From experimental evaluation results, our proposed evolution control methods can accelerate IEC search significantly.

The remainder of this paper is structured as follows. Section 2 presents an overview of human model in computation. Section 3 presents an overview of kernel method and introduces our linear classifier design method by kernel classification in detail. Some kernel functions used in our study are described, in which they are used to project an original search space to different feature space. Section 4 proposes an evolution control method by using human model in feature space. Fitness landscape in feature space is studied and discussed. Evaluations are conducted in Section 5 by using four Gaussian mixture models as pseudo-IEC user to evaluate our proposed methods. Some discussions of our proposed method and evaluation results and several open topics are presented in Section 6. Section 7 concludes the whole paper while some future work is presented.

2. Human Model in Computation

2.1. Human Related Computation. Some computation mechanisms need human assistance to complete a certain task, since the human’s capability and preference were introduced into computational process. The key words that relate to these researches and applications are games, interactive optimization, and human computation interaction, and so forth. On the one hand, human has intelligence, such as non-linear thing, productivity, innovation, hypothesis, that the computer or computation cannot simulate and compute. Human can therefore compensate these drawbacks of computer or computation. On the other hand, the computer has powerful and huge computation capability. The computer can help a certain user to complete their works in computational way of releasing their workloads and fatigues. The crucial issue is type and way in designing these human and computation cooperations.

The prototype and mechanism related to human and computation can be categorized into three perspectives. Firstly, it uses human’s intelligence, computational capabilities, and advantages of computer to compensate each sides’ limitations. Human and computer work together for a certain task. It is the subjective of human computer cooperation. Secondly, it obtains human’s potential or unknown knowledge (e.g., psychological, physiological, or intelligent knowledge) from a computational process. It belongs to the topics of humanized information extraction from interaction between human and computer. Thirdly, it enhances human cognitive competence by a computational process. It is intelligence amplification [17].

Human model is an essential research subject in researches of computational process related to human, such as human computation, awareness computing, and IEC. In human computation technique, two prominent human-computation techniques, games with a purpose and microtask crowd sourcing, can help resolve semantic technology related tasks, including knowledge representation, ontology alignment, and semantic annotation [18]. Human model can improve these computations and be applied to further commercial outlook. In awareness computing, human model can be introduced into its computational process to analyze

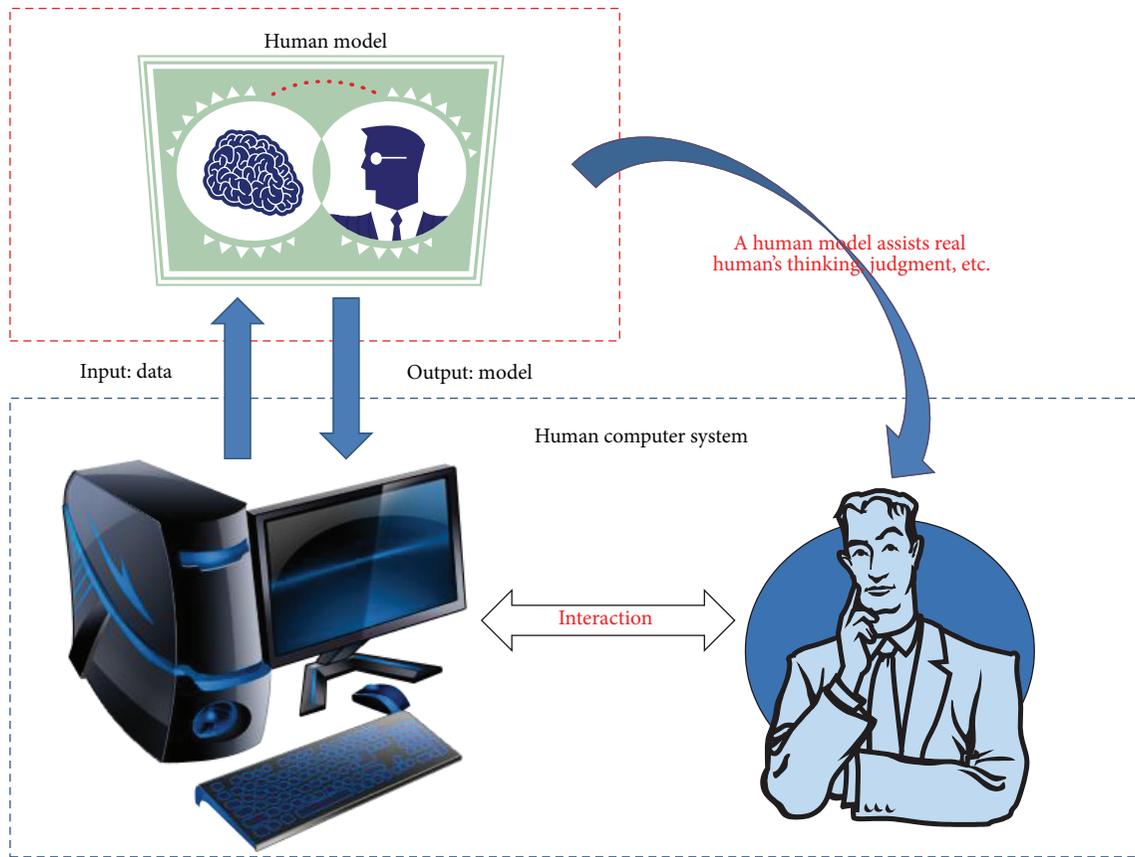


FIGURE 1: Conceptual diagram of a human model, which can assist a real human in a human computer system. From the interaction between human and computer, we can obtain the relationship of human's preference, thinking, knowledge, and so forth, and its related elements in computer system. And then, we use these data to establish a human model in the computer to assist human to solve a certain task or problem.

human awareness mechanism [19]. In IEC, human model can be used either to analyze human cognitive knowledge in human side or to enhance IEC search in the computer side for releasing user fatigue. This is the subject of this paper as well.

Figure 1 demonstrates a conceptual diagram of human model that can assist and extend the human's capability in a human computer system. There are two components in this diagram: one is a human model and the other is a human computer system. The human computer system supports learning data to a human model system, which can establish a human model to help a real human to complete a certain work in an environment of interaction between human and computer. The implementations of human model present its research philosophy.

2.2. Human Model. Model and simulation are important for theoretical study in human related computation. Any success of practical applications comes from fundamental research with the necessary assistance of model and simulation. When a system relates to a real human, it is essential to establish a human model to simulate characteristics of a human for research. Generally, the Turing machine can be considered as the first human model in the history of computer science

[20]. There are three aspects in a human model, that is, perceptual model, cognitive model, and physical model [21]. They correspond to concepts of sensation and perception, consciousness, and behavior in psychological research.

We should clearly define psychological and physiological characteristics of each layer to establish a human model, which is as well as a research issue in ergonomics research. In every aspect of each layer, there are different research scales for a human model. In perceptual model, it can be separated into vision model, auditory model, olfactory model, and gustatory model. In cognitive model, it can be separated into space model, time model, motion model, emotion model, and so forth. A study on human emotion and cognition recognition was conducted by soft computing techniques [22]. In physical model, it can be separated by functions or organs, and so forth. A human organ model was established to better understand human physiological activities and disasters themselves [23].

Human model is a synthesis model of perceptual and cognitive model mentioned above, which is established for a certain computational tasks. It has a variety of implementations, whatever the purposes and the forms come from. Not only human but also animal has computational capability in the world [24, 25]. The study scale of human model can be extended into the life world as a life model to recognize

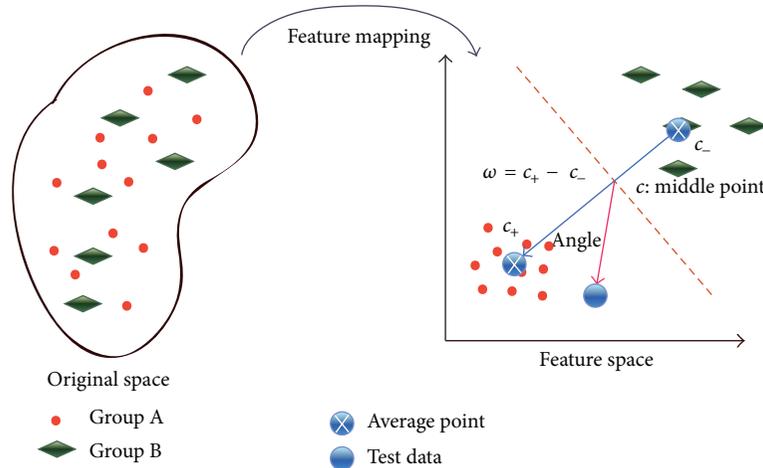


FIGURE 2: Human model by a binary classifier in our experiments. These two categories cannot be separated linearly in original space; however, after projecting these data into a higher dimensional feature space, they can be separated by a binary classifier. The linearity characteristic shows the advantage of proposed human model.

and understand the behaviour of the natural world. Some of primary discussable issues in natural computing are related to this topic [26]. The prior issue of establishing a human model is to distinguish differences between pure computing and human thinking [27].

3. Kernel Methods

3.1. *Kernel Trick.* Kernel methods present a series of data transformation techniques in machine learning that projects original space data into another higher dimensional space, that is, *feature space*, in which we can establish a linear model to reduce complexity of data relation. Typically, kernel methods are applied in classification and regression problems [28].

In general, linearity is a special characteristic, and no model of a real system is actually linear. However, linear relations have been focused in many research areas. If a model is nonlinear, we can project it into a feature space for obtaining linear relation, but not trying to fit a nonlinear model in an original space [12]. This kind of techniques are known as kernel trick.

The kernel trick was originally proposed in [29]. Mercer’s theorem is its mathematical result, which presents that any continuous, symmetric, positive semidefinite kernel function $K(x, y)$ can be expressed as an inner product $\langle x, y \rangle$ form in a high dimensional space [30]. Suppose that there are sample data (1) in a measurable space P , the kernel is positive semidefinite (2). There must be a function $\varphi(x)$, that is, *feature map*, whose range is in an inner product space Y of high dimension, shown in (3). This transformation process can be expressed in (4):

$$\text{SampleData} = x_0, x_1, \dots, x_n \in P, \tag{1}$$

$$\sum_{i,j} K(x, y) c_i c_j \geq 0, \tag{2}$$

$$K(x, y) = \langle x \cdot y \rangle, \tag{3}$$

$$P \rightarrow \varphi(x) \rightarrow Y. \tag{4}$$

There are several advantages of kernel methods. First, the kernel methods define a similarity measurement among sample data and present original space complex information in a simple form in feature space. Second, its computational complexity depends on the kernel function only and does not utilize feature map and feature space explicitly. Third, the kernel methods use training data in the form of kernel function and kernel matrix rather than the training data themselves, because there is no need to conduct a feature map explicitly in a high dimensional feature space.

3.2. *Kernel Classification.* Mercer theorem presents that kernel function corresponds to some feature space, and its mathematical result was presented in [30]. Since it is proposed, kernel methods were used in wide research areas, which include classification [31], principal component analysis [32], pattern analysis [33], support vector machine [34], and so forth.

Kernel classification processes the data that is difficultly distinguished in an original space. It projects them into higher dimensional space by kernel function to design proper linear classifier for solving classification problems. In our proposal, we consider human model as a simple binary classification problem as in Figure 2. The human model is applied to IEC for modelling characteristics of an IEC human user and enhancing IEC optimization.

In Figure 2, two category data properties are hard to be separated into two groups by a linear model in an original space. By a feature map, the data in original space is transferred into a high dimensional feature space, where they can be separated into two groups by a linear model easily. The

binary classification problem in the original space and feature space is described by (5) and (6), respectively,

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \in R^d \times \{+1, -1\}, \quad (5)$$

$$\{(\varphi(x_1), y_1), (\varphi(x_2), y_2), \dots, (\varphi(x_n), y_n)\} \in H \times \{+1, -1\}. \quad (6)$$

In feature space, we can obtain a vector ω (9), which is from one group vector's center vector (7) to the other (8), and its middle point is shown in (10). When new unknown data comes, we can judge its category through the angle of ω and the vector from ω 's middle point to the unknown point. The concrete judgement process is shown in (11), where sgn is a sign function. If the result is positive, the unknown data belongs to positive group; conversely, it belongs to the other group (negative group),

$$c_+ = \frac{1}{m_+} \sum_{i=1}^{m_+} \varphi(x_i^+), \quad (7)$$

$$c_- = \frac{1}{m_-} \sum_{j=1}^{m_-} \varphi(x_j^-), \quad (8)$$

$$\omega = c_+ - c_-, \quad (9)$$

$$c = \frac{1}{2} (\omega) = \frac{1}{2} (c_+ - c_-), \quad (10)$$

$$y = \text{sgn} (\langle \varphi(x) - c, \omega \rangle), \quad (11)$$

$$= A - B - \frac{1}{2} (C - D). \quad (12)$$

In feature space, we can judge the unknown data's category through (11). However, there is not an explicit feature map φ in kernel classification method, we must establish the feature map φ with some kernel function form. Equation (13) shows the concrete algorithm of (11) with the forms of kernel function,

$$\begin{aligned} A &= \frac{1}{m_+} \sum_{i=1}^{m_+} K(x_i^+, x), \\ B &= \frac{1}{m_-} \sum_{j=1}^{m_-} K(x_j^-, x), \\ C &= \frac{1}{m_+^2} \sum_{i=1}^{m_+} \sum_{i=1}^{m_+} K(x_i^+, x_i^+), \\ D &= \frac{1}{m_-^2} \sum_{j=1}^{m_-} \sum_{j=1}^{m_-} K(x_j^-, x_j^-). \end{aligned} \quad (13)$$

3.3. Kernel Functions. The selection of kernel function is a crucial issue for the success of all kernel algorithms, because the kernel function constitutes prior knowledge that is available about a task. Accordingly, there is no free lunch in kernel function selection. In our proposed human models and evolution control methods, we use three well known kernel

functions with different parameters in our experimental evaluation. They are linear kernel, polynomial kernel and Gaussian kernel (radial basis function, i.e., RBF kernel). Equations (14), (15), and (16) show their concrete forms:

$$K(x, z) = \langle x, z \rangle, \quad (14)$$

$$K(x, z) = (\langle x, z \rangle + 1)^r, \quad (15)$$

$$K(x, z) = \exp \frac{-|x - z|^2}{2\sigma^2}, \quad \sigma \in R - \{0\}. \quad (16)$$

4. Kernel Classification Based Human Model

4.1. Concept of the Proposal. In an original search space, we cannot basically utilize a linear classifier to judge an individual's property by fitness, which is preference of a certain user in IEC. That means it is impossible to establish a linear human model in original search space. Reference [9] reported a dynamic fitness threshold technique to ensure fitness increasing from one generation to the next. However, there is possibility to lead to local optima due to the fact that classifier model is linear in an original search space. Reference [35] proposed a constructive mapping genetic algorithm (CMGA) to implement this mechanism.

In Figure 2, suppose that circles show the better fitness area and rhombuses are the worse fitness area. If we use a dynamic fitness threshold technique (such as CMGA), which is linear in original search space, to filter new offspring, many individuals with better fitness will be drawn up so that algorithm performance will become worse. However, in feature space, this dynamic fitness threshold technique can be implemented by a linear classifier thanks to projecting them into a higher dimensional search space by kernel method. All individuals can be separated clearly and exactly in feature space, and this is beneficial to obtain a fitness landscape in high dimensional feature space. For IEC, it is a human model that is implemented by a linear classifier in feature space.

4.2. Evolution Control Method by a Human Model. Accordance with the primary motivation and kernel classification method, we design an evolution control method by establishing a human model in feature space to enhance IEC search for relieving user's fatigue. The proposed algorithm is shown in Algorithms 1 and 2. Algorithm 1 is a framework of kernel method based GA, and Algorithm 2 is one of its implementations by using a human model.

4.2.1. Training Data Selection. It is crucial to select labelled training data in an original search space to distinguish the individuals' category (i.e., human preference). We choose n and m different individuals with the better fitness and the worse fitness as training data for kernel function, which shows ones are near the optimum and the others are beyond the optimum. The parameters $n = 3$ and $m = 3$ are in experimental evaluation. The performance of selection method depends on kernel function design corresponding to a certain fitness landscape in feature space or a certain human user preference.

```

(1)  $G = 0$ 
(2) Initialize  $P(G)$ 
(3) Computing Fitness( $P(G)$ )
(4) while Non-Termination do
(5)   Selecting Training Data
(6)   Training Linear Classifier
(7)   Selection( $P(G)$ )
(8)   Kernel Based Crossover( $P(G)$ )
(9)   Kernel Based Mutation( $P(G)$ )
(10)   $G = G + 1$ 
(11)  Computing Fitness( $P(G)$ )
(12) end while

```

ALGORITHM 1: Kernel based genetic algorithm (G : generation and $P(G)$: population of the G th generation).

```

(1) ( $Pa$ ) = RandomChoose( $P(G)$ )
(2) ( $OffS$ )=Operator( $Pa$ )
(3) if  $LC(OffS)$  near  $GOptima$  then
(4)   Put  $OffS$  into  $P(G + 1)$ 
(5) end if

```

ALGORITHM 2: Kernel based operators (G : generation, $P(G)$: population, Pa : parent, $OffS$: offspring, $F(x)$: fitness function, $LC(x)$: linear classifier, $GOptimum$: global optimum, operator: crossover or mutation).

4.2.2. Human Model Implementation by a Linear Classifier Based on Kernel Method. After we obtain the labelled training data as input data for kernel function, we project these labelled training data (individuals and their fitness) into high dimensional feature space by a kernel function. In our experimental evaluation, we use three kernel functions, which are shown in (14), (15), and (16). Parameter setting is that $r = 2, 3, 5, 7$ is in polynomial kernel, and $\sigma = 1, 3, 5, 7$ is in RBF kernel for comparing their optimization performance. The training and implementation of linear classifier are from one generation to the next. It is an online training and utilization method, which can adapt IEC user preference from one generation to the next.

The utilization of different kernel function is conducting an operation to map individuals into a different dimensional and structural feature space, so the performance of linear classifier (human model) design is decided by the selection of kernel function and its parameter setting. In Algorithm 1, Step (6) shows a linear classifier training in every generation. For a certain fitness landscape in an original search space, there must be an optimal linear classifier (human model) design with a certain kernel function and its parameter setting. It is a promising study topic to obtain this optimal human model for IEC search. We will conduct this research topic in the future.

4.2.3. Human Model Utilization. In a high dimensional feature space, we establish a human model by a linear classifier

to support preference of a human user that linear classifier in an original search space cannot support. When IEC obtains a new offspring, that is, an object for evaluation, we use the designed human model to classify its category (near or beyond global optimum, i.e., human subjective preference) and then to judge whether to put it into the next generation (Algorithm 2). Because this processing is conducted by computer automatically, it does not increase human evaluation workload, but can enhance IEC search significantly. The training and utilizing human model can be applied every generation or several generation once. In our experimental evaluation, we evaluate our proposal with the first method, that is, training and utilizing human model every generation.

5. Experimental Evaluation

5.1. Simulation Experimental Design. User fatigue is a considerable factor in the IEC optimization evaluation. Experimental evaluations frequently require many repeated experiments under the same conditions, and in this case it is necessary to perform the evaluations using an IEC user model rather than with a real human IEC user. We need to evaluate acceleration methods by analyzing the load of a single evaluation along with the convergence characteristics through IEC simulation. After that we must conduct a human subjective evaluation to evaluate user fatigue and acceleration performance synthetically and thus conclude our evaluation of methods proposed here. This paper deals with IEC simulation of the first stage.

Gaussian mixture model is modelled as a pseudo IEC user in our simulation in Section 5.2. We conducted simulation evaluations to compare the characteristics of several methods with multiple different initializations under the same experimental conditions. A constructive mapping genetic algorithm (CMGA) is introduced as a comparison algorithm in our experiment in Section 5.3. We explain our proposed algorithms, their parameter setting, and evaluation metrics, such as several statistical tests in Section 5.4. Some evaluation results and observations are initially summarized in Section 5.5,

$$\text{GMM}(x) = \sum_{i=1}^k a_i \exp\left(-\sum_{j=1}^n \frac{(x_{ij} - \mu_{ij})^2}{2\sigma_{ij}^2}\right), \quad (17)$$

$$\sigma = \begin{pmatrix} 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix}, \quad (18)$$

$$\mu = \begin{pmatrix} -1 & 1.5 & -2 & 2.5 & -1 & 1.5 & -2 & 2.5 & -1 & 1.5 \\ 0 & -2 & 3 & 1 & 0 & -2 & 3 & 1 & 0 & -2 \\ -2.5 & -2 & 1.5 & 3.5 & -2.5 & -2 & 1.5 & 3.5 & -2.5 & -2 \\ -2 & 1 & -1 & 3 & -2 & 1 & -1 & 3 & -2 & 1 \end{pmatrix}, \quad (19)$$

$$a_i = (3.1, 3.4, 4.1, 3)^T. \quad (20)$$

TABLE 1: Mean and standard variance of final results. The number in blanket is standard variance, and the bold font shows better result among the proposed methods.

F	N	Linear	Poly2	Poly5	Poly7	Poly10	RBF1	RBF5	RBF7	RBF10
3-D	4.09 (0.87)	4.91 (0.62)	4.81 (0.68)	4.67 (1.10)	4.57 (0.79)	4.66 (0.91)	4.74 (0.73)	4.65 (0.76)	4.81 (0.76)	4.91 (0.59)
5-D	1.68 (0.38)	2.29 (0.25)	2.28 (0.32)	2.27 (0.25)	2.26 (0.32)	2.26 (0.22)	2.26 (0.31)	2.16 (0.39)	2.21 (0.35)	2.27 (0.26)
7-D	0.63 (0.20)	1.00 (0.28)	0.96 (0.30)	1.16 (0.38)	1.06 (0.30)	1.09 (0.31)	0.99 (0.30)	0.88 (0.18)	1.10 (0.39)	1.08 (0.30)
10-D	0.06 (0.01)	0.27 (0.07)	0.35 (0.10)	0.35 (0.10)	0.38 (0.14)	0.24 (0.04)	0.28 (0.18)	0.25 (0.06)	0.29 (0.05)	0.25 (0.04)

TABLE 2: GA parameters setting.

Parameter	Value or setting
Coding	Binary number
Number of generation	20
Population size	20
Selection	Roulette wheel and elite Dynamic fitness threshold
Crossover	One-point
Crossover rate	80%
Mutation rate	10%

5.2. *Gaussian Mixture Model as a Pseudo-IEC User.* Reference [36] discussed some limits of human brain with respect to information processing. In particular, this research had found that people are unable to keep up with more than 5–9 different chunks of information at one time. Gaussian mixture model (GMM) with less dimensional setting can well simulate it and some features of evaluation when a human conducts an IEC experiment, that is, relative and discrete fitness evaluations. GMM consists of different mean, variance, and peak together to express the characteristics when a human user conducts IEC evaluation experiments [37].

We use GMM as a pseudo-IEC user to evaluate our proposed methods. We choose four Gaussian mixture models as the basis function of mixture model. For each single model, we set $k = 4$ and dimension as 3, 5, 7, and 10. The GMM is shown in (17), and parameters σ (18), μ (19), and a_i (20) are set as follows.

5.3. *Comparison Method: CMGA.* In the experiments, we use genetic algorithm (GA) as an optimization method to evaluate the proposed methods. We compare our proposed acceleration methods with CMGA. Algorithm 3 shows the primary process of CMGA. The difference between canonical genetic algorithm and CMGA is the judgement (steps (9)–(11)). CMGA can proceed with the next generation once average fitness of the current population is better than that of the last one.

5.4. *Experimental Conditions.* The parameter setting is in Table 2. We test with 30 trial runs of 20 generations for each GMM with different dimension setting, and apply statistical

```

(1)  $G = 0$ 
(2) Initialize  $P(G)$ 
(3) Computing Fitness( $P(G)$ )
(4) while Non-Termination do
(5) Selection( $P(G)$ )
(6) Crossover( $P(G)$ )
(7) Mutation( $P(G)$ )
(8) Computing Fitness( $P(G)$ )
(9) if Ave fitness of  $P(G) >$  Ave fitness of  $P(G - 1)$  then
(10)    $G = G + 1$ 
(11) end if
(12) end while

```

ALGORITHM 3: Constructive mapping genetic algorithm. (G : generation and $P(G)$: population of the G th generation).

tests (sign test, Friedman test, and Bonferroni-Dunn test) to evaluate the significance of our proposals with their comparison algorithm. All these GMM tasks are posed as maximization problems with the optimal solution, which is the point with higher value.

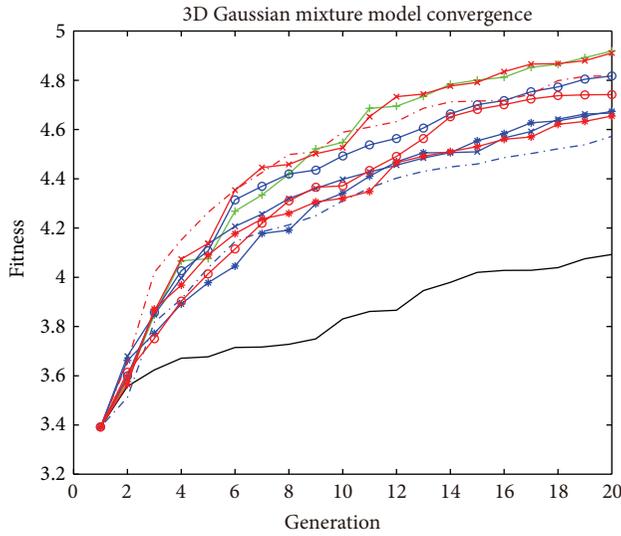
We abbreviate the GA where the evolution control methods are by the linear kernel as GA-Linear, where the evolution control methods are by the polynomial kernel (15) with parameter setting with 2, 5, 7, and 10 as GA-Poly2, GA-Poly5, GA-Poly7, and GA-Poly10, where the evolution control methods are by the RBF kernel (16) with parameter setting with 1, 5, 7, and 10 as GA-RBF1, GA-RBF5, GA-RBF7, GA-RBF10, and CMGA as GA-N. These abbreviations are also used in Figures 3, 4, and 5, Tables 1 and 3.

5.5. *Experimental Results.* Figure 3 shows the average convergence curves of the best fitness values of 30 trial runs of GA-N, GA-Linear, GA-Poly2, GA-Poly5, GA-Poly7, GA-Poly10, GA-RBF1, GA-RBF5, GA-RBF7, and GA-RBF10. For different dimension GMM, Table 1 shows the mean and standard variance and Figure 4 shows their sign tests at each generation. From these results, we can obtain the following results.

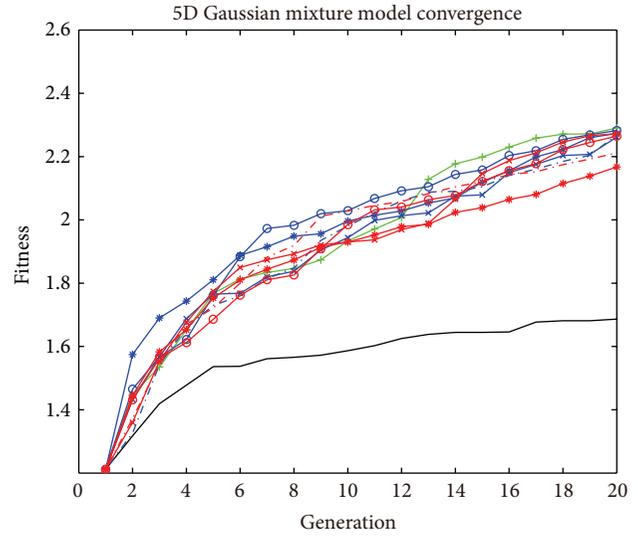
- (1) Our proposed methods can significantly accelerate all of the GMM well.

TABLE 3: Algorithm ranking by Friedman test ($P < 0.05$).

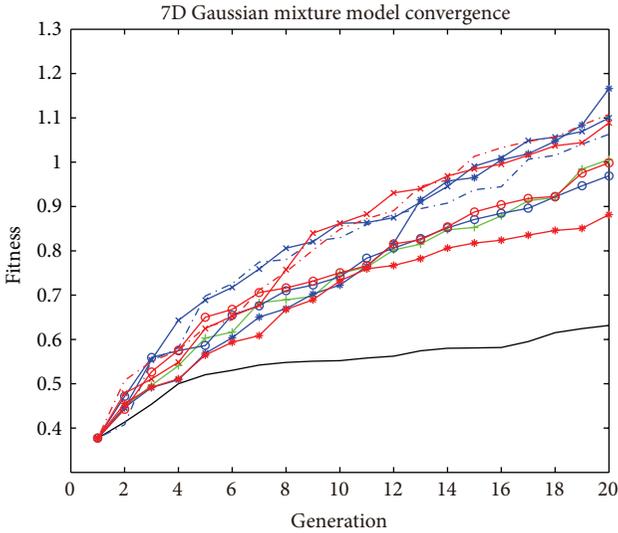
F	N	Linear	Poly2	Poly5	Poly7	Poly10	RBF1	RBF5	RBF7	RBF10
3-D	2.53	6.22	6.03	5.40	5.52	5.15	5.63	6.42	5.65	6.45
5-D	2.87	5.97	6.00	5.97	5.93	5.57	5.83	6.13	5.07	5.67
7-D	2.67	5.80	5.13	6.83	5.87	6.33	5.67	6.27	4.40	6.03
10-D	1.50	5.27	6.13	6.83	7.03	5.30	5.23	6.37	5.53	5.80
Average	2.39	5.81	5.83	6.26	6.09	5.58	5.59	6.30	5.16	5.99
Rank	10	6	5	2	3	8	7	1	9	4



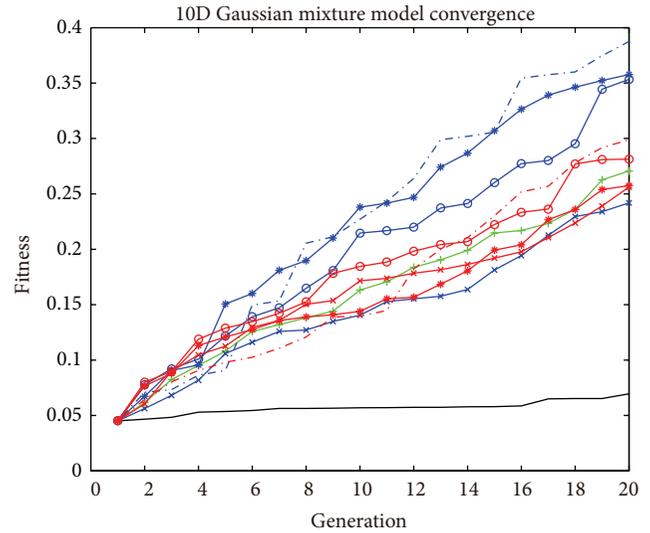
(a)



(b)



(c)



(d)



FIGURE 3: Average convergence curves of 30 trial runs with 20 generations for (a) 3D Gaussian Mixture Model, (b) 5D Gaussian mixture model, (c) 7D Gaussian mixture model, (d) 10D Gaussian mixture model, the performance of our proposed method is better than CMGA.

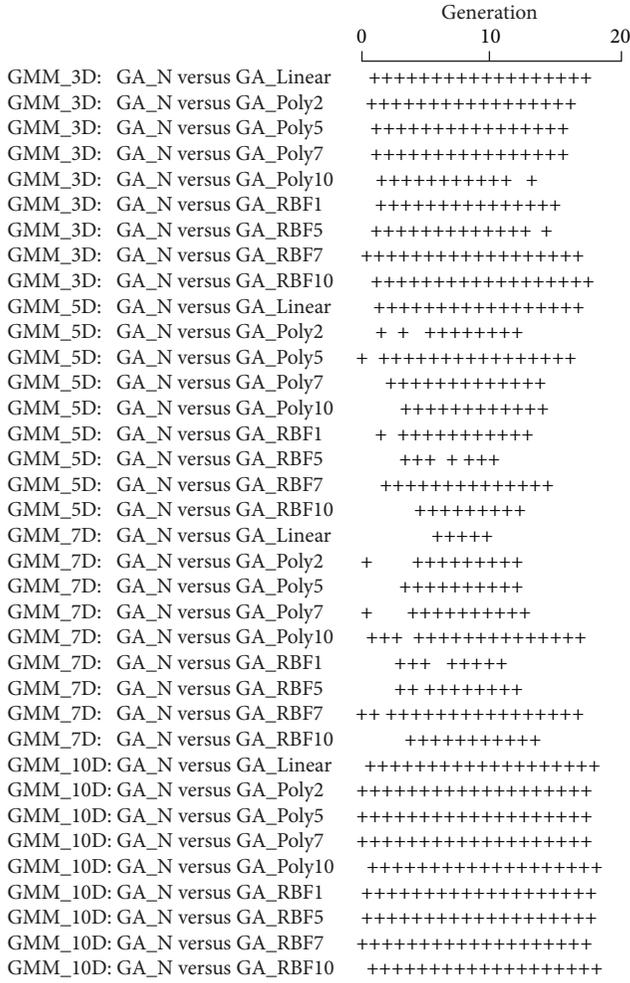


FIGURE 4: Sign test results for 30 trial runs of GA-N versus Linear, GA-N versus GA-Poly2, GA-N versus GA-Poly5, GA-N versus GA-Poly7, GA-N versus GA-Poly10, GA-N versus GA-RBF1, GA-N versus GA-RBF5, GA-N versus GA-RBF7, and GA-N versus GA-RBF10. The + mark means that a propose method converges significantly better than CMGA, respectively ($P < 0.05$). There are no cases where the proposed methods are significantly poorer than CMGA.

- (2) The performances of nine proposed algorithms are better than that of the CMGA.
- (3) Linear kernel method (GA-Linear) and RBF kernel methods (GA-RBF) seem to have a better acceleration performance for the lower dimensional task (GMM with 3 dimensions); however, polynomial kernel method (GA-Poly) have a better acceleration performance for the higher dimensional task (GMM with 10 dimensions).
- (4) Most of the cases, RBF kernel method (GA-RBF) and polynomial kernel method (GA-Poly) have the same acceleration performance to GMM with 5 and 7 dimensions.

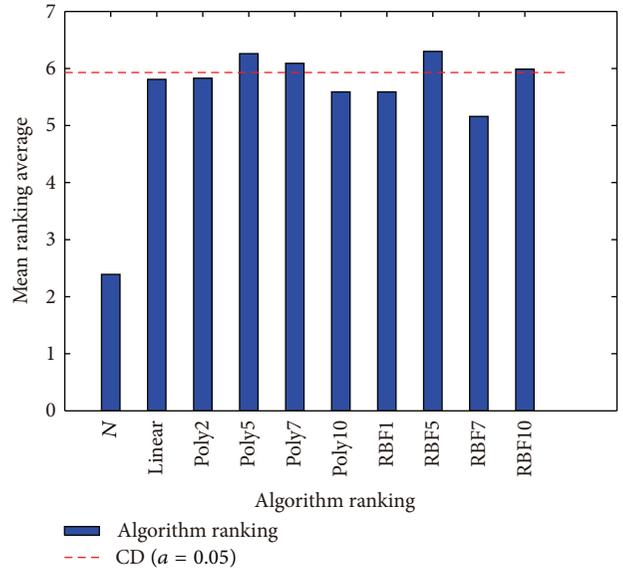


FIGURE 5: Rankings obtained through the Friedman test and graphical representation of the Bonferroni-Dunn's procedure (Taking GA-N as a control method). CD means critical difference.

- (5) Polynomial kernel method (GA-Poly) is better than RBF kernel method (GA-RBF) in the 10-dimensional GMM.

6. Discussions

6.1. Optimization Performance of the Proposal. Experimental evaluation results show that our proposed evolution control method with human model can assist and enhance IEC search significantly. The results indicate that kernel classification is a powerful tool to establish a human model that distinguishes property of individual, which is near or beyond the global optimum (human preference). However, its performance depends on the tasks, training data, and kernel function.

From the sign test (Figure 4), we can observe that our proposed methods are more effective to the GMM with 3 and 10 dimensions than that with 5 and 7 dimensions. This result indicates that feature space projected by linear, polynomial, and RBF kernel for GMM with 3 and 10 dimensions can separate near or beyond the global optimum clearly than that for GMM with 5 and 7 dimensions. It indicates that linear, polynomial, and RBF kernel are better methods to implement human model by a linear classifier with the fitness landscape characteristics as the GMM with 3 and 10 dimensions.

From the average convergence result (Figure 3), we compare four Subfigures (a) to (d) from lower dimension to high dimension. It sketches acceleration performance of polynomial kernel seems to become better along with the GMM dimensions' increasing. It concludes that polynomial kernel methods have better performance for high dimensional tasks.

We apply Friedman test ($P < 0.05$) to rank the algorithms, which we use in the experimental evaluations (Table 3). We can observe that GA-Poly methods and GA-RBF methods

almost have the same optimization performance from the ranking metric. To evaluate the significant level of all the algorithms, we apply an additional Bonferroni-Dunn test to calculate critical difference (CD in (21)) for comparing their differences in significant level of $\alpha < 0.05$:

$$CD = q * \sqrt{\frac{k * (k + 1)}{6 * N}}. \quad (21)$$

In (21), parameters k and N are the number of algorithms and number of benchmark tasks, respectively. They are $k = 10$ and $N = 4$ in the experimental evaluations. When $\alpha < 0.05$, q is 3.261 from Table B16 (two-tailed $\alpha(2)$) of [38]. Figure 5 sketches the results of Bonferroni-Dunn test. There is a significant difference between GA-N methods and some of our proposed methods. We can conclude that some of our proposed algorithms by embedding a human model can enhance IEC search and better than normal method (GA-N) significantly.

Our proposal is to obtain fitness landscape in feature space where the correct or accurate preference information may not be obtained in an original search space and establish a human model by these obtained information. In our proposed evolution control methods, we choose a linear classifier as a human model to obtain preference information that original search space cannot support correctly. From the experimental results, our proposed methods obtain better performance than that of the dynamic fitness threshold technique (CMGA) that conducts search strategy only based on original search space fitness landscape. In a high dimensional feature space, the important fitness landscape information is not only individuals' classification, but also the search direction or the global optimum's location, which can directly guide IEC search. If we can obtain more of such information from the designed human model in feature space, IEC performance must be improved significantly. This is the final objective of our proposal.

6.2. Human Model Design in Feature Space. It is crucial to design an accurate human model in feature space to obtain better performance for distinguishing individuals' property. There are three aspect issues to be considered. First is training data selection, second is kernel function selection, and the third is kernel parameter setting.

6.2.1. Training Data. Training data selection decides the correct classification in feature space. In our experimental evaluation, we only choose three individuals with related better fitness and three individuals with related worse fitness as the labelled data for training the human model. From the result, it can be as one of the selection methods; however, if we can obtain more original search space information to decide how to select training data, it must improve human model performance by a linear classifier and reduce computational cost.

The number of training data depends on population size and linear classifier learning capability. If we separate population into two groups with better and worse fitness as the training data, the overtraining problem may happen. So

how to decide the proper training data number to construct a human model is a promising topic in our future work.

6.2.2. Kernel Function. Kernel function selection decides higher feature space topology. From the Mercer theorem, any kernel function corresponds to some feature spaces. It decides the individuals' distribution and classification capability of linear classifier in feature space. On one hand, in a set of well known kernel function, for a fixed search space of IEC task, there must be a kernel function with optimal classification performance. It is a promising study topic on how to select a proper kernel function for a concrete application. On the other hand, if we can obtain some a priori knowledge from an original search space, we also can design a new kernel function to transfer individuals into the desired feature space, where the fitness landscape is more beneficial for IEC search. There is not an absolute rule for selecting a proper kernel function. The design and selection of a kernel function should be adapted to a concrete search space in IEC. Some principles should be considered.

- (1) Kernel function selection or design should consider feature space structure, original search space's prior knowledge and training sample data.
- (2) Kernel function should induce a priori knowledge and present an original search space's information structure.
- (3) Kernel function selection and design must keep information structure in feature space, that is, linear characteristic.

6.2.3. Kernel Parameter Setting. Kernel parameter setting decides the topology of a high dimensional feature space. A well-known study topic is the parameter setting and tuning, after we decide a kernel function for a concrete IEC application. There is still not a mathematical conclusion which is the best parameter setting for a kernel function. It depends on the experimental result and experience from a concrete IEC application. However, it is a valuable study topic on designing a better human model to obtain better performance in feature space for IEC application.

6.2.4. Other Design Issues. In our experimental evaluation, we design a human model by a binary classifier in feature space, which is a little imprecise. The possible result is that individuals in the same generation may have the same fitness value. It decreases the pressure of selecting superior individuals in GA, which is one of drawbacks of our designed binary classifier human model. Human model design can be implemented by a multiclass classifier to solve this issue. If there is more a priori knowledge by a certain IEC human user and IEC task, we also can improve the design method of the human model to use a multi class classifier in the evolution control method. In this way, we can obtain more novel human models and evolution control methods to obtain better performance of IEC search. We will conduct this subject in our future work.

6.3. *Computational Complexity.* From (13) and Algorithm 2, it presents a concrete algorithm that conducts interactive search by using a kernel method based human model. The process needs more algebraic operations, so it is costly. For a concrete IEC application, the time used in computing is less than that of human's subjective evaluation. However, we should also consider actual time cost in kernel computing and reduce it.

In our experimental evaluation, we conduct evolution control method every generation. However, in a concrete IEC application, we can conduct this strategy several generation once to save the computational cost rather than conducting it in every generation. It is necessary to consider the time cost for this method in a real world application.

6.4. *Proposed Methods in Other EC Algorithms and Human Related Computation.* In our experimental evaluation, we apply a human model with an evolution control method in four GMM for enhancing IEC search. In general, our proposed human model can be used in all human related computing, such as awareness computing, human computing, which obtains preference information of a human user by kernel method. When applying this human model and evolution control method in another EC algorithm, we need to consider the linear classifier design method mentioned above and make sure obtained information is correct. Otherwise, the wrong information obtaining can lead to worse performance of the normal IEC search or other human related computing applications.

7. Conclusion and Future Work

In this study, we propose a method to relieve IEC user fatigue by establishing a human model to obtain preference information by kernel classification. In high dimensional feature space, we design a linear classifier to judge an individual property corresponding to human preference. Based on the obtained fitness landscape, we propose a human model design method and an evolution control method to enhance IEC search. The experimental evaluation with four different dimension GMM as a pseudo-IEC user shows that our proposed methods are effective. We also analyze the performance and limitation of our proposed methods. Some open topics and further opportunities are discussed.

Our further plan of this research is to evaluate our proposed methods to a concrete IEC application using a real human user to obtain a practical conclusion of the proposal. Other issues are to continue designing an efficient search strategy based on obtained human model in high dimensional fitness landscape to improve the human model by classifier design for obtaining better enhancement performance, and so forth. We will conduct these research topics in the future.

Conflict of Interests

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

References

- [1] C. Caldwell and V. S. Johnston, "Tracking a criminal suspect through face-space with a genetic algorithm," in *Proceedings of the 4th International Conference on Genetic Algorithm*, pp. 416–421, Morgan Kaufmann, San Diego, Calif, USA, 1991.
- [2] K. Sims, "Artificial evolution for computer graphic," in *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)*, pp. 319–328, 1991.
- [3] M. Herdy, "Evolutionary optimisation based on subjective selection—evolving blends of coffee," in *Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT '97)*, pp. 640–644, Aachen, Germany, 1997.
- [4] A. Kosorukoff, "Human based genetic algorithm," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 3464–3469, Tucson, Ariz, USA, October 2001.
- [5] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.
- [6] X. Sun, D. Gong, Y. Jin, and S. Chen, "A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 685–698, 2013.
- [7] X. Sun, D. Gong, and W. Zhang, "Interactive genetic algorithms with large population and semi-supervised learning," *Applied Soft Computing*, vol. 12, no. 9, pp. 3004–3013, 2012.
- [8] D. Gong, X. Ji, J. Sun, and X. Sun, "Interactive evolutionary algorithms with decision-maker's preferences for solving interval multi-objective optimization problems," *Neurocomputing*, vol. 137, pp. 241–251, 2014.
- [9] Y. Pei and H. Takagi, "A survey on accelerating evolutionary computation approaches," in *Proceedings of the International Conference of Soft Computing and Pattern Recognition (SoC-PaR '11)*, pp. 201–206, Dalian, China, 2011.
- [10] Y. Pei and H. Takagi, "Comparative study on fitness landscape approximation with fourier transform," in *Proceedings of the 6th International Conference on Genetic and Evolutionary Computing (ICGEC '12)*, pp. 400–403, IEEE, Kitakyushu, Japan, August 2012.
- [11] Y. Pei and H. Takagi, "Fourier analysis of the fitness landscape for evolutionary search acceleration," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '12)*, pp. 1–7, June 2012.
- [12] H. Takagi, T. Ingu, and K. Ohnishi, "Accelerating a GA convergence by fitting a single-peak function," *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, vol. 15, no. 2, pp. 219–229, 1993.
- [13] Y. Pei and H. Takagi, "Accelerating evolutionary computation with elite obtained in projected one-dimensional spaces," in *Proceedings of the 5th International Conference on Genetic and Evolutionary Computing (ICGEC '11)*, pp. 89–92, Jimmen, Taiwan, September 2011.
- [14] Y. Pei and H. Takagi, "Accelerating IEC and EC searches with elite obtained by dimensionality reduction in regression spaces," *Evolutionary Intelligence*, vol. 6, no. 1, pp. 27–40, 2013.
- [15] Y. Pei, K. Chao, B. Fu, Y. J. Lin, H. Chen, and Y. Xu, "Method and system for determining recommended passage place sequence," CN Patent 102158799(B), 2013.
- [16] Y. Pei and H. Takagi, "A novel traveling salesman problem solution by accelerated evolutionary computation with approximated cost matrix in an industrial application," in *Proceedings*

- of the International Conference of Soft Computing and Pattern Recognition (SoCPaR '11), pp. 39–44, IEEE, October 2011.
- [17] W. R. Ashby, *An Introduction to Cybernetics*, Chapman & Hall, London, UK, 1956.
- [18] E. Law and L. von Ahn, *Human Computation*, Synthesis Lectures on Artificial Intelligence and Machine Learning, 2011.
- [19] H. Takagi, “Interactive evolutionary computation for analyzing human awareness mechanisms,” *Applied Computational Intelligence and Soft Computing*, vol. 2012, Article ID 694836, 8 pages, 2012.
- [20] A. M. Turing, “On computable numbers, with an application to the entscheidungsproblem,” *Proceedings of the London Mathematical Society*, vol. 42, no. 2, pp. 230–265, 1936.
- [21] S. K. Card, T. P. Moran, and A. Newell, “The model human processor: an engineering model of human performance,” in *Handbook of Human Perception*, vol. 2, pp. 1–35, 1986.
- [22] Y. Zhao, X. Wang, M. Goubran, T. Whalen, and E. M. Petriu, “Human emotion and cognition recognition from body language of the head using soft computing techniques,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 4, no. 1, pp. 121–140, 2013.
- [23] M. C. Finlay, L. Xu, P. Taggart, B. Hanson, and P. D. Lambiase, “Bridging the gap between computation and clinical biology: validation of cable theory in humans,” *Frontiers in Physiology*, vol. 4, article 213, 2013.
- [24] C. Chandrasekaran, L. Lemus, A. Trubanova, M. Gondan, and A. A. Ghazanfar, “Monkeys and humans share a common computation for face/voice integration,” *PLoS Computational Biology*, vol. 7, no. 9, Article ID e1002165, 2011.
- [25] A. Whiten, “Humans are not alone in computing how others see the world,” *Animal Behaviour*, vol. 86, no. 2, pp. 213–221, 2013.
- [26] Y. Pei, “Chaotic evolution: fusion of chaotic ergodicity and evolutionary iteration for optimization,” *Natural Computing*, vol. 13, no. 1, pp. 79–96, 2014.
- [27] P. Naur, “Computing versus human thinking,” *Communications of the ACM*, vol. 50, no. 1, pp. 85–94, 2007.
- [28] Y. Pei and H. Takagi, “Fitness landscape approximation by adaptive support vector regression with opposition-based learning,” in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '13)*, pp. 1329–1334, October 2013.
- [29] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer, “Theoretical foundations of the potential function method in pattern recognition learning,” *Automation and Remote Control*, vol. 25, no. 6, pp. 821–837, 1964.
- [30] J. Mercer, “Functions of positive and negative type, and their connection with the theory of integral equations,” *Philosophical Transactions of the Royal Society of London. Series A. Containing Papers of a Mathematical or Physical Character*, vol. 209, pp. 415–446, 1909.
- [31] R. Herbrich, Ed., *Learning Kernel Classifiers: Theory and Algorithms*, The MIT Press, Cambridge, Mass, USA, 2001.
- [32] B. Schoelkopf, A. Smola, and K.-R. Mueller, “Nonlinear component analysis as a kernel eigenvalue problem,” Tech. Rep., Max-Planck-Institut für biologische Kybernetik Arbeitsgruppe Bülthoff, 1996.
- [33] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [34] B. Schoelkopf, C. J. C. Bugar, and A. Smola, Eds., *Advances in Kernel Methods: Support Vector Learning*, The MIT Press, Cambridge, Mass, USA, 1998.
- [35] M. K. Maiti and M. Maiti, “Multi-item shelf-space allocation of breakable items via genetic algorithm,” *Journal of Applied Mathematics and Computing*, vol. 20, no. 1-2, pp. 327–343, 2006.
- [36] G. A. Miller, “The magical number seven, plus or minus two: some limits on our capacity for processing information,” *The Psychological Review*, vol. 63, no. 2, pp. 81–97, 1956.
- [37] H. Takagi and D. Pallez, “Paired comparison-based interactive differential evolution,” in *Proceedings of the 1st World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 475–480, Coimbatore, India, December 2009.
- [38] J. H. Zar, *Biostatistical Analysis*, Pearson Prentice-Hall, Upper Saddle River, NJ, USA, 5th edition, 2007.

Research Article

Unsupervised Spectral-Spatial Feature Selection-Based Camouflaged Object Detection Using VNIR Hyperspectral Camera

Sungho Kim

Yeungnam University, Gyeongsan, Gyeongbuk 712-749, Republic of Korea

Correspondence should be addressed to Sungho Kim; sunghokim@ynu.ac.kr

Received 12 June 2014; Accepted 28 August 2014

Academic Editor: Shifei Ding

Copyright © 2015 Sungho Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The detection of camouflaged objects is important for industrial inspection, medical diagnoses, and military applications. Conventional supervised learning methods for hyperspectral images can be a feasible solution. Such approaches, however, require a priori information of a camouflaged object and background. This letter proposes a fully autonomous feature selection and camouflaged object detection method based on the online analysis of spectral and spatial features. The statistical distance metric can generate candidate feature bands and further analysis of the entropy-based spatial grouping property can trim the useless feature bands. Camouflaged objects can be detected better with less computational complexity by optical spectral-spatial feature analysis.

1. Introduction

The development of an image sensor and optical dispersion technology has made it possible to capture hyperspectral image data with lower prices, such as SPECIM or Honeywell products [1]. Therefore, it is possible to inspect and develop algorithms from acquired hyperspectral images instead of a public database of remote hyperspectral images, such as AVIRIS and Hyperion [2]. Currently, hyperspectral images are used frequently in a range of areas to detect important parts such as cavities in medical applications and crime in forensic applications [1, 3].

Although spectral information can be useful for discriminating camouflaged or abnormal regions, the high dimensionality of the hyperspectral data leads to a huge increase in computational time, and the highly correlated bands contain a degree of redundancy, which might have a negative impact on detection. For example, if a single scan of a hyperspectral cube contains 1,392 pixels (samples), 1,000 pixels (scan length), and 1,040 (bands) with a 2 bytes A/D resolution, the total data size was approximately 3 GBytes. This is 600 times larger than the size of the full HD image (6 MBytes). Therefore, the key problem for the detection of

hyperspectral abnormal regions is to reduce the computation complexity without degrading the detection accuracy. Therefore, reducing the dimensionality by the spectral band selection is often adopted to reduce computational cost and improve the accuracy.

Band selection can be achieved by supervised or unsupervised learning. The former requires a set of labeled training databases and produces the high accuracy of detection performance [4–7]. On the other hand, the number of training samples is limited in most hyperspectral applications. The latter requires no training images and detects abnormal regions directly from a test hypercube. Therefore, this study adopted the unsupervised learning-based band selection scheme for its convenience in automatic camouflaged or abnormal region detection. Recently, several studies proposed a range of band selection or elimination methods in unsupervised learning approaches focusing only on spectral analysis. Previous techniques of unsupervised band selection for hyperspectral images can be classified broadly into two categories: ranking-based methods [8, 9] and clustering-based methods [10, 11]. The ranking-based methods evaluate the relevance of each band independently to estimate the quality of the attributes depending on how well their values

help classify the patterns using either the information divergences [9] or similarity-based band analysis [8]. On the other hand, clustering-based methods perform clustering on bands to group them according to their correlation and selects one band from each cluster representing the whole group, such as mutual information [11] or affinity propagation [10].

In the first stage of spectral feature analysis, a new statistical distance measure in the ranking-based method instead of the band clustering method was proposed due to the high computational complexity. Spectral analysis can generate candidate bands that maximize the statistical distance. In the second stage, an entropy-based measure was proposed to quantify the uncertainty of spatial segmentation. The bands that generate high entropy value (noisy spatial segmentation) can be reduced. Therefore, the first contribution is the proposition of a novel band selection method by considering both spectral and spatial analysis without prior knowledge. The second contribution is the automatic detection of a camouflaged or abnormal region without a training process. Therefore, the detected results can be obtained without human intervention if any kinds of hyperspectral test images are applied to the inspection system. Section 2 introduces the proposed camouflaged target detection method using spectral-spatial feature analysis. Section 3 validates the proposed method according to various band selection schemes and Section 4 concludes the study.

2. Proposed Camouflaged Object Detection Method

2.1. Overview of the Proposed Inspection System. Figure 1 summarizes the overall flow of the proposed hyperspectral inspection system. Given a test hypercube image, the automatic band selection block is activated by consecutive spectral and spatial analysis. Statistical distance analysis of each spectral band generates a discriminating curve. The candidate spectral bands can be obtained through the local maxima of the curve. Segmented regions can be obtained using each band with cluster labels. The underlying assumption is that good band segments the input image into two regions: camouflaged and background regions. The number of segmented regions was quantized using entropy. Therefore, entropy can represent the complexity of regions. Based on entropy, the optimal bands are selected. The final detection results were obtained using *K*-means clustering with the selected bands.

2.2. Spectral-Spatial Analysis-Based Band Selection

2.2.1. Hypercube Acquisition System (Table 1). The spectral image acquisition system consists of a SPECIM VNIR camera (PS-FW-11-V10E) mounted on a linear stage, LED, or halogen lamps and a target to inspect, as shown in Figure 2(a). Figure 2(b) shows sample spectral band images.

2.2.2. Spectral Analysis. A camouflaged object detection problem can be regarded as selecting suitable spectral bands that discriminate interesting region in normal background.

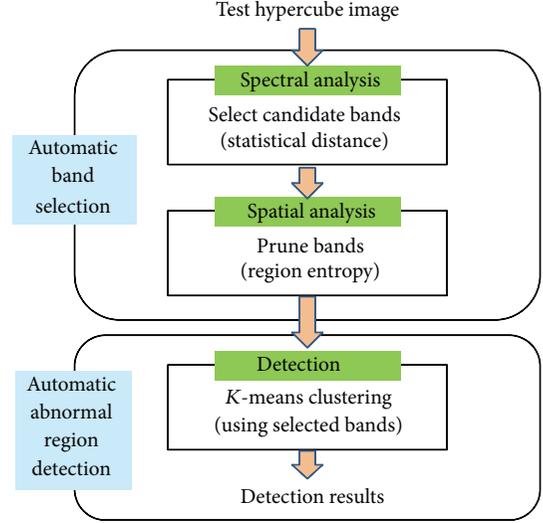


FIGURE 1: Overall inspection flow of the proposed system.

TABLE 1: Specifications of the hyperspectral image acquisition system.

Item	Specifications
Spectral range	400–1000 nm (VNIR)
Spectrograph	ImSpector V10E 30 μ slit, 2.8 nm spectral resolution
Camera	Kappa 1,392 \times 1,040 pixels, 12 bits, 11 fps, FireWire interface
Lamp	150 W A08975, SCHOTT
Scanner	Linear stage, length 750 mm

The proposed statistical distance metric can be useful to generate candidate bands because a hypercube image provides enough samples (about millions) of spectral profiles and statistical distance can measure the distinctiveness of spectral bands, which leads to easy detection of camouflaged objects. For example, if a test hypercube consists of a real leaf and a printed leaf, the spectral profile and specific band image are obtained as shown in Figure 3. Statistical distance-based, candidate band selection is motivated by the observation of band image analysis, as shown in Figure 4(a). The distribution histogram can be made for a hypothesized band b [nm]. According to the distribution, two Gaussian distribution functions (foreground and background) parameterized from the means (μ_1, μ_2) and standard deviations (σ_1, σ_2) can be fitted. The class discriminability measure is defined as

$$D(b) = \frac{|\mu_1(b) - \mu_2(b)|}{\sigma_1(b) + \sigma_2(b)}. \quad (1)$$

By applying the aforementioned equation to each band, the band discriminability curve can be obtained according to the wavelength, as shown in Figure 4(b). The candidate bands can be selected by applying local maxima or global maxima to the curve.

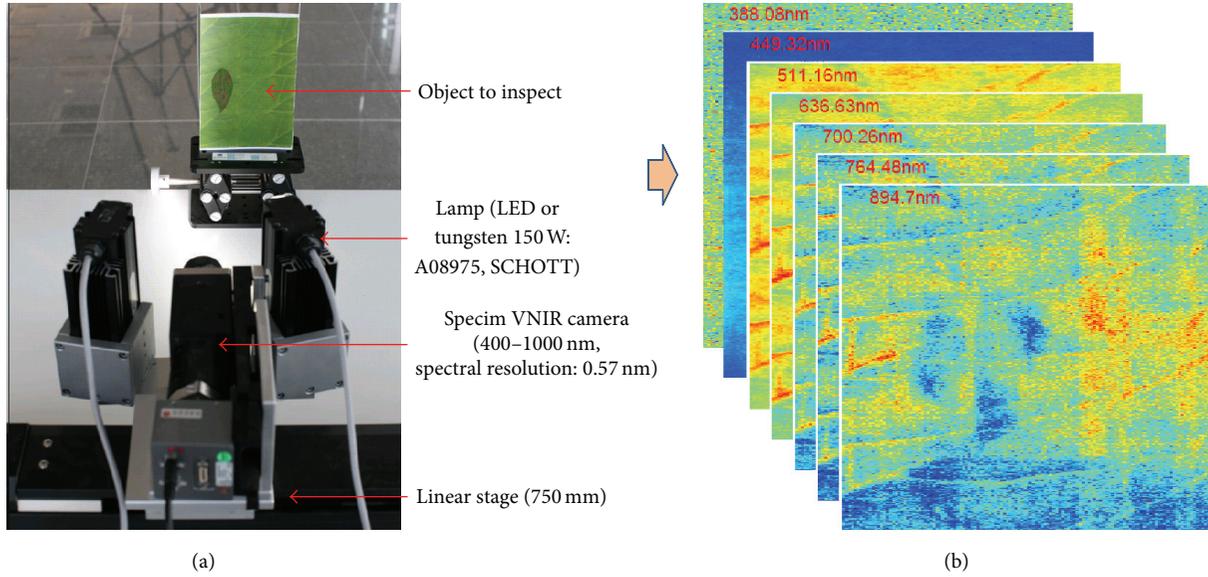


FIGURE 2: Hypercube: (a) VNIR hyperspectral image acquisition system and (b) sample spectral images.

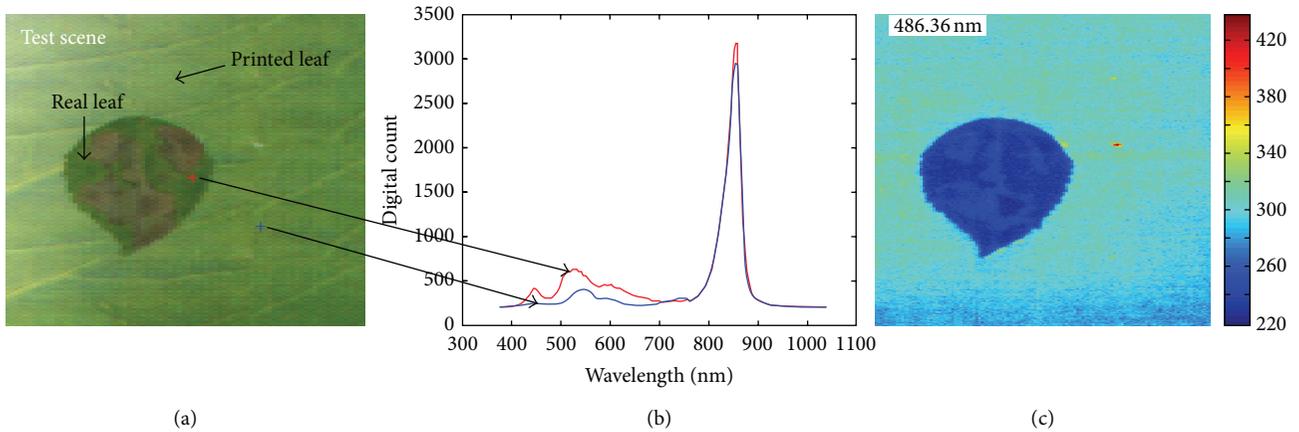


FIGURE 3: Spectral analysis: (a) test hypercube, (b) spectral profiles at the selected points, and (c) band image at 486.36 nm.

2.2.3. *Spatial Analysis.* K -means clustering can effectively cluster data using feature distance [12]. If K -means clustering ($K = 2$) with b th band image is performed, the discriminability value can be obtained as mentioned above. At the same time, a segmented image using the class labels in image space can be acquired. If a hypercube image has the size of samples (S) \times scan length (L) \times bands (B), the complexity of segmented regions at the b th band can be quantified using entropy. Entropy can measure the complexity of spatial region distribution. In the camouflaged object detection problem, the ideal number of regions is just two (foreground and background). Therefore, high entropy can represent large number of segmented regions. The region entropy is defined as

$$H(b) = -\sum_{i=1}^M p_i(b) \log p_i(b), \quad (2)$$

where $p_i(b)$ is the probability of the pixels belonging to i th region. This is defined as $p_i(b) = N_i(b)/(S \times L)$. M denotes the total number of segmented regions and $N_i(b)$ denotes the number of pixels belonging to the i th region at the band image b . Ideally, the detection results consist of one abnormal region and the other background region. If the number of segmented region increases, the region entropy increases. Therefore, a threshold is applied for the region entropy to reduce the candidate bands that generate many small regions. Figure 5 shows the region segmentation results according to the different region entropy values. The region entropy threshold around 1 is normally used.

3. Experimental Results

The proposed method was validated in terms of the band selection scheme using the same K -means clustering (unsupervised classifier). The baseline band selection method was

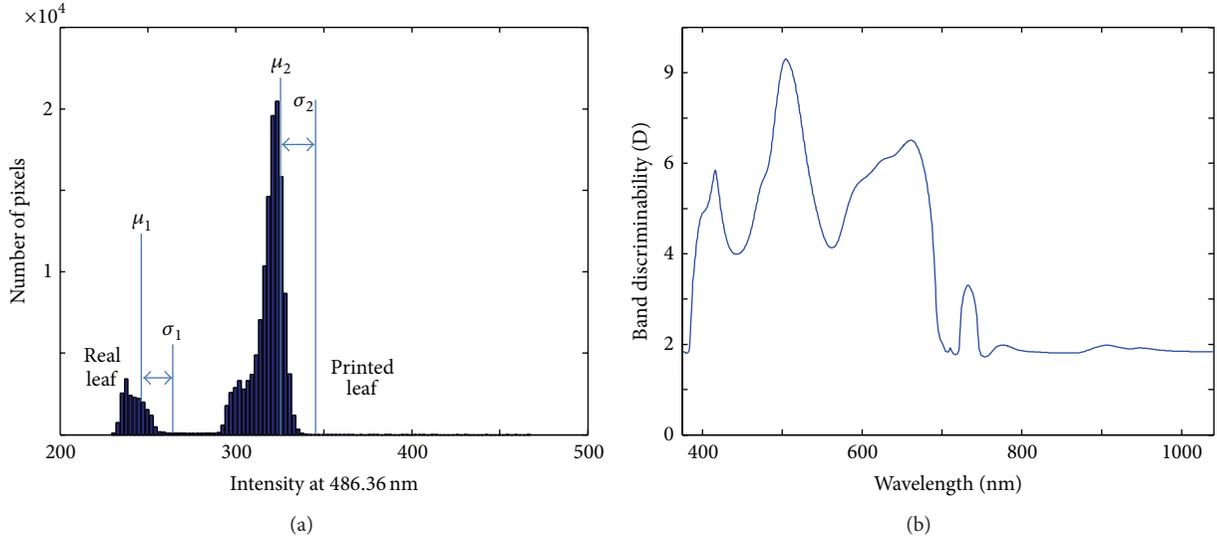


FIGURE 4: Spectral analysis: (a) test hypercube, (b) band image at 486.36 nm, (c) pixel distribution, and (d) proposed band discriminability graph.

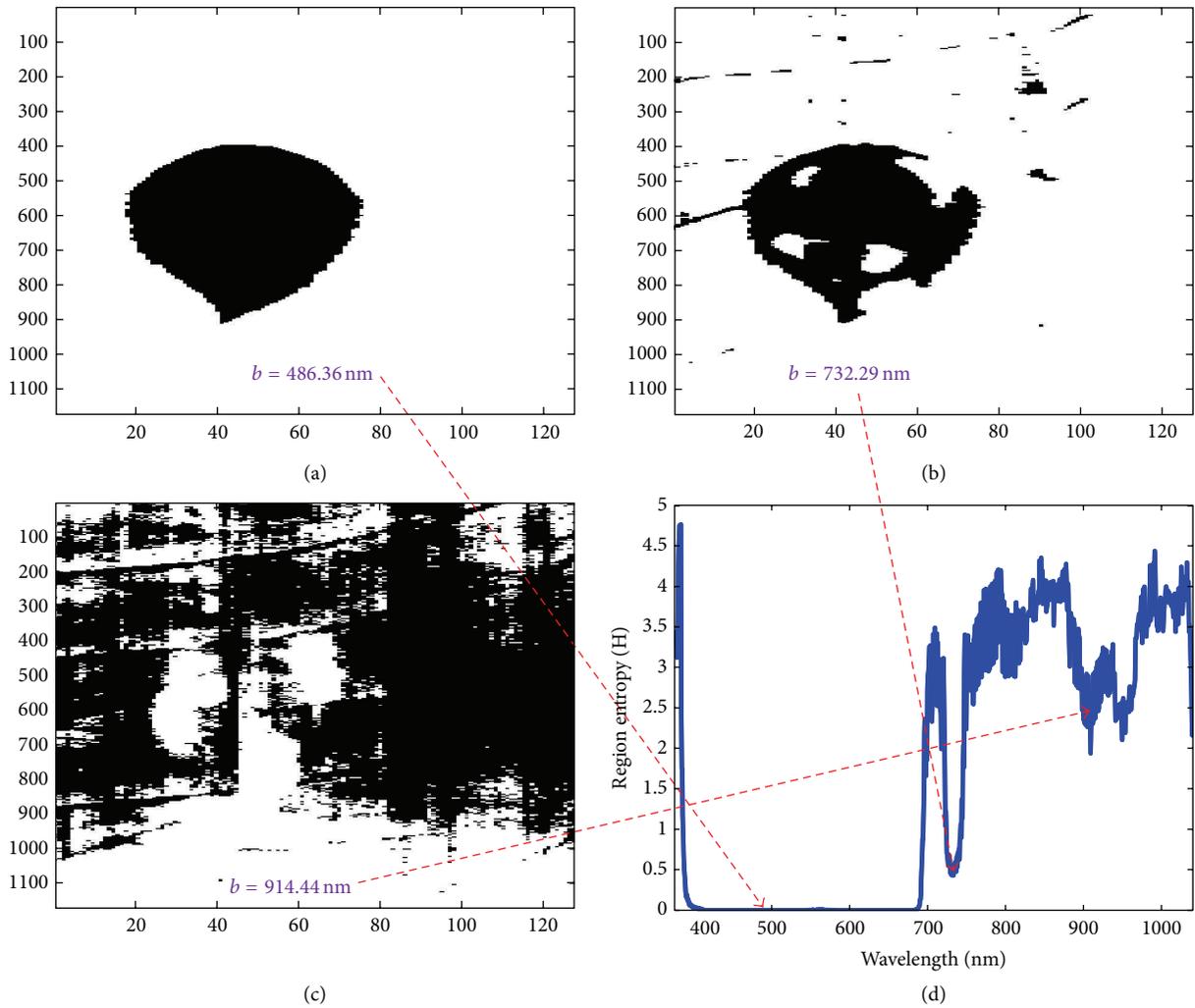


FIGURE 5: Spatial analysis: the large number of regions produces high region entropy score and two segmented regions produce lowest region entropy score.

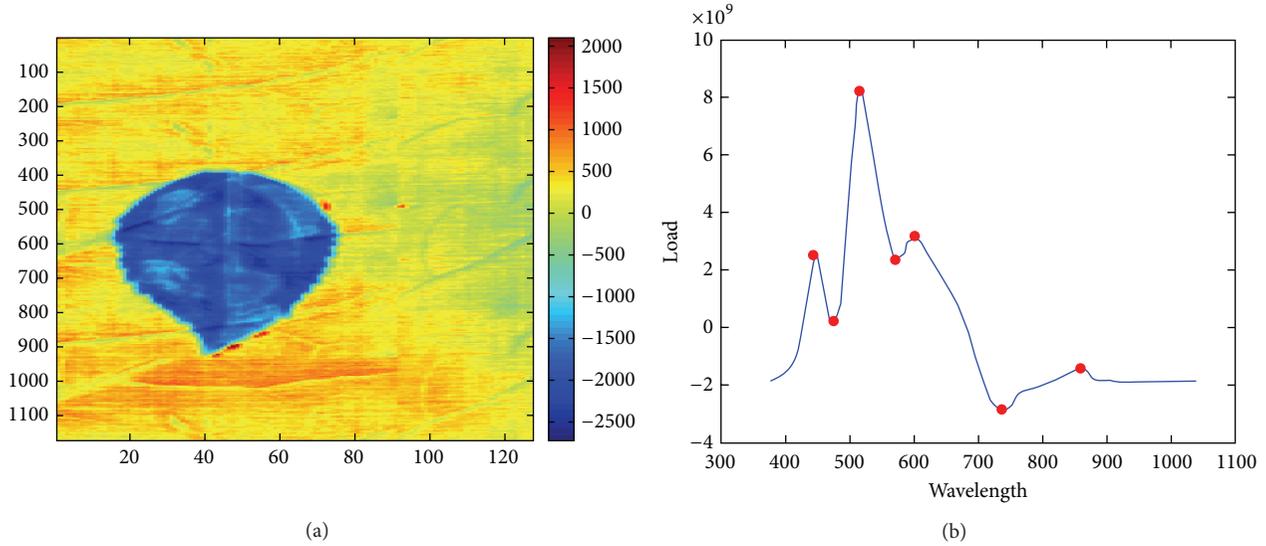


FIGURE 6: Baseline method of band selection by PCA [6]: (a) principal component: PC2 and (b) load function curve of PC2.

principle components analysis (PCA), which is an effective data reduction technique that is used frequently in hyperspectral data analysis [6]. In PCA, a human manually selects a principle component (i.e., PC2 as shown in Figure 6(a)) that visualizes the abnormal region clearly. The optimal set of bands can be selected using the local maxima/minima from the loading curve of PC2 as shown in Figure 6(b).

As a second baseline method, the entire spectrum curve, where all the bands are selected, is used [13]. These conventional methods are compared with the proposed band selection methods, such as band selection by spectral analysis (Proposed 1) and by spectral + spatial analysis (Proposed 2). The detection rate (DR), false alarm rate (FAR), and the number of bands used for quantitative comparison are used. Table 2 lists the overall performance comparison of the leaf database. PCA method selected 7 bands (447.4, 475.2, 517.3, 572.9, 600.6, 740.0, and 858.0). PCA and profile methods showed similar detection results with a high false alarm rate of 45%. The Proposed 1 method selected 9 bands (416.2, 503.7, 660.7, 732.3, 776.8, 905.2, 948.1, 1000.6, and 1027.3 nm) and showed 100% of DR with 0.008% of FAR. The Proposed 2 method with 4 selected bands (416.2, 503.7, 660.7, and 732.3 nm) showed the optimal performance with the fewest number of bands. Figures 7(c)–7(f) show the qualitative performance comparison results for a given test hypercube (Figure 7(a)) and a ground truth image (Figure 7(b)). The Proposed 2 method could detect the camouflaged region perfectly. In terms of detection time complexity, the Proposed 2 method took only 0.66 seconds which is 9.1 times faster than the PCA and 84.1 times faster than the profile. The space complexity is proportional to the number of bands. So, the Proposed 2 method occupies the smallest memory space.

Another test was conducted to validate the proposed method for the hair database, which consists of a wig and hair. Table 3 summarizes the overall performance comparison for the leaf database. PCA, profile, and Proposed 1 methods

TABLE 2: Comparison of abnormal region detection methods for the leaf database (DR: detection rate, FAR: false alarm rate, Proposed 1: spectral analysis, Proposed 2: spectral + spatial analysis).

Method	DR (%)	FAR (%)	Number of bands	Detection time (s)
PCA [6]	71.6	45.8	10	6.03
Profile [13]	73.5	45.2	1040	55.53
Proposed 1	100.0	0.0008	9	0.93
Proposed 2	100.0	0.0	4	0.66

TABLE 3: Comparison of the abnormal region detection methods for the hair database.

Method	DR (%)	FAR (%)	Number of bands
PCA [6]	92.1	3.1	5
Profile [13]	92.8	2.2	1040
Proposed 1	92.4	2.4	8
Proposed 2	99.9	0.2	2

showed similar detection results with a DR of 92% and FAR of 2~3%. The Proposed 2 method with 2 selected bands (945.5 and 1017.3 nm) showed the best performance with fewest number of bands. Figures 8(c)–8(f) show the qualitative performance comparison results for a given hair test image (Figure 8(a)) and a ground truth image (Figure 8(b)). As shown in Figure 8(a), detection errors occurred at the specular regions.

4. Conclusions

This letter proposed a novel band selection and abnormal region detection method in a completely unsupervised manner. From a test input hypercube, the proposed system generates candidate bands based on statistical distance analysis.

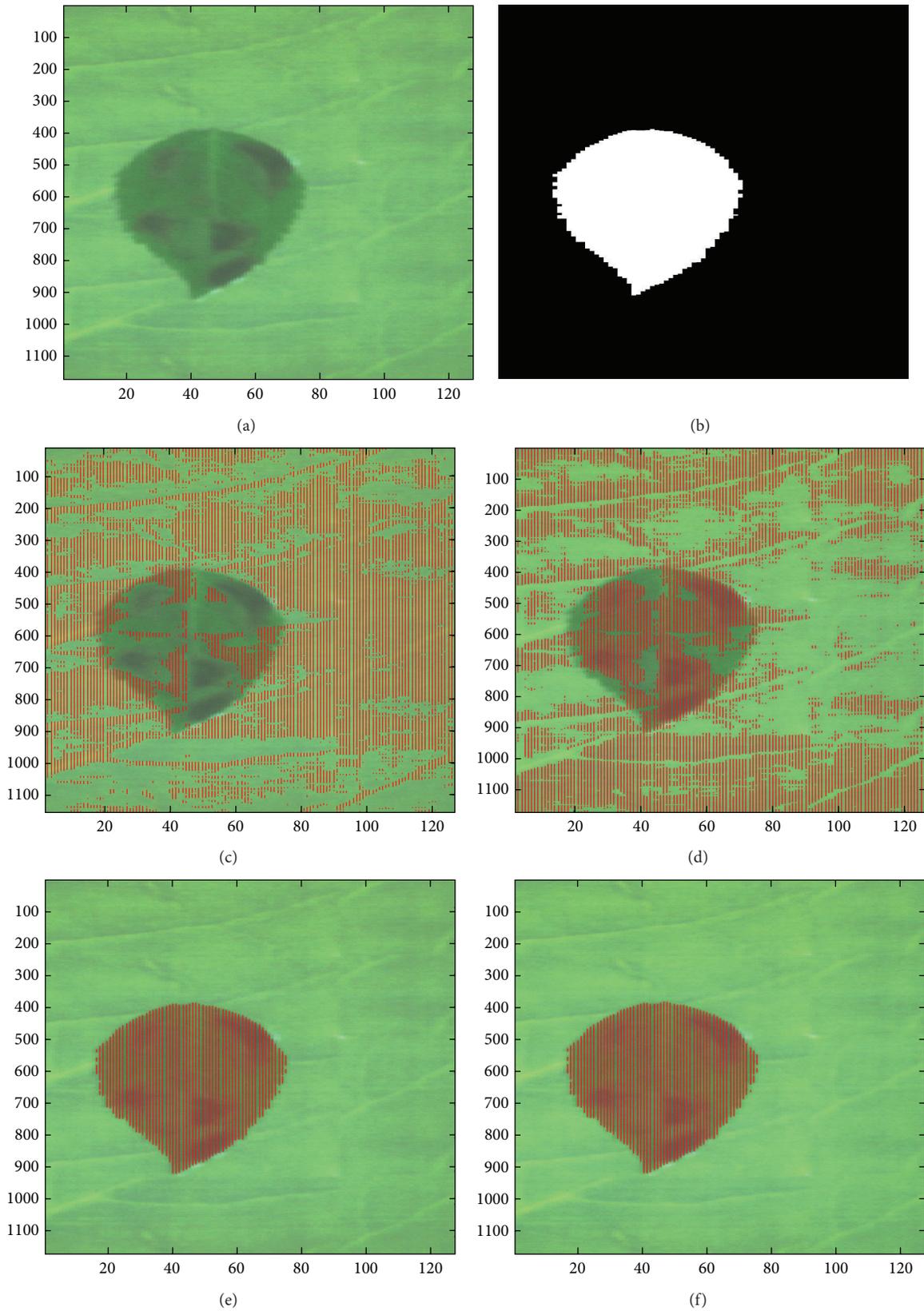


FIGURE 7: Abnormal region detection results: (a) test leaf image, (b) ground truth, (c) PCA method, (d) spectral profile, (e) Proposed 1 band selection by spectral analysis, and (f) Proposed 2 band selection by spectral-spatial analysis.

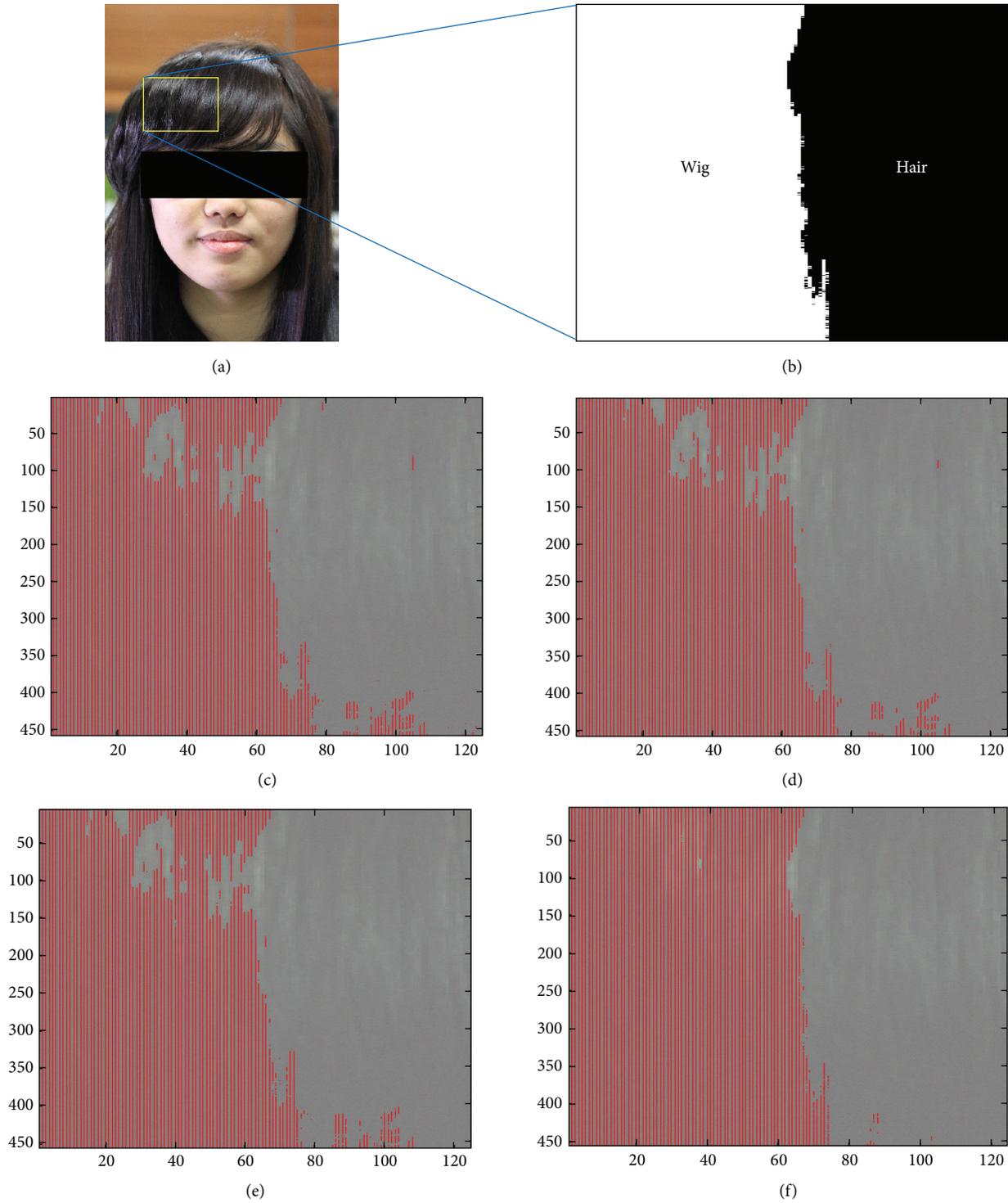


FIGURE 8: Abnormal region detection results: (a) test hair image, (b) ground truth, (c) PCA method, (d) spectral profile, (e) Proposed 1 band selection by spectral analysis, and (f) Proposed 2 band selection by spectral-spatial analysis.

The system removes bands that generate a number of region segments based on the region entropy measure. Experimental comparisons with the baseline methods validated the outperformance of the proposed method in terms of the detection

rate and false alarm rate with a minimal number of bands for a real test set. The best abnormal region detection result with a few selected bands (2–4) can be obtained without human intervention in both band selection and detection.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT, and Future Planning (NRF-2014R1A2A2A01002299).

References

- [1] J. Kuula, I. Polonen, H.-H. Puupponen et al., "Using VIS/NIR and IR spectral cameras for detecting and separating crime scene details," *Proceedings of SPIE*, vol. 8359, 83590 pages, 2013.
- [2] J. Cipar, T. Cooley, and R. Lockwood, "A comparison of forest classification using Hyperion and AVIRIS hyperspectral imagery," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '06)*, pp. 1956–1959, Denver, Colo, USA, August 2006.
- [3] C. Zakian, I. Pretty, and R. Ellwood, "Near-infrared hyperspectral imaging of teeth for dental caries detection," *Journal of Biomedical Optics*, vol. 14, no. 6, Article ID 064047, 2009.
- [4] S. Li, J. Qiu, X. Yang, H. Liu, D. Wan, and Y. Zhu, "A novel approach to hyperspectral band selection based on spectral shape similarity analysis and fast branch and bound search," *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 241–250, 2014.
- [5] X. Bian, T. Zhang, L. Yan, X. Zhang, H. Fang, and H. Liu, "Spatial-spectral method for classification of hyperspectral images," *Optics Letters*, vol. 38, no. 6, pp. 815–817, 2013.
- [6] Q. Lü, M. J. Tang, J. R. Cai, J. W. Zhao, and S. Vittayapadung, "Vis/NIR hyperspectral imaging for detection of hidden bruises on kiwifruits," *Czech Journal of Food Sciences*, vol. 29, no. 6, pp. 595–602, 2011.
- [7] B. Guo, S. R. Gunn, R. I. Damper, and J. D. B. Nelson, "Band selection for hyperspectral image classification using mutual information," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, pp. 522–526, 2006.
- [8] Q. Du and H. Yang, "Similarity-based unsupervised band selection for hyperspectral image analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 4, pp. 564–568, 2008.
- [9] C.-I. Chang and S. Wang, "Constrained band selection for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 6, pp. 1575–1585, 2006.
- [10] Z. Ji, S. Jia, and L. Shen, "Unsupervised band selection for hyperspectral imagery classification without manual band removal," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 531–543, 2012.
- [11] A. Martínez-Usó, F. Pla, J. M. Sotoca, and P. García-Sevilla, "Clustering-based hyperspectral band selection using information measures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 4158–4171, 2007.
- [12] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.
- [13] A. Bal, M. S. Alam, M. N. Islam, and M. A. Karim, "Hyperspectral target detection using Gaussian filter and post-processing," *Optics and Lasers in Engineering*, vol. 46, no. 11, pp. 817–822, 2008.

Research Article

Fast Adapting Ensemble: A New Algorithm for Mining Data Streams with Concept Drift

Agustín Ortíz Díaz,¹ José del Campo-Ávila,² Gonzalo Ramos-Jiménez,² Isvani Frías Blanco,¹ Yailé Caballero Mota,³ Antonio Mustelier Hechavarría,¹ and Rafael Morales-Bueno²

¹Department of Computer Science, University of Granma, 85100 Granma, Cuba

²Department of Language and Computer Science, University of Málaga, Complejo Tecnológico, 29071 Málaga, Spain

³Department of Computer Science, University of Camagüey, 70100 Camagüey, Cuba

Correspondence should be addressed to Agustín Ortíz Díaz; aortizd@udg.co.cu

Received 26 June 2014; Accepted 15 September 2014

Academic Editor: Shifei Ding

Copyright © 2015 Agustín Ortíz Díaz et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The treatment of large data streams in the presence of concept drifts is one of the main challenges in the field of data mining, particularly when the algorithms have to deal with concepts that disappear and then reappear. This paper presents a new algorithm, called Fast Adapting Ensemble (FAE), which adapts very quickly to both abrupt and gradual concept drifts, and has been specifically designed to deal with recurring concepts. FAE processes the learning examples in blocks of the same size, but it does not have to wait for the batch to be complete in order to adapt its base classification mechanism. FAE incorporates a drift detector to improve the handling of abrupt concept drifts and stores a set of inactive classifiers that represent old concepts, which are activated very quickly when these concepts reappear. We compare our new algorithm with various well-known learning algorithms, taking into account, common benchmark datasets. The experiments show promising results from the proposed algorithm (regarding accuracy and runtime), handling different types of concept drifts.

1. Introduction

Classification algorithms that learn from data streams in the presence of concept drifts have received a lot of attention in recent years. They are very important because of their application in different areas such as bioinformatics, medicine, economics and finance, industry, the environment, and many other fields of application. For instance, Gama et al. [1] have grouped the applications requiring adaptation into four categories: monitoring and control, management and strategic planning, personal assistance and information, and ubiquitous environment applications.

Within incremental learning [2], the problem of classification is generally defined for a sequence (possibly infinite) of examples (also known as instances) $S = e_1, e_2, \dots, e_i, \dots$ arriving over time, normally one at a time and not necessarily time-dependent. Each training example $e_i = (\vec{x}_i, y_i)$ is formed by a vector \vec{x}_i and a discrete value y_i , named label which is

taken from a finite set Y named class. Each vector $\vec{x}_i \in \vec{X}$ has the same dimensions, each dimension is named attribute and each component $x_{i,j}$ is an attribute value (numeric or symbolic). It is assumed that there is an underlying function $y = f(\vec{x}_i)$ and the goal is to obtain a model from S that approximates f as \hat{f} in order to classify or predict the label of nonlabeled examples (also known as observations), so that \hat{f} maximizes the prediction accuracy [3]. Sometimes it is assumed that the examples arrive in batches of the same size. Let us consider concept as the term that refers to the whole distribution of the problem at a certain point in time [4]. This concept can be characterized by the joint distribution $P(\vec{X}, Y)$.

In the real world, concepts are often unstable and change over time. The underlying data distribution may change as well. Often these changes make the model built on old data inconsistent with the new data and an updating of the model is necessary. This problem, known as concept

drift, complicates the task of learning a model from data and requires an additional mechanism in order to maintain the learning model up-to-date with respect to the current concept [5].

According to Tsymbal [5], an ideal concept drift handling system should be able to (1) quickly adapt to concept drift, both abrupt and gradual; (2) be robust to noise and be able to distinguish it from concept drift; and (3) recognize and treat recurring contexts. However, often the mechanisms, used to favor a fast adaptation to concept drifts, like, for example, the use of base classifiers that individually adapt to that change and to their rapid substitution with current classifiers, make the correct treatment of the recurring concepts more difficult.

On the other hand, today's ensemble systems have gained in importance as they provide a mechanism that effectively combines a set of classifiers to obtain not only a more complex but also a more accurate classification model [6].

In this paper, we present Fast Adapting Ensemble (FAE), an algorithm that adapts very quickly to both abrupt and gradual concept drifts and has been specifically built to deal with recurring concepts.

2. Related Work

Gama et al. [7] distinguish two categories in which strategies are positioned to address the problem of concept drift: strategies in which learning adapts at regular time intervals without considering that there has been a change in the concept and strategies in which a concept drift is first detected, and then learning adapts to this change. Ensembles are usually included within the first strategy, as they have mechanisms (to update existing classifiers, to eliminate low-performance classifiers, to insert classifiers, etc.) that allow them to evolve without having to directly detect concept drift. However, recent research proposes different mechanisms of direct detection of changes that are inserted into the ensembles. One advantage of incorporating a drift detector is to exploit the capacity of ensembles to adapt to gradual changes, combined with the natural working mode of the detector during abrupt changes.

2.1. Ensembles for Data Stream Mining. One of the first proposals for data stream mining was the Streaming Ensemble Algorithm (SEA) [8]. SEA divides the training dataset into batches of the same size and a new base classifier is built from each one of these batches and added to the ensemble. The algorithm has a maximum number of classifiers that, when reached as an adaptation mechanism, requires the replacement of previous base classifiers by following certain criteria. To unify the predictions of the base classifiers, SEA uses unweighted-majority voting. SEA adapts to gradual changes well, but its adaptation is not as good for abrupt changes. According to Kolter and Maloof [9], these results are influenced by the voting mechanism used and also because classifiers stop learning once they have been created. An algorithm which follows a similar scheme to SEA is MultiCIDIM-DS, proposed by del Campo-Ávila [6].

Under the same division scheme of the training dataset, Wang et al. [10] proposed a new method, called Accuracy Weighted Ensemble (AWE). To combine the response of base classifiers, the proposal uses a weighted-majority voting. The weighting of the base classifiers depends on the accuracy obtained by them when using the examples from the current training batch. As SEA, it adapts to gradual changes, but it has trouble adapting to abrupt concept drifts. One of the reasons for this inefficiency is that AWE has to wait for the next batch in order to update the weights of base classifiers. Unfortunately, reducing the size of the batch does not solve the problem because that would result in lower overall system accuracy.

The Batch Weighted Ensemble algorithm (BWE) [11] is an ensemble that takes the AWE algorithm as its basic precursor. This proposal is one of those included within the second strategy proposed by Gama et al. [7]; therefore, it incorporates a drift detector inside the model; this detector is called the Batch Drift Detection Method (BDDM) and uses a regression model to determine the presence of concept drift. The drift detector is basically used to determine whether to create a new base classifier due to concept drifts, or whether the concept is stable and the ensemble has not been modified. The idea is to combine the ability of the ensembles to adapt to gradual changes with the natural working mode of the drift detector for detecting abrupt changes.

According to Gonçalves and Barros [12], the Accuracy Updated Ensemble (AUE) [13] is an enhancement of AWE. Both use classifier ensembles and are associated with weights that are updated as data arrive. The main difference between them is the usage of incremental classifiers instead of static ones; it proposes a simpler weighting function to avoid zeroing the weight of all classifiers, a possible situation in AWE, and updates classifiers only if they have been highly accurate in recent data.

Another idea for data stream mining is to use the training examples one by one as they arrive, online. An algorithm that uses this system to update its base classifiers is the Dynamic Weighted Majority (DWM), proposed by Kolter and Maloof [9]. DWM is based on the Weighted Majority Algorithm (WMA) [14], which takes the idea of working with a group of experts, to which an initial weight is automatically assigned. Then, when a new example arrives, the base algorithm receives a prediction from each expert and makes a final decision by combining the predictions and the weights of each expert; finally, if an expert makes an incorrect prediction, then its weight is reduced by a multiplicative constant between 0 and 1. In order to adapt to working with data streams and to handle concept drifts, DWM includes mechanisms to add, update, and delete base classifiers. At each given moment p , a test is performed and a new classifier is added with a weight value equal to 1 if the system output is incorrect; moreover, the system deletes each base classifier, whose weight falls below a threshold of θ . One of the potential problems of this algorithm is that it penalizes base classifiers when they fail but it does not reward them when they are right; this makes the base classifiers' weights fall quickly and they only remain a short while within the ensemble; this,

coupled with the fact that DWM steadily updates the base classifiers, does not make it suitable for the treatment of recurring concepts.

Kolter and Maloof also proposed an algorithm called the Additive Expert Ensemble (AddExp) [15]. This system is very similar to DWM and both have common mechanisms such as the type of voting, the way of inserting new classifiers, and the mechanism, to quickly remove multiple classifiers simultaneously. They differ in the fact that they propose two distinct methods for replacing the classifiers: the first one is based on removing the old ones, for which a constant that controls how long the expert has been within the ensemble is included, and the second one is based on the weakest classifier, as it deletes the one with the lowest weight. Similar to DWM, AddExp inherits the same deficiencies in the treatment of recurring concepts.

With the same work strategy with the data stream, the Ensemble Classification Algorithm for Incremental Data Streams (ICEA) was proposed [16]. The idea of this proposal is that each base classifier learns incrementally, automatically adding the result of their learning as quickly as possible. According to the authors, a faster detection of the concept drift is obtained, when compared to some batch-based algorithms. ICEA uses adapting mechanisms similar to those of DWM. As in DWM, classifiers may be only a short time within the ensemble, which makes it inefficient to handle recurring concepts, this in addition to the fact that base classifiers are steadily readapted, forgetting the old concepts.

The DWM-WIN algorithm [17] proposed some modifications to the DWM algorithm. The first modification is based on a characteristic of the version of the Winnow algorithm implemented by Blum [18]. Winnow is similar to WMA in the idea of changing the weight of the experts according to their individual prediction; the difference is that it includes a new multiplicative constant ($\eta > 1$) to reward the expert weight when the prediction is correct. By adding this feature, DWM-WIN ensures that each expert is more likely to stay within the ensemble if their behavior improves over time; this makes it more flexible when dealing with recurring concepts. Another modification is that, in some variants of the proposed algorithm, when removing experts, their age is taken into account.

A new ensemble for incremental learning, named Diversity for Dealing with Drifts (DDD), is proposed by Minku and Yao [19]. DDD maintains several ensembles with different levels of diversity. If the presence of concept drifts is not detected in the data, the system will consist of two ensembles, one with a low diversity and one with a high diversity. When a concept drift is detected, two new ensembles are built, one with a low diversity and one with a high diversity. According to the authors, old ensembles are maintained because this ensures a better exploitation of diversity, the use of the information learned from old concepts and robustness against false alarms. The four ensembles are maintained while two conditions that check the change status are met; otherwise, using a combination mechanism, a working model with two ensembles starts again. The authors report that DDD is able to maintain a better accuracy than other proposals such as DWM.

Finally, there has been a recent addition to proposals that adapt the well-known algorithms Bagging [20] and Boosting [21] for data stream mining. Using a heuristic and a weighted majority voting, Bagging and Boosting are algorithms which create intermediate models that are the basis for a single final model whose accuracy improves the accuracy of any one of them. According to the Bagging algorithm, the final model is made from the most common rules within several individual models, and according to the Boosting algorithm, multiple classifiers, which are voted according to their error rate, are generated, but unlike the Bagging algorithm, they are not obtained from different samples but rather sequentially on the same training set. Incremental versions of the Bagging and Boosting algorithms have been proposed by Oza and Russell since 2001 [22]. But, other versions that adapt to concept drifts have appeared more recently.

The OzaBagADWIN algorithm proposed by Bifet et al. [23] is a Bagging algorithm adaptation. The idea of this proposal is to add a drift detector called Adaptive Windowing (ADWIN) [24] to the incremental version of the Bagging algorithm [22]. The adaptation mechanism is based on replacing the worst of the classifiers in an instant of time with a new base classifier created more recently.

The Adaptive Boosting Ensemble Classifier (ACS) [25] is an adaptive version of Boosting algorithm proposed by Wankhade and Dongre. This new version uses the Boosting algorithm for an ensemble method combined with an adaptive sliding window and a Hoeffding tree to detect concept drifts, and if necessary, add a new base classifier; this mechanism improves the functioning of the ensemble. According to the author, the algorithm works well in environments with concept drifts, as it adapts dynamically and quickly to changes and it also requires little memory to operate.

Another adaptive version of the Boosting algorithm was proposed by Dongre and Malik [26]. The new adaptation follows a similar idea to ACS but combines the well-known Boosting algorithm with the ADWIN drift detector [24]. The proposal uses the Boosting algorithm as the ensemble method and ADWIN to detect concept drifts and if necessary handle the input data window and add new base classifiers. The results show that the proposed method takes less time, uses less memory, and is more accurate than other known methods (OzaBag, OzaBoost, and OzaBagADWIN).

None of the aforementioned classifiers take into account the possible presence of recurring concepts, so they have not been adapted to work with them.

2.2. Systems for the Treatment of Recurring Concepts. The online learning system should be able to recognize and handle recurring concepts. If a concept has appeared before, previous successful classifiers should be used. Using many classifiers built from old concepts is one possible way to handle recurring concepts.

The Adaptive Classifiers Ensemble (ACE) [27] is a system, published by Nishida et al., which is able to handle recurring concepts better than a conventional system. This ensemble is accompanied by four elements: first, a single classifier that uses the input data one by one incrementally; this classifier

replaces the ensemble for the prediction work when abrupt concept drifts take place because the ensemble takes a long time to update as it has to wait for the next batch to arrive to do so; second, a drift detector; third, a sliding window used to store the results of predictive accuracy and confidence intervals of each classifier on the most recent data, and finally, a buffer used to store recent training examples and to build the new classifiers.

An ensemble especially for the treatment of recurring concepts was presented by Ramamurthy and Bhatnagar [28]. This approach builds a historical global set of classifiers (decision trees) from sequential data chunks of same size. Each individual classifier for this committee represents a different concept. So a new classifier is only built when the concept in the data stream changes and when this concept is not represented by a classifier in the historical global set. These historic classifiers are never deleted because the concept that one represents may reappear. Not all the classifiers participate in the classification process at the same time. The system uses a filter which screens the existing classifiers and allows only those relevant to the current concept to participate in the classification process. This approach, like AWE, has to wait for the next chunk in order to update all the mechanisms of the system.

Although not an ensemble, the algorithm, Recurring Concept Drifts (RCD) [12], is included here because it is able to handle recurring concepts. RCD is not a simple classifier nor an ensemble, but rather a collection of classifiers from which the one to be used is selected at any time based on the distribution of the current data; for this, nonparametric statistical tests are used. A new classifier and a significant sample of the data used to create it are added to the collection each time a new detected concept fails to match any of the previously stored concepts. The authors state that their results are superior to those of other algorithms when faced with abrupt changes and they get similar results when addressing gradual changes.

Finally, we have included two other approaches that are not ensembles but are able handle recurring concepts. Li et al. [29] proposed a classification algorithm called REDLLA for data streams with recurring concept drifts and limited labeled data. It was built for semisupervised learning and it adopts a decision tree as the classification model. When growing a tree, a clustering algorithm based on k-means is installed to produce concept clusters and to label unlabeled data at leaves. In the presence of deviations between historical concept clusters and new ones, potential concept drifts are distinguished and recurring concepts are maintained. According to the authors, REDLLA algorithm is efficient and

effective for mining recurring concept drifts even in cases with a large volume of unlabeled data.

Gama and Kosina [30] present a method that memorizes learnt decision models whenever a concept drift is signaled. The system uses meta-learning techniques that characterize the domain of applicability of previous learnt models. The meta-learner can detect the reoccurrence of contexts and take proactive action by activating previously learnt models. According to the authors, the main benefit of this approach is that the proposed meta-learner is capable of selecting similar historical concepts, if indeed such exist, without the knowledge of true classes of examples.

3. Fast Adapting Ensemble: A New Ensemble Method

As shown in the previous section, there are a few proposals that use ensembles to treat recurring concepts. The use of base classifiers which individually adapt to change and the little time they sometimes remain within the ensemble favor a fast adaptation to concept drifts but make the correct treatment of recurring concepts more difficult.

FAE is an ensemble designed to quickly adapt to concept drifts and specializes in the treatment of recurring concepts. Like RCD [12], this proposal has a set of classifiers that represents several of the concepts analyzed; although it differs in that, these classifiers are organized into active and inactive, according to their behavior when testing current data. FAE is an ensemble that takes its global decision from the partial decision of the active classifiers, while retaining a group of inactive classifiers as a warehouse of old concepts, which ease the treatment of recurring concepts. These inactive classifiers are activated very quickly if the concept that they represent reappears. Reactivation of classifiers and insertion of new updated classifiers, if necessary, ensure rapid adaptation, especially if the concepts are recurring.

Like several of the algorithms analyzed [6, 8, 10, 11], FAE divides the training data stream into blocks of the same size and builds, if necessary, a new base classifier, which adds to the ensemble; thus naturally, it obtains knowledge from large datasets. The algorithm sets a maximum limit of classifiers to store, which, when reached as an adaptation mechanism, requires replacing previous classifiers following certain base criteria.

FAE associates a weight to each base classifier and uses weighted-majority voting to unify the partial votes. Like Wang et al. [10], in order to update the weights, it uses the precision obtained by each of the base classifiers when testing the current training set but differs in that FAE proposes a new formula for adjusting the weights and it also does not have to wait for the new training block to be completed but continues updating the weights of the base classifier with parts of the block. Due to the characteristics of the formula for updating, the weight associated with each base classifier may decrease

$S = e_1, e_2, \dots, e_i, \dots$: Data stream.
 $e_i = (\vec{x}_i, y_i)$: Each training example is formed by a vector \vec{x}_i and a discrete value y_i , named label and taken from a finite set Y named class.
 \vec{bc} : Vector of base classifiers.
 \vec{w} : Vector of weights of base classifiers.
 \vec{status} : Vector of status of base classifiers (Active or inactive).
 $\vec{concept}$: Vector of concepts associated with base classifiers.
 $E = \{\vec{bc}, \vec{w}, \vec{status}, \vec{concept}\}$: Ensemble.
block: Set of examples necessary for building a new base classifier.
t_block: Set of examples necessary for testing the ensemble.
ne: Number of examples necessary for building a new base classifier.
nt: Number of examples necessary for testing the ensemble.

General FAE algorithm

Initialization_ensemble

```

While (next example) // Start the training of the ensemble.
  block  $\leftarrow$  block  $\cup$  {example}
  t_block  $\leftarrow$  t_block  $\cup$  {example}
  i  $\leftarrow$  i + 1
  if (i mod nt = 0) // each nt examples weights and status are updated.
    Update_base_classifier_weight
    Update_base_classifier_status
    t_block  $\leftarrow$   $\emptyset$ 
  end if // End of the update block
  if (i mod ne = 0) //each ne examples creating a new base classifier is analyzed.
    Add_new_base_classifier
    block  $\leftarrow$   $\emptyset$ 
  end if
end while
  
```

ALGORITHM 1

```

concept: current concept.
Initialization_ensemble
concept  $\leftarrow$  1
block  $\leftarrow$   $\emptyset$ 
For i = 1, ..., ne
  next example
  block  $\leftarrow$  block  $\cup$  {example}
end for
bc1  $\leftarrow$  build_base_classifier (block) // a new base classifier is built
w1  $\leftarrow$  1
status1  $\leftarrow$  active
concept1  $\leftarrow$  concept
block  $\leftarrow$   $\emptyset$ 
t_block  $\leftarrow$   $\emptyset$ 
i  $\leftarrow$  0
  
```

ALGORITHM 2

or increase depending on its behavior when testing the new data.

This proposal is included in the second strategy mentioned by Gama et al. [7] because it incorporates a drift detector to the model. As discussed by Deckert [11], the drift detector is used to determine when to create a new base

classifier according to the presence or absence of concept drifts; if the concept is stable, an unnecessary new classifier is not created, which contributes to saving memory and favors previous base classifiers representing other concepts remaining within the ensemble. The purpose of this idea is to take advantage of the capacity of ensembles to adapt to gradual changes combined with the natural work of the

drift detector for abrupt changes. Because of this, FAE is able to manipulate both gradual and abrupt concept drifts (see Algorithm 1).

3.1. Initialization of the Ensemble. For the ensemble to be functional, though, of course, not properly trained, it is necessary to ensure that there is at least one active base classifier. For this reason, the initial step is to create, with the first block of training, a base classifier with its active status and initial weight equal to 1. In addition, the first concept to be analyzed is initialized (see Algorithm 2).

3.2. Update the Weights and Status of the Base Classifiers. The weights and status of base classifiers are updated each nt period. The value nt is the number of examples needed to update the weights and status of base classifiers; this value should be less than that defined for a set of examples (number of examples needed to create a new base classifier), in order not to wait unnecessarily for a block to be completed to update the base classifier weight. This is one of the shortcomings of the algorithm proposed by Wang et al. [10], which is why it was difficult to detect abrupt concept drifts.

The formula used to update the weights is inspired by studies in disciplines such as telecommunications [31], specifically formulas for smoothing to calculate a stable measure of the usability of communication lines. This way of updating the weights of the classifiers allows them to be increased or decreased according to the behavior of the classifiers when testing the current training set. It is intended that the base classifiers can remain longer within the ensemble.

Preset constants β_1 and β_2 ($\beta_1 + \beta_2 = 1$) represent the level of importance they have given to the behavior of base classifiers over old data and current data, respectively. A high value of β_1 (compared to the value of β_2) means that more importance will be given to the historical behavior of the classifier than to its behavior over current data; change adaptation will be a little slower but the process will be more robust over noisy data. A high value of β_2 (compared to the value of β_1) means that more importance will be given to the current behavior of base classifier than to its historical behavior; change adaptation will be much faster but it is likely to be affected by noisy data. Hence, the importance of assigning values to β_1 and β_2 is directly related to the balance desired between sensibility to concept drifts or noisy data (see Algorithm 3).

It is important to note the fact that inactive classifier weights are not decreased; they are only increased if current classifier behavior improves. The purpose of this procedure is not to unnecessarily reduce the weight of a classifier, about which it is known that it has not been identified with the current concept (for this reason it is inactive), and thus rapid activation is ensured when the concept that it represents appears again.

The base classifier can have two statuses, active or inactive. An active classifier is one that keeps its weight above a preset threshold θ . For predicting in an instant of time,

only active classifiers are used, as they are considered the best adapted to the current concept.

An inactive classifier is one that keeps its weight below the preset threshold θ . Inactive classifiers are not involved in predictions but remain stored as long as possible, as they represent old concepts. The weights of inactive classifiers are also updated every nt examples (only if it is improved) (see Algorithm 4).

Whenever weights of base classifiers are updated; afterwards, statuses are updated, too; thus the activation-inactivation of the classifiers in the ensemble is ensured.

There are two implementation details not reflected in the pseudocode: first, when updating the statuses of base classifiers, it is always ensured that at least one classifier remains in the active status and also that it has the best current behavior (highest weight); second, in the experiments, a somewhat higher value than the threshold θ is used to activate a base classifier; with it, subsequent changes of activation-inactivation or vice versa, which are annoying and harmful to predictions, decrease.

3.3. Adding a New Base Classifier. The first thing to be considered in order to add a new base classifier is the information provided by the drift detector used. The drift detector must have as output three possible alerts: no change, possible change (warning), and change (drift). The known drift detectors DDM (drift detection method) [32] proposed by Gama et al. and EDDM (early drift detection method) [33] proposed by Baena et al. have these features. A new base classifier is only created with the last two alerts (warning or drift); with the “possibly change” alert, the new base classifier is associated with the current concept and, with the “change” alert, it is associated with a new concept. If the alert is “no change,” the ensemble remains unchanged.

The drift detector used was DDM. This approach detects changes in the probability distribution of examples. The main idea of this method is to monitor the error-rate produced by a classifier. Statistical theory states that error decreases if the distribution is stable. When the error increases, it means that the distribution has changed [32].

This procedure ensures that new classifiers are only added when necessary; thus, within the ensemble, old concepts remain longer, which allows for a better treatment of recurring concepts (see Algorithm 5).

3.4. Deleting a Base Classifier. The algorithm deletes a base classifier when a new classifier is to be added and the maximum limit of classifiers to store has been reached. The removal process takes into account the following aspects: status of classifiers (active-inactive), age and weight of the classifiers, and the number of classifiers associated with a concept. It is always about first deleting an inactive classifier; in its absence (all classifiers are active), it proceeds to remove an active classifier. To avoid a cumbersome pseudocode, explanations are included (see Algorithm 6).

Option 1. The oldest inactive classifier that belongs to a concept which has more than one base classifier associated

```

n: Number of base classifiers in the ensemble in each moment.
 $\beta_1, \beta_2$ : Factors to adjust the weights associated with base classifiers ( $\beta_1 < 0$ ;  $\beta_2 < 0$ ;  $\beta_1 + \beta_2 = 1$ ).
Update_base_classifier_weight
for  $j \leftarrow 1, \dots, n$ 
   $w \leftarrow w_j * \beta_1 + accuracy(bc_j, t\_block) * \beta_2$ 
  if ((statusj = active) or ( $w > w_j$ ))
     $w_j \leftarrow w$ 
  end if
end for

```

ALGORITHM 3

```

 $\theta$ : Threshold used to delete base classifiers from the ensemble.
Update_base_classifier_status
for  $j \leftarrow 1, \dots, n$ 
  if ( $w_j > \theta$ )
    statusj  $\leftarrow active$ 
  else
    statusj  $\leftarrow inactive$ 
  end if
end for

```

ALGORITHM 4

```

max: maximum limit of classifiers to store in the ensemble.
n: Number of base classifiers in the ensemble in each moment.
Add_new_basic_classifier
alert  $\leftarrow Drift\_detector()$ 
if ((alert = "warning") or (alert = "drift"))
  if ( $n = max$ ) // fullensemble
    delete_base_classifier
     $n \leftarrow n - 1$ 
  end if
   $n \leftarrow n + 1$ 
   $cb_n \leftarrow build\_base\_classifier(block)$  // a new classifier is built.
   $w_n \leftarrow 1$ 
  statusn  $\leftarrow active$ 
  if (alert = "drift")
    concept  $\leftarrow concept + 1$ 
  end if
  conceptn  $\leftarrow concept$ 
end if

```

ALGORITHM 5

```

Delete_base_classifier.
if (there are inactive classifiers)
  Delete an inactive classifiers // Option 1.
else if (there are no inactive classifiers)
  Delete an active classifiers // Option 2.

```

ALGORITHM 6

with it is deleted. If all existing concepts are associated with a single base classifier, the oldest inactive classifier is removed, and, with it, the concept itself is also deleted.

Option 2. Active classifier with less weight is removed.

When a base classifier is deleted from the ensemble, all of its associated values (weight, status, and concept) are also deleted.

It is always about deleting the oldest classifier but taking into account the highest number of concepts remaining represented within the ensemble. Often the oldest classifier is not deleted, in order to keep a classifier, which is the only representative of a concept within ensemble.

One purpose of this procedure of deleting base classifiers is to maintain representation of all the concepts analyzed, within the ensemble, for as long as possible. This approach favors the treatment of recurring concepts and maintains a high diversity of concepts within the ensemble.

To identify the best configuration with which to test the ensemble, different values of the following parameters were used: β_1 , β_2 , θ , ne, nt, and max.

To identify the best parameter set, approximately 500 different configurations were tested for these parameters. The best and more robust configuration found was $\beta_1 = 0.5$; $\beta_2 = 0.5$; ne = 500; nt = 50, and max = 15.

3.5. Analyzing Spatial and Temporal Complexity. In the context of machine learning, it is important to analyze the time and space complexity of the algorithms. This study is even more necessary when the learning process is done from data streams, online, with the possibility of having nonending datasets.

At this point, now the FAE algorithm has been described and we present a detailed analysis of this complexity. We must note that the algorithm can be configured with different base classifiers, so the final details about complexity will depend on the final base classifier used. In this case, the common configuration that we propose uses the Hoeffding Tree or VFDT [34] and the analysis of complexity is done under this assumption (additional studies for different base classifiers can be easily derived).

Spatial complexity is basically determined by the maximum number of base classifiers stored in the ensemble (max) and their maximum size. In our case, we have used decision trees, so the maximum size for such a model is a completely expanded decision tree. If we assume that the dataset is defined by a finite number of attributes (n_{attr}) with a maximum number of symbolic values (n_{values}), the spatial complexity is $O(\max \cdot n_{attr}^{n_{values}})$, which is polynomial. This reasoning is extensible to numerical attributes. In worst case, there will be as many different values as different examples in the set used to build the base classifier (ne). In addition, it is easy to have fewer values because discretization methods can be applied [35]. Clearly, this is the

worst case. In general, the amount of space needed is much lower.

Temporal complexity, in this context, cannot be studied for the whole process, because it is continuous. It is usual to study the processing time per example or, in our case, per block of examples (ne or nt). Therefore, the analysis can be done according to two situations: building new base classifiers or updating them. In the first case, the temporal complexity depends on the temporal complexity for the selected base classifier. In our case, we have configured FAE to use VFDT, so it requires constant time to process each example [34] (which will depend on the size of a completely expanded tree), that is, $O(ne \cdot n_{attr}^{n_{values}})$. In the second case, the updating process, each example in the testing block (with size nb) is tested with each base classifier in order to update all the weights and statuses. In the worst case scenario, the classifiers are completely expanded decision trees (whose branches have as nodes as attributes, n_{attr}), so the temporal complexity is $O(nb \cdot \max \cdot n_{attr})$.

4. Experimental Results

In this section, we present the algorithms used in the tests, parameters, information about the datasets, and an empirical study of the results obtained.

The proposal presented here was implemented using the Massive Online Analysis (MOA) framework [36], developed at Waikato University, New Zealand. MOA is a framework for mining data streams. It offers a collection of machine learning algorithms, evaluation tools, and dataset generators commonly used in data stream research.

Experiments were performed on a computer using an Intel, Pentium CPU, P6000 1.87 GHz with a RAM of 4 GB.

Algorithms. In the experiments, we used the following algorithms: AWE (Wang et al., 2003), AUE (Brzezinski and Stefanowski, 2011), OzaBagAdwin (Oza and Russell, 2009), and decision trees with Hoeffding bounds or Very Fast Decision Tree (Hoeffding Tree or VFDT, Domingos and Hulten, 2000); all the algorithms have freely available implementations in the MOA framework.

The base classifier used in the experiments for all algorithms (ensembles) was a Hoeffding Tree or VFDT [34]. So we can exclude the influence of the base classifier in the comparison between ensembles. It is also important to compare the Hoeffding Tree with the rest of the algorithms to verify whether they improve it or not.

The parameters used for each of the algorithms (AWE, AUE, OzaBagAdwin, and Hoeffding Tree) in the experiments are the default values defined in the MOA framework.

Artificial Datasets. We selected two artificial datasets to perform the experiments: LED, proposed by Breiman et al. in 1984 and SEA proposed by Nick Street and Kim in 2001. They are commonly used in the concept drift research area [1] and are freely available from the MOA framework.

The LED dataset is composed of 24 categorical attributes; 17 of which are irrelevant, and one categorical class with ten

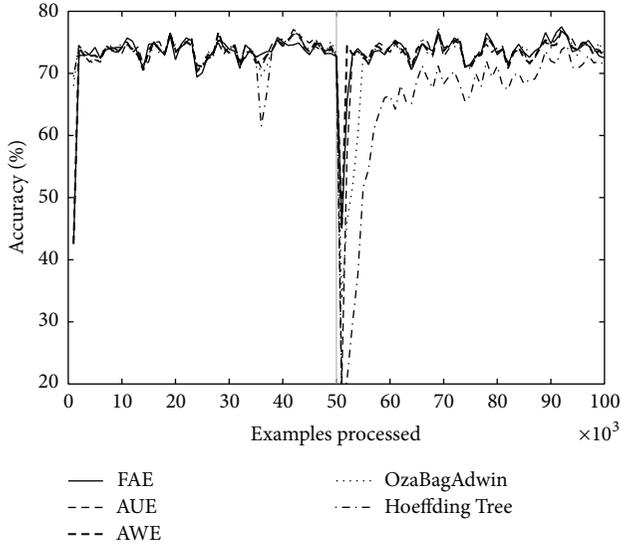


FIGURE 1: LED concept, 100 000 examples. One change point: $t_0 = 50000$, $w = 0$.

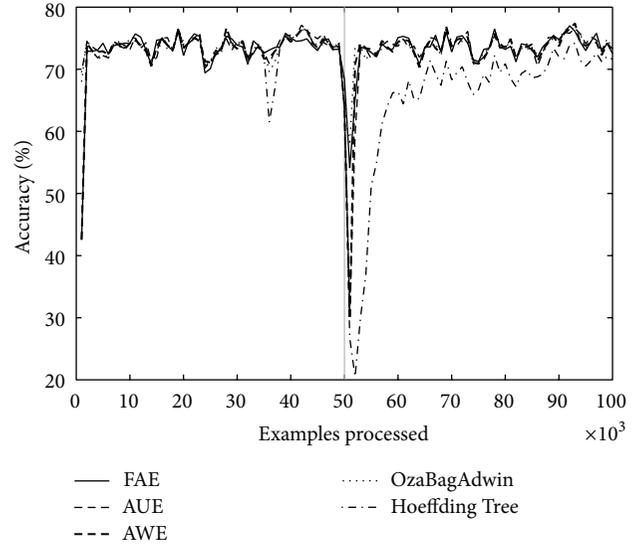


FIGURE 2: LED concept, 100 000 examples. One change point: $t_0 = 50000$, $w = 1000$.

possible values. The goal is to predict the digit displayed on a seven-segment LED display, where each attribute has 10% probability of being inverted (noise) [36]. We used a version of LED available at MOA that includes concept drifts in the datasets by simply changing the attribute positions.

The SEA dataset is generated using three attributes, where only the first two are relevant. All three attributes have values between 0 and 10. The points of the dataset are divided into 4 different concepts. The classification is done using $f_1 + f_2 \leq \alpha$, where f_1 and f_2 represent the first two attributes and α is a threshold value [36]. The most frequent values of α are 9, 8, 7, and 9.5. We also used a version of SEA concept available at MOA. Table 1 shows general characteristics of LED and SEA datasets.

To create abrupt or gradual concept drifts in data stream, the MOA framework uses a sigmoid function, as a practical solution for defining the probability that each new example of the stream will belong to the new concept after the drift. In this sigmoid model, only two parameters need to be specified: t_0 , the point of change, and w , the length of change [36] (number of examples in the transition between concepts).

4.1. Abrupt and Gradual Change. In a first phase of the experiments to test the behavior of the algorithms under consideration, over different concept drifts, the following schemas were used: 100000 examples, half of the examples of the first concept and the second concept of a remainder ($t_0 = 50000$). The length of change (w) takes four possible values 0, 100, 500, and 1000 to simulate an abrupt change from ($w = 0$) to more gradual changes ($w = 1000$).

Table 2 shows the results of testing on the LED dataset. The first 50000 examples were generated with a number of drifting attributes equal to 1 and the other 50000 with a number of drifting attributes equal to 7.

Table 2 shows that the accuracy significantly reduced around the change point (transition between concepts). FAE always reported results between the two best accuracy values taken around the change point for all values of w (0, 100, 500, and 1000). The same applies to the other two results, final accuracy and time.

Figures 1 and 2 correspond to the results of Table 2 ($w = 0$, columns A and $w = 1000$, columns D). We can see that the graphs show accuracy falls around the change point. We consider it important to draw attention to the depth and width of each of these falls; depth indicates by how much accuracy falls for each algorithm around the change point and the width indicates how long it takes to recover. FAE reports results with a low loss of accuracy both in plotted results and in the rest of the experiments (not plotted) and a low recovery time compared to the other algorithms.

Very similar results to those described above occur when testing on data generated according to SEA concept. The first 50000 examples were generated with the first classification function ($f_1 + f_2 \leq 9$) and the others with the fourth classification function ($f_1 + f_2 \leq 9.5$) (see Table 3).

Over both datasets, LED and SEA, and over different types of changes, abrupt and gradual ones, FAE shows promising results with regard to accuracy fall depth around the change point, recover time from accuracy fall (see Figures 1 and 2), final accuracy, and runtime.

4.2. Recurring Concept Drift. In the second phase of the experiments to test the behavior of the algorithms over recurring concepts, we built 8 datasets, each one with 100000 examples and in the presence of recurring concepts.

Dataset 1 and 2. LED concept, three change points, every 25000 examples, we change the number of attributes with

TABLE 1: General characteristics of LED and SEA datasets.

Dataset	Attribute number	Relevant attribute	Irrelevant attribute	Total sample number
LED	24	7	17	100 000
SEA	3	2	1	100 000

TABLE 2: LED concept, 100 000 examples. One change point: $t_0 = 50 000$. Columns A, $w = 0$; columns B, $w = 100$; columns C, $w = 500$; and columns D, $w = 1000$.

Classifiers	Lowest accuracy around change point (%)				Final accuracy (%)				Time (s)			
	A	B	C	D	A	B	C	D	A	B	C	D
FAE	45,1	50,9	55,1	54,1	73,15	73,2	73,16	73,15	63,01	64,55	66,75	68,09
AUE	19,9	15,6	26,6	31	72,84	72,99	72,95	72,89	128,12	118,81	109,25	125,49
AWE	45,2	44,7	27,5	29,7	73,13	73,12	72,97	72,87	234,52	238,35	230,43	204,13
OzaBagAdwin	33,6	61,6	61,1	57,8	72,68	73,53	73,48	73,44	101,57	102,13	101,43	102,68
Hoeffding Tree	16,9	18	21,7	26,7	69,24	69,24	69,26	69,24	9,05	8,95	8,36	9,05

drift (we follow the scheme 1, 7, 1, 7; number of attributes with drift). Dataset 1, $w = 0$, and dataset 2, $w = 1000$.

Dataset 3 and 4. SEA concept, three change points, every 25000 examples, we change the threshold value α (we follow the scheme 9,5, 9, 9,5, 9 threshold value). Dataset 3, $w = 0$, and dataset 4, $w = 1000$.

Dataset 5 and 6. LED concept, seven change points, every 12500 examples, we change the number of attributes with drift (we follow the scheme 1, 3, 5, 7, 1, 3, 5, 7; number of attributes with drift). Dataset 5, $w = 0$, and dataset 6, $w = 1000$.

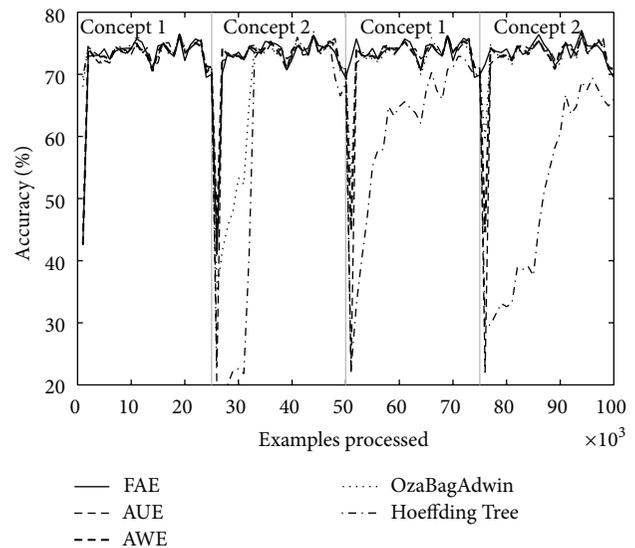
Dataset 7 and 8. SEA concept, seven change points, every 12500 examples, we change the threshold value α (we follow the scheme 7, 8, 9,5, 9, 7, 8, 9,5, 9 threshold value). Dataset 7, $w = 0$, and dataset 8, $w = 1000$.

Tables 4 and 5 show the results of evaluating each of the algorithms over the eight datasets defined above. Table 4 shows the results when there are four concepts and Table 5 when there are eight. Each table comes with four figures (Table 4 with Figures 3, 4, 5, and 6; Table 5 with Figures 7, 8, 9, and 10) which plot accuracy values as functions of processed examples.

Labels, concept 1, concept 2..., were added only to the figures in order to visually highlight when a concept is recurrent. The same concept is labeled with the same label.

According to the results shown in Tables 4 and 5, it is interesting to note the following.

The Hoeffding Tree algorithm (it is not an ensemble) always achieves better results than other algorithms in terms of runtime; however, it gets the worst final accuracy values in all cases. By contrast, the algorithm AWE has the worst results regarding runtime in all cases; however, its final accuracy values are comparably good and are included among the best two results in several experiments. The algorithms AUE and OzaBagAdwin achieve comparably good final accuracy values too. The algorithm OzaBagAdwin achieves values very similar to those of FAE algorithm in terms of runtime, although with lower results.

FIGURE 3: Dataset 1. LED concept, three change points: every 25 000 examples, $w = 0$.

As seen in Tables 4 and 5, FAE always reported results between the two best accuracy and runtime values. These values show that the new approach achieves better results than the rest of the algorithms over the datasets with the proposed features (concept drifts and recurring concepts).

In each figure, we consider it important to draw attention to the second half starting from the 50000th example, when previously analyzed concepts reappear (recurring concepts). We can see that FAE shows practically no falls in the accuracy values compared to the rest of the algorithms. Differences are more notable over the LED dataset. FAE is an algorithm built to deal with recurring concepts and this is precisely what the results show. According to the results of the experiments, FAE is able to handle abrupt and gradual concept drifts; moreover, it is far superior to the rest of the algorithms in the treatment of recurring concepts.

TABLE 3: SEA concept, 100 000 examples. One change point: $t_0 = 50\,000$. Columns A, $w = 0$; columns B, $w = 100$; columns C, $w = 500$; and columns D, $w = 1000$.

Classifiers	Lowest accuracy around change point (%)				Final accuracy (%)				Time (s)			
	A	B	C	D	A	B	C	D	A	B	C	D
FAE	78,3	79,5	79,6	80,5	88,1	88,14	88,14	88,07	8,19	8,17	8,03	8,05
AUE	78,2	78,6	79,3	79,5	88,11	88,15	88,32	88,09	15,33	14,76	14,57	15,46
AWE	80,7	80,8	81,7	82	87,69	87,68	87,73	87,72	31,04	29,89	26,64	27,85
OzaBagAdwin	78,5	78,8	79,2	79,2	87,99	88,07	87,83	88,25	12,73	13,68	13,88	10,9
Hoeffding Tree	78,1	78,5	79	79,1	86,81	86,8	86,8	86,8	1,08	1,12	1,09	1,09

TABLE 4: LED and SEA concepts, 100 000 examples. Three change points: every 25 000 examples. Columns A, $w = 0$ and columns B, $w = 1000$.

Classifiers	LED				SEA			
	Final accuracy (%)		Time (s)		Final accuracy (%)		Time (s)	
	A	B	A	B	A	B	A	B
FAE	72,65	72,36	83,9	82,98	87,52	87,48	14,6	11,72
AUE	71,82	71,79	142,23	143,58	87,51	87,23	15,12	15,23
AWE	72,43	71,52	233,14	239,77	87,3	87,17	31	30,84
OzaBagAdwin	71,39	72,6	99,23	84,26	86,97	87,21	13,29	13,42
Hoeffding Tree	61,22	61,79	8,07	5,32	85,61	85,6	1,05	1,15

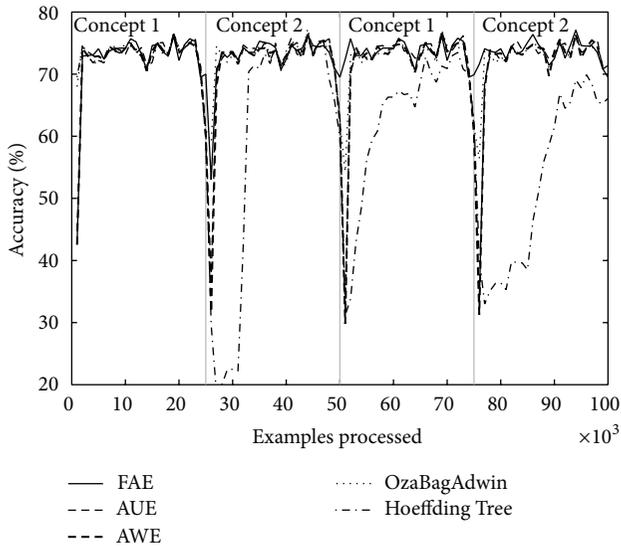


FIGURE 4: Dataset 2. LED concept, three change points: every 25 000 examples, $w = 1000$.

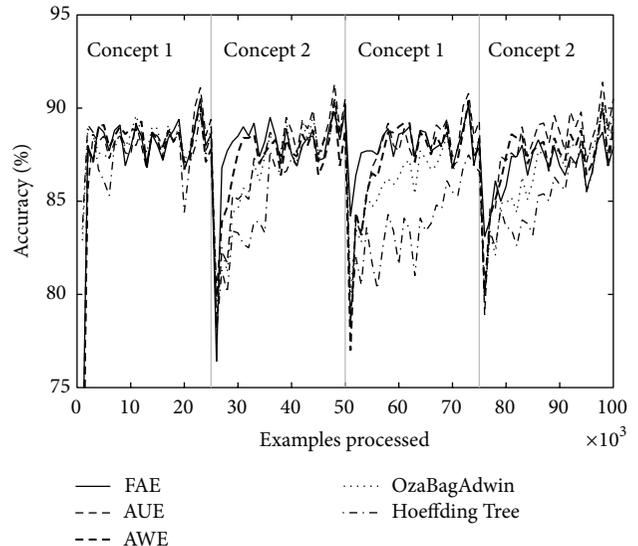


FIGURE 5: Dataset 3. SEA concept, three change points: every 25 000 examples, $w = 0$.

4.3. *Real Dataset.* The real world dataset we work with in this section has been used in several studies about concept drift [37]. For this dataset, there is no strong claim about any presence or type of change. In this dataset, we evaluate the algorithms by processing the examples in their temporal order.

Electricity dataset (Elec2) was described by Harries and analyzed by Gama. This dataset 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 Elec2 dataset contains 45,312 examples. The class label identifies the change of the price relative to a moving average of the last 24 hours.

As seen in Table 6 and in Figure 11, FAE reported results between the two best accuracy values again. However, it is important to note that the OzaBagAdwin algorithm achieved the best results over Electricity dataset.

TABLE 5: LED dataset, 100 000 examples. Seven change points: every 12 500 examples. Columns A, $w = 0$ and columns B, $w = 1000$.

Classifiers	LED				SEA			
	Final accuracy (%)		Time (s)		Final accuracy (%)		Time (s)	
	A	B	A	B	A	B	A	B
FAE	72,27	71,95	74,69	81,32	87,46	86,95	11,37	10,67
AUE	71,52	71,59	133,54	120,84	85,99	86,4	15,09	14,96
AWE	70,38	70,06	236,58	210,79	86,58	86,5	30,62	30,17
OzaBagAdwin	71,68	72,08	86,67	91,37	86,13	86,31	12,99	13,23
Hoeffding Tree	62,65	62,71	8,5	7,92	84,31	84,35	1,09	1,19

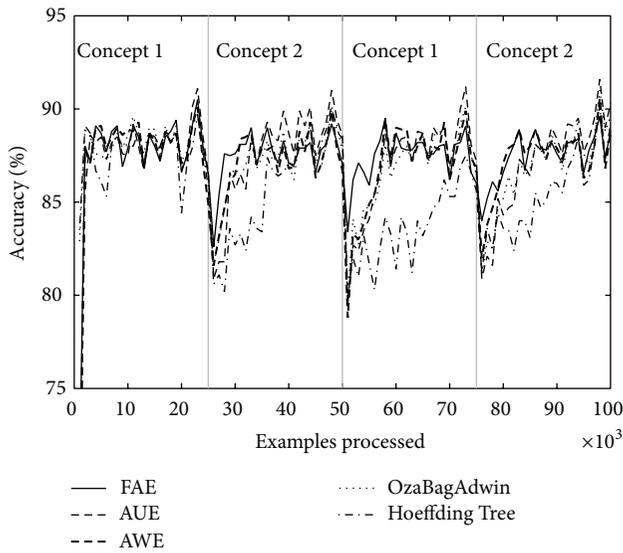


FIGURE 6: Dataset 4. SEA concept, three change points: every 25 000 examples, $w = 1000$.

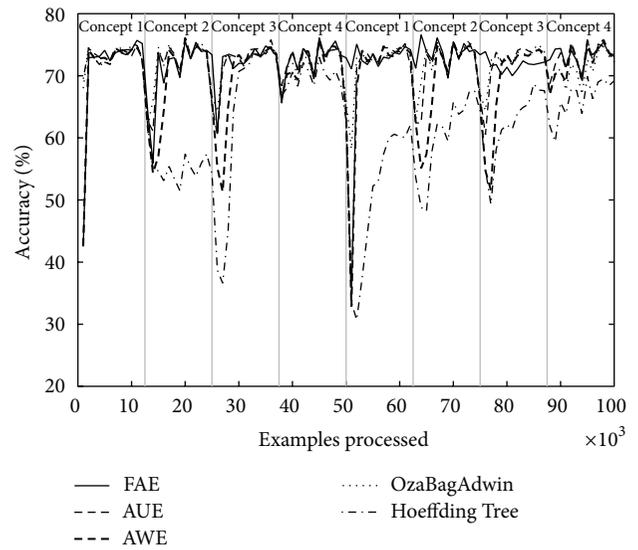


FIGURE 8: Dataset 6. LED concept, seven change points: every 12 500 examples, $w = 1000$.

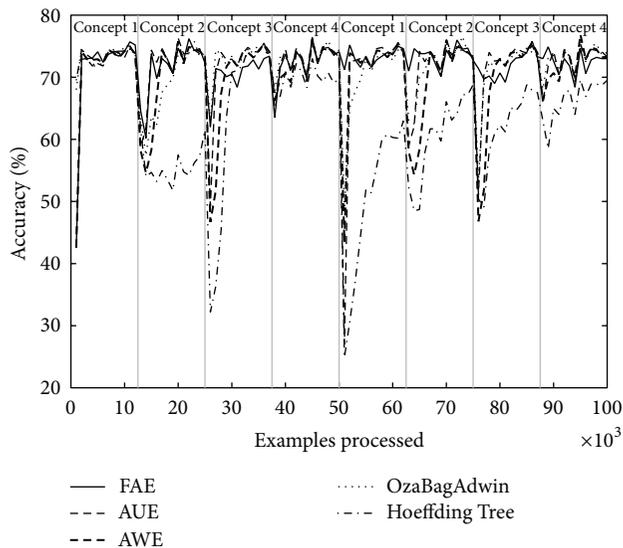


FIGURE 7: Dataset 5. LED concept, seven change points: every 12 500 examples, $w = 0$.

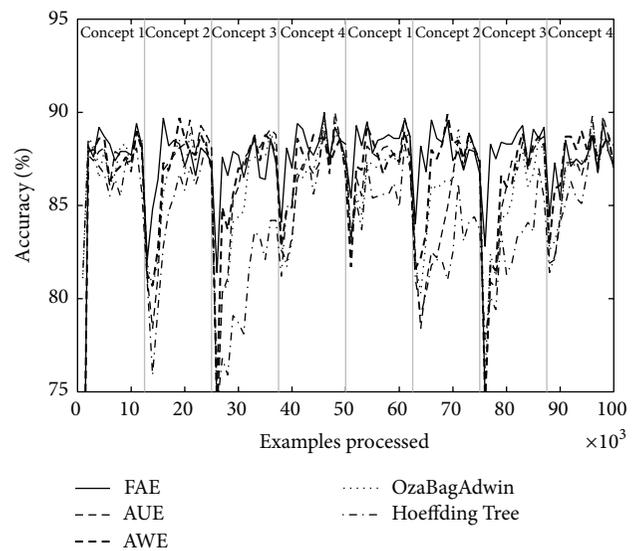


FIGURE 9: Dataset 7. SEA concept, seven change points: every 12 500 examples, $w = 0$.

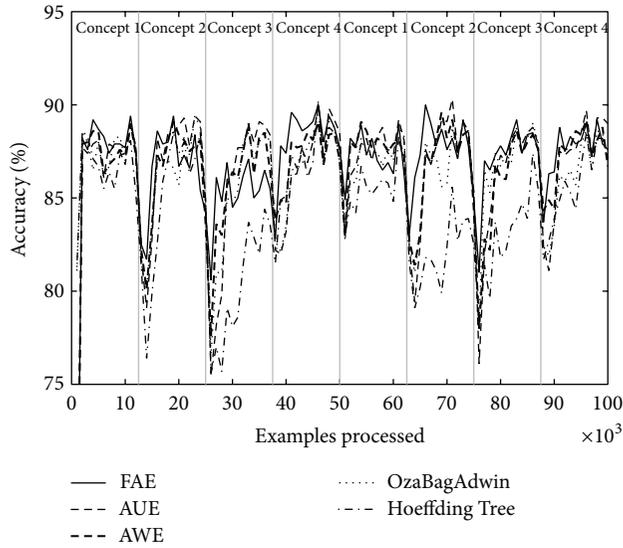


FIGURE 10: Dataset 8. SEA concept, seven change points: every 12 500 examples, $w = 1000$.

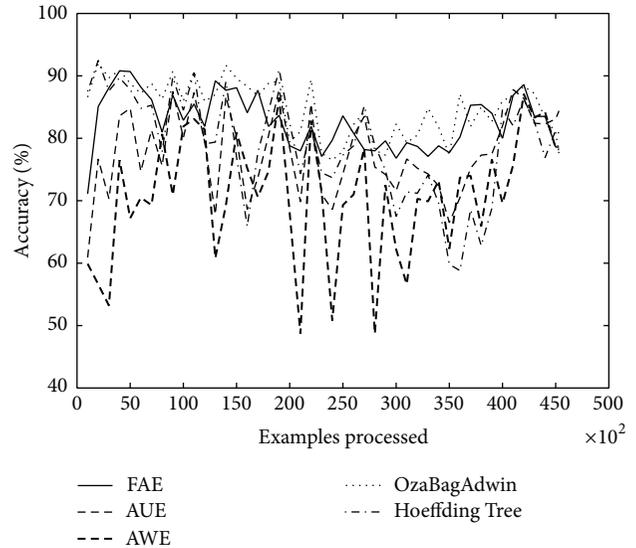


FIGURE 11: Electricity dataset, 45 312 examples.

TABLE 6: Electricity dataset, 45 312 examples.

Classifiers	Electricity (elec2)	
	Final accuracy (%)	Time (s)
FAE	82,4	9,34
AUE	77,7	10,62
AWE	71,13	12,89
OzaBagAdwin	84,81	7,77
Hoeffding Tree	78,92	1,53

5. Conclusions

The treatment of large data streams in the presence of concept drifts is one of the main challenges in the data mining area, specifically when the algorithms have to deal with concepts that disappear and then reappear. Most algorithms concentrate their efforts on the current data, by deleting or modifying previously constructed models that represent concepts that have disappeared. Many times when these concepts reappear, algorithms have to repeat work already done. This paper has presented FAE, an algorithm that adapts very quickly to both abrupt and gradual concept drifts, and has been specifically built to deal with recurring concept drifts.

FAE stores a set of inactive base classifiers (while, in this status, they are not used for prediction) which represent old concepts that were analyzed and then disappeared. These classifiers change to active status very quickly when the concept that they represent reappears.

FAE uses a drift detector (often DDM is used) to decide when to build and add a new base classifier. This mechanism allows adding new base classifiers only when necessary, thus contributing to saving memory which is used to keep other models. Using a drift detector favors the treatment of abrupt concept drifts, which is combined with the natural treatment of ensembles for gradual concept drifts.

FAE uses a weighted majority vote to obtain the global ensemble decision and proposes a formula for adjusting the weights of the base classifiers that allows the algorithm to increase or decrease them in relation to their actual performance. This mechanism allows the base classifiers to remain longer within the ensemble.

The experiments carried out show promising results of the proposed algorithm over datasets generated according to LED and SEA concepts and the real world dataset. Both abrupt and gradual concept drifts as well as the existence of recurring concepts have been simulated.

Conflict of Interests

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

References

- [1] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, article 44, 2014.
- [2] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, no. 1, pp. 69–101, 1996.
- [3] F. J. Ferrer-Troyano, J. S. Aguilar-Ruiz, and J. C. Riquelme, "Incremental rule learning and border examples selection from numerical data streams," *Journal of Universal Computer Science*, vol. 11, no. 8, pp. 1426–1439, 2005.
- [4] L. L. Minku, A. P. White, and X. Yao, "The impact of diversity on online ensemble learning in the presence of concept drift," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 730–742, 2010.
- [5] A. Tsymbal, "The problem of concept drift: definitions and related work," Tech. Rep. TCD-CS-2004-15, Department of Computer Science, Trinity College, Dublin, Ireland, 2004.

- [6] J. del Campo-Ávila, *Nuevos Enfoques en Aprendizaje Incremental [Ph.D. thesis]*, Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, 2007.
- [7] J. Gama, P. Medas, G. Castillo, and P. Rodríguez, "Learning with drift detection," in *Proceedings of the 17th SBIA Brazilian Symposium on Artificial Intelligence*, pp. 286–295, Sao Luis, Brazil, September–October, 2004.
- [8] W. Nick Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," in *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)*, pp. 377–382, New York, NY, USA, August 2001.
- [9] J. Kolter and M. Maloof, "Dynamic weighted majority: a new ensemble method for tracking concept drift," in *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM '03)*, pp. 123–130, Melbourne, Australia, November 2003.
- [10] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pp. 226–235, Washington, DC, USA, August 2003.
- [11] M. Deckert, *Batch Weighted Ensemble for Mining Data Streams with Concept Drift*, Springer, Berlin, Germany, 2011.
- [12] P. Gonçalves and R. Barros, *RCD: A Recurring Concept Drift Framework*, Centro de Informática, Universidad Federal de Pernambuco, Ciudad Universitaria, Recife, Brasil, 2013.
- [13] D. Brzezinski and J. Stefanowski, "Accuracy updated ensemble for data streams with concept drift," in *Hybrid Artificial Intelligent Systems*, E. Corchado, M. Kurzynski, and M. Wozniak, Eds., vol. 6679 of *Lecture Notes in Computer Science*, pp. 155–163, Springer, Berlin, Germany, 2011.
- [14] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Information and Computation*, vol. 108, no. 2, pp. 212–261, 1994.
- [15] J. Z. Kolter and M. A. Maloof, "Using additive expert ensembles to cope with concept drift," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, pp. 449–456, August 2005.
- [16] S. Yue, M. Guojun, L. Xu, and L. Chunnian, "Mining concept drifts from data streams based on multiclassifiers," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW '07)*, Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Beijing, China, 2007.
- [17] D. Mejri, R. Khanchel, and M. Limam, "An ensemble method for concept drift in nonstationary environment," *Journal of Statistical Computation and Simulation*, vol. 83, no. 6, pp. 1115–1128, 2013.
- [18] A. Blum, "Empirical support for winnow and weighted-majority algorithms: results on a calendar scheduling domain," *Machine Learning*, vol. 26, no. 1, pp. 5–23, 1997.
- [19] L. L. Minku and X. Yao, "DDD: a new ensemble approach for dealing with concept drift," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 4, pp. 619–633, 2012.
- [20] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [21] Y. Freund, "Boosting a weak learning algorithm by majority," in *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, 1990.
- [22] N. Oza and S. Russell, "Online bagging and boosting," in *Artificial Intelligence and Statistics 2001*, pp. 105–112, Morgan Kaufmann, 2001.
- [23] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 139–147, Paris, France, July 2009.
- [24] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 7th SIAM International Conference on Data Mining*, pp. 443–448, April 2007.
- [25] K. K. Wankhade and S. S. Dongre, "A new adaptive ensemble boosting classifier for concept drifting stream data," *International Journal of Modeling and Optimization*, vol. 2, no. 4, pp. 493–497, 2012.
- [26] S. Dongre and L. Malik, "Algorithm to handle concept drifting in data stream mining," *IJCSN International Journal of Computer Science and Network*, vol. 2, no. 1, 2013.
- [27] K. Nishida, K. Yamauchi, and T. Omori, "ACE: adaptive classifiers-ensemble system for concept-drifting environments," in *Multiple Classifier Systems*, vol. 3541 of *Lecture Notes in Computer Science*, pp. 176–185, Springer, Heidelberg, Germany, 2005.
- [28] S. Ramamurthy and R. Bhatnagar, "Tracking recurrent concept drift in streaming data using ensemble classifiers," in *Proceedings of the 6th International Conference on Machine Learning and Applications (ICMLA '07)*, pp. 404–409, December 2007.
- [29] P. Li, X. Wu, and X. Hu, "Mining recurring concept drifts with limited labeled streaming data," *Journal of Machine Learning Research—Proceedings Track*, pp. 241–252, 2010.
- [30] J. Gama and P. Kosina, *Tracking Recurring Concepts with Meta-Learners*, Springer, Berlin, Germany, 2009.
- [31] A. Tanenbaum, *Computer Networks*, Prentice-Hall, 2nd edition, 1988.
- [32] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Proceedings of the SBIA Brazilian Symposium on Artificial Intelligence*, pp. 286–295, 2004.
- [33] M. Baena, J. del Campo, R. Fidalgo, A. Bifet, R. Gavaldà, and R. M. Bueno, "Early drift detection method," in *Proceedings of the 4th International Workshop on Knowledge Discovery from Data Streams*, 2006.
- [34] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '00)*, pp. 71–80, August 2000.
- [35] L. I. Mora, I. Fortes, R. Morales-Bueno, and F. Triguero, "Dynamic discretization of continuous values from time series," in *Book "ECML'00"*, pp. 280–291, 2000.
- [36] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: massive online analysis," *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010.
- [37] J. Bártolo, *Learning recurring concepts from data stream in ubiquitous environments [Ph.D. thesis]*, Universidad Politécnica de Madrid, 2011.

Research Article

Negative Correlation Learning for Customer Churn Prediction: A Comparison Study

Ali Rodan,¹ Ayham Fayyumi,² Hossam Faris,¹ Jamal Alsakran,¹ and Omar Al-Kadi¹

¹King Abdulla II School for Information Technology, The University of Jordan, Amman 11942, Jordan

²College of Computer and Information Sciences, Al Imam Mohammad Ibn Saud Islamic University, Riyadh 11432, Saudi Arabia

Correspondence should be addressed to Ali Rodan; a.rodan@ju.edu.jo

Received 17 June 2014; Revised 23 August 2014; Accepted 7 September 2014

Academic Editor: Shifei Ding

Copyright © 2015 Ali Rodan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, telecommunication companies have been paying more attention toward the problem of identification of customer churn behavior. In business, it is well known for service providers that attracting new customers is much more expensive than retaining existing ones. Therefore, adopting accurate models that are able to predict customer churn can effectively help in customer retention campaigns and maximizing the profit. In this paper we will utilize an ensemble of Multilayer perceptrons (MLP) whose training is obtained using negative correlation learning (NCL) for predicting customer churn in a telecommunication company. Experiments results confirm that NCL based MLP ensemble can achieve better generalization performance (high churn rate) compared with ensemble of MLP without NCL (flat ensemble) and other common data mining techniques used for churn analysis.

1. Introduction

Technological improvements have enabled data driven industries to analyze data and extract knowledge. Data mining techniques facilitate the prediction of certain future behavior of customers [1]. One of the most important issues that reduces profit of a company is customer churn, which is also known as customer attrition or customer turnover [2]. Customer churn can also be defined as the business intelligence process of finding customers that are about to switch from a business to its competitor [3].

In today's industries, abundance of choices helps customers get advantage of a highly competitive market. One can choose a service provider that offers better service than others. Therefore, the profitmaking organizations which compete in saturated markets such as banks, telecommunication and internet service companies, and insurance firms strongly focused more on keeping current customers than acquiring new customers [4]. Moreover, maintaining current customers is proven to be much less expensive than acquiring new customers [5].

In order to keep their customers, companies need to have a deep understanding of why churn happens. There are several reasons to be addressed, such as dissatisfaction

from the company, competitive prices of other companies, relocation of the customers, and customers' need for a better service which can lead customers to leave their current service provider and switch to another one [6].

Among the previous studies for churn analysis, one of the most frequently used method is artificial neural networks (ANNs). In order to fine-tune the models developed, several topologies and techniques were investigated with ANNs, such as building medium-sized ANN models which were observed to perform the best and making experiments in many domains such as pay-TV, retail, banking, and finance [7]. These studies indicate that a variety of ANN approaches can be applied to increase prediction accuracy of customer churn. In fact, the use of neural networks in churn prediction has a big asset in respect of other methods used, because the likelihood of each classification made can also be determined. In neural networks each attribute is associated with a weight and combinations of weighted attributes participate in the prediction task. The weights are constantly updated during the learning process. Given a customer dataset and a set of predictor variables the neural network tries to calculate a combination of the inputs and to output the probability that the customer is a churner.

On the other hand, data collected for churn prediction is usually imbalanced, where the instances in the nonchurner customer outnumber the instances in the churning class. This is considered as one of the most challenging and important problems since common classification approaches tend to get good accuracy results for the large classes and ignore the small ones [8]. In [9], the authors discussed different approaches for handling the problem of imbalanced data for churn prediction. These approaches include using more appropriate evaluation metrics, using cost-sensitive learning, modifying the distribution of training examples by sampling methods, and using Boosting techniques. In [8], authors added ensemble classifiers as a fourth category of approaches for handling class imbalance. It has been shown that ensemble learning can offer a number of advantages over a single learning machine (e.g., neural network) training. Ensemble learning has a potential to improve generalization and decrease the dependency on training data [10].

One of the key elements for building ensemble models is the “diversity” among individual ensemble members. Negative correlation learning (NCL) [11] is an ensemble learning technique that encourages diversity explicitly among ensemble members through their negative correlation. However, few studies addressed the impact of diversity on imbalanced datasets. In [12], authors indicated that NCL brings diversity into ensemble and achieve higher recall values for minority class comparing NCL to independent ANNs.

Motivated by these possible advantages of NCL for class imbalance problems, in this paper, we apply the idea of NCL to an ensemble of multilayer perceptron (MLP) and investigate its application for customer churn prediction in the telecommunication market. Each MLP in the ensemble operates with a different network structure, possibly capturing different features of the input stream. In general, the individual outputs for each MLP of the ensemble are coupled together by a diversity-enforcing term of the NCL training, which stabilizes the overall collective ensemble output.

Moreover, the proposed ensemble NCL approach will be assessed using different evaluation criteria and compared to conventional prediction techniques and other special techniques proposed in the literature for handling class imbalance cases.

The paper has the following organization. Section 2 gives a background on data mining techniques used in the literature for churn analysis. In Section 3 we introduce negative correlation learning and how to use it to generate an ensemble of MLP with “diverse” members. Churn dataset description is given in Section 4. Experiments and results are presented in Section 5. Finally, our work is concluded in Section 6.

2. Related Work

Data mining techniques that are used in both researches and real-world applications generally treat churn prediction as a classification problem. Therefore, the aim is to use past customer data and classify current customers into two classes, namely, prediction churn and prediction nonchurn [7]. There

have also been made a few academic studies on clustering and association rule mining techniques.

After having done the necessary data collection and data preprocessing tasks and labeling the past data, features that are relevant to churn need to be defined and extracted. Feature selection, also known as dimension reduction, is one of the key processes in data mining and can alter the quality of prediction dramatically. Preprocessing and feature selection are common tasks that are applied before almost every data mining technique. There are different widely used algorithms for churn analysis, for instance, decision trees; by its algorithm nature, the obtained results can easily be interpreted and therefore give the researcher a better understanding of what features of customer are related to churn decision. This advantage has made decision trees one of the most used methods in this field. Some applications of decision trees in churn prediction include building decision trees for all customers and building a model for each of the customer segments [13]. Another relatively new classification algorithm is support vector machine (SVM). It is widely used in data mining applications, particularly in complex situations. SVM algorithm is proven to outperform several other algorithms by increasing the accuracy of classification and prediction of customers who are about to churn [14, 15].

Some other algorithms might be appropriate for customer churn prediction, as the artificial neural networks (ANN), which is another supervised classification algorithm that is used in predicting customer turnover. However, this algorithm is expected to give more accurate results when used in a hybrid model together with other algorithms or with another ANN classifier [16]. Another example could be genetic programming (GP). Genetic programming is an evolutionary method for automatically constructing computer program. These computer programs could be classifiers or regressors represented as trees. The authors in [17] used a GP based approach for modeling a churn prediction problem in telecommunication industry.

Moreover, Bayesian classification is one of the techniques which was also mentioned to be used in churn prediction. This method depends on calculating probabilities for each feature and its effect on determining the class value, which is the customer being a churning or not. However, Bayesian classification may not give satisfactory results when the data is highly dimensional [18].

Lastly, it is worth to mention k -nearest neighbor (k -NN) and random forests as another two classification methods applied in literature for churn prediction. k -NN is a classification method where an instance is classified by a majority vote of its neighbors, where, on the other hand, Random forests are an ensemble of decision trees which are generated from the bootstrap samples. The authors in [19] applied both k -NN and random forests to evaluate the performance on sampled and reduced features churn dataset. For some of related work for churn prediction methods, see Table 1.

3. Negative Correlation Learning (NCL)

NCL has been successfully applied to training multilayer perceptron (MLP) ensembles in a number of applications,

TABLE 1: Related work for churn prediction methods.

Author	Method	Description
Idris et al. [17]	GP	GP is applied with AdaBoost for churn prediction
Tsai and Lu [16]	ANN with BP	Applied as a hybrid approach in two stages (i.e., reduction and prediction)
Wang and Niu [14]	SVM	Least squares support vector machine (LS-SVM) is applied to establish a prediction model of credit card customer churn
Eastwood and Gabrys [31]	IBK	Authors apply simple k -nearest neighbor algorithm on a nonsequential representation of sequential data for churn prediction
Kraljević and Gotovac [32]	Decision trees	Decision trees (DT) were applied and compared with ANN and logistic regression. DT results outperform other models
Verbraken et al. [33]	Naive Bayes	Number of Bayesian Network algorithms, ranging from the Naive Bayes classifier to General Bayesian Network classifiers are applied for churn prediction

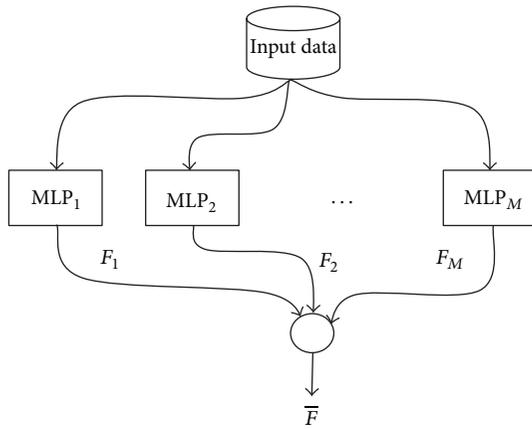


FIGURE 1: Ensemble of MLP networks.

including regression problems [20], classification problems [21], or time series prediction using simple autoregressive models [11].

In NCL, all the individual networks are trained simultaneously and interactively through the correlation penalty terms in their error functions. The procedure has the following form. Given a set of M networks and a training input set s , the ensemble output $F(t)$ is calculated as a flat average over all ensemble members (see Figure 1) $F_i(t)$:

$$F(t) = \frac{1}{M} \sum_{i=1}^M (F_i(t)). \quad (1)$$

In NCL the penalised error function to be minimised reads as follows:

$$E_i = \frac{1}{2} (F_i(t) - y(t))^2 + \lambda p_i(t), \quad (2)$$

where

$$p_i(t) = (F_i(t) - F(t)) \sum_{i \neq j} (F_j(t) - F(t)), \quad (3)$$

and $\lambda > 0$ is an adjustable strength parameter for the negative correlation enforcing penalty term p_i . It can be shown that

$$E_i = \frac{1}{2} (F_i(t) - y(t))^2 - \lambda (F_i(t) - F(t))^2. \quad (4)$$

Note that when $\lambda = 0$, we obtain a standard decoupled training of individual ensemble members. Standard gradient-based approaches can be used to minimise E by updating the parameters of each individual ensemble member.

3.1. Ensembles of MLPs Using NCL. Negative correlation learning (NCL) has been successfully applied to training MLP ensembles [10, 11, 20, 21]. We apply the idea of NCL to the ensemble of multilayer perceptron (MLPs) for predicting customer churn in a telecommunication company. Each MLP neural network operates with a different hidden layer, possibly capturing different features of the customer churn data. Crucially, the individual trained weights of the ensemble are coupled together by a diversity-enforcing term of the NCL training, which stabilises the overall collective ensemble output.

4. Dataset Description

The data used in this work is provided by a major Jordanian cellular telecommunication company. The data set contains 11 attributes of randomly selected 5000 customers for a time interval of three months. The last attribute indicates whether the customer churned (left the company) or not. The total number of churners is 381 (0.076 of total customers). The attributes along with their description are listed in Table 2.

The data is normalized by dividing each variable by its standard deviation. Normalization is recommended when data variables follow different dynamic ranges. Therefore, to eliminate the influence of larger values, normalization is applied to make all variables lie in the same scale.

4.1. Evaluation Criteria. In order to assess the developed model and compare it with different data mining techniques used for churn analysis, we use the confusion matrix shown in Table 3 which is the primary source for evaluating classification models. Based on this confusion matrix, the following three different criteria are used for the evaluation:

- (1) accuracy: measuring the rate of the correctly classified instances of both classes,

$$\text{Accuracy} = \frac{(tp + tn)}{(tp + fn + fp + tn)}, \quad (5)$$

TABLE 2: List of attributes.

Attribute name	Description
3G	Subscriber is provided with 3G service (yes, no)
Total consumption	Total monthly fees (calling + sms) (JD)
Calling fees	Total monthly calling fees (JD)
Local sms fees	Monthly local sms fees (JD)
International sms fees	Monthly fees for international sms (JD)
International calling fees	Monthly fees for international calling (JD)
Local sms count	Number of monthly local sms
International sms count	Number of monthly international sms
International MOU	Total of international outgoing calls in minutes
Total MOU	Total minutes of use for all outgoing calls
On net MOU	Minutes of use for on-net-outgoing calls
Churn	Churning customer status (yes, no)

TABLE 3: Confusion matrix.

	Predicted	
	Nonchurn	Churn
Actual nonchurn	tp	fn
Actual churn	fp	tn

(2) hit rate: measuring the rate of predicted churn in actual churn and actual nonchurn,

$$\text{Hit rate} = \frac{tn}{(fn + tn)}, \quad (6)$$

(3) actual churn rate: measuring the rate of predicted churn in actual churn,

$$\text{Churn rate} = \frac{tn}{(fp + tn)}. \quad (7)$$

5. Experiments and Results

5.1. Experimental Setup. In literature, some authors studied the effect of ensemble size on the performance. For example Hansen and Salamon in [22] suggested that ensembles with a small size as ten members were adequate to sufficiently reduce test-set error [23]. In our current paper we used empirical approach to investigate the appropriate size of the ensemble. For ensemble of networks, we tried (4, 6, 8, 10, 12, . . . , 20) networks in the hidden layer and then checked the performance each time. The best performance reached without

TABLE 4: Selected ensemble of MLP using NCL parameters based on 5-fold cross validation.

Parameter	Value
Hidden layers	1
Ensemble size (M)	10
Decay	0.001
hidden Layer nodes (N)	10
Activation function	$1/(1 + e(-x))$
Learning rate (η)	0.3
Momentum	0.2
λ	0.5

overfitting was for an ensemble of size of 10 networks. Therefore, the ensemble used in all our experiments consists of $M = 10$ MLP networks. In all experiments we use MLPs with hidden layer of $N = 10$ units. We used NCL training via backpropagation learning algorithm (BP) on E with learning rate $\eta = 0.3$. The output activation function of the MLP is sigmoid logistic.

We optimize the penalty factor λ and the number of hidden nodes using 5-fold cross validation, and λ is varied in the range $[0, 1]$ (step size 0.1) [10]. The number of hidden nodes is varied from 1 to 20 (step 1). Based on 5-fold cross validation, the details of the selected ensembles of MLPs using NCL parameters are presented in Table 4. Note that the selected parameters for the flat ensembles of MLPs (without NCL) are the same as with NCL except that there are no NCL λ parameters. The ensembles used in our experiments are also compared with the following common data mining techniques used in the literature:

- (i) k -nearest neighbour (IBK),
- (ii) Naive Bayes (NB),
- (iii) random forest (RF),
- (iv) genetic programming,
- (v) single MLP neural network trained with BP,
- (vi) C4.5 decision trees algorithm,
- (vii) support vector machine (SVM).

As churn data is imbalanced, NCL is also compared with other special techniques proposed in the literature for handling class imbalance cases. These techniques include:

- (i) AdaBoost algorithm with C4.5 Decision Tree as base classifier (AdaBoost), [24].
- (ii) Bagging algorithm with C4.5 Decision Tree as base classifier (Bagging) [25],
- (iii) MLP for cost-sensitive classification (NNCS) [26],
- (iv) synthetic minority oversampling technique with MLP as base classifier (SMOTE + MLP) [27],
- (v) neighborhood cleaning rules with constricted particle swarm optimization as base classifier (NCR + CPSO) [28–30],
- (vi) neighborhood cleaning rules with MLP as base classifier (NCR + MLP) [28].

TABLE 5: Tuning parameters for data mining techniques used in the comparison study.

Method	Parameters
GP	Population size = 1000, Maximum number of generations = 100, functions = {*, /, -, +, IF, <, >}, tree max depth = 10, tree max length = 30 elites = 1, selection mechanism = tournament selection crossover point probability = 90%, mutation = probability 15%
ANN with BP	Activation function = Sigmoid, Epoches = 5000, Learning Rate = 0.3, Momentum = 0.2
SVM	Cost = 1, Gamma = 10000
IBK	Number of neighbours = 1, nearest neighbor search algorithm = linear search (brute force search)
AdaBoost	Number of classifiers = 10
Bagging	Number of classifiers = 10
NNCS	Hidden layers = 2, hidden nodes = 15
SMOTE	Number of neighbors = 5
NCR + CPSO	Number of neighbors = 5 for SMOTE, number of particles = 75 for CPSO

TABLE 6: Evaluation results (results of best five models are shown in bold).

Model	Accuracy	Actual churn rate	Hit rate
<i>k</i> -Nearest neighbour (IBK)	0.927	0.022	0.067
Naive Bayes (NB)	0.597	0.901	0.115
Random Forest (RF)	0.940	0.006	0.109
Genetic programming (GP)	0.759	0.638	0.142
Single ANN with BP	0.941	0.625	0.607
Decision tress (C4.5)	0.975	0.703	0.964
Support vector machine (SVM)	0.977	0.703	0.992
AdaBoosting	0.972	0.719	0.898
Bagging	0.975	0.703	0.954
MLP for cost-sensitive classification (NNCS)	0.496	0.819	0.113
SMOTE + MLP	0.722	0.724	0.177
NCR + CPSO	0.894	0.827	0.694
NCR + MLP	0.642	0.751	0.144
Flat ensemble of ANN	0.958	0.732	0.725
Ensemble of ANN using (NCL)	0.971	0.803	0.814

Table 5 presents the empirical settings of the selected models parameters of these common data mining techniques based on 5-fold cross validation. For SVM, cost and gamma parameters are tuned using simple grid search. In C4.5 algorithm, the attributes are computed using the discriminant $P(\omega_0 | A = 0)$ where $P(\cdot)$ is the conditional probability and ω_i is the classification type. For naive Bayesian, the transcendent is $p(x | w_i)P(w_i) - p(x | w_j)P(w_j) = 0$ where w_i and w_j are the classification types, $p(x | w)$ is the condition probability density, and $P(w)$ is the transcendent probability.

5.2. Results and Comparison. Table 6 summarizes the results of negatively correlated ensemble of MLPs, independent ensemble of MLPs ($\lambda = 0$), and other data mining models used in the literature for customer churn prediction. In order to assess the improvement achieved by using a genuine NCL training against independent training of ensemble members ($\lambda = 0$), the MLP networks are initialized with the same

weight values in both cases (see Table 4). The MLP ensemble trained via NCL outperformed independent ensemble of MLPs model and all the other models used in this study.

Note that the two MLP ensemble versions we study share the same number of free parameters, with the sole exception of the single diversity-imposing parameter λ in NCL based learning.

According to Table 6, it can be noticed that NCL achieved high accuracy rate (97.1%) and ranked among the best five techniques (results are shown in bold font) with a slight difference (i.e, 0.6%) behind the best which is SVM. The flat ensemble ANN comes after with 95.8% accuracy. As mentioned earlier, accuracy is not the best evaluation metric when the examined problem is highly imbalanced. Therefore, we need to check other criteria; in our case, they are the churn rate and the hit rate. Looking at the obtained churn rates, NCL comes second with 80.3% churn rate and significant increase of 10% over SVM. It is important to indicate here that

although NB is the best in churn rate with 90.1%, it is the 2nd worst in terms of accuracy rate which is 59.7%. This gives a great advantage of NCL over NB. Finally, the hit rates of IBK, NB, RF, GP, NNCS, SMOTE + MLP, and NCR + MLP show very poor results so they can be knocked off the race. On the other hand, NCL and ensemble ANN show acceptable results for hit rate of 81.4% and 72.5%, respectively.

6. Conclusion

Customer churn prediction problem is important and challenging at the same time. Telecommunication companies are investing more in building accurate churn prediction model in order to help them in designing effective customer retention strategies. In this paper we investigate the application of an ensemble of multilayer perceptron (MLPs) trained through negative correlation learning (NCL) for predicting customer churn in a telecommunication company. Experimental results confirm that NCL ensembles achieve better generalization performance in terms of churn rate prediction with highly acceptable accuracy error rate compared with flat ensembles of MLPs and other common machine learning models from the literature.

Conflict of Interests

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

Acknowledgment

The authors would like to thank the Jordanian Mobile Telecommunication Operator for providing the required technical assistance and the data for this developed research work.

References

- [1] E. W. T. Ngai, L. Xiu, and D. C. K. Chau, "Application of data mining techniques in customer relationship management: a literature review and classification," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2592–2602, 2009.
- [2] J. Hadden, A. Tiwari, R. Roy, and D. Ruta, "Computer assisted customer churn management: state-of-the-art and future trends," *Computers and Operations Research*, vol. 34, no. 10, pp. 2902–2917, 2007.
- [3] S. A. Qureshi, A. S. Rehman, A. M. Qamar, and A. Kamal, "Telecommunication subscribers' churn prediction model using machine learning," in *Proceedings of the 8th International Conference on Digital Information Management (ICDIM '13)*, pp. 131–136, September 2013.
- [4] G.-E. Xia and W.-D. Jin, "Model of customer churn prediction on support vector machine," *System Engineering—Theory & Practice*, vol. 28, no. 1, pp. 71–77, 2008.
- [5] A. Keramati and S. M. S. Ardabili, "Churn analysis for an Iranian mobile operator," *Telecommunications Policy*, vol. 35, no. 4, pp. 344–356, 2011.
- [6] Z. Kasiran, Z. Ibrahim, and M. S. M. Ribuan, "Mobile phone customers churn prediction using elman and Jordan Recurrent Neural Network," in *Proceedings of the 7th International Conference on Computing and Convergence Technology (ICCT '12)*, pp. 673–678, December 2012.
- [7] A. Sharma and P. K. Panigrahi, "A neural network based approach for predicting customer churn in cellular network services," *International Journal of Computer Applications*, vol. 27, no. 11, pp. 26–31, 2011.
- [8] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 42, no. 4, pp. 463–484, 2012.
- [9] J. Burez and D. van den Poel, "Handling class imbalance in customer churn prediction," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4626–4636, 2009.
- [10] G. Brown and X. Yao, "On the effectiveness of negative correlation learning," in *Proceedings of the 1st UK Workshop on Computational Intelligence*, 2001.
- [11] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399–1404, 1999.
- [12] S. Wang, K. Tang, and X. Yao, "Diversity exploration and negative correlation learning on imbalanced data sets," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '09)*, pp. 3259–3266, June 2009.
- [13] S.-Y. Hung, D. C. Yen, and H.-Y. Wang, "Applying data mining to telecom churn management," *Expert Systems with Applications*, vol. 31, no. 3, pp. 515–524, 2006.
- [14] N. Wang and D.-X. Niu, "Credit card customer churn prediction based on the RST and LS-SVM," in *Proceedings of the 6th International Conference on Service Systems and Service Management (ICSSSM '09)*, pp. 275–279, Xiamen, China, June 2009.
- [15] A. Rodan, H. Faris, J. Alsakran, and O. Al-Kadi, "A support vector machine approach for churn prediction in telecom industry," *International Journal on Information*, vol. 17, no. 8, pp. 3961–3970, 2014.
- [16] C.-F. Tsai and Y.-H. Lu, "Customer churn prediction by hybrid neural networks," *Expert Systems with Applications*, vol. 36, no. 10, pp. 12547–12553, 2009.
- [17] A. Idris, A. Khan, and Y. S. Lee, "Genetic programming and adaboosting based churn prediction for telecom," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '12)*, pp. 1328–1332, Seoul, Republic of Korea, October 2012.
- [18] B. Huang, M. T. Kechadi, and B. Buckley, "Customer churn prediction in telecommunications," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1414–1425, 2012.
- [19] A. Idris, M. Rizwan, and A. Khan, "Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1808–1819, 2012.
- [20] G. Brown, J. L. Wyatt, and P. Tiño, "Managing diversity in regression ensembles," *The Journal of Machine Learning Research*, vol. 6, pp. 1621–1650, 2005.
- [21] R. McKay and H. Abbass, "Analyzing anticorrelation in ensemble learning," in *Proceedings of the Conference on Australian Artificial Neural Networks and Expert Systems*, pp. 22–27, 2001.
- [22] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.
- [23] D. Opitz and R. Maclin, "Popular ensemble methods: an empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

- [24] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [25] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [26] Z. H. Zhou and X. Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63–77, 2006.
- [27] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [28] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," in *Proceedings of the 8th Conference on AI in Medicine in Europe (AIME '01)*, vol. 2001 of *Lecture Notes on Computer Science*, pp. 63–66, Springer, 2001.
- [29] T. Sousa, A. Silva, and A. Neves, "Particle swarm based data mining algorithms for classification tasks," *Parallel Computing*, vol. 30, no. 5–6, pp. 767–783, 2004.
- [30] H. Faris, "Neighborhood cleaning rules and particle swarm optimization for predicting customer churn behavior in telecom industry," *International Journal of Advanced Science and Technology*, vol. 68, pp. 11–12, 2014.
- [31] M. Eastwood and B. Gabrys, "A non-sequential representation of sequential data for churn prediction," in *Knowledge-Based and Intelligent Information and Engineering Systems*, J. D. Velásquez, S. A. Ríos, R. J. Howlett, and L. C. Jain, Eds., vol. 5711 of *Lecture Notes in Computer Science*, pp. 209–218, Springer, Berlin, Germany, 2009.
- [32] G. Kraljević and S. Gotovac, "Modeling data mining applications for prediction of prepaid churn in telecommunication services," *Automatika*, vol. 51, no. 3, pp. 275–283, 2010.
- [33] T. Verbraken, W. Verbeke, and B. Baesens, "Profit optimizing customer churn prediction with Bayesian network classifiers," *Intelligent Data Analysis*, vol. 18, no. 1, pp. 3–24, 2014.

Research Article

An Approach to Model Based Testing of Multiagent Systems

Shafiq Ur Rehman and Aamer Nadeem

Center for Software Dependability, Mohammad Ali Jinnah University, Islamabad 44000, Pakistan

Correspondence should be addressed to Shafiq Ur Rehman; shafiq.rehman@gmail.com

Received 22 June 2014; Revised 11 September 2014; Accepted 14 September 2014

Academic Editor: Shifei Ding

Copyright © 2015 S. Ur Rehman and A. Nadeem. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Autonomous agents perform on behalf of the user to achieve defined goals or objectives. They are situated in dynamic environment and are able to operate autonomously to achieve their goals. In a multiagent system, agents cooperate with each other to achieve a common goal. Testing of multiagent systems is a challenging task due to the autonomous and proactive behavior of agents. However, testing is required to build confidence into the working of a multiagent system. Prometheus methodology is a commonly used approach to design multiagents systems. Systematic and thorough testing of each interaction is necessary. This paper proposes a novel approach to testing of multiagent systems based on Prometheus design artifacts. In the proposed approach, different interactions between the agent and actors are considered to test the multiagent system. These interactions include percepts and actions along with messages between the agents which can be modeled in a protocol diagram. The protocol diagram is converted into a protocol graph, on which different coverage criteria are applied to generate test paths that cover interactions between the agents. A prototype tool has been developed to generate test paths from protocol graph according to the specified coverage criterion.

1. Introduction

Autonomous agents possess features like reactivity and proactivity, and they are able to interact with each other in order to perform certain tasks. Multiagents systems are used in complex application due to agent's unique features. Agents perceive their environment and respond accordingly to meet their goal. Autonomy is the agent's ability to operate independently, without the need for human guidance or intervention [1]. Application of multiagent systems is seen in many domains like e-commerce, banking, air traffic control, information management, and so forth. There are many agent development methodologies in which agent based systems can be modeled; one of them is Prometheus methodology [2]. Prometheus agent oriented software engineering methodology has a well-developed process from system specification to architectural design and then detailed design leading easily to code.

The term autonomy refers to the goal oriented behavior of agents. Autonomous agents are programmed to perform automatically in order to achieve certain goals. All of their activities converge towards achieving their defined goals. There are certain commercial agent applications presented in

[3] which show the sensitivity of agent applications as they are meant to solve the real life problems in almost every domain. Real-time response and dynamism make testing of such application very hard. Performance and accuracy of results must be checked and this can be achieved with the effective testing of agent applications.

Padgham and Winikoff show that agent systems provide great flexibility, with over a million ways to achieve a given goal using only a relatively small hierarchy of goals and plans [4]. Because agents are autonomous and flexible, agent systems can be difficult to test. Therefore an approach is necessary that can test an agent system effectively and efficiently.

Prometheus is a methodology for designing intelligent agents from specification to detailed design and implementation [2]. One can model the agent using the Prometheus methodology starting from system specification to detailed design which includes identifying environment or external actors and scenarios with details of actions and percepts involved. Scenarios have actions and percepts associated with them. Different agents are responsible for different goals and different plans are associated with different goals [5]. Prometheus also supports the design via tool named

Prometheus Design Tool (PDT) [6] in which design activities can be modeled. We can capture the relationship between goals and plans of an agent by goal-plan diagram. We have demonstrated this relationship in our paper [7]. Interaction protocol in detailed design is captured by interaction pattern/sequence between the agents in a certain scenario. These interactions occurred between agents and actors in form of messages, actions, and percepts. Agent systems to perform correctly these interactions must be tested and their occurrence in protocol must be verified with test data. Based on autonomous agents testing we have two research questions which we will cover in our proposed testing framework.

(i) How can design artifacts be used to test the interactions in a multiagent system?

This involves illustrating how design artifacts are chosen to be used to test the different agent interactions. Each interaction is carried out to meet some defined goals. We can extract goal-plan diagram as we discussed in our earlier research [7] and use the flow between goals and plans with respect to agent interaction. Interactions between agents and actor include message, action, and percept. Only message interactions are covered in [8]; actions and percepts have not been covered. We use the protocol diagram and convert it to protocol graph to test all sort of interactions between the agents and actor in specific protocol.

(ii) How can the process of generating such tests be guided by coverage criteria?

Define the scope coverage of the testing framework; identify additional coverage criteria with existing criteria discussed in [8] and probably identifying the additional coverage criteria which will cover action and percepts as well in any interactions diagram.

Our aim is to test the interactions of agents using their model specified in terms of interaction protocol. We have developed a tool which generates test paths based on specified coverage criterion that will test the interaction between the agents via some protocol. In order to achieve a goal there can be interactions between the agent and environment as well.

An agent achieves its goal with the help of plans specified. A main goal may have some subgoals contributing their part in achieving the objective. A goal-plan diagram can be used to describe the behavior of the agent showing all relevant actions, percepts, messages, and subgoals to be performed during the execution. Section 2 describes modeling methodologies and how Prometheus is a better approach to design multiagent system. Section 3 focuses on related work done in testing of autonomous agents. Section 4 describes the details of testing framework for model based testing of autonomous agents. In Section 5 a case study has been presented by applying our testing framework. Section 6 describes the conclusion and future work; references are shown at the end.

2. Modelling Methodologies

There are several agent-oriented software engineering methodologies, for example, Gaia (Generic Architecture for Information Availability) [9], Multi-Agent Systems Engineering (MaSE) [10, 11], MESSAGE [12], Prometheus [2], Tropos [13], CoMoMAS [14], SODA (Societies in Open and Distributed

Agent spaces) [15], DESIRE [16], MAS-CommonKADS [17], and Belief-Desire-Intention (BDI) Model [18]. A methodology is collection of activities used to develop the system. Additionally methodology can be supported by the tool as well.

Agent architecture shows the behavior of agents, one of which is Belief-Desire-Intention (BDI) architecture [19]. BDI agents have certain goals to achieve. Belief-Desire-Intention properties are used to program intelligent agents. BDI agents have been widely used since last two decades and various researchers have explored their behavior. The agents whom we will discuss and use in our research are BDI agents. We consider multiagent systems developed by using Prometheus methodology. Padgham and Winikoff present Prometheus as an agent oriented methodology based on BDI agents [2].

Requirements are assumed to be known in Gaia methodology which forms the basis of analysis and design phases. Gaia is a methodology which distinguishes between analysis and design phases. It has Role Model and Interaction Model in analysis phase and Agent Model, Services Model, and Acquaintance Model in Design phase. Gaia has no tool support [9]. MaSE is an extension of the object-oriented approach that has two phases of analysis and design. MaSE does not have the view that agents should be autonomous and instead it assumes agents as only software which interacts with other softwares, that is, agents. Analysis contains three steps, that is, Capturing Goals, Applying Use Cases, and Refining Roles and design contains four steps, that is, Creating Agent Classes, Constructing Conversations, Assembling Agent Classes, and System Design [10]. MESSAGE adopts the life-cycle model of the Rational Unified Process (RUP) and is limited to analysis and design activities only. It uses UML as modeling language. It has five different views, for example, Organization view, Goal/Task view, Agent/Role view, Interaction view, and Domain view [12].

The Prometheus methodology [2] is a detailed AOSE methodology, which aims to cover all of the major activities required in the developing agent systems from system specification to architectural design and detailed design as well. Tropos is an AOSE methodology whose main distinction is the early requirement analysis. Agent related concepts like goals, plans, and tasks are included in all phases. No detailed information is available for last process defining agent types and mapping them to capabilities. The methodology does not appear to provide heuristics for any phase [18].

CoMoMAS focus in knowledge engineering problem arises in multiagent systems and provides extension in Cooperation Modeling Language for agents [14]. SODA focus on social inter-agent aspects of agent systems and that employs the concept of coordination models [15, 20]. DESIR contains expertise model and agents. Once analysis phase has been done, DESIRE could be used for specifying the design and implementation [16].

3. Related Work

To gain confidence on a multiagent system, it must be properly tested. Testing of software agent is an important and critical task as agents possess dynamic behavior. Basic

agent-oriented concepts, for example, autonomy, mental attitudes, pro-activeness, and so forth, have been covered in the above discussed methodologies but there are several exceptions. Tropos was not perceived as being easy to use whilst MESSAGE and GAIA were both ranked weakly on adequacy and expressiveness. MaSE does not provide detailed design. Prometheus methodology is rich enough to provide detailed design and tool support as well for developers [18]. There is a need of quality assurance issues to be addressed in multiagent systems designed in Prometheus methodology. We are aiming to fill the gap of providing quality assurance and testing support for the multiagent systems designed using Prometheus methodology.

Agents have run time response and adaptability. Coverage criteria for testing can be applied to both code and model [21]. Code base conform that all code are covered in term of statements, and so forth while model based coverage requires the different interaction from different states of the system, represented in specific model [22].

Low et al. consider test coverage criteria for BDI agents [23]. They derive two types of control-flow graphs: one with nodes, where node represents plans for BDI agent and arcs present messages or other events which initiate certain plan, and another CFG in which node presents statements within plans and arcs represent control-flow between statements (a standard control-flow graph). Several coverage criteria are defined, based on node, arc, and path coverage and some were based on the success or failure of executing statements and plans [23]. Different interactions between the modeling artifacts are not presented. Instead this approach is not considering interactions between agents; our approach considers agent interactions in multiagent systems.

Zhang et al. presented an approach for model based testing for agent system [24]. Testing framework caters the different sequence of agent program execution. Fault directed testing approach is used by first identifying appropriate units of the agent and testing the unit with the defined mechanism. It considers the plan as a single unit; then it is checked whether the plan is triggered by the appropriate event or not, and its precondition, cycles in plan, and plan completeness, and so forth are checked. Event testing is performed for numbers of applicable plans for the event. An electronic bookstore system has been used as the sample system; testing framework will execute test units in a sequence [24]. No coverage measures have been taken while considering interactions between agent and external agent or stub. We are considering interactions between multiagent systems through coverage measures.

Zheng and Alagar proposed a method for conformance testing of agent's BDI properties as alternative to formal verification [25]. Test cases are generated to check the implementation with respect to specification. Winikoff and Cranefield have analyzed the size of behavior space for BDI agent and found that failure handling has larger impact on size of behavior space than expected [1]. Failure handling has been introduced in context of agent's behavioral space [1]. Both techniques above do not consider interactions between agents neither have any coverage measures been taken even in unit testing.

Nguyen et al. build an approach in which autonomous agents are tested with the help of evolutionary algorithm techniques in which test cases are represented as chromosomes [26]. Soft goals are used as the evaluation criteria so that test cases will be developed keeping in mind to meet the identified soft goals criteria to test the agent [26]. Each test case is evaluated through a defined fitness function. Goals are represented by quality functions and new tests are selected by reproduction. A framework for testing of autonomous agents has been presented in [27]. Individual agents have been tested in [26] and genetic algorithm idea on testing has been presented. Above technique does not cover multi-agent systems neither interactions between agents. We will test multiagent system and cover interactions between them; we are inspired to use genetic algorithm in our future extension.

Miller et al. state that the interaction between the agents possesses complex behavior and therefore testing of interactions is important [8]. They defined two sets of test coverage criteria for multiagent interaction testing. The first uses only the protocol specification, while the second considers also the plans that generate and receive the messages in the protocol [8].

Existing model based testing techniques for multiagent systems do not cover every aspect of multi-gent systems, that is, dependencies and interactions. Interactions between agents in Prometheus methodology have action and percepts interactions between agents as well which have not been covered still in existing techniques. Our approach to multiagent system testing covers such interactions as well and testing coverage will be done.

4. Proposed Testing Framework

In this section, we discuss our proposed approach for testing of multiagent system using the Prometheus design artifact defined in Prometheus Design Tool (PDT). Our proposed testing framework will address the automated test case generation of multiagent system using design artifacts. Interaction protocols will be used to build a test model which covers messages, actions, and percepts in order to achieve certain goal. Coverage criteria have been defined on protocol graph, covering every possible interaction between agents. In future we will test generated test paths with test data. Test data generation will be done with evolutionary algorithms. An algorithm for automated test case generation will be proposed and tool has been developed which uses identified coverage criteria, keeping in mind the messages and percepts and interaction protocol, and generates test paths.

Figure 1 describes the testing framework of proposed technique. Our proposed technique has two main processes namely Protocol Graph Generator (Design Model) and test path generator. Design Model Generator uses Prometheus interaction protocol presented in protocol diagram (Figure 2) and generates a protocol graph (Figure 3) from it. The generated protocol graph gives a complete representation of all messages; percepts and action perform between the agents and actors in a specific protocol. Different coverage criteria will be defined focusing on percepts and actions as well along with messages and used as input to test path generator.

Input: Protocol Diagram with AUML syntax
Output: Protocol Graph.
 Let PG be the graph containing Percept, Action and Message Nodes.
 Start and End denotes the starting and ending states of Graph.
Step 1. For Each actor and Agent
Step 2. Make Percept, Action and Message Nodes.
Step 3. Link each nodes as defined in AUML notation
Step 4. If Loop Box
Step 5. add link last to first node in loop box
Step 6. End If
Step 7. If Box Alternative
Step 8. Add choice between nodes
Step 9. End If
Step 10. If Box Optional
Step 11. Add Different Path from start of optional to end of optional node.
Step 12. End If

ALGORITHM 1: Converting protocol diagram into protocol graph.

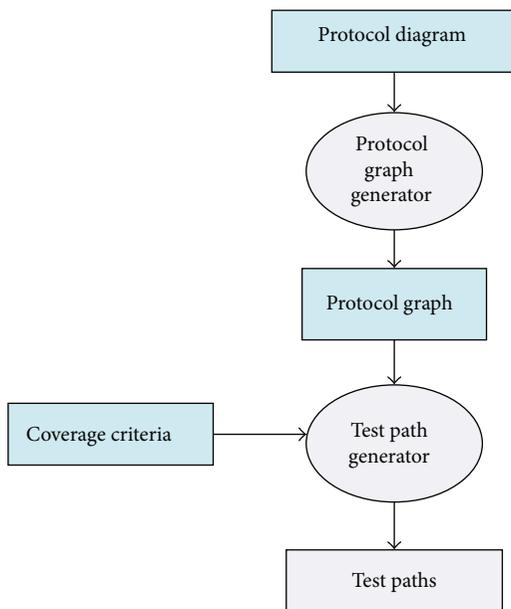


FIGURE 1: Proposed technique architecture.

Coverage criteria have been defined covering all possible interactions occurring in protocol graph. Test path generator uses protocol graph and applies different defined coverage criteria to generate test paths. Test paths will be generated using our test model, that is, protocol graph which will cater for interactions, messages, actions, and percepts in order to achieve certain goal.

Currently only message coverage criteria have been proposed by [8]. In a certain protocol, percepts and actions in an interaction have their importance and their coverage is necessary for effective testing. Our approach will uncover the interaction faults that would lie between the agents and actors.

4.1. Protocol Graph Generator. In our proposed testing framework, interaction protocol or protocol diagram is used as the design artifact which is transformed into a protocol graph. Protocol diagram contains messages, actions, and percepts interactions between agents and actors. Messages are passed only between the agents while actions and percepts interactions are performed between agents and actors.

4.1.1. Protocol Diagram to Protocol Graph. Protocol diagram shows details of how messages, action, and percepts are involved in a protocol. In our work we convert the protocol diagram into protocol graph. Protocol graph has been introduced by Miller et al. [8]. They defined two sets of test coverage criteria for multiagent interaction testing. The first uses only the protocol specification, while the second considers also the plans that generate and receive the messages in the protocol. Miller et al. [8] do not cover the actions and percepts during the interaction. We have extended the protocol graph with actions and percepts as they are a very important part of interaction protocol. Algorithm 1 is used to convert protocol diagram into protocol graph. We take protocol diagram as input and protocol graph has been produced by following Algorithm 1. Protocol diagram is represented in AUML representation as well. Code 2 shows AUML description of protocol graph. Protocol graph represents interaction protocol in nodes and vertices form, on which different coverage criteria have been applied.

Once we have successfully converted protocol diagram into protocol graph, we need to generate test paths from protocol graph. Algorithm 2 is used to generate test path from protocol graph.

4.2. Test Paths Generator. In this subsection we describe second process of our proposed approach named test path generator. Test path generator takes protocol diagram and coverage criteria as input and generates test paths for protocol. We have designed a test path generation Tool for

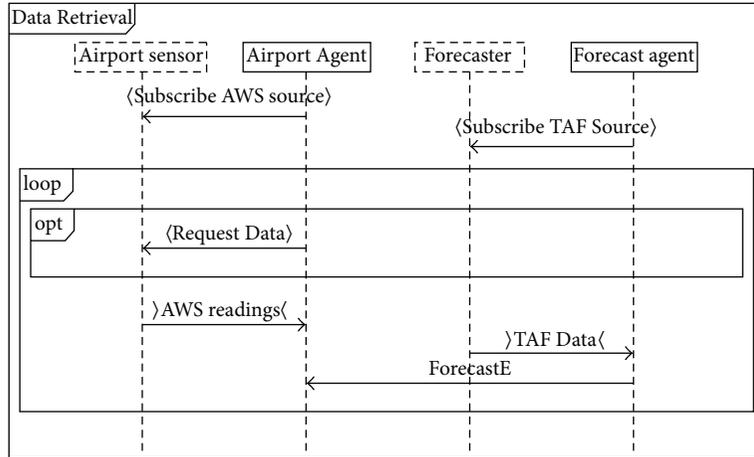


FIGURE 2: Data retrieval protocol diagram.

Input: Coverage Criteria (A set of defined coverage criteria), Graph (Set of nodes and edges)
Output: Test Paths
 Step 1. Build an **edge list** and **node list** of graph
 Step 2. Categorize node with respect to type
 Step 3. **if** all paths from graph = empty
 Step 4. **find_all_path from graph**
 Step 5. **End if**
 Step 6. Sort the paths in ascending order of the path length ending
 Step 7. **if** current path = selected coverage criteria
 Step 8. append (current path) in result
 Step 9. **End if**
 Step 10. Print Result

ALGORITHM 2: Test path generation from protocol graph.

automated test path generation. Coverage criteria have been defined in the following section.

4.2.1. Test Coverage Criteria. Our aim in this research paper is to test the interaction done in a protocol; those interactions can be in form of message, action, or percept. Miller et al. [8] have proposed some coverage criteria on protocol graph like message coverage and pair wise message coverage which are more likely the same.

Additional coverage criteria for protocol graph including actions and percepts have been defined in testing technique. We have defined the following coverage criteria that will cover all possible aspects of interactions between agents and actors in the form of message, action, and percept. Figure 4 shows hierarchy of test coverage criteria used to test multiagent system.

Test Path. A test path is a complete path in a protocol graph G that starts at node i and ends at node f . In following definitions of coverage criteria, M represents the set of all messages, P represents set of percepts and A represents set of all actions.

(1) **Message Coverage.** A set of test paths (TP) is said to satisfy message coverage criterion for a protocol graph G if each

message node m of graph G is included in at least one path $P \in TP$.

This coverage criterion ensures that every message in protocol has been traversed at least once. There exists path from start to traversing all messages in it.

(2) **Action Coverage.** A set of test paths (TP) is said to satisfy action coverage criterion for a protocol graph G if each action node “ a ” of graph G is included in at least one path $P \in TP$.

In this coverage criteria every action included in protocol graph must be included in generated test path for action coverage criterion.

(3) **Percept Coverage.** A set of test paths (TP) is said to satisfy percept coverage criterion for a protocol graph G if each percept node p of graph G is included in at least one path $P \in TP$.

In this coverage criteria every percept included in protocol graph must be included in generated test path for percept coverage criterion.

(4) **Message-Action Coverage.** A set of test paths (TP) is said to satisfy message-action coverage for protocol graph G if for each edge (m, a) in G , there is a test path $P \in TP$ that contains subpath (m, a) , where $m \in M$ and $a \in A$.

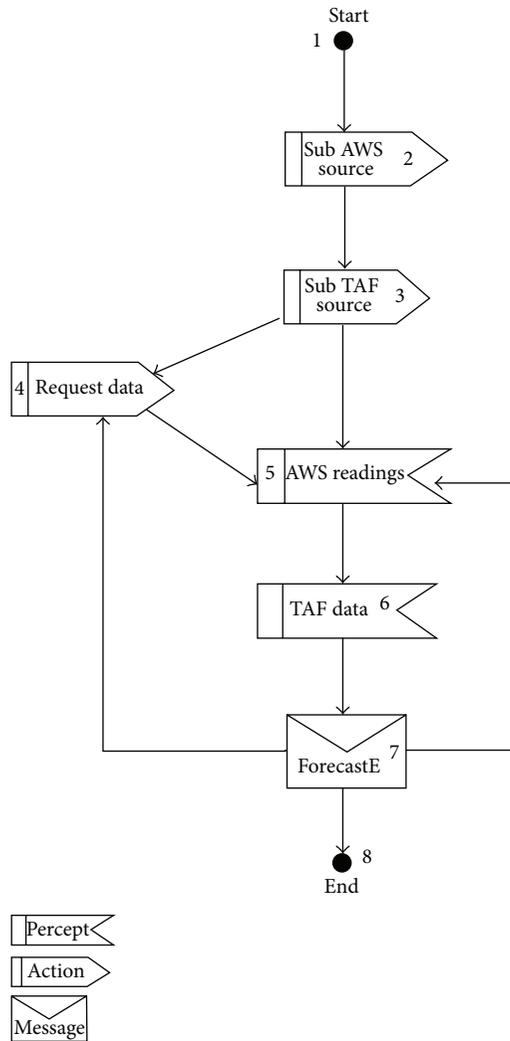


FIGURE 3: Protocol graph for data retrieval protocol diagram.

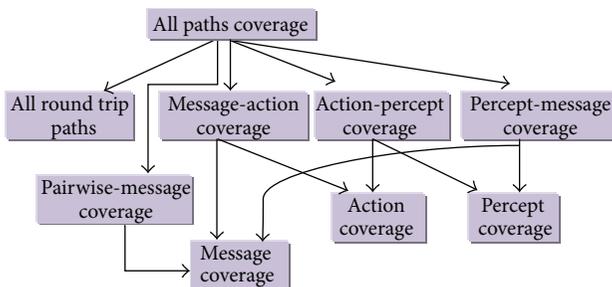


FIGURE 4: Test Coverage Criteria Hierarchy.

Messages are passed between the agents and actions are passed between the agent and actor. Agent sends a message to an agent and agents send the action to actor; this sort of interaction must also be covered assuring the message-action coverage criterion.

(5) *Action Percept Coverage.* A set of test paths (TP) is said to satisfy action-percept coverage for protocol graph G if for each edge (a, p) in G , there is a test path $P \in TP$ that contains subpath (a, p) , where $a \in A$ and percept $p \in P$.

Agents send an action to an actor in multiagent system demanding some task to be completed; in return actor sends the percept containing the required information or data, and this sort of communication is covered in action percept coverage criterion.

(6) *Percept-Message Coverage.* A set of test paths (TP) is said to satisfy percept-message coverage for protocol graph G if for each edge (p, m) in G , there is a test path $P \in TP$ that contains subpath (p, m) , where $p \in P$ and $m \in M$.

While receiving the percept from the actor, agents send a message to agent with necessary information; this sort of communication is covered in percept-message coverage criterion.

(7) *Pairwise-Message Coverage.* A set of test paths (TP) is said to satisfy pairwise-message coverage for protocol graph G if for each edge (m, n) in G , there is a test path $P \in TP$ that contains subpath (m, n) , where $m \in M$ and $n \in M$.

In protocol graph, all cases in one message can be followed by another message are covered in pairwise-message coverage. Addition of pairwise-message coverage assures arc coverage which is left in message coverage criterion.

(8) *All Round Trip Paths.* A set of test paths (TP) is said to satisfy all round trip paths coverage criterion for a protocol graph G if it loops back on same state in graph G in at least one test path $P \in TP$.

Interaction protocol describes the protocol in AUML protocol diagram which contains loops as well depending upon the protocol requirements. All round trip paths coverage criterion in protocol diagram traverse all loop at least once and include those paths which loops back on same state in generated test paths.

(9) *All Paths Coverage.* A set of test paths (TP) is said to satisfy all paths coverage criterion for protocol graph G if it traverses every complete path $P \in TP$ in G at least once.

All paths from start to end in a protocol graph are covered in all paths coverage criterion.

5. Case Study

In this research paper we have taken case study of multi-currency Bank Account system [28] which maintains bank accounts in nominated currencies and performs currency conversions to allow transactions against the accounts to occur in any currency. It consists of a BankAccount agent, a CurrencyExchange agent, and a Communicator agent which acts as an interface [28]. We have designed the system overview diagram of account case study using Prometheus Design Tool [6]. Figure 5 shows system overview diagram of multiagent system in which different agents have interacted with each other via account operation protocol. Each agent has actions, percepts, and messages associated with it.

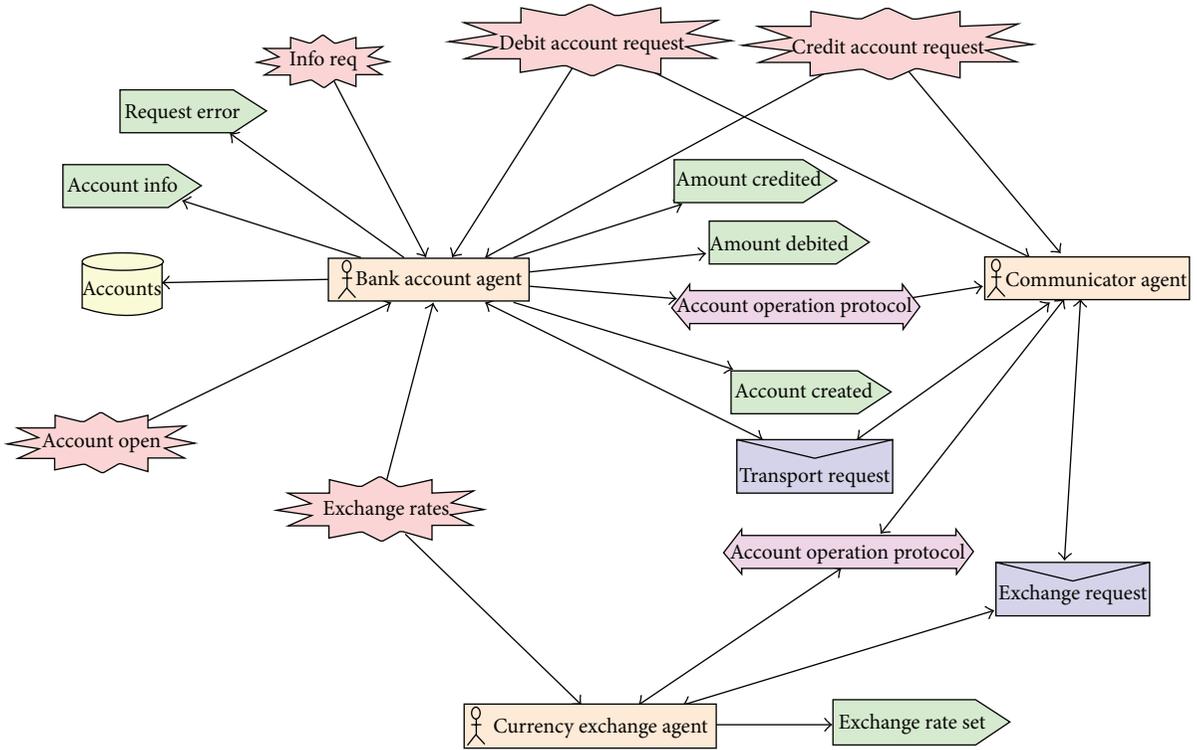


FIGURE 5: System overview diagram of multiagent system.

TABLE 1: Test paths for account operation protocol diagram.

S. #	Coverage criteria	Test paths
1	Message coverage	(i) 1 → 2 → 3 → 4 → 6 (message) → 7 (message) → 8 → 9 (message) → 11 → 13 → 14
2	Action coverage	(i) 1 → 2 → 3 (action) → 5 → 6 → 7 → 8 → 9 → 10 (action) → 13 (action) → 14 (ii) 1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 12 (action) → 14 (iii) 1 → 2 → 3 (action) → 4 → 6 → 7 → 8 → 9 → 11 (action) → 13 (action) → 14
3	Percept coverage	(i) 1 → 2 (percept) → 3 → 5 (percept) → 6 → 7 → 8 (percept) → 9 → 10 → 13 → 14 (ii) 1 → 2 (percept) → 3 → 4 (percept) → 6 → 7 → 8 (percept) → 9 → 11 → 13 → 14
4	Message action coverage	(i) 1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 (message) → 10 (action) → 13 → 14 (ii) 1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 (message) → 12 (action) → 14 (iii) 1 → 2 → 3 → 4 → 6 → 7 → 8 → 9 (message) → 11 (action) → 13 → 14
5	Action percept coverage	(i) 1 → 2 → 3 (action) → 5 (percept) → 6 → 7 → 8 → 9 → 10 → 13 → 14 (ii) 1 → 2 → 3 (action) → 4 (percept) → 6 → 7 → 8 → 9 → 11 → 13 → 14 (iii) 1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 12 (action) → 5 (Percept) → 6 → 7 → 8 → 9 → 12 → 14 (iv) 1 → 2 → 3 (action) → 4 (percept) → 6 → 7 → 8 → 9 → 11 → 13 → 4 → 6 → 7 → 8 → 9 → 11 → 13 → 14
6	Percept-message coverage	(i) 1 → 2 → 3 → 5 (percept) → 6 (message) → 7 → 8 (percept) → 9 (message) → 10 (action) → 13 → 14 (ii) 1 → 2 → 3 → 4 (percept) → 6 (message) → 7 → 8 (percept) → 9 (message) → 11 (action) → 13 → 14
7	Pairwise-message coverage	(i) 1 → 2 → 3 → 5 → 6 (message) → 7 (message) → 8 → 9 → 10 → 13 → 14
8	All round trip paths	(i) 1 → 2 → 3 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 10 → 13 → 5 → 12 → 14 (ii) 1 → 2 → 3 → 5 → 11 → 13 → 5 → 12 → 5 → 10 → 13 → 14
9	All paths coverage	(i) Infinite # of Paths

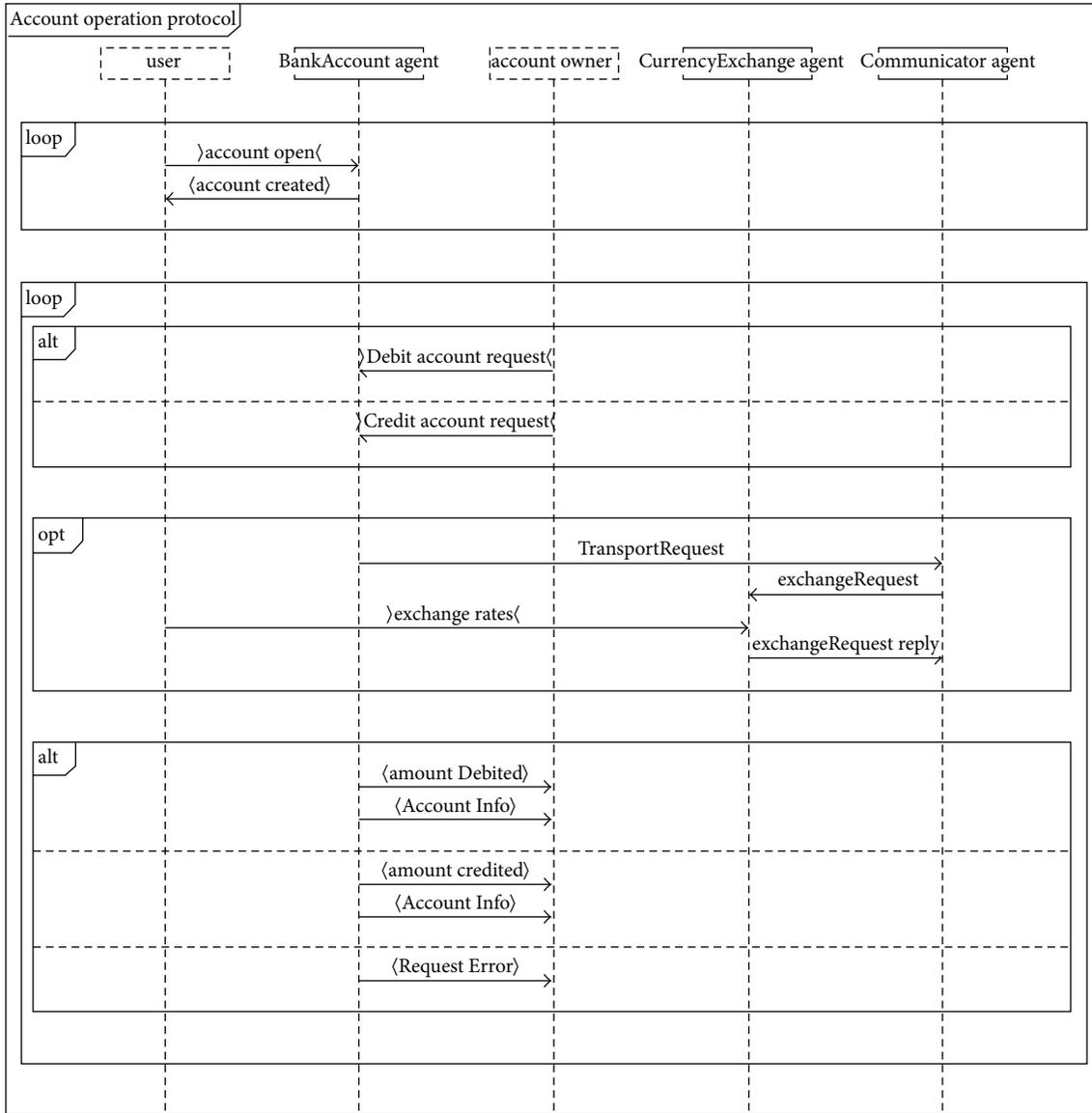


FIGURE 6: Account operation protocol diagram.

Different interactions between agents and actors are occurring through account operation protocol as depicted in Figure 5.

Each protocol includes different interactions between agents and actors to perform specific tasks; such interactions are modeled in protocol diagram. Content of protocol diagram includes alternatives and loops and other deviations from a simple sequence are depicted in AUMML using nested boxes [29]. Code 2 shows AUMML description of account operation protocol diagram used in Prometheus Design Tool.

Figure 6 shows details of account operation protocol diagram [30] that is further converted to protocol graph (Figure 7) by protocol graph convertor process.

Table 1 shows test path against each coverage criteria we have defined and applied on our case study.

5.1. Test Path generator Tool. We have developed a tool to illustrate our proposed approach. Protocol diagram converted to protocol graph on which different coverage criteria have been applied to generate paths with respect to coverage criteria defined above. Our tool takes protocol diagram as input and generates test paths. Test path generator tool has two main classes namely Graph Regeneration and Graph Parser. Graph Regeneration reads the input file and makes a graph object according to the file. This object is used in the program to produce the paths. Graph Parser searches all the possible paths according to the coverage criteria given to it. Table 2 shows input file for test path generator tool and Figure 8 shows the process of test path generation tool.

Different coverage criteria precondition and outcomes are programmed with respect to protocol graph. Code 1 shows

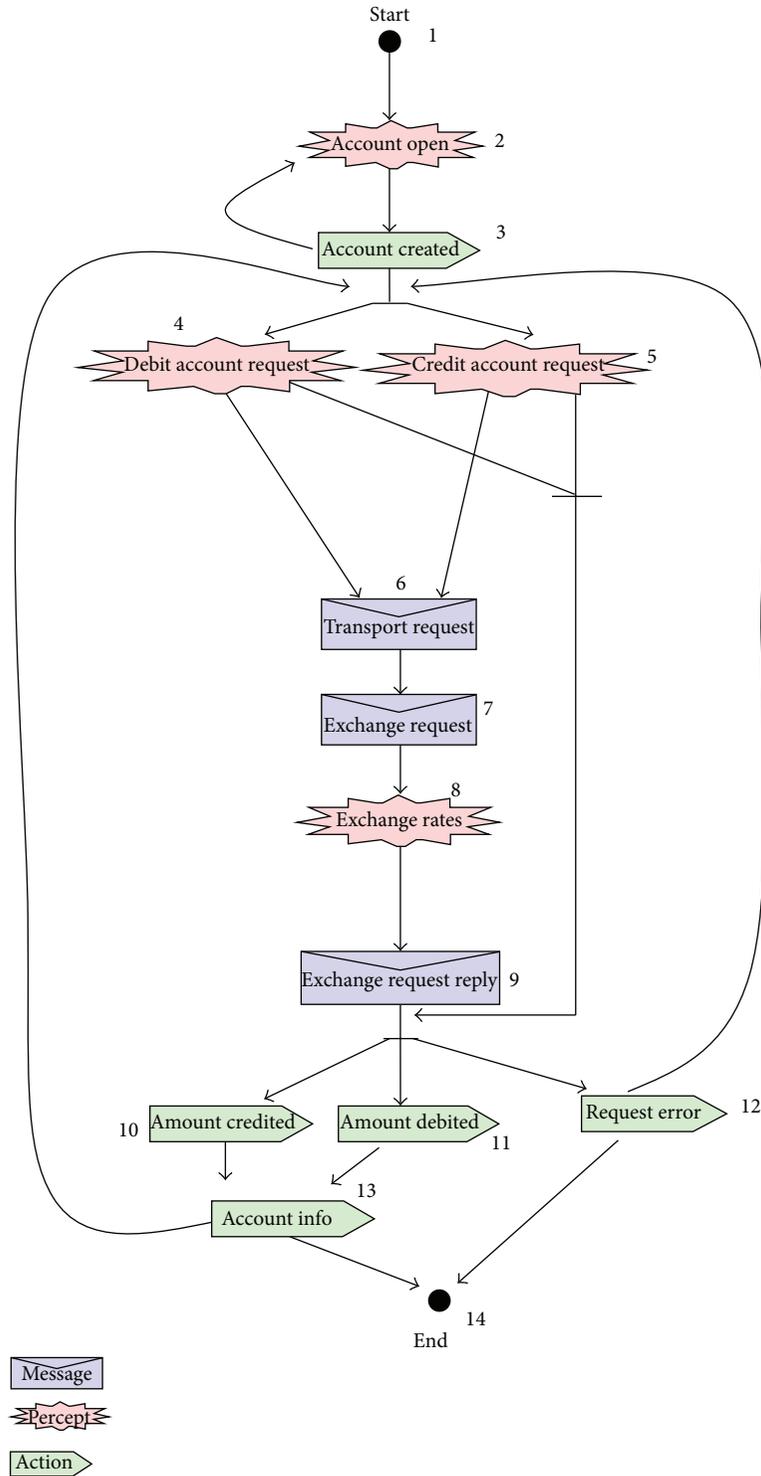


FIGURE 7: Protocol graph for account operation protocol diagram.

function which calculates all paths from protocol graph and coverage criteria are used to extract relevant path from all paths.

Protocol graph contains the sequence of percepts, action, and message as described in corresponding protocol diagram of a certain interaction protocol. Figure 9 shows the screen

shot of our tool which automates the test path generation from design artifact like protocol graph.

6. Conclusion and Future Directions

In this paper, we have proposed a novel approach to test multiagent systems based on design artifacts following

```

def find_all_paths (names, graph, start, end, pathof=
    ["start", "end", "message", "action", "precept"], path=[]):
    path = path + [start]
    if start == end:
        return [path]
    if not graph.has_key (start):
        return []
    paths = []
    for node in graph [start]:
        if names [node][1] in pathof:
            if path.count (node) < 2:
                newpaths = find_all_paths (names, graph, node, end, pathof, path)
                for newpath in newpaths:
                    paths.append (newpath)
    return paths

```

CODE 1: Find_all_paths function.

```

start account operation protocol
actor A user
agent B BankAccount agent
actor C account owner
agent D CurrencyExchange agent
agent E Communicator agent
box loop
    percept A B account open
    action B A account created
end loop
box loop
box alternative
    percept C B Debit account request
next
    percept C B Credit account request
end alternative
box opt
    message B E TransportRequest
    message E D exchangeRequest
    percept A D exchange rates
    message D E exchangeRequest reply
end opt
box alternative
    action B C amount Debited
    action B C Account Info
next
    action B C amount credited
    action B C Account Info
next
    action B C Request Error
end alternative
end loop
finish

```

CODE 2: AUML description of account operation protocol.

TABLE 2: Test path generator tool input file.

14	1,2	12,14
start, start	2,3	12,4
account open, precept	3,2	12,5
account created, action	3,4	13,14
debit account request, precept	3,5	13,4
credit account request, precept	4,6	13,5
transport request, message	4,10	
exchange request, message	4,11	
exchange rates, precept	4,12	
exchange request	5,6	
reply, message	5,10	
amount credited, action	5,11	
amount debited, action	5,12	
request error, action	6,7	
account info, action	7,8	
end, end	8,9	
27	9,10	
	9,11	
	9,12	
	10,13	
	11,13	

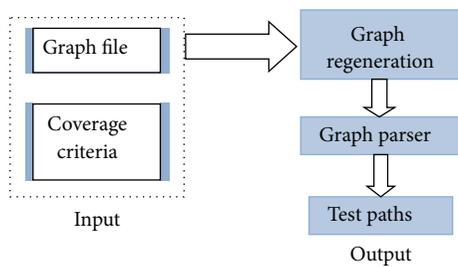


FIGURE 8: Test path generator tool process.

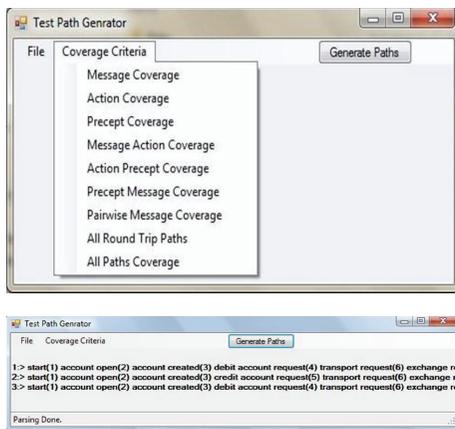


FIGURE 9: Test path generator tool.

Prometheus methodology. Testing a multiagent system is a challenging task due to dynamic behavior of agents. Agents interact with each other and actors via some protocol. Interaction protocol diagram contains all sorts of interactions between agents and actor like message, action, and precepts. We have proposed a testing framework which transforms the

interaction protocol diagram to a test model named protocol graph. The previously proposed protocol graph has been extended to include action and precepts along with messages.

Messages are passed between agents and precepts/actions are used as the interaction mechanism between agents and actors. We have identified different coverage criteria which include nodes and arcs of the protocol graph. These coverage criteria are used to generate test paths.

For future work, we plan to automate the generation of test data to execute the test paths. Test cases then will be applied to autonomous agents and will uncover the interaction faults. Evaluation to testing technique will show the benefits of applying novel approach in testing of autonomous agents with help of design artifacts following Prometheus methodology.

Conflict of Interests

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

References

- [1] M. Winikoff and S. Cranefield, "On the testability of BDI agents," in *Proceedings of the European Workshop on Multi-Agent Systems*, 2010.
- [2] L. Padgham and M. Winikoff, "Prometheus: a methodology for developing intelligent agents," in *Agent-Oriented Software Engineering III*, vol. 2585 of *Lecture Notes in Computer Science*, pp. 174–185, Springer, Berlin, Germany, 2003.
- [3] S. Munroe, T. Miller, R. A. Belecheanu, M. Pěchouček, P. Mcburney, and M. Luck, "Crossing the agent technology chasm: lessons, experiences and challenges in commercial applications of agents," *Knowledge Engineering Review*, vol. 21, no. 4, pp. 345–392, 2006.
- [4] L. Padgham and M. Winikoff, *Developing Intelligent Agent Systems: A Practical Guide*, John Wiley & Sons, New York, NY, USA, August 2004.
- [5] L. Padgham, J. Thangarajah, and M. Winikoff, "The prometheus design tool—a conference management system case study," in *Proceedings of the 8th International Conference on Agent-Oriented Software Engineering VIII*, pp. 197–211, 2008.
- [6] J. Thangarajah, L. Padgham, and M. Winikoff, "Prometheus design tool," in *Proceedings of the 4th International Conference on Autonomous Agents and Multi agent Systems (AAMAS '05)*, Utrecht, The Netherlands, July 2005.
- [7] S. U. Rehman and A. Nadeem, "AgentSpeak (L) bases testing of autonomous agents," in *Proceedings of the International Conference on Advanced Software Engineering & Its Applications (ASEA '11)*, pp. 11–20, Science and Engineering Research Support Society, Springer, Jeju Island, Korea, 2011.
- [8] T. Miller, L. Padgham, and J. Thangarajah, "Test coverage criteria for agent interaction testing," in *Agent-Oriented Software Engineering (AOSE) Workshop at AAMAS*, 2010.
- [9] M. Wooldridge, N. R. Jennings, and D. Kinny, "The Gaia methodology for agent-oriented analysis and design," *Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 3, pp. 285–312, 2000.

- [10] S. A. Deloach, M. F. Wood, and C. H. Sparkman, "Multiagent systems engineering," *International Journal of Software Engineering and Knowledge Engineering*, vol. 11, no. 3, pp. 231–258, 2001.
- [11] S. A. DeLoach, "Multiagent systems engineering: a methodology and language for designing agent systems," in *Proceedings of the Agent-Oriented Information Systems (AOIS '99)*, Seattle, Wash, USA, May 1998.
- [12] G. Caire, F. Leal, P. Chainho et al., "Agent oriented analysis using MESSAGE/UML," in *Proceedings of the 2nd International Workshop on Agent-Oriented Software Engineering (AOSE '01)*, M. Wooldridge, P. Ciancarini, and G. Weiss, Eds., pp. 101–108, Montreal, Canada, May 2001.
- [13] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini, "Troops: an agent-oriented software development methodology," Tech. Rep. DIT-02-0015, University of Trento, Department of Information and Communication Technology, 2002.
- [14] N. Glaser, *Contribution to knowledge modelling in a multi-agent framework (the CoMoMAS approach) [Ph.D. thesis]*, L'Universite Henri Poincare, 1996.
- [15] A. Omicini, "Societies and infrastructures in the analysis and design of agent-based systems," in *Proceedings of the 1st International Workshop on Agent-Oriented Software Engineering (AOSE '00)*, P. Ciancarini and M. J. Wooldridge, Eds., vol. 1957 of *Lecture Notes in Artificial Intelligence*, pp. 185–194, Springer, 2001.
- [16] F. M. T. Brazier, B. M. Dunin-Keplicz, N. R. Jennings, and J. Treur, "Desire: modelling multi-agent systems in a compositional formal framework," *International Journal of Cooperative Information Systems*, vol. 6, no. 1, pp. 67–94, 1997.
- [17] C. Iglesias, M. Garijo, J. C. Gonzales, and J. R. Velasco, "Analysis and design of multiagent systems using MAS-CommonKADS," in *Intelligent Agents IV Agent Theories, Architectures, and Languages: Proceedings of the 4th International Workshop, ATAL'97 Providence, Rhode Island, USA, July 24–26, 1997*, M. P. Singh, A. Rao, and M. J. Wooldridge, Eds., vol. 1365 of *Lecture Notes in Computer Science*, pp. 313–326, Springer, Berlin, Germany, 1998.
- [18] K. H. Dam, *Evaluating and comparing agent-oriented software engineering methodologies [Ph.D. thesis]*, School of Computer Science and Information Technology, RMIT University, Melbourne, Australia, 2003.
- [19] A. S. Rao, "AgentSpeak (L): BDI agents speak out in a logical computable language," in *Proceedings of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW '96)*, W. van de Velde and W. J. Perram, Eds., vol. 1038 of *Lecture Notes in Computer Science*, pp. 42–55, Springer.
- [20] S. J. Juneidi and G. A. Vouros, "Survey and evaluation of agent-oriented software engineering main approaches," *International Journal of Modelling and Simulation*, 2010.
- [21] A. Spillner, "Test criteria and coverage measures for software integration testing," *Software Quality Journal*, vol. 4, no. 4, pp. 275–286, 1995.
- [22] M. Utting and B. Legeard, *Practical Model-Based Testing: A Tools Approach*, Morgan-Kaufmann, San Francisco, Calif, USA, 2007.
- [23] C. K. Low, T. Y. Chen, and R. Rönquist, "Automated test case generation for BDI agents," *Autonomous Agents and Multi-Agent Systems*, vol. 2, no. 4, pp. 311–332, 1999.
- [24] Z. Zhang, J. Thangarajah, and L. Padgham, "Model based testing for agent systems," in *Software and Data Technologies, Communications in Computer and Information Science*, J. Filipe, B. Shishkov, M. Helfert, and L. A. Maciaszek, Eds., vol. 22, pp. 399–413, Springer, Berlin, Germany, 2009.
- [25] M. Zheng and V. S. Alagar, "Conformance testing of BDI properties in agent-based software system," in *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC '05)*, December 2005.
- [26] C. D. Nguyen, S. Miles, A. Perini, P. Tonella, M. Harman, and M. Luck, "Evolutionary testing of autonomous software agents," *Autonomous Agents and Multi-Agent Systems*, vol. 25, no. 2, pp. 260–283, 2012.
- [27] C. D. Nguyen, A. Perinirini, and P. Tonella, "Automated continuous testing of multi-agent systems," in *Proceedings of the 5th European Workshop on Multi-Agent Systems (EUMAS '07)*, 2007.
- [28] Jack intelligent agents, <http://aosgrp.com/products/jack/>.
- [29] M.-P. Huget and J. Odell, "Representing agent interaction protocols with agent UML," in *Proceedings of the 5th International Workshop on Agent Oriented Software Engineering (AOSE '04)*, July 2004.
- [30] RMIT, Agent Research Group, Australia, <http://www.cs.rmit.edu.au/agents/pdt/tutorial/Tutorial.html>.

Research Article

A Novel Clustering Algorithm Inspired by Membrane Computing

Hong Peng,¹ Xiaohui Luo,² Zhisheng Gao,¹ Jun Wang,³ and Zheng Pei¹

¹ Center for Radio Administration and Technology Development, Xihua University, Chengdu 610039, China

² School of Mathematics and Computer Engineering, Xihua University, Chengdu 610039, China

³ School of Electrical and Information Engineering, Xihua University, Chengdu 610039, China

Correspondence should be addressed to Hong Peng; ph.xhu@hotmail.com

Received 10 June 2014; Accepted 7 September 2014

Academic Editor: Shifei Ding

Copyright © 2015 Hong Peng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

P systems are a class of distributed parallel computing models; this paper presents a novel clustering algorithm, which is inspired from mechanism of a tissue-like P system with a loop structure of cells, called membrane clustering algorithm. The objects of the cells express the candidate centers of clusters and are evolved by the evolution rules. Based on the loop membrane structure, the communication rules realize a local neighborhood topology, which helps the coevolution of the objects and improves the diversity of objects in the system. The tissue-like P system can effectively search for the optimal partitioning with the help of its parallel computing advantage. The proposed clustering algorithm is evaluated on four artificial data sets and six real-life data sets. Experimental results show that the proposed clustering algorithm is superior or competitive to k -means algorithm and several evolutionary clustering algorithms recently reported in the literature.

1. Introduction

Data clustering is a fundamental conceptual problem in data mining, which describes the process of grouping data into classes or clusters such that the data in each cluster share a high degree of similarity while being very dissimilar to data from other clusters [1]. Over the past years, a large number of clustering algorithms have been proposed [2–4], which can be divided roughly in two categories: hierarchical and partitional. Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones or splitting larger clusters. Partitional clustering attempts to directly decompose a data set into several disjointed clusters based on similarity measure, for example, mean square error (MSE). Clustering algorithms have been used in a wide variety of areas, such as pattern recognition, machine learning, image processing, and web mining [5, 6]. In the present study, k -means algorithm [7, 8] has received wide attention because of the following two reasons: (i) k -means has been recently elected and listed among the top most influential data mining algorithms [9] and (ii) it is at the same time very simple and

quite scalable, as it has linear asymptotic running time with respect to any variable of the problem. However, k -means is sensitive to the initial centers and easy to get stuck at the local optimal solutions. Moreover, k -means takes large time cost to find the global optimal solution when the number of data points is large.

In recent years, some evolutionary algorithms have been introduced to overcome the shortcomings of k -means algorithm because of their global optimization capability. Several genetic algorithms- (GA-) based clustering algorithms have been proposed in the literature [10–14]. However, most of GA-based clustering algorithms can suffer from the degeneracy when numerous chromosomes represent the same solution. The degeneracy can lead to inefficient coverage of the search space as the same configurations of clusters are repeatedly explored. To overcome the shortcoming, particle swarm optimization- (PSO-) based or ant colony optimization- (ACO-) based clustering algorithms have been proposed. Kao et al. have proposed a hybrid technique based on combining the k -means and PSO for cluster analysis [15]. Shelokar et al. have introduced an evolutionary algorithm based on ACO for

clustering problem [16]. Niknam and Amiri have presented a hybrid evolutionary optimization algorithm based on the combination of PSO and ACO for solving the clustering problem [17].

The aim of membrane computing is to abstract computing ideas (data structures, operations with data, ways to control operations, computing models, etc.) from the structure and the functioning of a single cell and from complexes of cells, such as tissues and organs including the brain. There are three main classes of P systems investigated: cell-like P systems (based on a cell-like (hence hierarchical) arrangement of membranes delimiting compartments where multisets of chemicals evolve according to given evolution rules) [18], tissue-like P systems (instead of hierarchical arrangement of membranes, consider arbitrary graphs as underlying structures, with membranes placed in the nodes while edges correspond to communication channels) [19], and neural-like P systems [20]. Many variants of all these systems have been considered, for example, [21, 22] for cell-like P systems, [23, 24] for tissue-like P systems, and [25–30] for neural-like P systems. An overview of the field can be found in [31], with up-to-date information available at the membrane computing website (<http://ppage.psystems.eu/>). These efforts have addressed the parallel computing advantage of P systems as well as the high effectiveness of solving a variety of difficult problems; especially, P systems can solve a number of NP-hard problems in linear or polynomial time complexity [32] and even solve PSPACE problems in a feasible time [33, 34]. Moreover, membrane algorithms have demonstrated a powerful global optimization performance [35–37].

This paper focuses on application of membrane computing to data clustering. Our motivation is applying the specially designed elements and inherent mechanisms of P systems to realize a novel clustering algorithm, called the membrane clustering algorithm.

2. Data Clustering Problem

Clustering is the process of recognizing natural groups or clusters from a data set based on some similarity measure. Suppose that data set D has n sample points, x_1, x_2, \dots, x_n , $x_i \in R^d$ ($i = 1, 2, \dots, n$), and is partitioned into k clusters, C_1, C_2, \dots, C_k . Denote by z_1, z_2, \dots, z_k the corresponding centers. Usually, partitioning clustering algorithm searches for the optimal centers in the solution space according to some clustering measure in order to solve data clustering problem. A commonly used clustering measure is

$$M(C_1, C_2, \dots, C_k) = \sum_{i=1}^n \sum_{j=1}^k w_{ij} \|x_i - z_j\|, \quad (1)$$

where w_{ij} is the associate weight of point x_i with cluster j , which will be either 1 or 0 (if point x_i is allocated to cluster j , w_{ij} is 1, otherwise 0).

The clustering process, separating the objects into the clusters, is realized as an optimization problem. The goal of

the optimization problem is to find the optimal centers by minimizing objective function (1):

$$\min_{z_1, z_2, \dots, z_k} J = \min_{z_1, z_2, \dots, z_k} M(C_1, C_2, \dots, C_k). \quad (2)$$

In addition, the M value will be used to evaluate objects in the proposed clustering algorithm. If the M value of an object is the smaller one, the object is the better; otherwise, it is worse.

3. Proposed Membrane Clustering Algorithm

In this section the proposed membrane clustering algorithm is discussed in detail, which is inspired by the mechanism of membrane computing. A tissue-like P system with a loop structure of cells is designed as its optimization framework. The tissue-like P system with a loop structure of cells can be described as the following construct:

$$\Pi = (Z_1, \dots, Z_q, R_1, \dots, R_q, R', i_o), \quad (3)$$

where

- (1) Z_i ($1 \leq i \leq q$) is the set of m objects in cell i ;
- (2) R_i ($1 \leq i \leq q$) is the set of evolution rules in cell i , which contains three evolution rules: selection, crossover, and mutation rules;
- (3) R' is finite set of communication rules with the following forms:
 - (i) antiport rule: $(i, Z/Z', j)$, $i, j = 1, 2, \dots, q, i \neq j$. The rule is used to communicate the objects between a cell and its two adjacent cells;
 - (ii) symport rule: $(i, Z/\lambda, 0)$, $i = 1, 2, \dots, q$. The rule is used to communicate the objects between cell and the environment.
- (4) i_o indicates the output region of the system.

Figure 1 shows membrane structure of the tissue-like P system, which consists of q cells. The q cells are labeled by $1, 2, \dots, q$, respectively. The region labeled by 0 is the environment and is also output region of the system. The directed lines in Figure 1 indicate the communication of objects between the q cells. Moreover, the q cells will be arranged as a loop topology based on the communication rules described below. As usual in P system, the q cells, as parallel computing units, will run independently. In addition, the environment always stores the best object found so far in the system. When the system halts, the object in the environment will be regarded as the output of the whole system.

The role of the tissue-like P system is to evolve the optimal centers of clusters for a data set; thus each object in cells will express a group of (candidate) centers. Thus, each object in cells is considered as a $(k \times d)$ -dimensional real vector of the form

$$Z = (z_{11}, z_{12}, \dots, z_{1d}, \dots, z_{i1}, z_{i2}, \dots, z_{id}, \dots, z_{k1}, z_{k2}, \dots, z_{kd}), \quad (4)$$

where $z_{i1}, z_{i2}, \dots, z_{id}$ are d components of i th cluster center z_i , $i = 1, 2, \dots, k$. For simplicity, suppose that each cell has the same number of objects, which is denoted by m .

Initially, the system will randomly generate m initial objects for each cell. When an initial object Z is generated, $(k \times d)$ random real numbers are produced repeatedly to form it with the constraint of

$$A_1 \leq z_{i1} \leq B_1, \dots, A_j \leq z_{ij} \leq B_j, \dots, A_d \leq z_{id} \leq B_d, \quad (5)$$

where A_j and B_j are lower bound and upper bound of j th dimensional component of data points, respectively, $j = 1, 2, \dots, d$.

As usual, the tissue-like P system has two mechanisms: evolution and communication mechanisms. The two mechanisms will be described as follows.

3.1. Evolution Mechanism. The role of evolution rules is to evolve the objects in cells to generate new objects used in next computing step. During the evolution, each cell maintains the same size (the number of objects). In this work, three known genetic operations (selection, crossover, and mutation) [38, 39] are used as the evolution rules in cells. In a computing step, all objects (located in object pool) in each cell and the best objects (located in external pool) from its two adjacent cells constitute a matching pool. The objects in external pool are actually the best objects communicated from its two adjacent cells in previous computing step. The objects in matching pool will be evolved by executing selection, crossover, and mutation operations in turn. In order to maintain the size of objects in each cell, truncation operation is used to constitute new object pool according to the M values of objects. The objects in new object pool will be regarded as the objects to be evolved in next computing step. Figure 2 shows the evolution procedure of objects in a cell.

In this work, selection operation uses usual rotating wheel method, while crossover operation uses single-point crossover in which the position of crossover point is determined according to crossover probability p_c [39]. The single-point mutation is used to realize the mutations of objects. If v is a mutation point determined according to mutation probability p_m , its value becomes, after mutating,

$$v' = \begin{cases} v \pm 2\delta v, & v \neq 0 \\ v \pm 2\delta, & v = 0, \end{cases} \quad (6)$$

where the signs “+” or “-” occur with equal probability, and δ is real number in the range $[0, 1]$, generated with uniform distribution.

3.2. Communication Mechanism. The communication mechanism is used to exchange the objects between each cell and its two adjacent cells and update the best object found so far in the environment. The communication mechanism is realized by communication rules of two types: antiport rule $(i, Z/Z', j)$, which indicates that object Z is communicated from cell i to cell j and object Z' is communicated from cell j to cell i , and symport rule $(i, Z/\lambda, 0)$, which indicates that object Z is communicated from cell i to the environment.

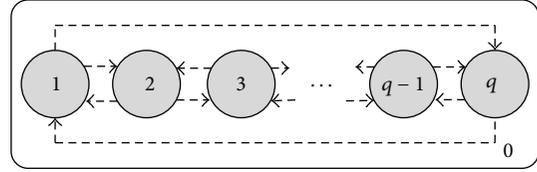


FIGURE 1: Membrane structure of the designed tissue-like P system.

The communication rules impliedly indicate the connection relationship between cells. Figure 3 shows the communication relation of objects between cells in the designed tissue-like P system. From a logical point of view, the communication relation shows that the cells form a loop topology, shown in Figure 3(a). Meanwhile, this also reflects a neighborhood structure of the communication of objects; namely, each cell only exchanges and shares the objects with its two adjacent cells, shown in Figure 3(b). After the objects are evolved, each cell (such as cell i) transmits its several best objects into adjacent cells (such as cells $i - 1$ and $i + 1$) and retrieves several best objects from adjacent cells (such as cells $i - 1$ and $i + 1$) by using the communication rule, constituting the matching pool of objects in next computing step. The special logical structure can bring the following benefits.

- (1) The coevolution of objects in the q cells can accelerate the convergence of the proposed clustering algorithm.
- (2) The object sharing mechanism of the local neighborhood structure can enhance the diversity of objects in the entire system.

The communication of objects not only occurs between cells, but also appears between cell and the environment. The global best object found so far in whole system is stored always in the environment. After objects are evolved, each cell communicates its best object found in current computing step into the environment to update the global best object. The update strategy is that if $f(Z) < f(G)$ then $G = Z$; otherwise, G retains unchanged, where Z is the current best object, G is the global best object, and $f(\cdot)$ is the fitness function (M value).

As usual in P system, the q cells, as parallel computing units, will run independently. In addition, the environment always stores the best object found so far in the system. In this work, maximum execution step number is used as the halting condition of the tissue-like P system; that is, the tissue-like P system will continue to run until it reaches the maximum execution step number. When the system halts, the object in the environment will be regarded as the output of whole system, namely, the found optimal centers.

Based on the tissue-like P system described above, the proposed membrane clustering algorithm is summarized in Algorithm 1.

4. Simulation Experiments

The proposed membrane clustering algorithm is evaluated on ten data sets and compared with classical k -means algorithm and several clustering algorithms based on evolutionary

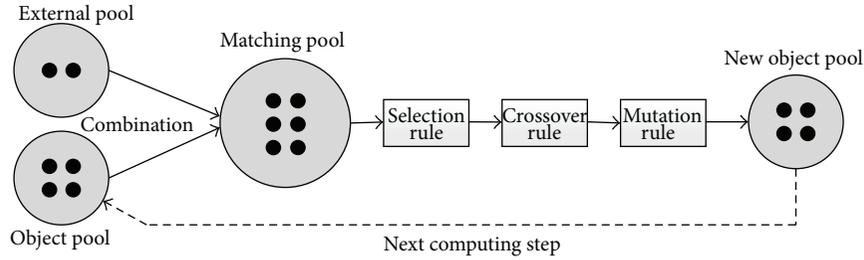


FIGURE 2: Evolution procedure of objects in a cell.

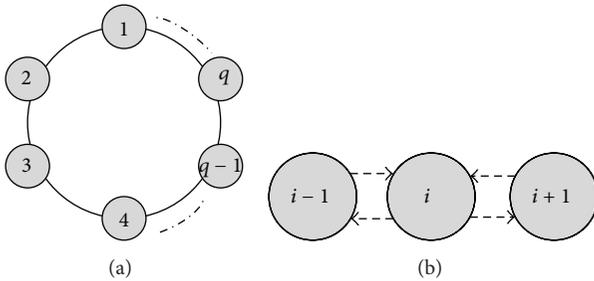


FIGURE 3: A loop topology structure of cells and the communication relation between adjacent cells.

TABLE 1: Properties of the test data sets.

	Data	Input	Class
<i>AD_5_2</i>	250	2	5
<i>Data_9_2</i>	900	2	9
<i>Square_4</i>	1000	2	4
<i>Sym_3_22</i>	600	2	3
<i>Iris</i> ,	150	4	3
<i>BreastCancer</i>	683	9	2
<i>Newthyroid</i>	215	5	3
<i>LungCancer</i>	32	56	3
<i>Wine</i>	178	13	3
<i>LiveDisorder</i>	345	6	2

algorithms, including GA [10], PSO [15], and ACO [16]. In order to test the robustness of these clustering algorithms, we repeat the experiments 50 times for each data set.

In the experiments, two kinds of data sets are used to evaluate these clustering algorithms. First is the four manually generated data sets used in the existing literatures, *AD_5_2*, *Data_9_2*, *Square_4*, and *Sym_3_22*, shown in Figure 4. Second is the six real-life data sets provided in UCI [40], including the *Iris*, *BreastCancer*, *Newthyroid*, *LungCancer*, *Wine*, and *LiveDisorder*. The sizes of the data sets can be found in Table 1.

The proposed membrane clustering algorithm will be compared with *k*-means and three evolutionary clustering algorithms recently reported in the literature, including GA, PSO, and ACO. These algorithms are implemented in Matlab 7.1 according to the following parameters.

- (1) Tissue-like P systems. Each cell contains 100 objects and communicates its first five best objects into two adjacent cells. The maximum computing step number is chosen to be 200. In the implementation, evolution rules use the adaptive crossover probability p_c and mutation probability p_m . In order to study performances of tissue-like P systems of different degrees, four cases are considered in the experiments: $q = 4, 8, 16, 20$.
- (2) GA [10]. In the rotating wheel method, single-point crossover and single-point mutation are used, where the crossover and mutation probabilities, p_c and p_m , are chosen to be 0.8 and 0.001, respectively. Let the population size be $N_{\text{swarm}} = 100$ and let maximum iteration number be $t_{\text{max}} = 200$.
- (3) PSO [15]. The w uses a linear decreasing inertia weight, where $w_{\text{min}} = 0.4$ and $w_{\text{max}} = 0.9$; $c_1 = c_2 = 2.0$, the population size $NP = 100$, and maximum iteration number is 200.
- (4) ACO [16]. The best parameter values are $\gamma_1 = \gamma_2 = 1.0$ and $\rho = 0.99$.

In the experiments, we realize four tissue-like P systems with degrees 4, 8, 16, and 20, respectively. The aim is to evaluate the effects of the number of cells (i.e., different degrees) on clustering quality. The four tissue-like P systems are applied to find out the optimal centers for the ten data sets, respectively. In this work, the M value is also used to measure the clustering quality of each clustering algorithm. Considering that the evolution rules in the designed tissue-like P system include stochastic mechanism, we independently execute the tissue-like P systems of the four degrees 50 times on each data set and then compute their mean values and standard deviations of the 50 runs. The mean values are used to illustrate the average performance of the algorithms while standard deviations indicate their robustness. Table 2 provides experimental results of the tissue-like P systems of four degrees on ten data sets, respectively. The results of degrees 16 and 20 are better than those of the other two degrees, namely, lower mean values and smaller standard deviations. It can be further observed that the tissue-like P system with degree 16 obtains the smallest mean values and standard deviations on most of data sets. The results illustrate that the tissue-like P system with degree 16 has good clustering quality and high robustness.

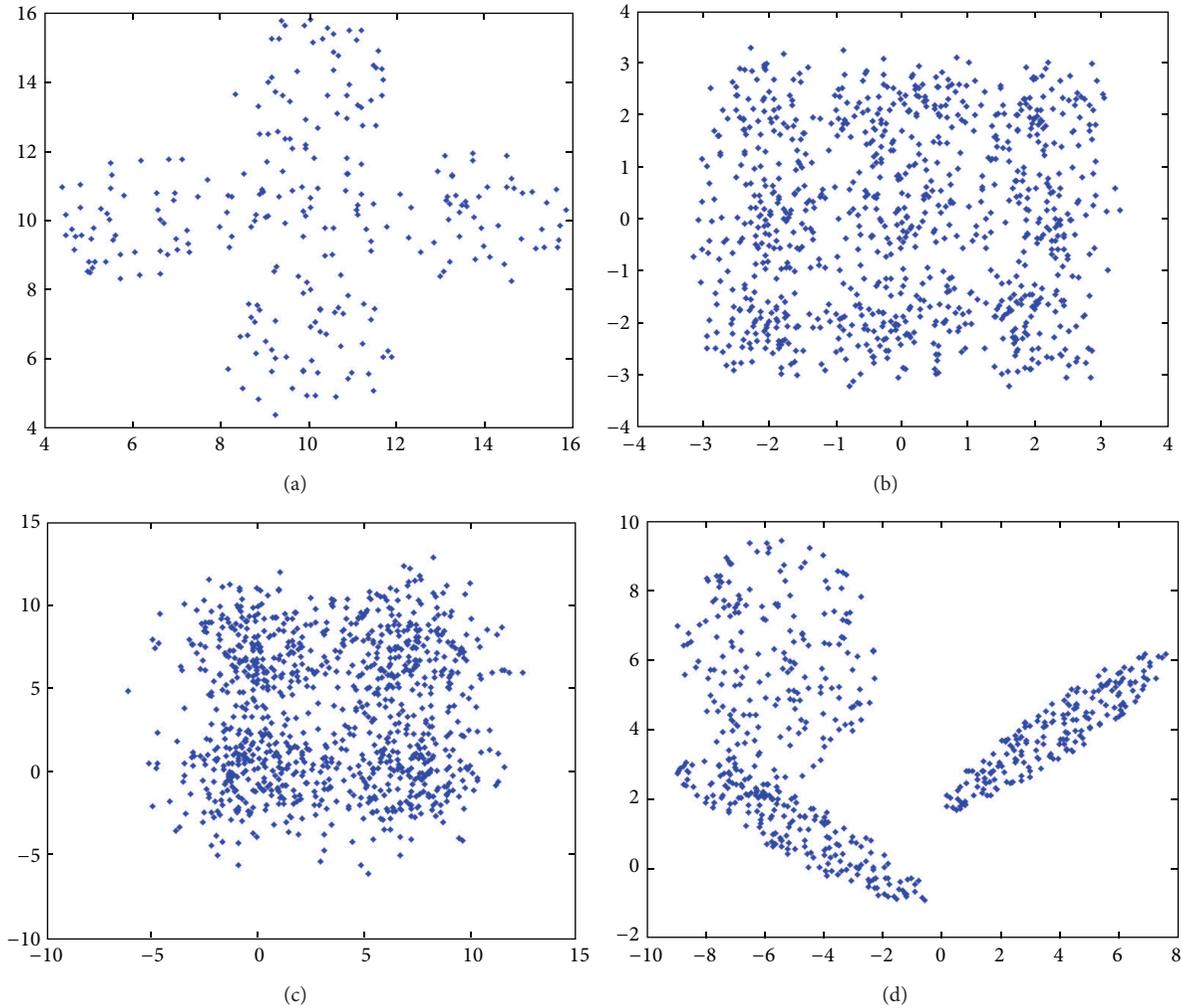


FIGURE 4: Four artificial data sets: (a) *AD_5_2*; (b) *Data_9_2*; (c) *Square_4*; (d) *Sym_3_22*.

TABLE 2: The performance comparisons of tissue-like P systems of different degrees.

Data set	4 cells	8 cells	16 cells	20 cells
<i>AD_5_2</i>	327.01 ± 0.0944	326.94 ± 0.0277	326.44 ± 0.0105	326.94 ± 0.0312
<i>Data_9_2</i>	591.11 ± 0.1331	591.12 ± 0.0510	591.06 ± 0.0280	591.03 ± 0.0537
<i>Square_4</i>	2380.25 ± 0.1334	2380.26 ± 0.0956	2379.74 ± 0.0189	2380.00 ± 0.0729
<i>Sym_3_22</i>	1248.31 ± 0.3156	1248.11 ± 0.0554	1247.72 ± 0.0105	1248.05 ± 0.0333
<i>Iris</i>	96.84 ± 0.0751	96.81 ± 0.0435	96.75 ± 0.0428	96.77 ± 0.0361
<i>BreastCancer</i>	2974.24 ± 1.5431	2971.14 ± 1.5287	2970.24 ± 1.1225	2969.06 ± 1.0970
<i>Newthyroid</i>	1885.69 ± 14.377	1870.37 ± 1.7355	1869.29 ± 0.9215	1871.18 ± 2.2496
<i>LungCancer</i>	124.69 ± 0.0045	124.69 ± 0.0012	124.69 ± 0.0011	124.69 ± 0.0035
<i>Wine</i>	16309.01 ± 2.5053	16303.42 ± 1.9595	16292.25 ± 0.1529	16301.97 ± 2.8563
<i>LiveDisorder</i>	9860.54 ± 5.7239	9859.02 ± 0.5116	9851.78 ± 0.0347	9857.08 ± 0.1043

In order to further evaluate clustering performance, the proposed membrane clustering algorithm is compared with GA-based, PSO-based, and ACO-based clustering algorithms as well as classical *k*-means algorithm. Table 3 gives the comparison results of the tissue-like P system of degree 16 with other four clustering algorithms on the ten data sets,

respectively. The comparison results show that the tissue-like P system provides the optimum average value and smallest standard deviation in comparison to those of other algorithms. For instance, the results obtained on the *AD_5_2* show that the tissue-like P system converges to the optimum of 326.4478 at almost times and PSO reaches to 326.44 in most

Input parameters: Data set, D , the number of clusters, k , the number of cell, q , the number of objects in each cell, m , maximum execution step number, S_{\max} , crossover rate, p_c , and mutation rate, p_m .

Output results: the optimal centers, G .

Step 1. Initialization

```

for  $i = 1$  to  $q$ 
  for  $j = 1$  to  $m$ 
    Generate  $j$ th initial object for cell  $i$ ,  $Z_{ij}$ ;
    Partition all data points into clusters,  $C_1, C_2, \dots, C_k$ ;
    Compute the  $M$  value of the object,  $M_{ij}$ ;
  end for
end for
Fill the global best object  $G$  using the best of all initial objects;
Set computing step  $s = 0$ ;

```

Step 2. Object evolution in cells

```

for each cell  $i$  ( $i = 1, 2, \dots, q$ ) in parallel do
  Evolve all object  $Z_{ij}$  ( $j = 1, 2, \dots$ ) in its mating pool using evolution rules;
  Use truncation operation to maintain its  $m$  best objects;
  for  $j = 1$  to  $m$ 
    Partition all data points into clusters,  $C_1, C_2, \dots, C_k$ ;
    Compute the  $M$  value of the object,  $M_{ij}$ ;
  end for
end for

```

Step 3. Object communication between cells

```

for each cell  $i$  ( $i = 1, 2, \dots, q$ ) in parallel do
  Transmit better objects in cell  $i$  to its two adjacent cells;
  Receive better objects from its two adjacent cells into its mating pool;
  Update  $G$  using the best object in cell  $i$ ;
end for

```

Step 4. Halt condition judgment

```

if  $s \leq S_{\max}$  is satisfied
   $s = s + 1$ ;
  goto Step 2;
end if
The system exports the global best object  $G$  in the environment and halts;

```

ALGORITHM 1: Membrane clustering algorithm: a clustering algorithm based on tissue-like P systems.

TABLE 3: The results obtained by the algorithms for 50 runs on the ten data sets.

Data set	P systems	GA	PSO	ACO	k -means
<i>AD_5_2</i>	326.44 ± 0.0105	332.31 ± 0.4792	326.44 ± 0.0128	326.45 ± 0.0344	332.47 ± 3.1286
<i>Data_9_2</i>	591.06 ± 0.0280	593.72 ± 0.2635	591.14 ± 0.0303	591.42 ± 0.0372	623.57 ± 3.1326
<i>Square_4</i>	2379.74 ± 0.0189	2380.33 ± 0.6319	2379.74 ± 0.0226	2379.79 ± 0.0428	2386.00 ± 4.5217
<i>Sym_3_22</i>	1247.72 ± 0.0105	1249.36 ± 1.2163	1247.72 ± 0.0149	1247.75 ± 0.0315	1255.45 ± 3.8725
<i>Iris</i>	96.75 ± 0.0428	99.83 ± 5.5239	97.23 ± 0.3513	97.25 ± 0.4152	104.11 ± 12.4563
<i>BreastCancer</i>	2970.24 ± 1.1225	3249.26 ± 229.734	3050.04 ± 110.801	3046.06 ± 90.500	3251.21 ± 251.143
<i>Newthyroid</i>	1869.29 ± 0.9215	1875.11 ± 13.5834	1872.51 ± 11.0923	1872.56 ± 11.1045	1886.25 ± 16.2189
<i>LungCancer</i>	124.69 ± 0.0011	129.52 ± 4.4961	127.23 ± 1.1528	127.31 ± 1.2936	139.40 ± 7.3136
<i>Wine</i>	16292.25 ± 0.1529	16298.42 ± 2.1523	16292.25 ± 0.1531	16292.25 ± 0.1672	16312.43 ± 9.4269
<i>LiveDisorder</i>	9851.73 ± 0.0347	9856.14 ± 1.9523	9851.73 ± 0.0356	9851.74 ± 0.0692	9868.32 ± 7.9274

of runs, while ACO, GA, and k -means attain 326.45, 322.31, and 332.47, respectively. The standard deviations of M values for the tissue-like P system, PSO, and ACO are 0.0105, 0.0128, and 0.0344, respectively, which are significantly smaller than the other two algorithms. For the results on the *Iris*, the optimum value is 96.75, which is obtained in most of runs of

the tissue-like P system; however, the other four algorithms fail to attain the value even once within 50 runs. The results on the *Newthyroid* also show that the tissue-like P system provides the optimum value of 1869.29 while the PSO, ACO, GA, and k -means obtain 1872.51, 1872.56, 1875.11, and 1886.25, respectively. In addition, the tissue-like P system obtains

TABLE 4: The results of P values produced by Wilcoxon's rank sum test.

P systems	GA	PSO	ACO	k -means
<i>AD_5_2</i>	$4.1321e-3$	$2.3256e-2$	$2.6351e-2$	$3.4273e-3$
<i>Data_9_2</i>	$4.0536e-3$	$2.2734e-2$	$2.7932e-2$	$3.2963e-3$
<i>Square_4</i>	$3.9275e-3$	$2.1482e-2$	$2.8175e-2$	$3.5387e-3$
<i>Sym_3_22</i>	$3.7894e-3$	$2.4357e-2$	$2.8529e-2$	$3.4416e-3$
<i>Iris</i>	$4.0968e-3$	$3.5823e-2$	$3.2634e-2$	$3.6528e-3$
<i>BreastCancer</i>	$3.9235e-3$	$2.9527e-2$	$2.8192e-2$	$3.4632e-3$
<i>Newthyroid</i>	$3.8864e-3$	$2.5162e-2$	$2.9355e-2$	$3.5381e-3$
<i>LungCancer</i>	$3.8575e-3$	$2.7346e-2$	$2.7358e-2$	$3.5138e-3$
<i>Wine</i>	$3.7639e-3$	$3.2189e-2$	$2.7963e-2$	$3.6348e-3$
<i>LiveDisorder</i>	$3.8398e-3$	$2.4671e-2$	$2.8846e-2$	$3.5822e-3$

smallest standard deviation on each data set in comparison to the other four algorithms, which illustrates that it has high robustness.

Wilcoxon's rank sum test is a nonparametric statistical significance test for independent samples. The statistical significance test has been conducted at the 5% significance level in the experiments. We create five groups for the ten data sets, which are corresponding to the five clustering algorithms (tissue-like P system, GA, PSO, ACO, and k -means), respectively. Each group consists of the M values produced by 50 consecutive runs of the corresponding algorithms. In order to illustrate if the goodness is statistically significant, we have completed a statistical significance test for these clustering algorithms. Table 4 gives the P values provided by Wilcoxon's rank sum test for comparison of two groups (one group corresponding to the tissue-like P system and another group corresponding to some other method) at a time. The null hypothesis assumes that there is no significant difference between the mean values of two groups, whereas there is significant difference in the mean values of two groups for the alternative hypothesis. It is evident from Table 4 that all P values are less than 0.05 (5% significance level). This is a strong evidence against the null hypothesis, establishing significant superiority of the proposed membrane clustering algorithm.

5. Conclusion

In this paper, we discuss a membrane clustering algorithm, a novel clustering algorithm in the framework of membrane computing. Distinguished from the existing evolutionary clustering techniques, two inherent mechanisms of membrane computing are exploited to realize the membrane clustering algorithm, including evolution and communication mechanisms. For this purpose, a tissue-like P system consisting of q cells is designed, in which each cell as parallel computing unit runs in maximally parallel way and each object of the system represents a group of candidate centers. Moreover, the communication rules impliedly realize a local neighborhood structure; namely, each cell exchanges and shares the best objects with its two adjacent cells. Under the control of evolution and communication mechanisms of objects, the tissue-like P system is able to search for the

optimal centers for a data set to be clustered. In addition, the local neighborhood structure can guide the exploitation of the optimal object and enhance the diversity of evolution objects.

Conflict of Interests

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

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant nos. 61170030 and 61472328), the Chunhui Project Foundation of the Education Department of China (nos. Z2012025 and Z2012031), and the Sichuan Key Technology Research and Development Program (no. 2013GZX0155), China.

References

- [1] J. A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, 1975.
- [2] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ, USA, 1988.
- [3] R. Xu and D. Wunsch II, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.
- [4] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [5] B. Everitt, S. Landau, and M. Leese, *Cluster Analysis*, Arnold, London, UK, 2001.
- [6] S. Saha and S. Bandyopadhyay, "A symmetry based multiobjective clustering technique for automatic evolution of clusters," *Pattern Recognition*, vol. 43, no. 3, pp. 738–751, 2010.
- [7] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k -means clustering algorithms: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.
- [8] D. Steinley, "K-means clustering: a half-century synthesis," *The British Journal of Mathematical and Statistical Psychology*, vol. 59, no. 1, pp. 1–34, 2006.
- [9] X. Wu, *Top Ten Algorithms in Data Mining*, Taylor & Francis, Boca Raton, Fla, USA, 2009.

- [10] S. Bandyopadhyay and U. Maulik, "An evolutionary technique based on K-means algorithm for optimal clustering in \mathbb{R}^N ," *Information Sciences*, vol. 146, no. 1-4, pp. 221-237, 2002.
- [11] S. Bandyopadhyay and S. Saha, "GAPS: a clustering method using a new point symmetry-based distance measure," *Pattern Recognition*, vol. 40, no. 12, pp. 3430-3451, 2007.
- [12] M. Laszlo and S. Mukherjee, "A genetic algorithm that exchanges neighboring centers for k -means clustering," *Pattern Recognition Letters*, vol. 28, no. 16, pp. 2359-2366, 2007.
- [13] D. X. Chang, X. D. Zhang, and C. W. Zheng, "A genetic algorithm with gene rearrangement for K-means clustering," *Pattern Recognition*, vol. 42, no. 7, pp. 1210-1222, 2009.
- [14] C. D. Nguyen and K. J. Cios, "GAKREM: a novel hybrid clustering algorithm," *Information Sciences*, vol. 178, no. 22, pp. 4205-4227, 2008.
- [15] Y. T. Kao, E. Zahara, and I. W. Kao, "A hybridized approach to data clustering," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1754-1762, 2008.
- [16] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "An ant colony approach for clustering," *Analytica Chimica Acta*, vol. 509, no. 2, pp. 187-195, 2004.
- [17] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and k -means for cluster analysis," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 183-197, 2010.
- [18] G. Păun, "Computing with membranes," *Journal of Computer and System Sciences*, vol. 61, no. 1, pp. 108-143, 2000.
- [19] C. Martin-Vide, G. Păun, J. Pazos, and A. Rodríguez-Patón, "Tissue P systems," *Theoretical Computer Science*, vol. 296, no. 2, pp. 295-326, 2003.
- [20] M. Ionescu, G. Păun, and T. Yokomori, "Spiking neural P systems," *Fundamenta Informaticae*, vol. 71, no. 2-3, pp. 279-308, 2006.
- [21] G. Păun, "P systems with active membranes attacking NP-complete problems," *Journal of Automata, Languages and Combinatorics*, vol. 6, no. 1, pp. 75-90, 2001.
- [22] L. Pan and T. Ishdorj, "P systems with active membranes and separation rules," *Journal of Universal Computer Science*, vol. 10, no. 5, pp. 639-649, 2004.
- [23] G. Păun, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "Tissue P systems with cell division," *International Journal of Computers, Communications and Control*, vol. 3, no. 3, pp. 295-303, 2008.
- [24] L. Pan and M. J. Pérez-Jiménez, "Computational complexity of tissue-like P systems," *Journal of Complexity*, vol. 26, no. 3, pp. 296-315, 2010.
- [25] L. Pan and G. Păun, "Spiking neural P systems with anti-spikes," *International Journal of Computers, Communications and Control*, vol. 4, no. 3, pp. 273-282, 2009.
- [26] L. Pan, G. Păun, and M. J. Pérez-Jiménez, "Spiking neural P systems with neuron division and budding," *Science China*, vol. 54, no. 8, pp. 1596-1607, 2011.
- [27] J. Wang, L. Zou, H. Peng, and G. Zhang, "An extended spiking neural P system for fuzzy knowledge representation," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 7, pp. 3709-3724, 2011.
- [28] H. Peng, J. Wang, M. J. Pérez-Jiménez, H. Wang, J. Shao, and T. Wang, "Fuzzy reasoning spiking neural P system for fault diagnosis," *Information Sciences*, vol. 235, pp. 106-116, 2013.
- [29] J. Wang, P. Shi, H. Peng, M. J. Perez-Jimenez, and T. Wang, "Weighted fuzzy spiking neural P systems," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 2, pp. 209-220, 2013.
- [30] J. Wang and H. Peng, "Adaptive fuzzy spiking neural P systems for fuzzy inference and learning," *International Journal of Computer Mathematics*, vol. 90, no. 4, pp. 857-868, 2013.
- [31] G. Păun, G. Rozenberg, and A. Salomaa, *The Oxford Handbook of Membrane Computing*, Oxford University Press, New York, NY, USA, 2010.
- [32] G. Păun and M. J. Pérez-Jiménez, "Membrane computing: brief introduction, recent results and applications," *BioSystems*, vol. 85, no. 1, pp. 11-22, 2006.
- [33] A. Alhazov, C. Martín-Vide, and L. Pan, "Solving a PSPACE-complete problem by recognizing P systems with restricted active membranes," *Fundamenta Informaticae*, vol. 58, no. 2, pp. 67-77, 2003.
- [34] T. Ishdorj, A. Leporati, L. Pan, X. Zeng, and X. Zhang, "Deterministic solutions to QSAT and Q3SAT by spiking neural P systems with pre-computed resources," *Theoretical Computer Science*, vol. 411, no. 25, pp. 2345-2358, 2010.
- [35] G. Zhang, J. Cheng, M. Gheorghe, and Q. Meng, "A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems," *Applied Soft Computing Journal*, vol. 13, no. 3, pp. 1528-1542, 2013.
- [36] H. Peng, J. Wang, M. J. Pérez-Jiménez, and P. Shi, "A novel image thresholding method based on membrane computing and fuzzy entropy," *Journal of Intelligent and Fuzzy Systems*, vol. 24, no. 2, pp. 229-237, 2013.
- [37] H. Peng, J. Wang, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "The framework of P systems applied to solve optimal watermarking problem," *Signal Processing*, vol. 101, pp. 256-265, 2014.
- [38] E. Falkenauer, *Genetic Algorithms and Grouping Problems*, John Wiley & Sons, 1998.
- [39] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.
- [40] <http://www.ics.uci.edu/~mllearn/MLRepository.html>.

Research Article

Primary Path Reservation Using Enhanced Slot Assignment in TDMA for Session Admission

Suresh Koneri Chandrasekaran,¹ Prakash Savarimuthu,²
Priya Andi Elumalai,³ and Kathirvel Ayyaswamy⁴

¹Department of Computer Science and Engineering, Tagore Engineering College, Chennai 600127, India

²Department of Electronics and Communication Engineering, Jerusalem College of Engineering, Chennai 600100, India

³HCL Technology, Chennai, India

⁴Department of Computer Science and Engineering, Anand Institute of Higher Engineering and Technology, Chennai, India

Correspondence should be addressed to Suresh Koneri Chandrasekaran; kcsuresh84@gmail.com

Received 16 May 2014; Revised 15 September 2014; Accepted 28 October 2014

Academic Editor: Ahmad T. Azar

Copyright © 2015 Suresh Koneri Chandrasekaran et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile ad hoc networks (MANET) is a self-organized collection of nodes that communicates without any infrastructure. Providing quality of service (QoS) in such networks is a competitive task due to unreliable wireless link, mobility, lack of centralized coordination, and channel contention. The success of many real time applications is purely based on the QoS, which can be achieved by quality aware routing (QAR) and admission control (AC). Recently proposed QoS mechanisms do focus completely on either reservation or admission control but are not better enough. In MANET, high mobility causes frequent path break due to the fact that every time the source node must find the route. In such cases the QoS session is affected. To admit a QoS session, admission control protocols must ensure the bandwidth of the relaying path before transmission starts; reservation of such bandwidth noticeably improves the admission control performance. Many TDMA based reservation mechanisms are proposed but need some improvement over slot reservation procedures. In order to overcome this specific issue, we propose a framework—PRAC (primary path reservation admission control protocol), which achieves improved QoS by making use of backup route combined with resource reservation. A network topology has been simulated and our approach proves to be a mechanism that admits the session effectively.

1. Introduction

The strength of MANET [1] lies in its ability to form self-organized network which seems to be an interesting fact. A MANET provides a practical way to rapidly build a decentralized communication network in areas where there is no existing infrastructure or where temporary connectivity is needed. This property makes these networks highly flexible. There exist some practical design issues in MANET such as limited bandwidth, dynamic nature of topology, and decentralized coordination. Due to these design issues, the quality factors like bandwidth, delay, and jitter [2] get affected. Certain real time applications such as audio and video expect additional bandwidth in order to provide QoS [3–5]. The goal of QoS is to achieve more deterministic network behaviour so that the

information carried by the network can be better delivered and the resources can be better utilized. To provide QoS, we need certain frameworks such as reservation, scheduling, admission control, and routing algorithms [6–9].

Admission control can avoid the network congestion by estimating whether the new session is admissible or not [10, 11]. Various admission control methods have been proposed in many articles in last few years. However the admission control based protocols have certain pitfalls such as improper handling of session. Different mechanism has been proposed for session admission, of which backup mechanism has a noticeable achievement [12, 13]. Admission control algorithm combined with backup path solves the frequent path break due to mobility; even such mechanism is not utilized properly. The reservation mechanism ensures the QoS [8, 14]. There are

many reservation based protocols, such as hop reservation multiple accesses (HRMA) [15] and five-phase reservation protocol (FPRP) [16, 17]. In addition, contention based protocols, such as carrier sensing multiple accesses with collision avoidance (CSMA/CA), and hybrid protocols, such as TDMA protocol based on contention and reservation [18], exist. The reservation based TDMA protocols have several advantages over contention based protocols, of which conflict free and maximum spatial reuse efficiency is acquainted to the domain [14, 19, 20]. For network with dynamic topology FPRP is a suitable one. Even though the performance aspect of FPRP is improved, still the time slot utilization was not improved.

Considering all the issues of admission control, reservation and backup path, we propose a QoS framework named primary path reservation admission control (PRAC). PRAC framework focuses on admission control and reservation, in order to achieve quality of service. Using backup route aided admission control protocols, such as StAC [13], MACMAN [12], introduces too many control overheads, whereas our proposed protocol minimizes the control packets. In the reservation mechanism such as TDMA, the slot assignment procedure for any request does not consider the neighboring slot assignment. Due to the fact that bandwidth is commonly shared by all carrier sensing nodes, it affects the current node transmission. So it is mandatory to consider the neighboring slot assignment when calculating the bandwidth, which further tries to avoid collisions. By using PRAC, we minimize the delay and maximize the throughput for any admitted session, thereby increasing the overall performance.

2. Background and Related Works

2.1. Admission Control. The admission control (AC) mechanism provides a way that should ensure admissible path from one node to another. Such mechanism determines admission of new data flows by keeping track of available bandwidth; it determines the residual capacity of any individual node and also of the carrier sensing neighbor (n_{cs}) nodes [10]. The AC Protocol starts with calculation of local nodes capacity. The local capacity of any node can be estimated by any of the quality aware routing (QAR) protocols [21–23]. This protocol discovers the routes which have sufficient resource that satisfies the requirements of a session. Apart from these protocols, the basic routing protocols like DSR [21] can also be used to estimate the local capacity (bandwidth), where the local capacity is the unconsumed bandwidth at a given node. Various methods have been proposed to calculate the local capacity in [11–13], among which channel ideal time (CIT) is suitable for admission control and can be calculated using channel ideal time and link transmission in use [12, 13]. The residual capacity calculation alone does not serve the need of admission control, so the second phase continues with ensuring the adequate capacity of the n_{cs} . This adequate capacity can be calculated by several methods. In [10], the first proposed method, referred to as CACP-Multihop, gathers information about the residual capacities can be obtained from the neighbors using admission request packets which are

flooded through the network with a radius of two hops. The second proposed method, referred to as CACP-Power, uses a high power transmission mechanism to send the admission request packet to all nodes within the n_{cs} . The other proposed method in adaptive admission control [7] and SoftMAC [11] sends hello packets to calculate the CIT value.

2.2. Related Works. We have analyzed a wide range of protocols before landing up in a concept to provide a new way of admission control. The works done by various authors depict that the core concentration is on providing QoS and to address the issues related to them. In-depth analysis of [7, 24] paves way to get a clear picture classifying the admission control protocols and these proposals provide many useful information pieces for forming a more efficient protocol.

In [13], multipath admission control (MACMAN) was proposed. The paper discussed the priority given to QoS by maintaining backup routes, which thereby enhances the performance (by trying to avoid path or packet loss) of the overall network. Even though this method proves optimal in increasing the performance, the concealed fact is that maintaining the backup routes increases control overhead too. An improved version of MACMAN staggered backup (StAC-backup) [13] proposes a technique that involves only partial disjoint set for admission control which reduces the path identification control overhead when compared to other protocols discussed. As the control overhead due to the beacon messages increases, the network data transfer rate decreases. In [25], distributed admission control protocol (DACP) was proposed. This paper contributes mechanisms to provide QoS by making use of bandwidth reservation. In order to reserve bandwidth, DACP estimates local and neighbour capacity. So mentioned protocol does not consider the hidden and exposed terminal problem since reservation of bandwidth may not be accurate and also the paper does not consider backup route based routing.

In [8], an on-demand bandwidth reservation QoS routing protocol for mobile ad hoc networks was proposed, which considers revising the bandwidth estimation and reservation procedure. For route discovery, an approach min-max was used to satisfy the bandwidth requirement. Also the reservation procedure used estimates the weight of its neighbor's slot for availability. The time slot with lowest weight will be reserved. The proposed method admission control and bandwidth reservation (ACBR) in [26] suffers from a limitation that estimates the available capacity of the neighboring nodes, using one-hop distance only. In addition, it does not take the contenting nodes in the interference range into account. In other words, this scheme considers only the contention of nodes within transmission range.

In [18], the proposed TMMR protocol performs bandwidth reservation in order to attain multihop packet transmissions. It also provisions the node mobility through fast fault node detection. TDMA based reservation mechanism is exploited but still it does not improve time slot selection for reservation. The novel idea of time slot utilization proposed in [20] does not consider the impact of hidden terminal problem and it lacks effectiveness in session admission. Considering

the above discussed facts, our proposed method exhibits the flow of admission in an efficient reservation, combined with a backup path mechanism. The effect of such an improvised mechanism shows enhanced performance.

3. Primary Path Reservation Admission Control Protocol (PRAC)

3.1. Backup Route Discovery. Studies on backup route discovery emphasize that many backup routes have to be maintained to overcome re-routing process after a route failure. The disadvantage of the above stated method is that since many control packets are sent through and forth, there is a consistent increase in the network overhead. In PRAC, we maintain a primary path and a single backup path. From previous analysis, dynamic source routing (DSR) [21] proves to be a suitable routing protocol for finding backup routes. The backup route discovery process finds the route in which the nodes involved in the process are in a complete disjoint set. The backup path discovered should not include the nodes that were in the primary path [12]. Results from [13] which include many backup routes prove that the nodes in primary and backup routes can be sufficiently disjoint and are not required to be in full disjoint sets. Consider

$$|R_{\text{primary}} \cap R_{\text{backup}}| \leq \frac{|R_{\text{primary}}|}{2}, \quad (1)$$

where R_{primary} is the primary path and R_{backup} is the backup path. But in PRAC since only one backup route is maintained, the backup route nodes should be a complete disjoint set with that of the primary route nodes:

$$|R_{\text{primary}} \cap R_{\text{backup}}| = 0. \quad (2)$$

If many backup paths are maintained, the condition for partial or sufficient disjoint sets may yield better performance, whereas, in the case of a single backup, if the primary path fails then the probability of backup failure is also feasible. Hence the backup path nodes should be a complete disjoint set, thereby reducing the risk of failure. The capacity constraint route discovering process is explained in Figure 1. The route discovery process starts with the request for session. In this, the source node broadcasts the RREQ packet to all its neighbouring nodes. Each node calculates its own residual capacity, which if satisfies the capacity requested by a session, will rebroadcast the RREQ packet to their neighbours. The session capacity requirement BW_{req} can be calculated as follows:

$$BW_{\text{req}} = b * n_{\text{cs}}, \quad (3)$$

This equation is used to calculate the session capacity requirement at any node. Here b is the number of slots required and n_{cs} is the number of contenting nodes in the carrier sensing range. MACMAN [12] implements high power method that builds up n_{cs} sets, which in turn increases the beacon overhead making it a probable disadvantage. Our proposed PRAC method overcomes it by the use of admission request packet

(similar to CACP-Multihop), flooded through the network, covering a two-hop radius at the time of session admission, depending upon the session capacity; a node may confirm the session admission or deny it. In the case of primary route failure, backup route discovery is initiated, in which the process is decoupled from the normal route discovery mechanism and only admission control mechanism takes place.

3.2. Validation of Single Backup Path over Multiple Backup Paths. Though multiple backup path approach does make sure the backup route availability and better manages the throughput drop, it adds additional overhead in the network by sending probe packet to manage the paths. Moreover the single backup path provides a better way in managing the overall traffic in a given network by avoiding unintended interference that happened on the session routes. As there is only one backup path, the maintenance quotient is virtually lower compared to the multiple backup method (that periodically engages the routes for the probing). Though the problem of handling the mobility/path-break appears to exist in this single backup path, it provides a way to find out another path if initial backup path fails to establish the connection. So in any eventual case the single backup path does make sure that at any point of time there is a path existing for the data transmission. We assume h is the number of hops between the source and the destination. The single paths can achieve reduced probe packet as demonstrated in shown calculation below. The analytical verification and the robustness of the single backup path are based on the simple routing model. The probability of packet control overheads denoted P_c between the source and the destination in the single backup path routing:

$$P_c = 1 - (1 - \mu)^h, \quad (4)$$

where μ is average control packet and h is the number of hops. Now we can find that the probability of control packets in multiple backup paths maintained is

$$P_c = 1 - \left\{ (1 - \mu)^h \right\}^m, \quad (5)$$

where m is the number of disjoint-backup paths. As the number of hops, h , and the multiple backup paths, m increases the probability of the overhead in the network increases as depicted in the formula (5), whereas in single backup path, the only parameter that causes the overhead to increase is the number of hops and similarly the number of control packets to send also increases. This does make sure that the network traffic is managed better.

3.3. Reservation. The purpose of backup route discovery is to find the residual capacity of each and every node in a route. Such a mechanism discovers many routes, among which the efficient routes are selected (the mechanism involved in backup route discovery has been discussed in Section 3.1). The reservation approach in our proposal focuses only on the primary route. When the destination node receives the first RREQ packet of a session, it is considered as the primary

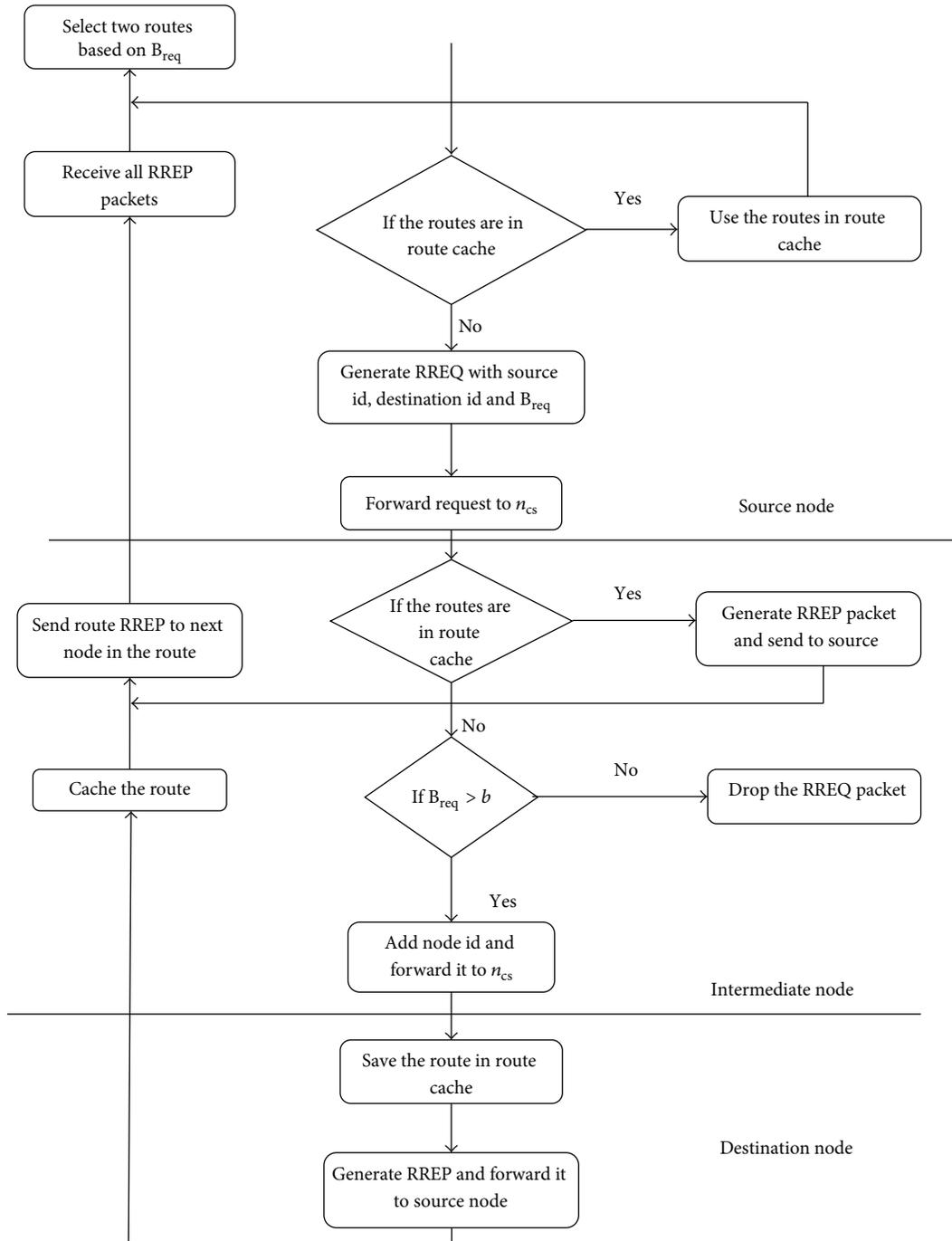


FIGURE 1: Capacity constraint route discovering process.

path and reservation reply is sent along the reverse direction of the same primary path. When the second RREQ packet is received, this path is not reserved but it will be considered as a backup path thereby sending a normal reply. This reservation process is coupled with backup route discovery. The above facts discussed address how the reservation process integrates the route discovery stage. The reason behind using TDMA for reservation in this paper is that it tolerates the radio interference problem and it holds good in our scenario, since all the nodes share the single common channel. We propose a new

approach in TDMA which considers both hidden terminal and exposed terminal problem. A glance on the following example helps us to understand the problem in making reservation. In Figure 2, consider the path from A to C. Here the grey slots depict that they are busy and similarly the white slots are free. Between A and B there are five matching free slots {1, 2, 3, 4, 5} and between B and C there are four matching free slots {3, 4, 5, 6}. If we reserve slots {1, 2, 3} for A to transmit and slot {4, 5, 6} for B to transmit, then the path bandwidth is only three. Suppose that if the other pair D and E

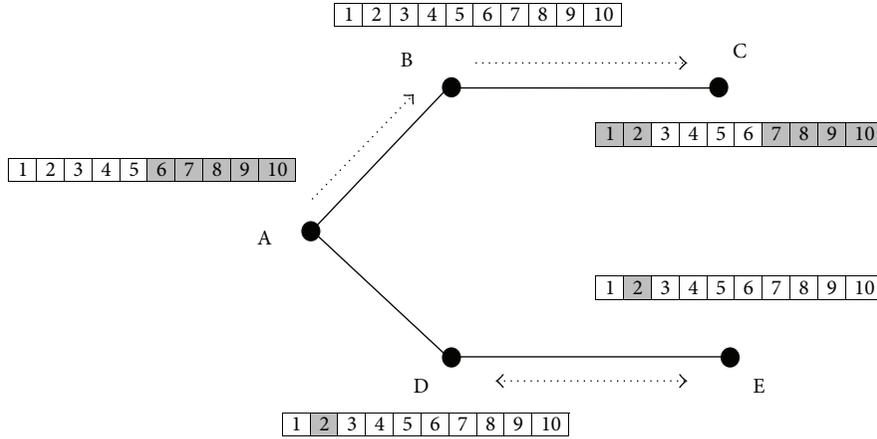


FIGURE 2: Bandwidth calculation interfered by hidden and exposed terminal problem.

are currently using slot 2 to communicate, two possible cases will arise.

Case 1. If D is the receiver on slot 2, then A will not allow sending on slot 2, so collision occurs at D. This is called hidden terminal problem since common free slots between A and B are reduced to {1, 3, 4, 5}.

Case 2. Due to Case 1, the bandwidth of the path A to C degrades to 2 slots. If D is the sender on slot 2 then A will not allow sending on slot 2. This is called exposed terminal problem.

3.3.1. System Model

- (1) Each node maintains the neighbor’s information (including available slot). This information is gathered by hello message.
- (2) TDMA frames are fixed length; time frame had 16 time slots, with 5 ms for each time slot.
- (3) Nodes shared single common channel.
- (4) Each frame consists of two subframes: the data frame and control frame. The data frame consists of fixed number of data slots. When a node wants to transmit or receive data packets, it may use its control slot to reserve the desired data slots. In the control frame, each node has a committed control slot and therefore avoids contention and collision.
- (5) Control and data slot handshake: the control slot holds control packets such as the route request, hello packet, the route reply, and the route error. When a node wants to transfer data it may use the control packets to reserve the required path. Once it is reserved, it can be used for both transmitting and receiving data.

3.3.2. Data Structure

- (i) We maintain sending slot table (SS), receiving slot table (RS), and hop count table (H) in every node. The

sending slot table $SS_X[1 \dots n, 1 \dots s]$: of node X records the time slots of all the nodes within 2 hops from X and is having sending activities. So $SS_X[i, j] = 1$ if slot j of node i has been reserved for transmission; otherwise, $SS_X[i, j] = 0$. The receiving slot table $RS_X[1 \dots n, 1 \dots s]$: of node X records the time slots of all the nodes which are within 2 hops from node X and are having receiving activities. Similarly, $RS_X[i, j] = 1$ if slot j of node i has been reserved for receiving; otherwise, $RS_X[i, j] = 0$. The hop count table $H_X[1 \dots n, 1 \dots n]$: of a node X maintains a record of the mutual distances between nodes in X’s neighborhood. Similarly, for each node i that is within 1 hop from X; $H_X[i, j] = 1$ if node j is within 1 hop from i; otherwise, $H_X[i, j] = \infty$.

- (ii) The RREQ has following parameters: $RREQ(S, D, id, X, b, path, NH)$ where S is source node and D is the destination node and id is the session identity issued by source; b is bandwidth requirement, which can be represented by the number of slots. X is the node that is currently relaying RREQ. The path is the partial path, with the available slot, that has been discovered so far. It has the format $((h_1, l_1), (h_2, l_2), \dots, (h_k, l_k))$. Here h_i is node identity, where $i = 1, \dots, k$ and each l_i contains the total b slots that are found to be available for h_i to transmit to h_{i+1} . NH is neighbour hop list $((h'_1, l'_1), (h'_2, l'_2), \dots)$. Here each node h'_i may serve as the next hop of node X that extends the current partial path, only if h'_i has sufficient slots and also l'_i contains b slots that can be used by X to transmit to h'_i .
- (iii) The RREP has following parameters: $RREP(S, D, id, path)$. When a route is found at the destination D, we need to initiate a packet RREP to the source node S. This packet traverses through the reverse path and reserves the slots on the path.

```

Step 1.
    if Y is not listed in NH then exits this procedure
    else Construct a list  $path\_temp = path|(X, l'_i)$ 
    // | -mean that current node is attached/concatenated with path
Step 2.
    (i) Construct two temporary table
        SS_temp [n, s] & RS_temp [n, s] then
        Copy SS [n, s] in to SS_temp [n, s]
        Copy RS [n, s] in to RS_temp [n, s]
    (ii) Let  $path = ((h_1, l_1), (h_2, l_2), \dots, (h_k, l_k))$ 
        for  $i = 1, \dots, k - 1$ 
            for every time slot  $t$  in the list  $l_i$ 
                SS_temp [ $h_i, t$ ] = 1
                RS_temp [ $h_{i+1}, t$ ] = 1
            for every time slot  $t$  in the list  $l_k$ 
                SS_temp [ $h_k, t$ ] = 1
                RS_temp [ $X, t$ ] = 1
    (iii) for every time slot  $t$  in the list  $l'_i$ 
        SS_temp [ $X, t$ ] = 1
        RS_temp [ $Y, t$ ] = 1
Step 3.
    if NH_temp = empty
    for each 1-hop neighbour Z of Y do
        L = slot_selection(Y, Z, b, SS_temp, RS_temp)
        if L ≠ empty
            NH_temp = NH_temp|(Z, L)
    else if NH_temp ≠ empty then
        broadcast RREQ(S, D, id, b, Y, path_temp, NH_temp)
Subroutine: slot_selection(Y, Z, b, SS_temp, RS_temp)
for each slot  $i$ , where  $1 \leq i \leq s$  following condition will be checked
Con 1: (SS_temp [Y, i] = 0) & (RS_X [Y, i] = 0) & (SS_temp [Z, i] = 0) & (RS_X [Z, i] = 0)
Con 2: for all ( $H_Y [Y, W] = 1$ ) then RS_temp [W, i] = 0
Con 3: for all ( $H_Y [Z, W] = 1$ ) then SS_temp [W, i] = 0

```

ALGORITHM 1

Lemma 1. A slot t can be used by a node X to send to another node Y without causing collision, if the following conditions are satisfied.

- (1) Slot t is not scheduled to send/receive node in neither X nor Y .
- (2) For any 1-hop neighbor Z of X , slot t is not scheduled to receive data in Z .
- (3) For any 1-hop neighbor Z or Y , slot t is not scheduled to send data in Z .

3.3.3. Reservation Procedure

(A) *Route Request Phase.* When a node Y receives a broadcast packet RREQ($S, D, id, b, X, path, NH$) from a neighbour node X and if Y has not received the same packet before, then Algorithm 1 will be executed. As shown in Step 1, if NH does not have the node Y listed in it, then a $path_temp$ will be created. Hence the corresponding parameter l'_i that contains b time slots can be used by X to transmit to h'_i without collision. In Step 2, the temporary tables for sending and receiving SS_temp and RS_temp are used during the probing stage, in

which the confirmed list is stored in original SS and RS . As the path is propagated for each slot of a node, the confirmed slot information is saved in the respective temporary tables SS_temp and RS_temp . This confirmed slot in respective node is carried over to next node that is visited and this will continue till the destination node.

(1) When RREQ arrives at intermediate nodes (see Algorithm 1), in Step 3 the information of $path$ and NH will be saved in the temporary tables that we have discussed in Step 2. The $slot_selection(Y, Z, b, SS_temp, RS_temp)$ routine is called to check if there is any slot available for Y to send Z . In case of any slot available in one hop to extend the current path, then the RREQ will be rebroadcasted. The same routine is used for node Y to choose the free time slots b in order to send data to Z . The slot selection procedure is based on Lemma 1. In case, if the required free slot is available, then the algorithm will proceed further to know if there are more free slots to occupy. This is mainly to increase the channel reuse which is essential in the case of wireless communication. Selection of these free slots is done, giving high priority to the ones (node) without the hidden and exposed terminal problems. To achieve this, we give the valid time slot i , an

```

// for each intermediate node  $X = h_i$  the following steps to be executed.
for  $j = i - 2$  to  $i + 2$  do
  for each time slot  $t$  in  $l_j$ 
     $SS_X [h_j, t] = 1$ 
  for each time slot  $t$  in  $l_{j-1}$ 
     $RS_X [h_j, t] = 1$ 
end for

```

ALGORITHM 2

increased priority so that $SS_temp[W, i] = 1$ to the neighbour W of Z .

(2) When RREQ arrives at destination node, a confirmed path is formed once the destination D receives the RREQ($S, D, id, b, X, path, NH$). The destination node D can still accept the request RREQ or choose ignoring it, based on following condition. When a node D receives a broadcast packet RREQ from a node X , then if NH does not have D listed in it, then a $path_temp$ will be created. The corresponding parameter l'_i contains b time slots that can be used by X to transmit to h'_i without collision. Then RREP will be sent to the source S .

- (i) Let (h'_i, l'_i) be the entry in NH such that $h'_i = D$.
- (ii) $path_temp = path \mid (X, l'_i)$.
- (iii) Send RREP($S, D, id, path_temp$) to source S .

(B) *Route Reply Phase*. As a part of the reply sequence, the RREP packet will be sent in the reverse direction of $path$ in unicast manner with each intermediate node relaying the packet. Also the “receive” and “send” information will be saved in the respective table in each node where the packet is traversed. Assuming that $path$ contains $((h_1, l_1), (h_2, l_2), \dots, (h_k, l_k))$, each intermediate node will update the sending and receiving table (confirmed one) with available time slot information (see Algorithm 2).

3.4. Route Maintenance. The fact behind route maintenance is that the process monitors the primary and backup routes on a regular basis, checking for QoS requirements (session requirements) [12, 13]. In our proposed method, when the primary route fails, the control moves towards the backup path. Here, PRAC makes use of BRQ (backup route query) message that continuously monitors the stability and effectiveness of the backup path. Previous studies show that many other mechanisms use multiple backup routes, which accounts for increase in reliability but also increases the control overhead. But in our model, the possibility of the primary path failure is very low because of the reservation mechanism used, as described before. The control overhead in PRAC model is very low because path monitoring does not comprise many backup paths and instead is done for a single backup path only. In Figure 3 we describe overall route maintenance process. In case of failure of the primary path, in order to

maintain the reliability of the network, the backup path is taken. This backup path is herewith considered to be primary and starts reservation process. Since our proposed model uses a single backup strategy, when the backup is taken as primary (in case of failure), then we lack the existence of a backup path (PRAC requires the presence of a primary and backup path always). Hence the source finds a backup path from route cache. As mentioned before, with the use of BRQ message, the route cache is analysed for the best route that accomplishes the session request. When a node receives a BRQ message, it calculates the contention difference (CD). In our maintenance model, we avoid the calculation of contention count. The reason is that the transmission flow along the primary path is likely to reduce the measured available bandwidth along the backup path. Calculating contention count may end up in insufficient metric; hence the contention difference proves to be an optimal method:

$$CD = |CS_{neighbor} \cap R_{backup}| - |CS_{neighbor} \cap R_{primary}|. \quad (6)$$

The CD should hold a constraint in which $BW_{avail} > CD \cdot BW_{req}$. If the above condition fails, then it sends a BRQF message to the source, thereby removing the specific route from route cache. The advantage of PRAC route maintenance is that in the case of both primary and backup path being failed, the source does not go for route recovery process again and instead it makes use of the route cache information [21].

4. Performance Evaluation

4.1. Simulation Environments. We use NS-2 network simulator to verify the PRAC’s performance. Based on our analysis, set of simulations involves a larger network with random mobility. Following are the specifications that have been followed while simulating the PRAC protocol. 100 nodes are randomly placed in a 1000 m × 1000 m area. By setting the node transmission range of 250 m and a CS-Range of 550 m, multihop routes have been created and also allowed all types of collisions to occur. 50 nodes are randomly chosen as sources of traffic to 50 other nodes. Each session was CBR traffic flows that were used with a packet size of 512 bytes and a bit rate of 128 kbps. The nodes move according to the random waypoint mobility model and the bandwidth of the channel is 2 Mbps. The two ray ground propagation model was employed to avoid wireless channel errors. Backup route query (BRQ) interval is 2 s (Table 1).

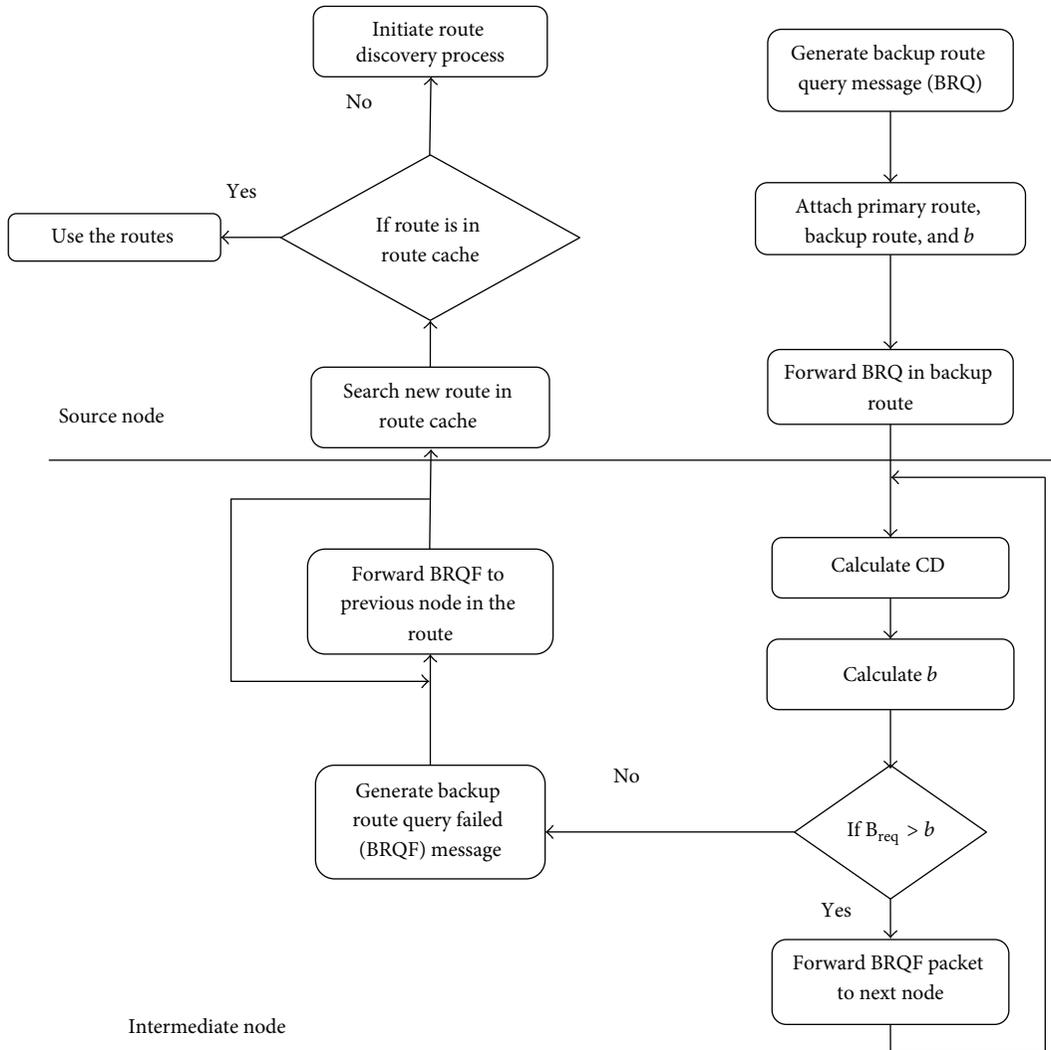


FIGURE 3: Route maintenance.

TABLE 1: Parameter settings.

Parameters	Values
Propagation model	Two-ray ground
Reception range	250 m
Carrier sensing range	550 m
Data packet size	512 bytes
CBR data rate	128 kbps
Network area	1000 m × 1000 m
Mobility model	Random way point
Backup route query (BRQ)	2 s
Channel bit rate	2 Mbps
Number of nodes	100
TDMA frame length	16 time slots
Slot time	5 ms
Simulation time	100 sec
Maximum number of sessions	50

4.2. Results and Discussions. Figure 4 shows the throughput of each session attained by PRAC. In the results, the number of admitted sessions is more or less similar; in addition, the throughput of the admitted session is higher than other models during the simulation time. As shown, we obtain higher throughput while using PRAC. However, when St backup [13] and DACP [25] are used, only medium throughput is obtained and when TMMR [18] is used, only lower throughput is obtained. From the results it is clear that the proposed protocol is capable of reducing the number of unnecessary routing packets during route discovery, by making admission control decisions at every node in the network. Thus, DACP can use more resources in the network than other models to transmit data packets. The admission control without backup path (i.e., TMMR and DACP) model shows poor throughput, while the admission control with backup path (St backup and PRAC) shows high throughput. In addition, PRAC attains greater aggregated throughput than other models.

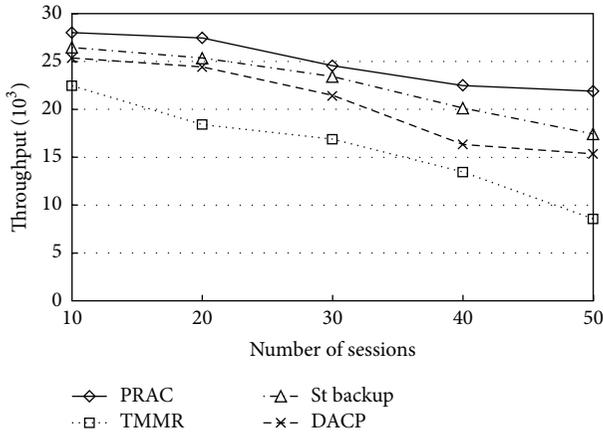


FIGURE 4: Throughput.

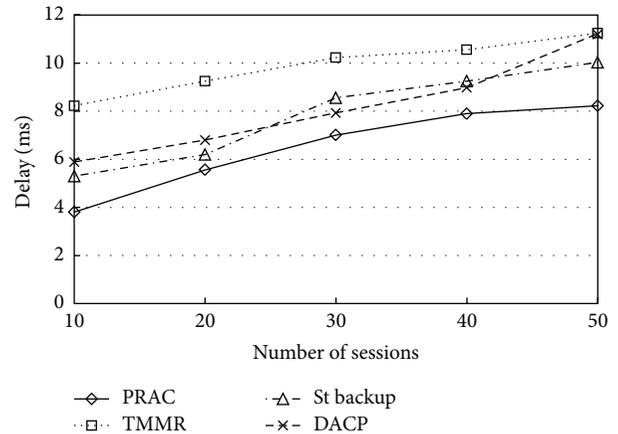


FIGURE 6: End to end delay.

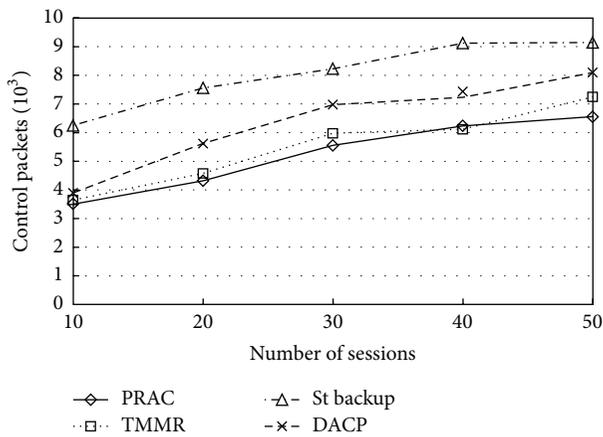


FIGURE 5: Control packets overheads.

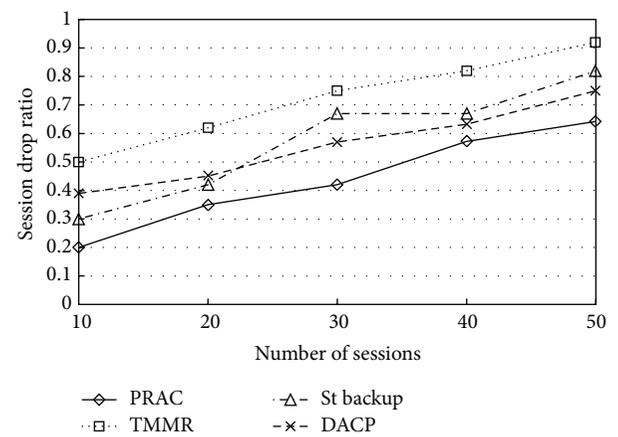


FIGURE 7: Session drop ratio.

It is also essential to inspect the overhead produced by the protocol since PRAC introduces additional control messages. Figure 5 demonstrates the average amount of control packets transmitted for the duration in the simulations. As expected, the graph demonstrates a notable increase in control traffic in the network when we use backup path. When compared to the existing protocols our PRAC protocol noticeably produces lesser control packets. As seen in Section 3.3, the actual amount of control packet overhead is an important factor for maintaining a single backup path. A tradeoff happens between the extra control packet overhead and the facility to quickly switch to a new path if the present one breaks down. However, there is an increase in overhead if the protocol uses backup routes that are to be maintained. But this increase is acceptable due to the improvement of the other performance parameters. At first, the overhead is low; this is because of minimum number of sessions that are admitted and also the backup route maintenance messages are produced on a per-flow basis for admitted flows.

Since any admission control protocols are mainly designed to put up real time applications that have requirements on end-to-end delay, it is essential to make sure that the additional overhead does not include added delay that go

above the delay bounds of the requests [27]. The end-to-end delay for the sessions can be seen in Figure 6. From this figure, PRAC is capable of providing an increase in data packet delivery and minimizing the end-to-end delay. Since there is no admission control performed in TMMR, the network becomes congested as new sessions are added to the network, resulting in decreased throughput and dramatically increasing the delay of the sessions. On the other hand, the throughput of the sessions shows significant degradation; also the delay rises as the number of sessions increases. The average end-to-end delay in the simulations that is achieved for all other three protocols is much higher than PRAC, indicating PRAC's ability to balance the network load.

Figure 7 shows the average number of times the sessions are stopped per simulation. When using TMMR, DACP, and St backup, a large number of sessions are stopped. When using PRAC, in many cases, these same sessions can switch to a backup route and retain the transmission. For all the other protocols, it is likely that as the number of admitted sessions increases so does the number of session breaks. However, PRAC is able to handle the session breaks even if large amount of session is admitted. The reduced number of session drops is likely to increase the QoS at the end user.

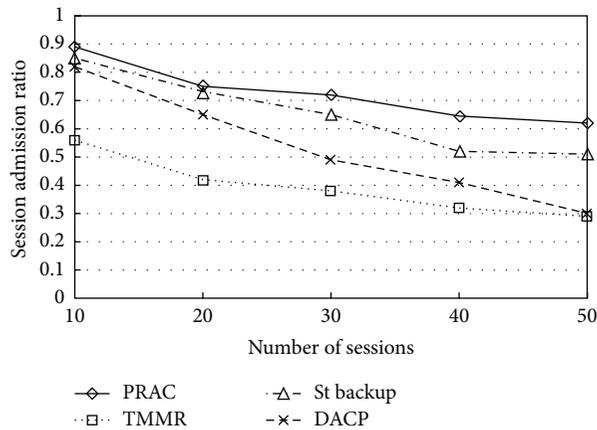


FIGURE 8: Session admission ratio.

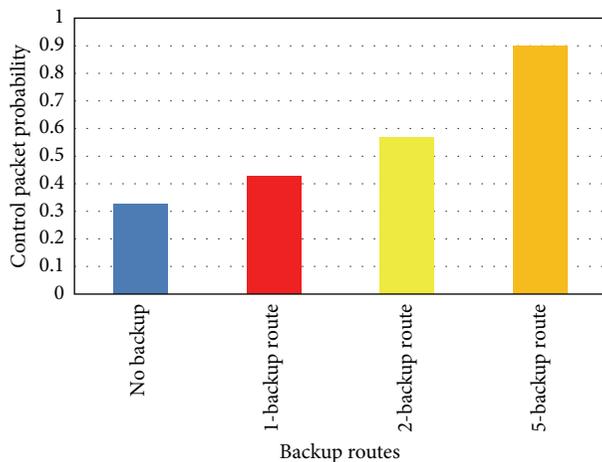


FIGURE 9: Identifying the overall control packet probability when using different backup path.

In Figure 8 we inspect the number of sessions that each protocol permits to be active at the same time. It can be agreed that when the number of sessions is minimum, the variance among the protocols is negligible. However, when a larger number of sessions are initiated, PRAC is able to sustain that more than all other three protocols. This contributes to the variances in delivery rate among the protocols and proves that PRAC is capable of using the resources in the network more efficiently.

Finally, as shown in Figure 9 we compare different backup path approach to identify the overall control packet probability. It is obvious that the probability is least for no Backup approach, as there is no much of backup transmission planned after the failure attempt in first time. Also the trend in the shown diagram depicts that the higher the backup routes, the larger the control packet probability in the network. To balance the overhead or the control packet probability and to manage the overall data transmission quality, single backup path will be the better choice as it has least probability next to no backup approach but still we have a backup path to handle

the earlier failure attempts thus making sure the continuous operation of data transmission.

5. Conclusion and Future Work

Pitching the performance of any network, keeping it uncompromised depends upon the QoS provided by the network. If that is the case, our proposed PRAC model minimizes the delay and maximized throughput for any admitted session thereby increasing the overall performance. However, the model considers the respective nodes capacity, analyzing its neighbor's capacity. The QoS provided eventually does not bring up any constrains on reliability. Considering our environment where mobility is high, our model ensures a tested backup path, which holds the key to revival from the path break. Providing many backup paths increases the control overheads, whereas PRAC considers only one tested backup path and this relatively reduces the control overheads. In general, in the mobile networks, resource reservation provides QoS. Our reservation procedure supports real time applications by avoiding the hidden and exposed terminal problems. Thus it provides collision free reservation and minimized delay. Our protocol PRAC, which includes admission control along with reservation mechanism, ensures much minimized delay. We attain a problem free reservation, but still connectivity poses a major problem in the mobile ad hoc network. In future, studies can be made on providing QoS without any connectivity issues, which paves way for many research enhancements.

Conflict of Interests

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

References

- [1] R. Ramanathan, J. Redi, and B. Technologies, "A brief overview of ad hoc networks: challenges and directions," *IEEE Communications Magazine*, vol. 40, no. 5, pp. 20–22, 2002.
- [2] H. Menouar, F. Filali, and M. Lenardi, "A survey and qualitative analysis of MAC protocols for vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 30–35, 2006.
- [3] T. B. Reddy, I. Karthigeyan, B. S. Manoj, and C. Siva Ram Murthy, "Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions," *Ad Hoc Networks*, vol. 4, no. 1, pp. 83–124, 2006.
- [4] L. Chen and W. B. Heinzelman, "QoS-aware routing based on bandwidth estimation for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 561–572, 2005.
- [5] S. Sivavakeesar and G. Pavlou, "Quality of Service aware MAC based on IEEE 802.11 for multihop ad-hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, Atlanta, Ga, USA, 2004.
- [6] X. Chen, H. M. Jones, and D. Jayalath, "Channel-aware routing in MANETs with route handoff," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 108–121, 2011.
- [7] L. Hanzo II and R. Tafazolli, "Admission control schemes for 802.11-based multi-hop mobile ad hoc networks: a survey," *IEEE*

- Communications Surveys and Tutorials*, vol. 11, no. 4, pp. 78–108, 2009.
- [8] C.-S. Hsu, J.-P. Sheu, and S.-C. Tung, “An on-demand bandwidth reservation QoS routing protocol for mobile ad hoc networks,” in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp. 198–207, June 2006.
- [9] K. C. Suresh and S. Prakash, “MAC and routing layer supports for QoS in MANET: a survey,” *International Journal of Computer Applications*, vol. 60, no. 8, pp. 40–46, 2012.
- [10] Y. Yang and R. Kravets, “Contention-aware admission control for ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, pp. 363–377, 2005.
- [11] H. Wu, Y. Liu, Q. Zhang, and Z.-L. Zhang, “SoftMAC: layer 2.5 collaborative MAC for multimedia support in multihop wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, pp. 12–25, 2007.
- [12] A. Lindgren and E. Belding-Royer, “Multi-path admission control for mobile ad hoc networks,” in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '05)*, IEEE, 2005.
- [13] L. Hanzo and R. Tafazolli, “QoS-aware routing and admission control in shadow-fading environments for multirate MANETs,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 622–637, 2011.
- [14] D. Zhang, W. Zhang, and K. Liu, “A dual channel reservation MAC protocol for mobile Ad Hoc networks,” *Communications in Information Science and Management Engineering*, vol. 3, no. 12, pp. 604–613, 2013.
- [15] Z. Tang and J. J. Garcia-Luna-Aceves, “Hop reservation multiple access for multichannel packet radio networks,” *Computer Communications*, vol. 23, no. 10, pp. 877–886, 2000.
- [16] C. Zhu and M. S. Corson, “A five-phase reservation protocol (FPRP) for mobile ad hoc networks,” in *Proceedings of the 17th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '98)*, pp. 322–331, San Francisco, Calif, USA, 1998.
- [17] Q. Yang, Y. Zhuang, and J. Shi, “An improved contention access mechanism for FPRP to increase throughput,” *ETRI Journal*, vol. 35, no. 1, pp. 58–68, 2013.
- [18] J.-R. Cha, K.-C. Go, J.-H. Kim, and W.-C. Park, “TDMA-based multi-hop resource reservation protocol for real-time applications in tactical mobile adhoc network,” in *Proceedings of the IEEE Military Communications Conference (MILCOM '10)*, pp. 1936–1941, November 2010.
- [19] I. Bekmezci and F. Alagöz, “Delay sensitive, energy efficient and fault tolerant distributed slot assignment algorithm for wireless sensor networks under convergecast data traffic,” *International Journal of Distributed Sensor Networks*, vol. 5, no. 5, pp. 557–575, 2009.
- [20] Y. Li, B. Sun, X. Luo, and N. Xiong, “A time slot reservation in modified TDMA-based ad hoc networks with directional antennas,” *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 636797, 12 pages, 2013.
- [21] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile Computing*, vol. 353, pp. 153–181, Kluwer Academic, 1996.
- [22] Y. Chang, Q. Liu, X. Jia, and K. Zhou, “Routing and transmission scheduling for minimizing broadcast delay in multi-rate wireless mesh networks using directional antennas,” *Wireless Communications and Mobile Computing*, 2012.
- [23] Y. Kim, H. Ko, S. Pack, W. Lee, and X. Shen, “Mobility-aware call admission control algorithm with handoff queue in mobile hotspots,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 8, pp. 3903–3912, 2013.
- [24] L. Khoukhi, H. Badis, L. Merghem-Boulaïhia, and M. Essegghir, “Admission control in wireless ad hoc networks: a survey,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, article 109, 2013.
- [25] J. Youn, S. Pack, and Y.-G. Hong, “Distributed admission control protocol for end-to-end QoS assurance in ad hoc wireless networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, article 163, 2011.
- [26] H. Zhu and I. Chlamtac, “Admission control and bandwidth reservation in multi-hop ad hoc networks,” *Computer Networks*, vol. 50, no. 11, pp. 1653–1674, 2006.
- [27] T. Liu, W. Liao, and J.-F. Lee, “Distributed contention-aware call admission control for IEEE 802.11 multi-radio multi-rate multi-channel wireless mesh networks,” *Mobile Networks and Applications*, vol. 14, no. 2, pp. 134–142, 2009.

Research Article

A Novel Multiobjective Evolutionary Algorithm Based on Regression Analysis

Zhiming Song,¹ Maocai Wang,^{1,2} Guangming Dai,¹ and Massimiliano Vasile²

¹*School of Computer, China University of Geosciences, Wuhan 430074, China*

²*Department of Mechanical & Aerospace Engineering, University of Strathclyde, Glasgow G1 1XJ, UK*

Correspondence should be addressed to Maocai Wang; cugwmc@gmail.com

Received 23 June 2014; Revised 15 September 2014; Accepted 30 December 2014

Academic Editor: Shifei Ding

Copyright © 2015 Zhiming Song et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As is known, the Pareto set of a continuous multiobjective optimization problem with m objective functions is a piecewise continuous $(m - 1)$ -dimensional manifold in the decision space under some mild conditions. However, how to utilize the regularity to design multiobjective optimization algorithms has become the research focus. In this paper, based on this regularity, a model-based multiobjective evolutionary algorithm with regression analysis (MMEA-RA) is put forward to solve continuous multiobjective optimization problems with variable linkages. In the algorithm, the optimization problem is modelled as a promising area in the decision space by a probability distribution, and the centroid of the probability distribution is $(m - 1)$ -dimensional piecewise continuous manifold. The least squares method is used to construct such a model. A selection strategy based on the nondominated sorting is used to choose the individuals to the next generation. The new algorithm is tested and compared with NSGA-II and RM-MEDA. The result shows that MMEA-RA outperforms RM-MEDA and NSGA-II on the test instances with variable linkages. At the same time, MMEA-RA has higher efficiency than the other two algorithms. A few shortcomings of MMEA-RA have also been identified and discussed in this paper.

1. Introduction

Evolutionary algorithm has become an increasingly popular design and optimization tool in the last few years [1]. Although there have been a lot of researches about evolutionary algorithm, there are still many new areas that needed to be explored with sufficient depth. One of them is how to use the evolutionary algorithm to solve multiobjective optimization problems. The first implementation of a multiobjective evolutionary algorithm dates back to the mid-1980s [2]. Since then, many researchers have done a considerable amount of works in the area, which is known as multiobjective evolutionary algorithm (MOEA).

Because of the ability to deal with a set of possible solutions simultaneously, evolutionary algorithm seems particularly suitable to solve multiobjective optimization problems. The ability makes it possible to search several members of the Pareto-optimal set in a single run of the algorithm [3]. Obviously, evolutionary algorithm is more effective than the

traditional mathematical programming methods in solving multiobjective optimization problem because the traditional methods need to perform a series of separate runs [4].

The current MOEA research mainly focuses on some highly related issues [5]. The first issue is the fitness assignment and diversity maintenance. Some techniques such as fitness sharing and crowding have been frequently used to maintain the diversity of the search. The second issue is the external population. The external population is used to record nondominated solutions found during the search. There have been some efforts on how to maintain and utilize such an external population. The last issue is the combination of MOEA and local search. Researches have shown that the combination of evolutionary algorithm and local heuristics search outperforms traditional evolutionary algorithms in a wide variety of scalar objective optimization problems [4, 6].

However, there are little researches focusing on the way to generate new solutions in MOEA. Currently, most MOEAs directly adopt traditional genetic operators such as crossover

and mutation. These methods have not fully utilized the characteristics of MOP when generating new solutions. Some researches show that MOEA fails to solve MOPs with variable linkages, and the recombination operators are crucial to the performance of MOEA [7]. It has been noted that under mild smoothness conditions, the Pareto set (PS) of a continuous MOP is a piecewise continuous $(m - 1)$ -dimensional manifold, where m is the number of the objectives. However, as analyzed in [8], this regularity has not been exploited explicitly by most current MOEA.

In 2005, Zhou et al. proposed to extract regularity patterns of the Pareto set by using local principal component analysis (PCA) [9]. They had also studied two naive hybrid MOEAs. In the two MOEAs, some trial solutions were generated by traditional genetic operators and others by sampling from probability models based on regularity patterns in 2006 [10].

In 2007, Zhang et al. conducted a further and thorough investigation along their previous works in [9, 10]. They proposed a regularity model-based multiobjective estimation of distribution algorithm and named it as RM-MEDA [5]. At each generation, the proposed algorithm models a promising area in the decision space by a probability distribution whose centroid is a $(m - 1)$ -dimensional piecewise continuous manifold. The local principal component analysis algorithm is used to build such a model. Systematic experiments have shown that RM-MEDA outperforms some other algorithms on a set of test instances with variable linkages.

In 2008, Zhou et al. proposed a probabilistic model based multiobjective evolutionary algorithm to approximate PS and PF (Pareto front) for a MOP in this class simultaneously and named the algorithm as MMEA [11]. They proposed two typical classes of continuous MOPs as follows. One class is that PS and PF are of the same dimensionality while the other one is that PF is a $(m - 1)$ -dimensional continuous manifold and PS is a continuous manifold with a higher dimensionality. There is a class of MOPs, in which the dimensionalities of PS and PF are different so that a good approximation to PF might not approximate PS very well. MMEA could promote the population diversity both in the decision spaces and in the objective spaces.

Modeling method is a crucial part for MOEA because it determines the performance of the algorithms. Zhang et al. built such a model by local principal component analysis (PCA) algorithm [5]. The test results show that the method has great performance over some instances with linkage variables. However, there are still some shortcomings about the method. The first shortcoming is that RM-MEDA needs extra CPU time for running local PCA at each generation. The second one is that the model is just linear fitting for all types of PS, including the one with nonlinear linkage variables, which enable that the result may be not accurate.

In the paper, we proposed a model-based multiobjective evolutionary algorithm with regression analysis, which is named as MMEA-RA. In MMEA-RA, a new modeling method based on regression analysis is put forward. In the method, least squares method (LSM) is used to fit a 1-dimensional manifold in high-dimensional space. Because least squares can fit any type of curves through its model,

the shortcomings of RM-MEDA can be avoided, especially for the instances with nonlinkage variables.

The rest of this paper is organized as follows. After defining the continuous multiobjective optimization problem in Section 2, the new model of multiobjective evolutionary algorithm based on regression analysis is put forward in Section 3. Then, a description of the test cases for MMEA-RA follows in Section 4. After presenting the results of the tests, the performance of MMEA-RA is analyzed and some conclusions are given in Section 5.

2. Problem Definition

In this paper, the continuous multiobjective optimization problem is defined as follows [5]:

$$\begin{aligned} \min \quad & F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T \\ \text{s.t.} \quad & x = (x_1, \dots, x_n)^T \in X, \end{aligned} \quad (1)$$

where $X \subset R^n$ is the decision space and $x = (x_1, \dots, x_n)^T$ is the decision vector. $F : X \rightarrow R^m$ consists of m real-valued continuous objective functions $f_i(x)$ ($i = 1, \dots, m$). R^m is the objective space.

Let $a = (a_1, \dots, a_n)^T \in R^n$ and $b = (b_1, \dots, b_n)^T \in R^n$ be two vectors, and a is said to dominate b , denoted by $a < b$, if $a_i \leq b_i$ for all $i = 1, \dots, n$, and $a \neq b$. A point $x^* \in X$ is called (globally) Pareto optimal if there is no $x \in X$ such that $F(x) < F(x^*)$. The set of all Pareto-optimal points, denoted by PS, is called the Pareto set. The set of all Pareto objective vectors is called the Pareto front, denoted by PF.

3. Algorithm

3.1. Basic Idea. Under certain smoothness assumptions, it can be induced from the Karush-Kuhn-Tucker condition that the PS of a continuous MOP defines a piecewise continuous $(m - 1)$ -dimensional manifold in the decision space [12]. Therefore, the PS of a continuous biobjective optimization problem is a piecewise continuous curve in R^2 .

The population in the decision space in a MOEA for (1) will hopefully approximate the PS and is uniformly scattered around the PS as the search goes on. Therefore, we can envisage the points in the population as independent observations of a random vector $\xi \in R^n$ whose centroid is the PS of (1). Since the PS is a $(m - 1)$ -dimensional piecewise continuous manifold, ξ can be naturally described by

$$\xi = \zeta + \varepsilon, \quad (2)$$

where ζ is uniformly distributed over a piecewise continuous $(m - 1)$ -dimensional manifold, and ε is an n -dimensional zero-mean noise vector. Figure 1 illustrates the basic idea.

3.2. Algorithm Framework. In this paper, a model-based multiobjective evolutionary algorithm based on regression analysis is put forward to solve continuous multiobjective

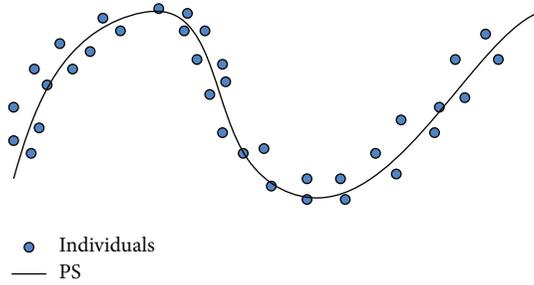


FIGURE 1: Individual solutions should be scattered around the PS in the decision space in a successful MOEA.

optimization problems with variable linkages. The algorithm is named as MMEA-RA. The algorithm works as follows.

MMEA-RA

Step 1 (initializing). Set $t = 0$. Generate an initial population $Pop(0)$ and compute the value F of each individual solution in $Pop(0)$.

Step 2 (stopping). If stopping condition is met, the algorithm stops and returns the nondominated solutions in $Pop(t)$, and their corresponding F vectors constitute an approximation to the PF.

Step 3 (modeling). Build the probability model in $Pop(t)$ to fit expression (2),

- (3.1) to compute the coefficients a_k for $k = 0, 1, \dots, j$ by solving the matrix in expression (10);
- (3.2) to compute the manifold $\psi = \{x = (x_1, \dots, x_n) \in R^n\}$ by expression (11);
- (3.3) to generate a n -dimensional zero-mean noise vector between $(-noise, noise)$ randomly based on expressions (13) and (14).

Step 4 (reproducing). Generate a new solution set Q from expression (2). Evaluate the value F of each solution in Q .

Step 5 (selecting). Select N individuals from $Q \cup Pop(t)$ to create $Pop(t + 1)$.

Step 6. Set $t = t + 1$ and go to Step 2.

In the following Section 3.3, the implementation of modeling, reproducing, and selecting of the above algorithm will be given in detail.

3.3. Modeling. Fitting expression (2) to the points in $Pop(t)$ is highly related to principal curve analysis, which aims at finding a central curve of a set of points in R^n [13]. However, most current algorithms for principal curve analysis are rather expensive due to the intrinsic complexity of their models. RM-MEDA uses the $(m - 1)$ -dimensional local principal component analysis (PCA) algorithm [14]; it is less complex compared with most algorithms for principal

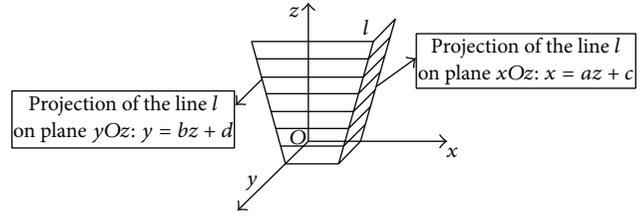


FIGURE 2: Illustration of the geometric meaning of expression (3).

curve analysis. However, it needs much more CPU time compared with the traditional evolutionary algorithms which adopt genetic recombination operators such as crossover and mutation. Moreover, the PS cannot be exactly described by local PCA because it only uses linear curves to approximate the model at one cluster of $Pop(t)$.

In this implementation, we do not make use of clustering method in the modeling process. We try to find the principal curve of the whole points in $Pop(t)$, not just the local part of them. As is known, least squares approach is a simple and effective method for linear curve fitting and nonlinear curve fitting, such as polynomial or exponential curve fitting. Then we consider whether this technique could be made use of to describe expression (2).

For the sake of simplicity, it could be assumed that the centroid of ξ is a manifold ψ in formula (2), and ζ is uniformly distributed on ψ . ψ is a $(m - 1)$ -dimensional hyperrectangle. Particularly, in the case of two objectives, ψ is a curve segment in R^2 .

A line in 3-dimensional space can be expressed as

$$x = az + c, \quad y = bz + d, \tag{3}$$

where $a, b, c,$ and d are the coefficients of the expression.

The geometric meaning of expression (3) is that a 3-dimensional line l can be seen as the intersecting line of two planes $m_1: x = az + c$ and $m_2: y = bz + d$. Figure 2 illustrates this meaning.

The expression $x = az + c$ can be seen as the projection of the line l in xOz plane, and $y = bz + d$ is the one in the yOz plane.

As a 3-dimensional line can be expressed by the intersecting of 2 planes, then a n -dimensional line can be expressed as the intersecting of $(n - 1)$ planes as

$$\begin{aligned} x_2 &= a_1 + b_1x_1, \\ x_3 &= a_2 + b_2x_1, \\ &\vdots \\ x_n &= a_{n-1} + b_{n-1}x_1. \end{aligned} \tag{4}$$

Expression $x_i = a_{i-1} + b_{i-1}x_1$ can be regarded as the projection of the n -dimensional line in x_iOx_1 plane.

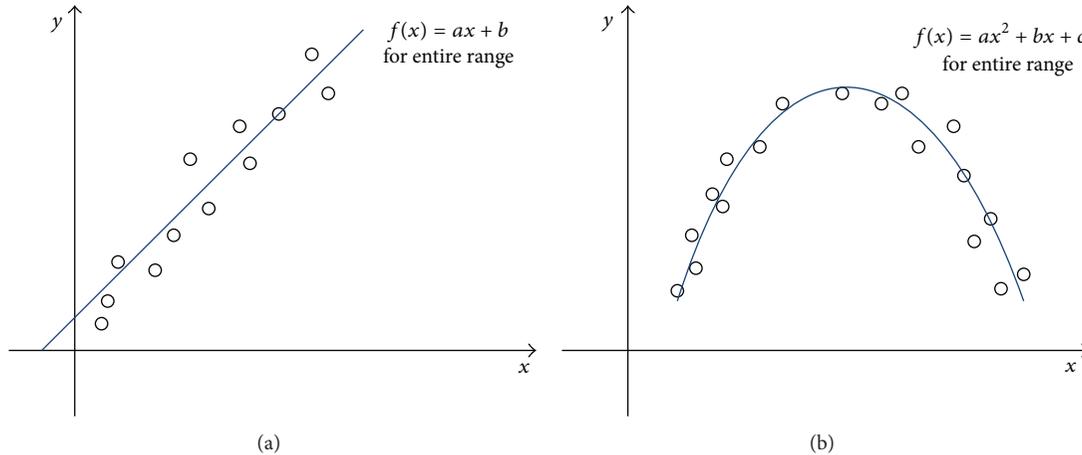


FIGURE 3: Illustration of the least squares approach (a) linear (b) nonlinear.

By expression (4), we further conclude that a n -dimensional curve can be regarded as the intersecting of $(n - 1)$ surface, and expression (5) shows this idea:

$$\begin{aligned}
 x_2 &= a_{1,0} + a_{1,1}x_1 + a_{1,2}x_1^2 + \dots + a_{1,p}x_1^p, \\
 x_3 &= a_{2,0} + a_{2,1}x_1 + a_{2,2}x_1^2 + \dots + a_{2,p}x_1^p, \\
 &\vdots \\
 x_n &= a_{n-1,0} + a_{n-1,1}x_1 + a_{n-1,2}x_1^2 + \dots + a_{n-1,p}x_1^p.
 \end{aligned}
 \tag{5}$$

Expression $x_j = a_{j-1,0} + a_{j-1,1}x_1 + a_{j-1,2}x_1^2 + \dots + a_{j-1,p}x_1^p$ can be regarded as the approximate projection of the n -dimensional curve in x_jOx_1 surface. Each expression is a p -order polynomial. (x_1, \dots, x_n) is a point on the n -dimensional curve. Then the thing that we need to do is to find out all the coefficients $a_{i,j}$, which could make the curve fit the population in the decision space well, and here we used least squares approach method to help us find out the best coefficients.

Least squares approach is mainly used to fit the curve, that is to say, to capture the trend of the data by assigning a single function across the entire range. Figure 3 shows the idea.

In Figure 3, Figure 3(a) looks linear in trend, so we can fit the curve by choosing a general form of the straight line $f(x) = ax + b$, and then the goal is to identify the coefficients a and b such that $f(x)$ fits the date well, the method to identify the two coefficients is called as linear regression. Figure 3(b) looks nonlinear, we use higher polynomial $f(x) = ax^2 + bx + c$, and the goal is to find out the coefficients a, b , and c such that $f(x)$ fits the date well. It is called as nonlinear regression compared with linear regression. In fact, there are a lot of functions with different shapes that depend on the coefficients. The methods to find out the best coefficients are just called as regression analysis (RA).

Consider the general form for a polynomial with order j :

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_jx^j = \sum_{k=0}^j a_kx^k. \tag{6}$$

How can we choose the coefficients that best fit the curve to the data? The idea of least squares approach is to find a curve that gives minimum error between data y and the fitting curve $f(x)$. As is shown in Figure 4, we can firstly add up the length of all the solid and dashed vertical lines and then pick curve with minimum total error. The general expression for any error using the least squares approach is

$$\begin{aligned}
 \text{err} = \sum_{i=1}^n (d_i)^2 &= (y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 \\
 &+ \dots + (y_n - f(x_n))^2.
 \end{aligned}
 \tag{7}$$

For expression (7), we want to minimize the error err . Replace $f(x)$ in expression (7) with the expression (6), and then we have

$$\text{err} = \sum_{i=1}^n \left(y_i - \sum_{k=0}^j a_kx_i^k \right)^2, \tag{8}$$

where n is the number of data points given, i is the current data points being summed, and j is the polynomial order. To find the best line means to minimize the square of the distance error between line and data points. Find the set of coefficients a_0, a_1, \dots, a_j , that is to say, to minimize expression (8).

In Figure 4, there are four data points and two fitting curves $f_1(x)$ and $f_2(x)$. Obviously, $f_1(x)$ is better than $f_2(x)$ because there is smaller error between the four points and the fitting curve $f_1(x)$.

To minimize expression (8), take the derivative with respect to each coefficient a_k for $k = 0, 1, \dots, j$, and set each to zero:

$$\frac{\partial \text{err}}{\partial a_0} = -2 \sum_{i=1}^n \left(y_i - \sum_{k=0}^j a_kx_i^k \right) = 0,$$

$$\frac{\partial \text{err}}{\partial a_1} = -2 \sum_{i=1}^n \left(y_i - \sum_{k=0}^j a_kx_i^k \right) x = 0,$$

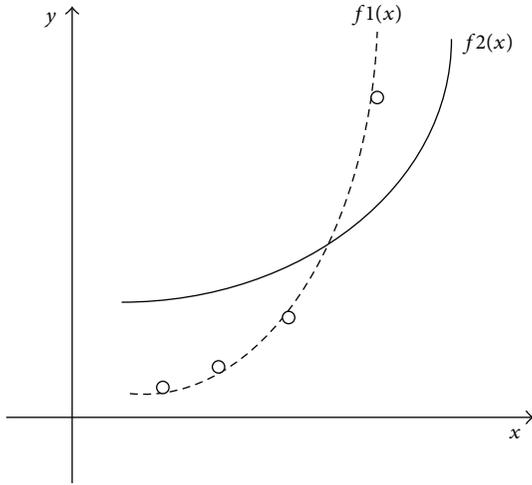


FIGURE 4: Four data points and two different curves.

$$\begin{aligned} & \vdots \\ \frac{\partial \text{err}}{\partial a_j} &= -2 \sum_{i=1}^n \left(y_i - \sum_{k=0}^i a_k x_i^k \right) x_i^j = 0. \end{aligned} \tag{9}$$

Rewrite these $j + 1$ equations, and put into matrix form:

$$\begin{pmatrix} n & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{j+1} \\ & & \vdots & & \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & \cdots & \sum x_i^{j+j} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_j \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum x_i y_i \\ \vdots \\ \sum x_i^j y_i \end{pmatrix}. \tag{10}$$

The coefficients a_k for $k = 0, 1, \dots, j$ can be solved by matrix computation.

With the above work, we can describe the 1-dimensional manifold ψ as

$$\begin{aligned} \psi &= \left\{ x = (x_1, \dots, x_n) \in R^n \mid \right. \\ & \left. x_i = \sum_{j=0}^p a_{i-1,j} x_1^j, \quad a_1 - 0.25(b_1 - a_1) \right. \\ & \left. \leq x_i \leq b_1 + 0.25(b_1 - a_1), \quad i = 2, \dots, n \right\}, \end{aligned} \tag{11}$$

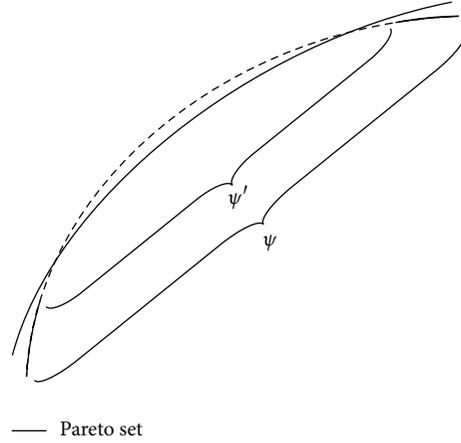


FIGURE 5: Illustration of extension.

where p is the polynomial order and a_1 and b_1 are the minimum and maximum values on x_1 :

$$a_1 = \min_{1 \leq j \leq N} x_1^j, \quad b_1 = \max_{1 \leq j \leq N} x_1^j. \tag{12}$$

In order to approximate the PS better, ψ is extended by 50% along x_1 . Figure 5 shows this idea. In Figure 5, ψ' could not approximate the PS very well, but its extension ψ can provide a better approximation.

When we find out the coefficients $a_{i,j}$ ($i = 1, \dots, n-1, j = 0, \dots, p$) based on least square approach above, we could get ζ in expression (2). ζ is generated over ψ uniformly and randomly.

In expression (2), ε is a n -dimensional zero-mean noise vector, and it is designed as the following description:

$$\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n), \tag{13}$$

where ε_i is a random number between $(-\text{noise}, \text{noise})$. The noise is changed from big to small as the generation goes on because big noise can accelerate the convergence of the population in the early generation and small noise can maintain the accuracy of the population in the end. Expression (14) shows the implementation:

$$\text{noise} = F_0 * 10^{\frac{(1-\text{maxGen})}{(\text{maxGen}+1-\text{curGen})}}, \tag{14}$$

where maxGen is the max generation of the algorithm and is set to be 200 and curGen is the current generation. F_0 is set to be 0.2 when the algorithm begins. Then the noise is changed from 0.2 to 0.02. The trends of the noise can be seen in Figure 6. As is shown in Figure 6, the noise decreases as the generation increases, and it will be stable after the 160th generation.

3.4. Reproducing. It is desirable that final solutions are uniformly distributed on the PS. Therefore, in order to maintain the diversity of the solution, in this paper, the new solution is generated uniformly and randomly as follows.

Step 1. Generate a point x' from ψ uniformly and randomly.

TABLE 1: Test instance.

Test case	Variables	Objectives
T1	$[0, 1]^n \times [0, 10]^{n-1}$	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - \sqrt{\frac{f_1(x)}{g(x)}} \right]$ $g(x) = \frac{1}{4000} \sum_{i=2}^n (x_i^2 - x_1)^2 - \prod_{i=2}^n \cos\left(\frac{x_i^2 - x_1}{\sqrt{i-1}}\right) + 2$

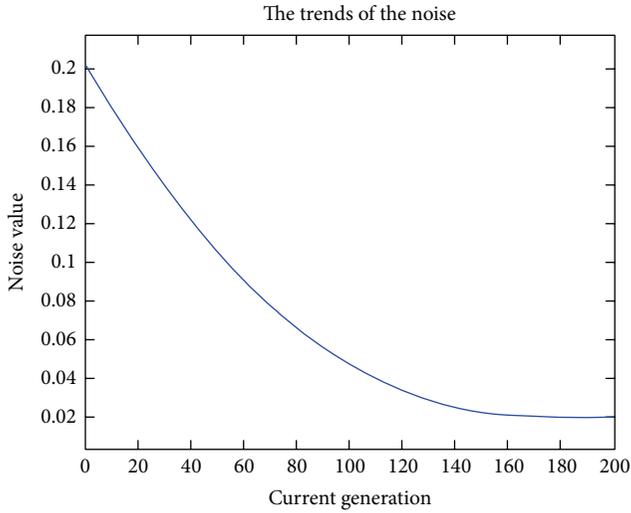


FIGURE 6: The trends of the noise.

Step 2. Generate a noise vector ε' in expression (13).

Step 3. Return $x = x' + \varepsilon'$.

In Step 3 of the algorithm framework of MMEA-RA, N new solutions can be produced by repeating above all steps N times.

3.5. *Selecting.* The selection procedure used in this paper is the same in the procedure used in [5], which is based on the nondominated sorting of NSGA-II [15]. The selection procedure is called as NDS-selection.

The main idea of NDS-selection is to divide $Q \cup \text{Pop}(t)$ into different fronts F_1, F_2, \dots, F_l such that the j th front F_j contains all the nondominated solutions in $\{Q \cup \text{Pop}(t)\} \setminus (\bigcup_{i=1}^{j-1} F_i)$. Therefore, there is no solution in $\{Q \cup \text{Pop}(t)\} \setminus (\bigcup_{i=1}^{j-1} F_i)$ that could dominate a solution in F_j . Roughly speaking, F_1 is the best nondominated front in $Q \cup \text{Pop}(t)$, and F_2 is the second best nondominated front, and so on. The detailed procedure of NDS-selection can be found in [5].

4. Test Case

4.1. *Performance Metric.* In this paper, the performance metric used to evaluate the solutions is the convergence metric γ , which is also the common performance metric in multiobjective optimization algorithm [16].

The metric γ measures that the solutions will be convergent to a known set of Pareto-optimal solutions. We find a set of 500 uniformly solutions from the true Pareto-optimal front in the objective space. And then to compute the minimum Euclidean distance of each solution from chosen solutions on the Pareto-optimal front. The average of these distances is used as the metric γ .

4.2. *General Experimental Setting.* There are three algorithms employed to solve the test instance for a comparison. These three algorithms are RM-MEDA, NSGA-II, and MMEA-RA, while MMEA-RA is the new algorithm proposed in this paper.

The three algorithms are implemented by C++. The machine used in the test is Core 2 Duo (2.4 GHz, 2.00 GB RAM). The experiment setting is as follows.

The number of new trial solutions generated at each generation is set to be 100 for all tests.

The number of decision variables is set to be 30 for all tests.

Parameter setting in RM-MEDA: the number of cluster K is set to be 5 in local PCA algorithm.

Parameter setting in MMEA-RA: the order is set to be 2.

We run each algorithm independently 10 times for the test instance. The algorithms stop after a given number of generations. The maximal number of generations in three algorithm is 1000.

Table 1 gives the test instance [5]. In the test instance, the feasible decision space is a hyperrectangle. There are nonlinear variable linkages in the test case. Furthermore, the test instance has many local Pareto fronts since its $g(x)$ has many locally minimal points. It also has some characteristics such as concave PF, nonlinear variable linkage, and multimodal with Griewank function.

If an element of solution x , sampled from MMEA-RA or RM-MEDA, is out of the boundary, we simply reset its value to a randomly selected value inside the boundary.

4.3. *Performance Analysis.* The evolution of the average γ -metric of the nondominated solutions for the test case is shown in Figure 7. It should be noted that the solutions of all three algorithms are stable when the iteration generation is more than 300. After the solutions are stable, the convergence values of the three algorithms are small than 0.1. Because we adopt the average of the minimum Euclidean distance of each

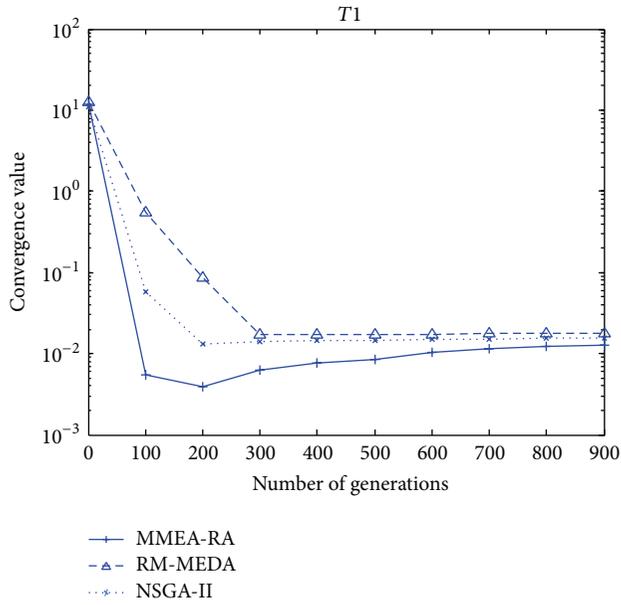


FIGURE 7: The evolution of the average γ -metric of the nondominated solutions in three algorithms for $T1$.

solution from chosen solutions as the metric γ , the smaller the convergence values, the better the convergence metric γ . As is shown by Figure 7, among the three algorithms, MMEA-RA has best convergence performance and NSGA-II and RM-MEDA follow.

Figure 8 shows the final nondominated solutions and fronts obtained by MMEA-RA on the test case. Figure 8(a) is the result with the lowest γ -metric obtained in 10 runs while Figure 8(b) is all the 10 fronts in 10 runs. It can be seen that the nondominated fronts with the lowest γ -metric are very close to the Pareto front, especially when f_1 tends to 0 and f_2 tends to 1. It can also be noted that the nondominated solutions in every run have some small fluctuations around the Pareto front.

The final nondominated solutions and fronts obtained by RM-MEDA on the test case are shown in Figure 9. Similarly, Figure 9(a) is the result with the lowest γ -metric obtained in 10 runs while Figure 9(b) gives all the 10 fronts in 10 runs. Similar to Figure 8, the nondominated solution(s) in Figures 9(a) and 9(b) are marked with red. The Pareto fronts are marked with blue. The Pareto fronts are given in Figures 9(a) and 9(b) only for comparing the quality of the nondominated solutions. It can be seen that the nondominated front with the lowest γ -metric is very consistent with the Pareto front although there are some differences between them. In particular, it should be noted that all results in 10 runs from RM-MEDA match the Pareto front better than MMEA-RA. But it also should be noted that there is an isolated point in the nondominated solutions for all 10 runs in Figure 9(b), maybe because RM-MEDA falls into a local minimum and could not jump out.

The final nondominated solutions and fronts obtained by NSGA-II on the test case are shown in Figure 10. Again, Figure 10(a) means the result with the lowest γ -metric

TABLE 2: The comparison of the running time (unit: ms).

	NSGA-II	RM-MEDA	MMEA-RA
The running time	79.368	127.543	92.771

obtained in 10 runs and Figure 10(b) means all 10 fronts in 10 runs. As is shown in Figure 10(a), the nondominated front with the lowest γ -metric is close to the Pareto front but different to the result obtained by MMEA-RA. The nondominated front with the lowest γ -metric in NSGA-II does not tend to the Pareto front very close. It does also not match the Pareto front as good as the result obtained by REMEDA. Similarly, the nondominated solutions in every run have some small fluctuations around the Pareto front.

The running time of the three algorithms are given in Table 2. From the point of the running time, as is shown in Table 2, among the three algorithms, NSGA-II is the best, then MMEA-RA follows, and RM-MEDA is the worst. This result is consistent with the main idea of the three algorithms. In RM-MEDA, local principal component analysis (PCA) is used to construct the model, and it needs extra CPU time for running local PCA at each generation. In MMEA-RA, the least squares method is used to construct the model, and it is easy to run the least squares by matrix computation. MMEA-RA is slower than NSGA-II because the selection in MMEA-RA is based on NSGA-II.

Obviously, it can be seen that the nondominated front with the lowest γ -metric obtained by MMEA-RA is the closest to the Pareto front in the three algorithms, which shows MMEA-RA is suitable to solve the problem with some characteristics such as concave PF, nonlinear variable linkage, and multimodal with Griewank function. In contrast, the results in 10 runs from RM-MEDA mostly match the Pareto front, which shows the performance of RM-MEDA is good in common.

5. Conclusion

In this paper, a model-based multiobjective evolutionary algorithm based on regression analysis (MMEA-RA) is put forward to solve continuous multiobjective optimization problems with variable linkages. MMEA-RA models a promising area whose centroid is a complete and continuous curve described by expression (8). Because of this feature, MMEA-RA does not need to cluster the population. The least squares approach is simple yet enough to describe the nonlinear principal curve using the polynomial model.

The less CPU time of MMEA-RA does not come without a price. MMEA-RA samples points uniformly around the PS in the decision variable space, and the centroid of the model is not piecewise but complete curve. This makes it very difficult for MMEA-RA to approximate the whole PF. The experimental results also reveal that MMEA-RA may fail in test instances with many local Pareto fronts.

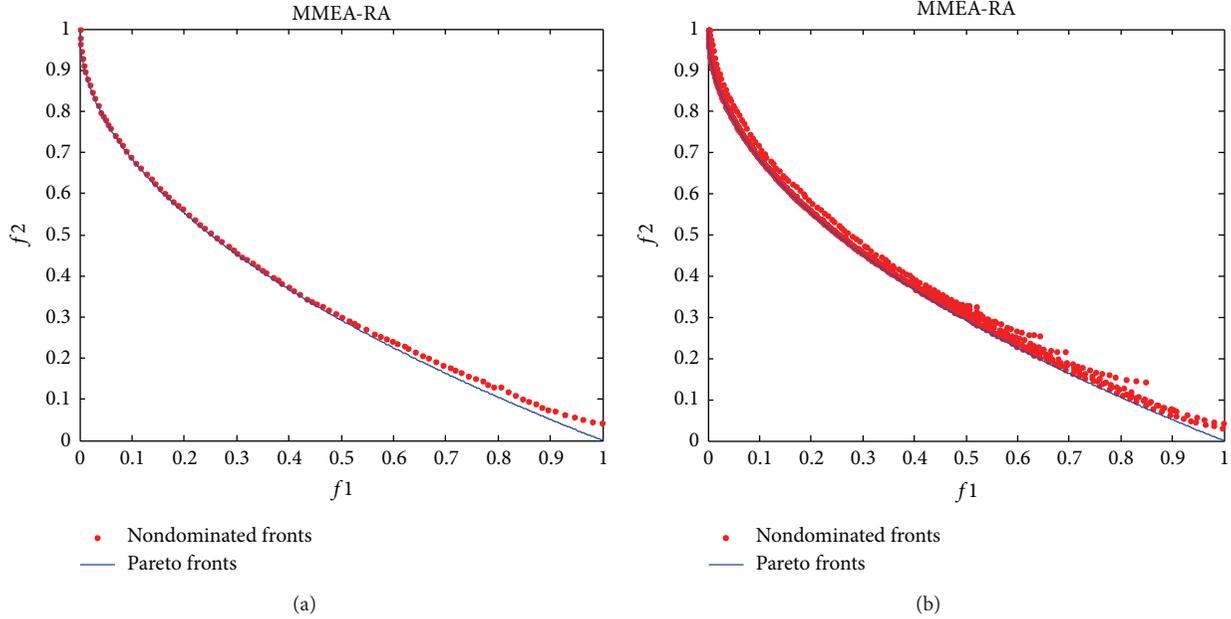


FIGURE 8: The final nondominated solutions and fronts found by MMEA-RA. (a) The result with the lowest γ -metric and (b) all the 10 fronts in 10 runs.

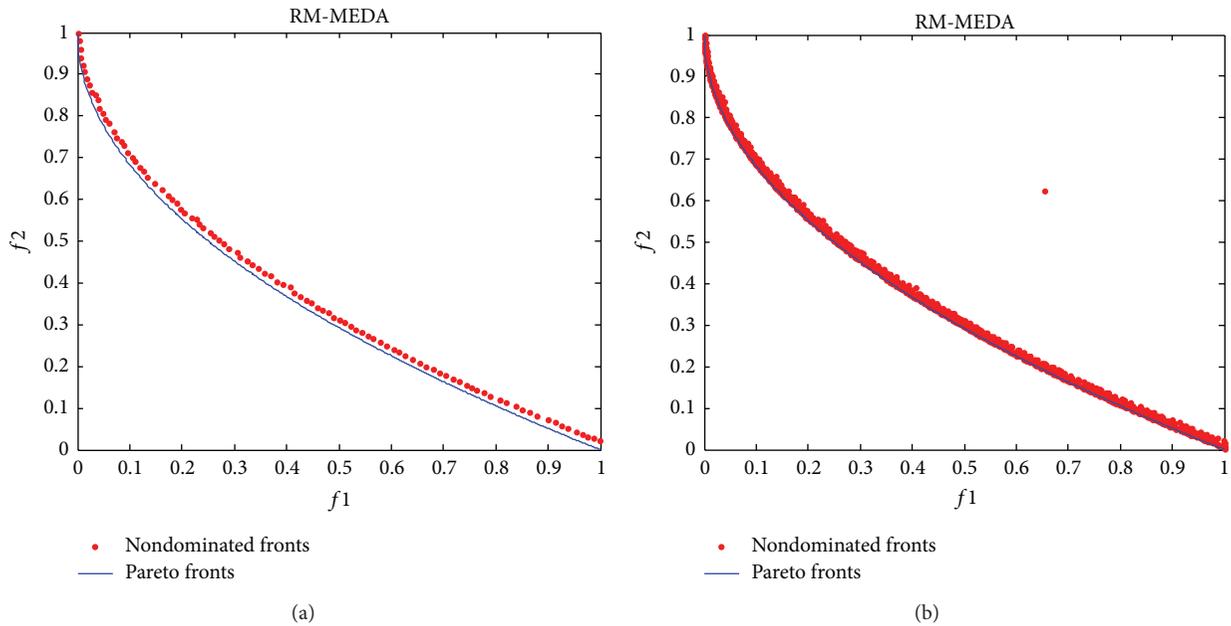


FIGURE 9: The final nondominated solutions and fronts found by RE-MEDA. (a) The result with the lowest γ -metric and (b) all the 10 fronts in 10 runs.

The future research topics along this line should include the following points:

- (1) designing an accurate model to describe the decision space: as the case of 3 objectives, the PS is a surface, so expression (8) cannot solve the problems with 3 objectives right now;
- (2) combining MMEA-RA with traditional genetic algorithms using operators such as crossover and mutation for accelerating the convergence of the algorithm;
- (3) improving the method to calculate random noise value to make the final population more convergent;
- (4) considering the distribution of the solutions in the objective space when sampling solutions from

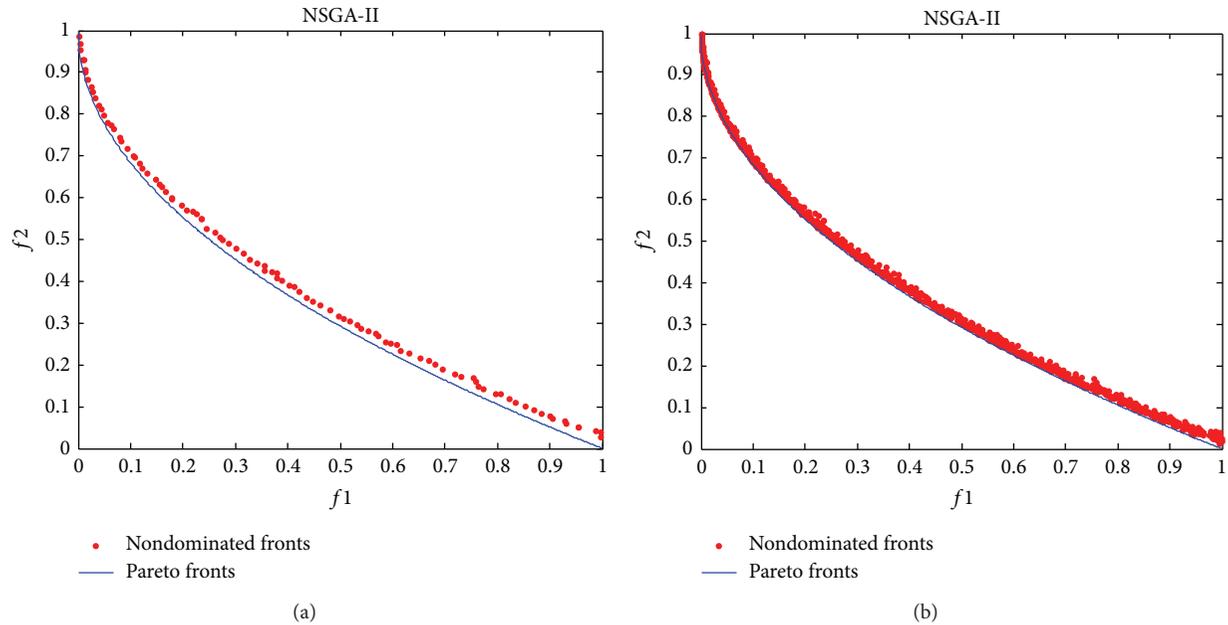


FIGURE 10: The final nondominated solutions and fronts found by NSGA-II. (a) The result with the lowest γ -metric and (b) all the 10 fronts in 10 runs.

the models to improve the performance of MMEA-RA on the instance;

- (5) incorporating effective global search techniques for scalar optimization into MMEA-RA in order to improve its ability for global search.

Conflict of Interests

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

Acknowledgments

Maocai Wang thanks the Special Financial Grant from China Postdoctoral Science Foundation (Grant no. 2012T50681), the General Financial Grant from China Postdoctoral Science Foundation (Grant no. 2011M501260), the Grant from China Scholarship Council (Grant no. 201206415018), and the Fundamental Research Funds for the Central Universities at China University of Geosciences (Grant no. CUG120114). Guangming Dai thanks the Grant from Natural Science Foundation of China (Grant nos. 61472375 and 60873107) and the 12th Five-Year Preresearch Project of Civil Aerospace in China.

References

- [1] T. Back, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*, Oxford University Press, Oxford, UK, 1997.
- [2] J. D. Schaffer, *Multiple objective optimization with vector evaluated genetic algorithms [Ph.D. thesis]*, Vanderbilt University, Nashville, Tenn, USA, 1984.
- [3] C. A. C. Coello, D. A. van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, New York, NY, USA, 2002.
- [4] C. A. C. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *Knowledge and Information Systems*, vol. 1, no. 3, pp. 269–308, 1999.
- [5] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.
- [6] A. Jaskiewicz, "Genetic local search for multi-objective combinatorial optimization," *European Journal of Operational Research*, vol. 137, no. 1, pp. 50–71, 2002.
- [7] K. Deb, A. Sinha, and S. Kukkonen, "Multi-objective test problems, linkages, and evolutionary methodologies," in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference (GECCO '06)*, pp. 1141–1148, Seattle, Wash, USA, July 2006.
- [8] Y. Jin and B. Sendhoff, "Connectedness, regularity and the success of local search in evolutionary multi-objective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '03)*, vol. 3, pp. 1910–1917, IEEE, Canberra, Australia, December 2003.
- [9] A. Zhou, Q. Zhang, Y. Jin, E. Tsang, and T. Okabe, "A model-based evolutionary algorithm for Bi-objective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '05)*, pp. 2568–2575, Edinburgh, UK, September 2005.
- [10] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 3234–3241, Vancouver, Canada, July 2006.
- [11] A. Zhou, Q. Zhang, and Y. Jin, "Approximating the set of pareto-optimal solutions in both the decision and objective spaces by

- an estimation of distribution algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1167–1189, 2009.
- [12] O. Schutze, S. Mostaghim, M. Dellnitz, and J. Teich, “Covering Pareto sets by multilevel evolutionary subdivision techniques,” in *Evolutionary Multi-Criterion Optimization: Second International Conference, EMO 2003, Faro, Portugal, April 8–11, 2003. Proceedings*, vol. 2632 of *Lecture Notes in Computer Science*, pp. 118–132, Springer, Berlin, Germany, 2003.
- [13] T. Hastie and W. Stuetzle, “Principal curves,” *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 502–516, 1989.
- [14] N. Kambhatla and T. K. Leen, “Dimension reduction by local principal component analysis,” *Neural Computation*, vol. 9, no. 7, pp. 1493–1516, 1997.
- [15] M. Wang, G. Dai, and H. Hu, “Improved NSGA-II algorithm for optimization of constrained functions,” in *Proceedings of the International Conference on Machine Vision and Human-Machine Interface*, pp. 673–675, Kaifeng, China, April 2010.
- [16] K. Deb and S. Jain, “Running performance metrics for evolutionary multiobjective optimization,” Tech. Rep. 2002004, KanGAL, India Institute of Technology, Kanpur, India, 2004.

Research Article

A Novel Psychovisual Threshold on Large DCT for Image Compression

Nur Azman Abu¹ and Ferda Ernawan^{1,2}

¹ Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, 76100 Melaka, Malaysia

² Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang 50131, Indonesia

Correspondence should be addressed to Ferda Ernawan; ferda1902@gmail.com

Received 21 February 2014; Revised 20 August 2014; Accepted 6 October 2014

Academic Editor: Ahmad T. Azar

Copyright © 2015 N. A. Abu and F. Ernawan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A psychovisual experiment prescribes the quantization values in image compression. The quantization process is used as a threshold of the human visual system tolerance to reduce the amount of encoded transform coefficients. It is very challenging to generate an optimal quantization value based on the contribution of the transform coefficient at each frequency order. The psychovisual threshold represents the sensitivity of the human visual perception at each frequency order to the image reconstruction. An ideal contribution of the transform at each frequency order will be the primitive of the psychovisual threshold in image compression. This research study proposes a psychovisual threshold on the large discrete cosine transform (DCT) image block which will be used to automatically generate the much needed quantization tables. The proposed psychovisual threshold will be used to prescribe the quantization values at each frequency order. The psychovisual threshold on the large image block provides significant improvement in the quality of output images. The experimental results on large quantization tables from psychovisual threshold produce largely free artifacts in the visual output image. Besides, the experimental results show that the concept of psychovisual threshold produces better quality image at the higher compression rate than JPEG image compression.

1. Introduction

Most digital cameras implement a popular block transform in image coding [1]. The sequential block-based coding is a popular technique since it is compact and easy to implement. In standard JPEG image compression, an image is compressed by one block of 8×8 pixels at a time. The block-based DCT coding has prevailed at reducing interpixel statistical redundancy [2]. However, in order to achieve high compression ratio, 8×8 image block size with default JPEG quantization tables produces discontinuity of intensities among adjacent image blocks. These discontinuities of the intensity image between two adjacent image blocks cause a visual artifact due to interblock correlations [3] in image reconstruction. Block transform coding always results in blocking artifact at low bit rate. Blocking artifact is one of the most annoying problems [4].

The blocking effect becomes visible within smooth regions where adjacent block is highly correlated with an

input image. Since the 8×8 blocks of image pixels are coded separately, the correlation among spatially adjacent blocks provides boundary blocks when the image is reconstructed [5]. The artifact images in the compressed output are introduced by two factors. First, the transform coefficients coming out of the quantization process are rounded and then inadequately dequantized. Second, the blocking artifacts appear by the pixel intensity value discontinuities which occur along block boundaries [6]. These blocking artifacts are often visually observable.

This research pays a serious attention to the role of block size in transform image coding. Referring to JPEG image compression, the block of 8×8 image pixels based line coding has provided a low computational complexity. Previously, a compression scheme on 16×16 block has been investigated by Pennebaker and Mitchell [7] for image compression. This scheme does not provide an improvement on the image compression due to lack of progress on the central processing

unit in terms of its computing power at the time. The larger 16×16 block requires an extra image buffering and a higher precision in internal calculations. Nowadays, the technology of the central processing unit grows rapidly in terms of its computing power. Therefore, two-dimensional image transform on larger blocks is now practically efficient to operate on image compression.

In the previous research, the psychovisual threshold has been investigated on 8×8 image block size [8–15]. This paper proposes psychovisual threshold on the large image block of 256×256 DCT in order to reduce significant blocking effect within the boundary of small image block. This paper also discusses the process and apparatus to generate 256×256 quantization tables via a psychovisual threshold.

The organization of this paper is as follows. The next section provides a brief overview on a psychoacoustic model. Section 3 discusses a brief description of the 256×256 discrete cosine transform. Section 4 explains the development of psychovisual threshold on the large discrete cosine transform in image compression. Section 5 discusses a quality measurement on compressed output images. Section 6 shows the experimental results of 256×256 quantization tables from psychovisual threshold in image compression. Lastly, Section 7 concludes this paper.

2. The Principle of Psychoacoustic Model

Psychoacoustics is the study on how humans perceive sound or human hearing. Psychoacoustic studies show that the sound can only be heard at certain or higher sound pressure levels (SPL) across frequency order [16]. The psychoacoustics indicates that human hearing sensation has a selective sensitivity to different frequencies [17]. In the noise-free environment, the human ear audibility requires different loudness across various frequency orders. The sound loudness that the human audibility can hear is called the absolute hearing threshold [18] as depicted in Figure 1.

The principle of the psychoacoustic model by incrementing the sound pressure level one bark at a time has been used to detect audibility of human hearing threshold. The same principle of the psychoacoustic technique has been used to measure the psychovisual threshold in image compression by incrementing image frequency signal one-unit scale at a time. A block of image signals as represented by a transform coefficient at a given frequency is incremented one at a time on each frequency order. A psychovisual threshold has been conducted for image compression by calculating the just noticeable difference (JND) of the compressed image from the original image. This research will investigate the contribution of the transformed coefficient on each frequency order to the compressed output image. The average reconstruction error from the contribution of the DCT coefficients in image reconstruction will be the primitive of psychovisual threshold in image compression. This quantitative experiment has been conducted on 256×256 image blocks.

3. Discrete Cosine Transform

The two-dimensional discrete cosine transform [19] has been widely used in image processing applications. The DCT is

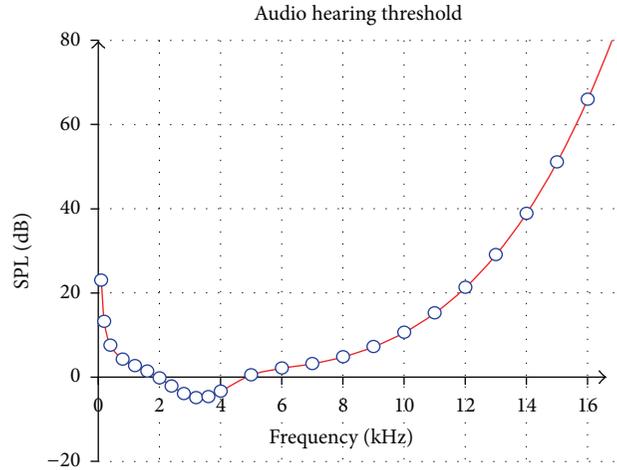


FIGURE 1: The absolute threshold of hearing under quiet condition.

used to transform the pixel values to the spatial frequencies. These spatial frequencies represent the detailed level of image information. The standard JPEG compression uses 8×8 DCT as shown in Figure 2 in image compression.

This paper proposes a large image block of $N \times N$ DCT set $C_n(x)$ of size $N = 256$ which can be generated iteratively as follows:

$$\begin{aligned}
 C_0(x) &= \frac{1}{\sqrt{N}}, \\
 C_1(x) &= \sqrt{\frac{2}{N}} \cos \frac{(2x+1)1\pi}{2N}, \\
 C_2(x) &= \sqrt{\frac{2}{N}} \cos \frac{(2x+1)2\pi}{2N}, \\
 C_3(x) &= \sqrt{\frac{2}{N}} \cos \frac{(2x+1)3\pi}{2N},
 \end{aligned} \tag{1}$$

for $x = 0, 1, 2, \dots, N-1$. The first four one-dimensional 256×256 DCT above are depicted in Figure 3 for visual purposes.

The kernel for the DCT is derived from the following definition [20]:

$$g = \lambda(u) \cos \frac{(2x+1)u\pi}{2N}, \tag{2}$$

where

$$\lambda(u) = \begin{cases} \frac{1}{\sqrt{N}}, & u = 0 \\ \sqrt{\frac{2}{N}}, & u > 0, \end{cases} \tag{3}$$

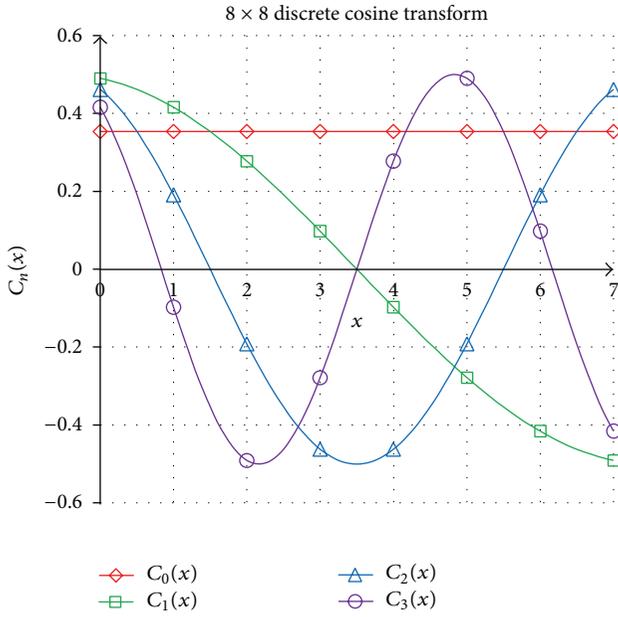


FIGURE 2: The first four 8 × 8 DCT of set $C_n(x)$ for $n = 0, 1, 2, 3$.

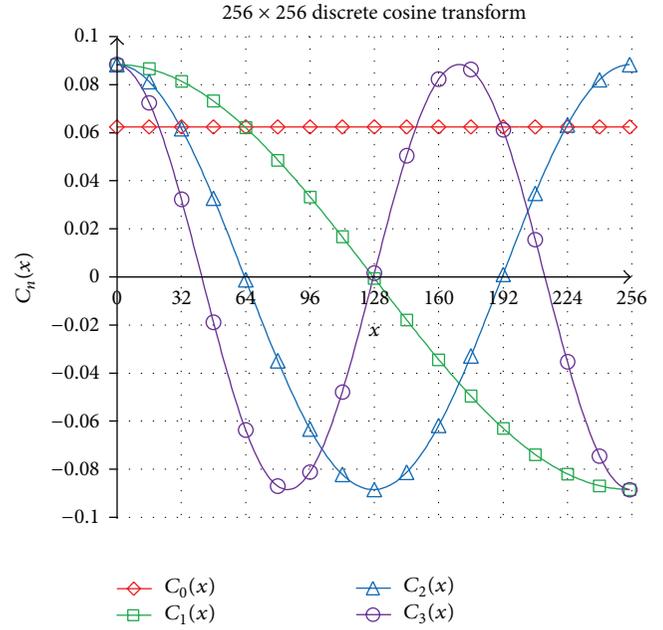


FIGURE 3: The first four 256 × 256 DCT of set $C_n(x)$ for $n = 0, 1, 2, 3$.

for $x = 0, 1, 2, \dots, N - 1$ and $u = 0, 1, 2, \dots, N - 1$. The definition of the two-dimensional DCT of an input image A and output image B is given as follows [19]:

$$B_{pq} = \alpha_p \beta_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad (4)$$

for $p = 0, 1, 2, \dots, M - 1$ and $q = 0, 1, 2, \dots, N - 1$ where

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & p > 0 \end{cases}, \quad \beta_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & q > 0. \end{cases} \quad (5)$$

The inverse of two-dimensional DCT is given as follows:

$$A_{pq} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \alpha_p \beta_q B_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad (6)$$

for $p = 0, 1, 2, \dots, M - 1$ and $q = 0, 1, 2, \dots, N - 1$. The image input is subdivided into $M \times N$ blocks of image pixels. The DCT is used to transform each pixel in the 256×256 image block pixel into the frequency transform domain. The outputs of transforming 256×256 image blocks of frequency signals are 65536 DCT coefficients. The first coefficient in the upper left corner of the array basis function is called the direct current (DC) coefficient and the rest of the coefficients are called the alternating current (AC) coefficients. DC coefficient provides an average value over the 256×256 block domain.

4. Psychovisual Threshold on Large Discrete Cosine Transform

In this quantitative experiment, the DCT coefficients on each frequency order are incremented concurrently one at a time from 1 to 255. The impact of incrementing DCT coefficients one at a time is measured by average absolute reconstruction error (ARE). The contribution of DCT coefficients to the quality image reconstruction and compression rate is analyzed on each frequency order. In order to develop psychovisual threshold on 256×256 DCT, ARE on each frequency order is set as a smooth curve reconstruction error. An ideal average reconstruction error score of an incrementing DCT coefficient on each frequency order on luminance and chrominance for 40 real images is shown in Figure 4.

An ideal finer curve of ARE on a given order from zero to the maximum frequency order 510 for luminance and chrominance channels is presented by the red curve and blue curve, respectively. These finer curves of absolute reconstruction error are set as the psychovisual threshold on 256×256 DCT. The contribution of an ideal error reconstruction for each frequency order is determined by two factors, its contribution to the quality on image reconstruction and the bit rates on image compression. The smooth curve of ARE is interpolated by a polynomial that represents the psychovisual error threshold on 256×256 DCT in image compression. With reference to Figure 4, this paper proposes the new psychovisual thresholds on 256×256 DCT basis function for luminance f_{VL} and chrominance f_{VR} which are simplified as follows:

$$f_{VL}(x) = -0.0000000000001435x^5 + 0.000000000011x^4 + 0.000000046x^3 - 0.000009x^2 + 0.00088x + 0.2352,$$

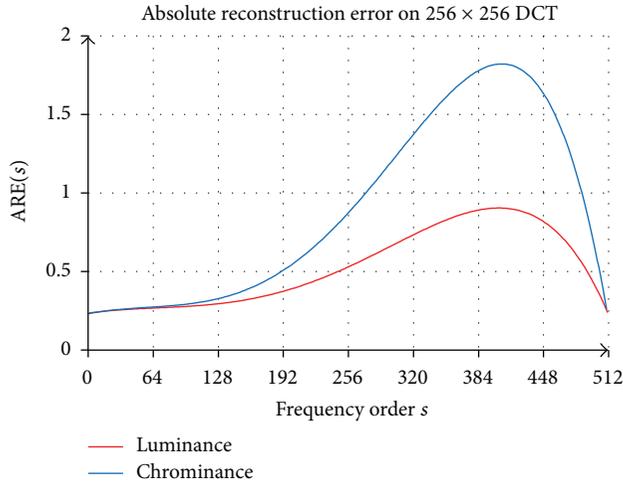


FIGURE 4: Average absolute reconstruction error of incrementing DCT coefficients on 256×256 DCT luminance and chrominance for 40 real images.

$$\begin{aligned}
 f_{VR}(x) = & -0.000000000000457x^5 \\
 & + 0.000000000156x^4 + 0.00000006x^3 \\
 & - 0.0000128x^2 + 0.0012x + 0.2309,
 \end{aligned} \quad (7)$$

for $x = 0, 1, 2, \dots, 510$, where x is a frequency order on 256×256 image block. Further, these thresholds are used to generate smoother 256×256 quantization values for image compression. The 256×256 quantization table has 511 frequency orders from order 0 until order 510. The order 0 resides in the top left most corner of the quantization matrix index of $Q(0,0)$. The first order represents the quantization value of $Q(1,0)$ and $Q(0,1)$. The second order represents $Q(2,0)$, $Q(1,1)$, and $Q(0,2)$. For each frequency order, the same quantization value is assigned to them.

Due to the large size of these new 256×256 quantization values, it is not possible to present the whole quantization matrix within a limited space in this paper. Therefore, the 256×256 quantization values are presented by traversing the quantization table on each frequency order in zigzag pattern as shown in Table 1. Table 1 presents one-dimensional index of quantization table on each frequency order. Each index represents the quantization value at those frequency orders. The new finer quantization tables from psychovisual threshold on 256×256 DCT for luminance and chrominance are shown in Tables 2 and 3, respectively. The visualization of the whole quantization values on each frequency order from the psychovisual threshold for luminance and chrominance channels is depicted in Figure 5.

These new finer 256×256 quantization tables have been generated from the psychovisual threshold functions in (7). Each traversing array in zigzag pattern represents the quantization value on each quantization order from order 0 to order 510. This new smoother 256×256 quantization

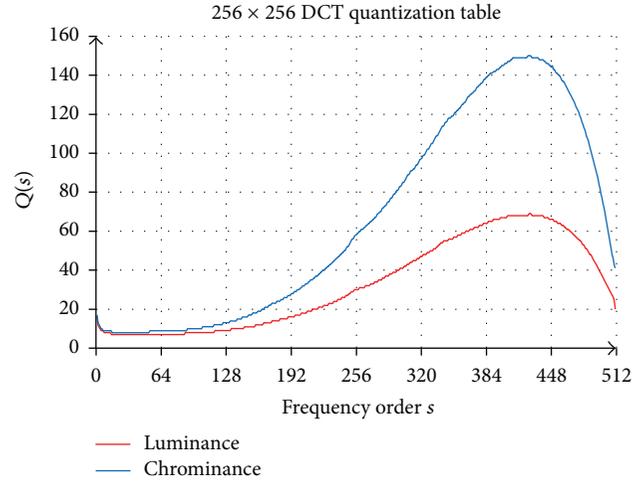


FIGURE 5: The 256×256 DCT quantization table for luminance and chrominance for image compression.

table for luminance is designed to take smaller value than a quantization table for chrominance. Any slight changes on respective frequency order in the luminance channel will generate significantly greater reconstruction error than a change in chrominance channel. The slight changes by the image intensity on the luminance channel will provide visible textures that can be perceived by human visual systems.

The contribution of the frequency signals to the reconstruction error is mainly concentrated in the low frequency order. Referring to Figure 1, the SPL values on frequency from 2 kHz to 5 kHz are significantly lower. These quantization tables follow the same pattern in order to capture the concept of the psychoacoustic absolute hearing threshold. The quantization values from the psychovisual threshold on chrominance channel are designed to be larger than the quantization values on luminance channel. The human visual system is less sensitive to the chrominance channels as they provide significantly irrelevant image information. The smoother quantization tables will be tested and verified in image compression.

5. Quality Measurement

Two statistical evaluations have been conducted in this research project to verify the performances of psychovisual threshold in image compression. In order to gain significantly better performance, the image compression algorithm needs to achieve a trade-off between average bit rates and quality image reconstruction. The conventional quality assessments are employed in this paper. They are average absolute reconstruction error (ARE), means square error (MSE), and peak signal to noise ratio (PSNR). The average reconstruction error can be defined as follows:

$$\text{ARE}(s) = \frac{1}{\text{MNR}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=0}^{R-1} |g(i, j, k) - f(i, j, k)|, \quad (8)$$

TABLE 1: The index of quantization value on each frequency order.

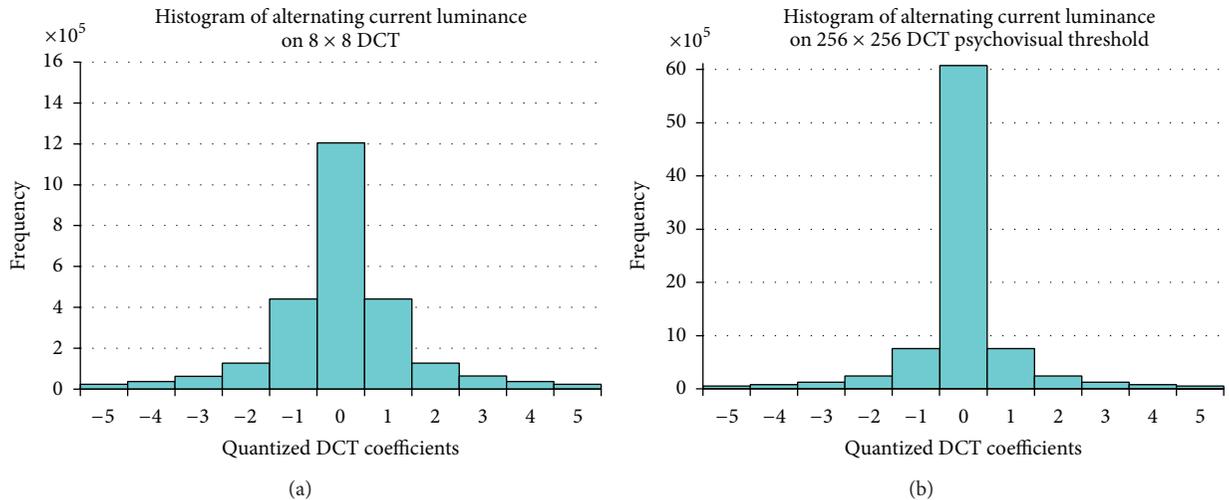
0	1	5	6	14	15	27	28	44	45	65	66	90	91	119	120	152	153	189	190	230	231	275
2	4	7	13	16	26	29	43	46	64	67	89	92	118	121	151	154	188	191	229	232	274	276
3	8	12	17	25	30	42	47	63	68	88	93	117	122	150	155	187	192	228	233	273	277	318
9	11	18	24	31	41	48	62	69	87	94	116	123	149	156	186	193	227	234	272	278	317	319
10	19	23	32	40	49	61	70	86	95	115	124	148	157	185	194	226	235	271	279	316	320	357
20	22	33	39	50	60	71	85	96	114	125	147	158	184	195	225	236	270	280	315	321	356	358
21	34	38	51	59	72	84	97	113	126	146	159	183	196	224	237	269	281	314	322	355	359	392
35	37	52	58	73	83	98	112	127	145	160	182	197	223	238	268	282	313	323	354	360	391	393
36	53	57	74	82	99	111	128	144	161	181	198	222	239	267	283	312	324	353	361	390	394	423
54	56	75	81	100	110	129	143	162	180	199	221	240	266	284	311	325	352	362	389	395	422	424
55	76	80	101	109	130	142	163	179	200	220	241	265	285	310	326	351	363	388	396	421	425	450
77	79	102	108	131	141	164	178	201	219	242	264	286	309	327	350	364	387	397	420	426	449	451
78	103	107	132	140	165	177	202	218	243	263	287	308	328	349	365	386	398	419	427	448	452	473
104	106	133	139	166	176	203	217	244	262	288	307	329	348	366	385	399	418	428	447	453	472	474
105	134	138	167	175	204	216	245	261	289	306	330	347	367	384	400	417	429	446	454	471	475	492
135	137	168	174	205	215	246	260	290	305	331	346	368	383	401	416	430	445	455	470	476	491	493
136	169	173	206	214	247	259	291	304	332	345	369	382	402	415	431	444	456	469	477	490	494	507
170	172	207	213	248	258	292	303	333	344	370	381	403	414	432	443	457	468	478	489	495	506	508
171	208	212	249	257	293	302	334	343	371	380	404	413	433	442	458	467	479	488	496	505	509	
209	211	250	256	294	301	335	342	372	379	405	412	434	441	459	466	480	487	497	504	510		
210	251	255	295	300	336	341	373	378	406	411	435	440	460	465	481	486	498	503				
252	254	296	299	337	340	374	377	407	410	436	439	461	464	482	485	499	502					
253	297	298	338	339	375	376	408	409	437	438	462	463	483	484	500	501						

TABLE 2: The quantization value on each frequency order for luminance.

16	12	9	9	7	7	7	7	7	7	7	7	8	8	9	9	11	11	16	16	23	23	34	
11	9	8	8	7	7	7	7	7	7	7	7	8	8	9	9	11	11	16	16	23	23	34	34
10	8	8	7	7	7	7	7	7	7	8	8	9	9	11	11	15	16	23	24	34	35	47	
8	8	7	7	7	7	7	7	7	8	8	9	9	11	12	15	16	22	24	33	35	46	47	
8	7	7	7	7	7	7	7	7	8	8	9	11	12	15	16	22	24	33	35	46	47	58	
7	7	7	7	7	7	7	7	8	8	9	11	12	15	17	22	24	33	35	46	48	58	58	
7	7	7	7	7	7	7	8	8	9	11	12	15	17	22	25	33	36	45	48	58	59	66	
7	7	7	7	7	8	8	9	11	12	15	17	22	25	32	36	45	48	57	59	66	66	66	
7	7	7	7	8	8	9	10	12	15	17	21	25	32	36	45	48	57	59	65	66	68	68	
7	7	8	8	9	10	12	14	17	21	26	32	37	44	49	57	60	65	66	68	69	66	65	
7	8	8	10	10	13	14	18	21	26	31	37	44	50	56	60	65	67	68	69	66	65	56	
8	8	10	10	13	14	18	20	26	31	38	43	50	56	60	65	67	68	68	66	64	57	56	
8	10	10	13	14	18	20	27	31	38	43	50	56	61	64	67	68	68	66	64	57	55	42	
10	10	13	14	18	20	27	31	38	43	51	56	61	64	67	68	68	66	64	58	55	43	41	
10	13	14	18	20	27	31	39	42	51	55	61	64	67	68	68	67	64	58	54	44	40	27	
13	14	19	20	28	31	39	42	51	55	61	64	67	68	68	67	63	59	53	45	39	28	26	
13	19	20	28	30	39	42	52	55	62	64	68	68	68	67	63	59	53	46	38	29	24		
19	19	29	30	39	41	52	55	62	63	68	68	68	67	63	60	52	47	37	30	20			
19	29	30	40	41	53	55	62	63	68	68	68	67	62	60	51	48	36	31					
29	30	40	41	53	54	62	63	68	68	68	68	62	61	51	48	35	32						
30	40	41	54	54	62	63	68	68	68	68	61	61	50	49	34	33							

TABLE 3: The quantization value on each frequency order for chrominance.

17	13	10	9	9	8	8	8	8	8	9	9	10	10	12	12	17	18	27	27	43	44	68
12	10	9	9	8	8	8	8	8	9	9	10	10	12	12	17	18	27	28	43	44	67	69
11	9	9	8	8	8	8	8	9	9	9	10	12	12	17	18	26	28	42	45	67	69	97
9	9	8	8	8	8	8	9	9	9	10	12	12	17	18	26	28	42	45	66	70	96	97
9	8	8	8	8	8	9	9	9	10	12	13	16	18	26	29	41	46	66	70	95	98	123
8	8	8	8	8	9	9	9	10	11	13	16	19	26	29	41	46	65	71	94	99	123	124
8	8	8	8	9	9	9	10	11	13	16	19	25	29	40	47	65	72	94	99	122	125	142
8	8	9	9	9	9	10	11	13	16	19	25	30	40	47	64	72	93	100	122	125	142	143
8	9	9	9	9	10	11	13	16	19	25	30	40	48	64	73	92	101	121	126	142	143	149
9	9	9	9	10	11	13	16	20	24	30	39	48	63	73	92	102	121	126	141	143	149	150
9	9	9	10	11	13	15	20	24	31	39	49	63	74	91	102	120	127	141	144	149	150	143
9	9	10	11	14	15	20	24	31	38	49	62	75	91	103	119	128	141	144	149	150	143	142
9	10	11	14	15	21	24	32	38	50	62	75	90	104	119	129	140	145	149	150	145	141	122
11	11	14	15	21	23	32	37	50	61	76	89	105	118	129	140	145	149	149	144	141	123	121
11	14	15	21	23	32	37	51	61	77	89	105	118	130	139	145	149	149	145	140	124	119	89
14	14	21	23	33	37	52	61	77	88	106	118	130	139	146	149	149	145	140	126	118	91	87
14	22	23	33	36	52	60	78	87	107	117	131	138	146	149	149	146	139	127	116	93	84	48
22	23	33	36	53	60	79	86	108	116	131	138	146	149	149	146	138	128	115	95	82	51	46
22	34	35	54	59	79	85	109	116	132	137	147	149	149	146	137	129	113	97	79	54	43	
34	35	55	59	80	85	109	115	132	136	147	149	149	147	137	130	112	99	77	57	41		
35	56	58	81	84	110	114	133	136	147	149	148	147	136	131	110	101	74	60				
56	58	81	83	111	114	134	135	148	149	148	148	135	132	108	103	72	63					
57	82	83	112	113	134	135	148	149	148	148	134	133	107	105	69	66						

FIGURE 6: Frequency distribution of the alternating current (AC) coefficients using 8×8 DCT quantization (a) and 256×256 DCT quantization from psychovisual threshold (b) on luminance for 40 real images.

where the original image size $M \times N \times R$ refers to the three RGB colors. The MSE calculates the average of the square of the error defined as follows [21]:

$$\text{MSE} = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=0}^{R-1} \|g(i, j, k) - f(i, j, k)\|^2. \quad (9)$$

The standard PSNR is used and calculated to obtain the measure of the quality of the image reconstruction. A higher

PSNR means that the image reconstruction is more similar to the original image [22]. The PSNR is defined as follows:

$$\text{PSNR} = 20 \log_{10} \left(\frac{\text{Max}_i}{\sqrt{\text{MSE}}} \right) = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right), \quad (10)$$

where Max_i is the maximum possible pixel value of the image. Structural similarity index (SSIM), another measurement of image quality, is a method to measure quality by capturing

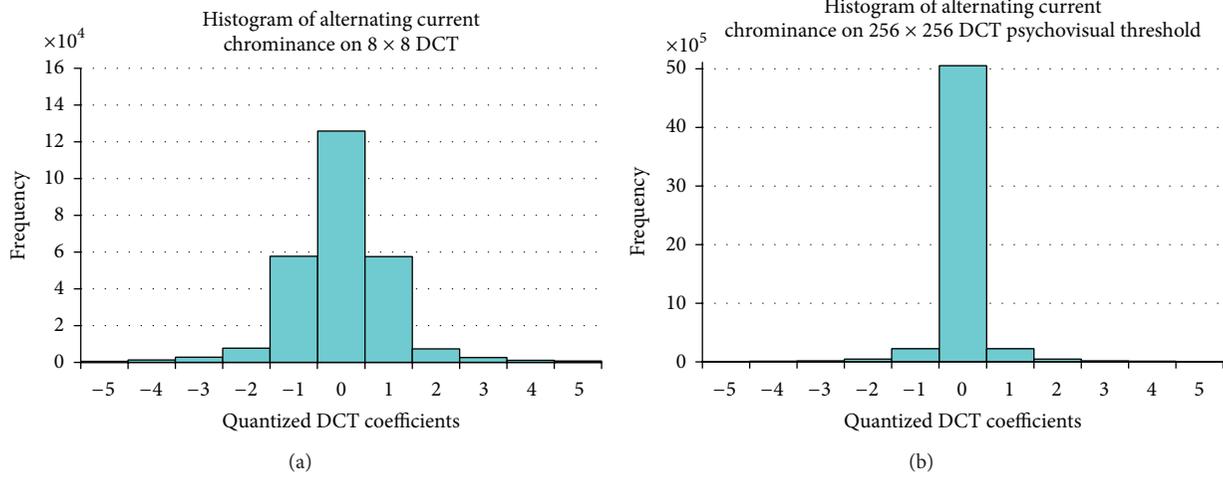


FIGURE 7: Frequency distribution of the alternating current (AC) coefficients using 8×8 default JPEG quantization tables (a) and 256×256 DCT quantization from psychovisual threshold (b) on chrominance for 40 real images.

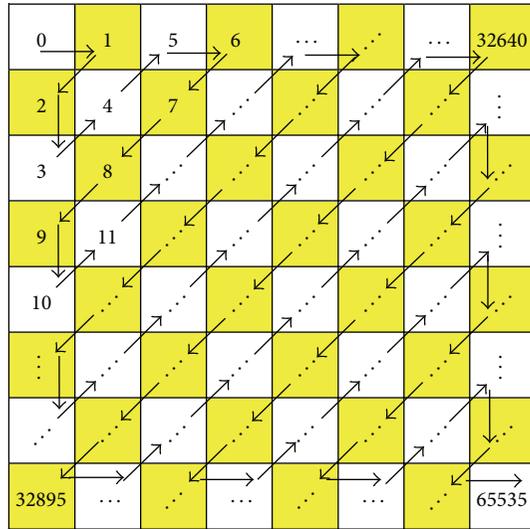


FIGURE 8: Zigzag order of 256×256 image block.

the similarity between original image and compressed image [23]. The SSIM is defined as follows:

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma, \quad (11)$$

where $\alpha > 0, \beta > 0, \gamma > 0$ are parameters to adjust the relative importance of the three components. The detailed description is given in [23].

6. Experimental Results

This quantitative experiment has been conducted to investigate the performance of a psychovisual threshold on the large image block. The new finer 256×256 quantization tables have been generated from the psychovisual threshold on 256×256

DCT. They are tested on 40 real and 40 graphical high fidelity images. An input image consists of 512×512 colour pixels. The RGB image components are converted to YCbCr color space. In this experiment, each image is divided into 256×256 image block pixels; thus each block is transformed into 256×256 DCT. The DCT coefficients are quantized by new finer 256×256 quantization tables from psychovisual coefficients after the quantization process is summarised by histograms in Figures 6 and 7. The histogram of the frequency distribution is obtained after quantization processes. Figures 6 and 7 show a histogram of the frequency distribution after quantization process of 8×8 default JPEG quantization tables and 256×256 quantization tables from psychovisual threshold for luminance and chrominance, respectively.

The compression rate focuses mainly on the contribution of the AC coefficients to image compression performance. The frequency coefficients after quantization process consist of many zeros. The frequency distribution of the AC coefficients given by its histogram may predict the compression rate. The higher zeros value on the histogram of frequency distribution means that the image compression output provides lower bit rate to present an image.

According to Figures 6 and 7, the distribution of the frequency coefficients after the quantization process by smoother 256×256 quantization tables from psychovisual thresholds produces significantly more zeros for both luminance and chrominance channels, respectively. These finer quantization tables produce a smaller standard deviation on AC coefficients from the large 256×256 image block than the small 8×8 image block. Hence, it is possible to code large transformed block using smaller number of bits for the same image.

The 256×256 transform coding consists of 65535 AC coefficients for each regular block. Most AC coefficients are naturally small coming out of quantization process. The

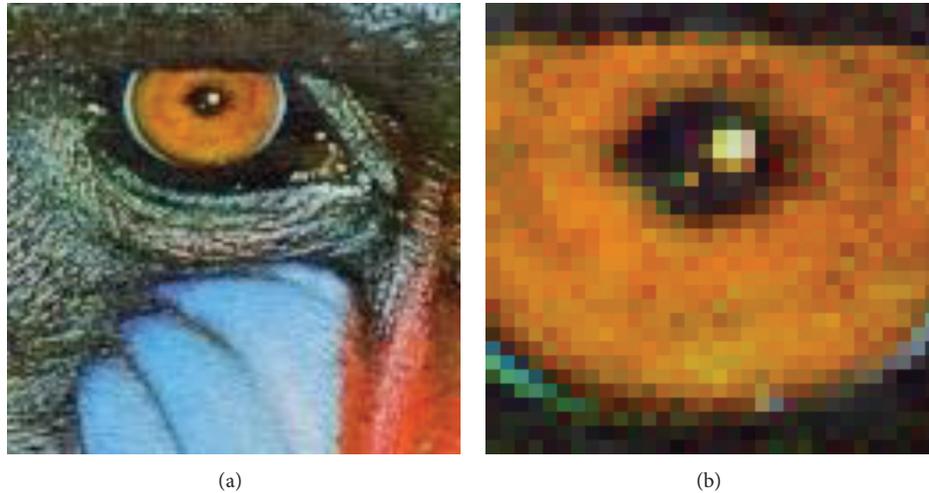


FIGURE 9: Original baboon image (a) zoomed in to 400% (b).

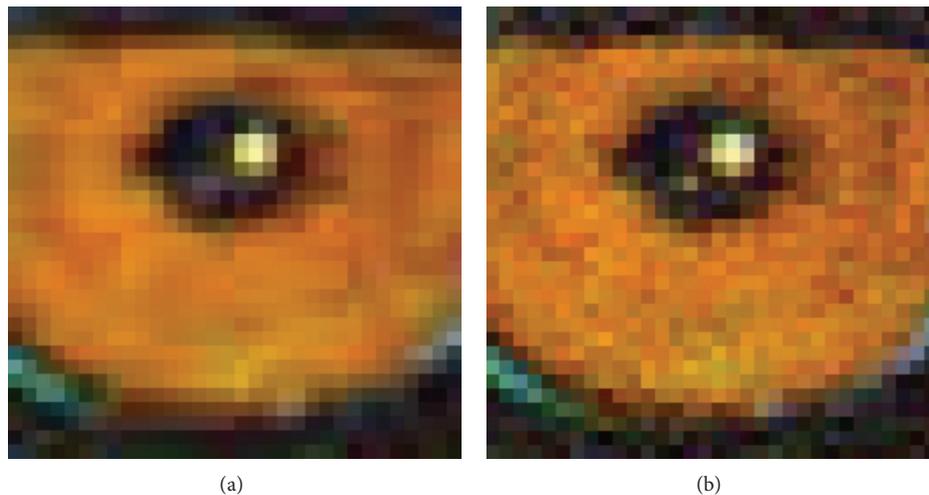


FIGURE 10: The comparison of visual outputs between 8×8 JPEG quantization table (a) and 256×256 quantization tables from psychovisual threshold (b) zoomed in to 400%.

psychovisual threshold on large DCT determines an optimal contribution of the AC coefficients of large transform coding. The new 256×256 quantization tables from psychovisual threshold are able to reduce down the irrelevant AC coefficients. The distribution of the transform coefficients gives an indication on how much transform coefficients will be encoded by a lossless Huffman coding.

An average Huffman code is calculated from the quantized transform coefficients. After the transformation and quantization of 256×256 image block are over, the direct current (DC) coefficient is separated from the AC coefficients. The AC coefficients are listed as a traversing array in zigzag pattern as shown in Figure 8.

Next, run length encoding is used to reduce the size of a repeating coefficient in the sequence of the AC coefficients. The coefficient values can be represented compactly by simply indicating the coefficient value and the length of its

run wherever it appears. The output of run length coding represents the symbols and the length of occurrence of the symbols. The symbols and variable length of occurrence are used in Huffman coding to retrieve code words and their length of code words. Using these probability values, a set of Huffman code of the symbols can be generated by Huffman tree. Next, the average bit length is calculated to find the average bit length of the AC coefficients.

There are only four DC coefficients under regular 256×256 DCT. The maximum code length of the DC coefficient is 16 bits. The DC coefficients are reduced down by 4 bits. The average bit length of the DC coefficients produces 12 bits after the quantization process in image compression. The average bit length of image compression based on default 8×8 JPEG quantization tables and the finer 256×256 quantization tables from psychovisual threshold is shown in Table 4. The experimental results show the new large quantization tables

TABLE 4: Average bit length of Huffman code of image compression using 8×8 JPEG compression and 256×256 JPEG compression using psychovisual threshold for 40 real images and 40 graphical images.

Average bit length of Huffman code	8 × 8 JPEG compression		256 × 256 JPEG compression	
	40 real images	40 graphic images	40 real images	40 graphic images
DC luminance	5.7468	5.6722	12	12
DC chrominance Cr	2.7941	3.8663	12	12
DC chrominance Cb	3.1548	4.0730	12	12
AC luminance	2.8680	2.9653	2.3031	2.6582
AC chrominance Cr	2.0951	2.5059	1.2931	2.1450
AC chrominance Cb	2.1845	2.5158	1.3656	2.1855

TABLE 5: The average image reconstruction error using 8×8 JPEG compression and 256×256 JPEG compression using psychovisual threshold for 40 real images and 40 graphical images.

Image measurement	8 × 8 JPEG compression		256 × 256 JPEG compression	
	40 real images	40 graphic images	40 real images	40 graphic images
ARE	5.535	5.648	5.074	5.050
MSE	70.964	92.711	51.841	59.625
PSNR	31.190	31.636	32.363	33.421
SSIM	0.956	0.957	0.961	0.960

from psychovisual threshold produce a lower average bit length of Huffman code than the default JPEG quantization tables.

The DCT coefficients from a large image block have been greatly discounted by quantization tables in image compression. An optimal amount of DCT coefficients is investigated by reconstruction error and average bit length of Huffman code. The effect of incrementing DCT coefficient has been explored from this experiment. The average reconstruction error from incrementing DCT coefficients is mainly concentrated in the low frequency order of the image signals.

The new 256×256 quantization table from the psychovisual threshold produces a lower average bit length of Huffman code in image compression as shown in Table 4. At the same time, the compressed output images produce a better quality image reconstruction than the regular 8×8 default JPEG quantization tables as listed in Table 5. The new design on quantization tables from psychovisual threshold performs better by producing higher quality in image reconstruction at lower average bit length of Huffman code.

The average bit size of image compression as presented in Table 6 shows that the finer 256×256 quantization tables from psychovisual threshold use fewer bits. Therefore, the compression ratio of the difference between a compressed image from the new large quantization table and the original image produces a higher compression ratio than standard JPEG image compression as shown in Table 7. In order to observe the visual quality of the output image, a sample of original baboon right eye is zoomed in to 400% as depicted on the right of Figure 9.

The image compression output of 256×256 quantization table from psychovisual threshold is shown on the right of Figure 10. A visual inspection on the output image using 256×256 quantization tables from psychovisual threshold

produces the richer texture on the baboon image. The psychovisual threshold on 256×256 image block gives an optimal balance between the fidelity on image reconstruction and compression rate. The experimental results show that the psychovisual threshold on large DCT provides minimum image reconstruction error at lower bit rates. The JPEG image compression output as depicted on the left of Figure 10 contains artifact image or blocking effect under regular 8×8 block transform coding. At the same time, the smoother 256×256 quantization tables from psychovisual threshold manage to overcome the blocking effects along the boundary blocks. These finer 256×256 quantization tables from psychovisual threshold provide high quality image with fewer artifact images. The psychovisual threshold is practically the best measure of an optimal amount of transform coefficients to the image coding.

7. Conclusion

This research project has been designed to support large block size in a practical image compression operation in the near future. A step-by-step procedure has been developed to produce psychovisual threshold on 256×256 block transform. The use of the psychovisual threshold on a large image block is able to overcome the blocking effect or artifact image which often occurs in standard JPEG compression. The psychovisual threshold on discrete transform has been used to determine an optimal amount of frequency transform coefficients to code by generating the much needed quantization tables. Naturally, a frequency transform on large image block is capable of reducing redundancy and better exploiting pixel correlation within the image block. The new set of quantization tables from psychovisual threshold produces

TABLE 6: The average bit size of 8×8 JPEG compression and 256×256 JPEG compression using psychovisual threshold.

8 × 8 JPEG compression		256 × 256 JPEG compression	
40 real images	40 graphic images	40 real images	40 graphic images
230.997 Kb	258.394 Kb	158.792 Kb	223.652 Kb

TABLE 7: The average compression ratio score of 8×8 JPEG compression and 256×256 JPEG compression using psychovisual threshold.

8 × 8 JPEG compression		256 × 256 JPEG compression	
40 real images	40 graphic images	40 real images	40 graphic images
3.3247	2.9722	4.8365	3.4339

better performance than JPEG image compression. These quantization tables from psychovisual threshold practically provide higher quality images at lower bit rate for image compression application.

Conflict of Interests

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

Acknowledgment

The authors would like to express a very special thanks to Ministry of Education, Malaysia, for providing financial support to this research project by Fundamental Research Grant Scheme (FRGS/2012/FTMK/SG05/03/1/F00141).

References

- [1] Y. Wang, X. Mao, and Y. He, "A dual quad-tree based variable block-size coding method," *Journal of Visual Communication and Image Representation*, vol. 21, no. 8, pp. 889–899, 2010.
- [2] L. Ma, D. Zhao, and W. Gao, "Learning-based image restoration for compressed images," *Signal Processing: Image Communication*, vol. 27, no. 1, pp. 54–65, 2012.
- [3] L. Wang, L. Jiao, J. Wu, G. Shi, and Y. Gong, "Lossy-to-lossless image compression based on multiplier-less reversible integer time domain lapped transform," *Signal Processing: Image Communication*, vol. 25, no. 8, pp. 622–632, 2010.
- [4] S. Lee and S. J. Park, "A new image quality assessment method to detect and measure strength of blocking artifacts," *Signal Processing: Image Communication*, vol. 27, no. 1, pp. 31–38, 2012.
- [5] J. Singh, S. Singh, D. Singh, and M. Uddin, "A signal adaptive filter for blocking effect reduction of JPEG compressed images," *International Journal of Electronics and Communications*, vol. 65, no. 10, pp. 827–839, 2011.
- [6] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. J. R. Liu, "Anti-forensics of JPEG compression," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10)*, pp. 1694–1697, Dallas, Tex, USA, March 2010.
- [7] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, Springer, New York, NY, USA, 1993.
- [8] F. Ernawan, N. A. Abu, and N. Suryana, "TMT quantization table generation based on psychovisual threshold for image compression," in *Proceedings of the International Conference of Information and Communication Technology (ICoICT '13)*, pp. 202–207, Bandung, Indonesia, March 2013.
- [9] F. Ernawan, N. A. Abu, and N. Suryana, "Adaptive tchebichef moment transform image compression using psychovisual model," *Journal of Computer Science*, vol. 9, no. 6, pp. 716–725, 2013.
- [10] F. Ernawan, N. A. Abu, and N. Suryana, "An adaptive JPEG image compression using psychovisual model," *Advanced Science Letters*, vol. 20, no. 1, pp. 26–31, 2014.
- [11] F. Ernawan, N. A. Abu, and N. Suryana, "An optimal tchebichef moment quantization using psychovisual threshold for image compression," *Advanced Science Letters*, vol. 20, no. 1, pp. 70–74, 2014.
- [12] F. Ernawan, N. A. Abu, and N. Suryana, "Integrating a smooth psychovisual threshold into an adaptive JPEG image compression," *Journal of Computers*, vol. 9, no. 3, pp. 644–653, 2014.
- [13] F. Ernawan, N. A. Abu, and N. Suryana, "A psychovisual threshold for generating quantization process in tchebichef moment image compression," *Journal of Computers*, vol. 9, no. 3, pp. 702–710, 2014.
- [14] N. A. Abu, F. Ernawan, and S. Sahib, "Psychovisual model on discrete orthonormal transform," in *Proceedings of the International Conference on Mathematical Sciences and Statistics (ICMSS '13)*, pp. 309–314, Kuala Lumpur, Malaysia, February 2013.
- [15] N. A. Abu, F. Ernawan, and N. Suryana, "A generic psychovisual error threshold for the quantization table generation on JPEG image compression," in *Proceedings of the IEEE 9th International Colloquium on Signal Processing and its Applications (CSPA '13)*, pp. 39–43, Kuala Lumpur, Malaysia, March 2013.
- [16] L. Hong-Fu, Z. Cong, and L. Rui-Fan, "Optimization of masking expansion algorithm in psychoacoustic models," in *Proceedings of the International Symposium on Intelligence Information Processing and Trusted Computing (IPTC '11)*, pp. 161–164, Hubei, China, October 2011.
- [17] H. Bao and I. M. S. Panahi, "Psychoacoustic active noise control based on delayless subband adaptive filtering," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10)*, pp. 341–344, Dallas, Tex, USA, March 2010.
- [18] H. Chen and T. L. Yu, "Comparison of psycho acoustic principles and genetic algorithms in audio compression," in *Proceedings of the 18th International Conference on Systems Engineering (ICSEng '05)*, pp. 270–275, Las Vegas, Nev, USA, August 2005.
- [19] N. Ahmed, T. Natrajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, 1993.
- [20] O. Hunt and R. Mukundan, "A comparison of discrete orthogonal basis functions for image compression," in *Proceedings of*

the Conference on Image and Vision Computing New Zealand (IVCNZ '04), pp. 53–58, November 2004.

- [21] A. Horé and D. Ziou, “Image quality metrics: PSNR vs. SSIM,” in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 2366–2369, Istanbul, Turkey, August 2010.
- [22] Y.-K. Chen, F.-C. Cheng, and P. Tsai, “A gray-level clustering reduction algorithm with the least PSNR,” *Expert Systems with Applications*, vol. 38, no. 8, pp. 10183–10187, 2011.
- [23] C. Yim and A. C. Bovik, “Quality assessment of deblocked images,” *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 88–98, 2011.

Research Article

Chaos Time Series Prediction Based on Membrane Optimization Algorithms

Meng Li,¹ Liangzhong Yi,² Zheng Pei,¹ Zhisheng Gao,¹ and Hong Peng¹

¹School of Radio Management Technology Research Center, Xihua University, Chengdu 610039, China

²School of Computer Science and Technology, Sichuan Police College, Luzhou 646000, China

Correspondence should be addressed to Zheng Pei; pqyz@263.net

Received 26 June 2014; Revised 27 August 2014; Accepted 27 August 2014

Academic Editor: Shifei Ding

Copyright © 2015 Meng Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper puts forward a prediction model based on membrane computing optimization algorithm for chaos time series; the model optimizes simultaneously the parameters of phase space reconstruction (τ, m) and least squares support vector machine (LS-SVM) (γ, σ) by using membrane computing optimization algorithm. It is an important basis for spectrum management to predict accurately the change trend of parameters in the electromagnetic environment, which can help decision makers to adopt an optimal action. Then, the model presented in this paper is used to forecast band occupancy rate of frequency modulation (FM) broadcasting band and interphone band. To show the applicability and superiority of the proposed model, this paper will compare the forecast model presented in it with conventional similar models. The experimental results show that whether single-step prediction or multistep prediction, the proposed model performs best based on three error measures, namely, normalized mean square error (NMSE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

1. Introduction

Chaotic time series is a kind of nonlinear dynamic phenomenon between certainty and randomness, in which Lyapunov exponent is adopted to decide whether a time series is chaos or not; that is, the time series is chaotic if its Lyapunov exponent is greater than zero [1]. Because it can be widely applied in real life, such as in the network traffic, earthquake prediction, and weather forecasting [2–5], chaotic time series prediction has become a hot spot, and many interesting results have been provided by a lot of researchers in recent years [6, 7].

Initially, the traditional statistical fitting methods, such as autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA) models, have been used in chaotic time series prediction. However, due to the inherent linearity assumptions, the above conventional mathematical tools are not well suited for dealing with ill-defined and uncertain systems. With the recent development in chaos theory, numerous nonlinear systems have been identified to be chaotic despite their random behaviors, in which the local

model is an important method for chaotic time series; the method projected chaotic time series into a multidimensional phase space, which is then divided into several subspaces where the mapping function is approximated by means of local approximation [8–10]. Chaotic time series prediction based on nonlinear systems shows in general superior performance over the traditional statistical fitting methods. As another alternative in dealing with nonlinear systems, support vector machine (SVM) was proposed in [11, 12] based on the principles of the statistical VC (Vapnik Chervonenkis) dimensional theory and structural risk minimization. SVM can better solve problems such as nonlinear, dimension disaster, and good performance for the small sample. It will be widely used in face recognition, speech recognition [13–15], and so forth. Because of its universal approximation capabilities, recently, least squares support vector machine (LS-SVM) [16] is applied to predict chaotic time series [17, 18]. In the model, firstly, the phase space reconstruction technique of chaotic theory is used to reconstruct the nonlinear data; then the least squares support vector machine regression is applied in multidimensional phase space.

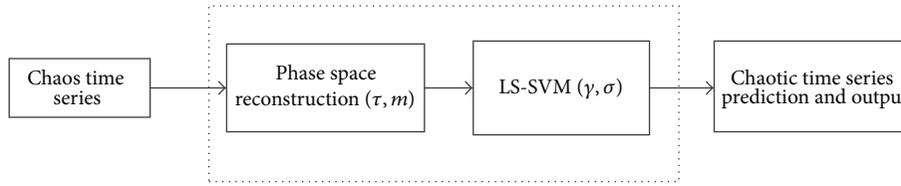


FIGURE 1: The flow chart of chaotic time series prediction.

Formally, phase space reconstruction method is succeeded by delay time and embedding dimension; that is, for a given time series x_1, \dots, x_{n-1}, x_n (n is the number of the data), by using delay time and embedding dimension, the phase points after reconstruction of the time series are $X_i = [x_{i-(m-1)\tau}, \dots, x_{i-\tau}, x_i]$ ($i = 1, \dots, M-1, M$), where τ is delay time, m is embedding dimension, and M is the number of phase space points [19]. Accordingly, the prediction value of next time $t+1$ based on LS-SVM can be expressed as

$$x_{t+1} = f(X_t), \quad (1)$$

where $f(\cdot)$ is regression estimates function.

In applications, there are two key problems in the prediction model based on LS-SVM. One is the choice of delay time (τ) and embedding dimension (m) in the process of phase space reconstruction. Another is the selection of kernel function and its relevant parameters [20]. The phase space reconstruction is used to express out the trace of the evolution of chaotic time series without singular; namely, chaotic time series is projected into a multidimensional phase space. Kernel function is associated with learning and modeling for the data set of phase space reconstruction to forecast accurately the future value. A large number of studies have shown that the selection of delay time (τ) and embedding dimension (m) in phase space reconstruction has a direct impact on prediction results of chaotic time series [21]. If τ is too small in the delay neighbor element of the phase space, there will be information redundancy. If it is too big, τ leads to loss of information; the track of signals will occur folding phenomenon. Similarly, if m is too small, it is not enough to show the detailed structure of chaotic systems. If m is too big, the calculation will become complicated and cause the impact of noise.

LS-SVM learning performance is largely dependent on the choice of kernel function. A large number of studies have shown that, with the lack of a priori knowledge of specific issues, the overall performance of the radial basis kernel function model is better than other kernel function models and hence this paper selects the radial basis kernel function as the kernel function of LS-SVM. So in the model, there are two parameters (cost factor (γ) and kernel parameter (σ)) that need to be identified; cost factor γ is generally used to control the model complexity and compromise of approximation error, which is commonly in $[1, 1000]$. Kernel parameter σ reflects the structure of high-dimensional feature space and affects the generalization ability of the system; when the value of σ is too small, it will occur over-learning phenomenon and poor generalization, while the value of σ is too large, it will emerge less learning phenomenon; the range

of σ is in $[0.1, 10000]$ [22]. Currently, there are mainly two ideas for optimization of the parameters of the phase space reconstruction (τ, m) and LS-SVM (γ, σ). One is that the parameters were optimized separately as shown in Figure 1, in which, firstly, optimal delay time (τ) and embedding dimension (m) in the phase space are selected independently [19, 23–28] or at the same time [27, 29, 30]; then parameters γ and σ of the LS-SVM are selected by gradient descent method [31], genetic algorithm (GA) [32] or particle swarm optimization (PSO) [33], and so forth. Another idea is to optimize jointly the parameters, that is, the parameters (τ, m, γ, σ) as a whole to carry on the optimization [34].

Membrane systems presented in [35], also called P systems, are bioinspired computing models belonging to a broader family of so-called biological or natural computing [36, 37], which is a distributed and parallel computing model with hierarchy. Recently, membrane systems are widely used in many fields, such as in gasoline blending scheduling, radar emitter signals analyzing, and images skeletonizing [38–40]. This paper uses a membrane computing (cell-like membrane computing optimization algorithm) to optimize simultaneously the parameters of the phase space reconstruction and LS-SVM (namely, τ, m, γ , and σ). It is an important basis for spectrum management to predict accurately the change trend of parameters in the electromagnetic environment, which can help decision makers to develop an optimal action program. Then, using the model presented in this paper to predict band occupancy rate of frequency modulation (FM) broadcasting band and interphone band.

The rest of this paper is organized as follows. Section 2 briefly reviews phase space reconstruction, LS-SVM regression, and membrane computing. Section 3 introduces specifically the algorithm of parameters joint optimization about prediction model. In Section 4 the prediction model presented in this paper will be used to predict the parameters of electromagnetic environment. Conclusions are given in Section 5.

2. Preliminaries

2.1. Phase Space Reconstruction and LS-SVM Regression. Let the time series be $\{x_1, \dots, x_{n-1}, x_n\}$; after the phase space reconstruction, the points in phase space can be expressed as [41, 42]

$$X_i = (x_i, \dots, x_{i+(m-2)\tau}, x_{i+(m-1)\tau}) \quad (i = 1, \dots, M-1, M), \quad (2)$$

where $M = n - (m-1)\tau$ is the number of phase space points, τ denotes the delay time, and m is embedding dimension.

Assume the given l samples data $\{(X_i, y_i) \mid i = 1, \dots, l - 1, l\}$, where $X_i \in R^d$ is the sample input, $y_i \in R$ is the sample output. The regression principle of LS-SVM can be explained as follows:

$$y(X) = \omega^T \Phi(X) + b, \tag{3}$$

where $\Phi(\cdot)$ is a nonlinear mapping from the input space to the feature space, ω is a vector of weight coefficients, and b is a bias constant.

The optimal hyperplane will be determined by the maximum geometry interval. Hence the LS-SVR problem can be transformed as follows [43]:

$$\min J(\omega, \zeta) = \frac{1}{2} \omega^T \omega + \frac{1}{2} \gamma \sum_{i=1}^l \zeta_i, \tag{4}$$

$$\text{s.t., } y_i = \omega^T \Phi(X_i) + \zeta_i + b, \quad i = 1, \dots, l - 1, l,$$

where ζ_i are the error variables and γ is hyperparameter. The process of finding the optimal decision function is to determine the process parameters ω and b .

Introducing Lagrange multipliers, one can establish Lagrange functions as follows:

$$L = \frac{1}{2} \omega^T \omega + \frac{1}{2} \gamma \sum_{i=1}^l \zeta_i - \sum_{i=1}^l \alpha_i (\omega^T \Phi(X_i) + \zeta_i + b - y_i), \tag{5}$$

where α_i ($i = 1, \dots, l - 1, l$) are the Lagrange multiplier. The conditions for optimality are given by

$$\frac{\partial L}{\partial \omega} = 0, \quad \frac{\partial L}{\partial b} = 0, \quad \frac{\partial L}{\partial \zeta_i} = 0, \quad \frac{\partial L}{\partial \alpha_i} = 0. \tag{6}$$

After elimination of the variables ω and ζ , a set of linear equations can be obtained:

$$\begin{pmatrix} 0 & I'^T \\ I' & \Omega + \gamma^{-1} I \end{pmatrix} \begin{pmatrix} b \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ Y \end{pmatrix}, \tag{7}$$

where $I' = (1, \dots, 1, 1)^T \in R^l$, $I \in R^{l \times l}$ denotes a unit matrix, $\alpha = (\alpha_1, \dots, \alpha_{l-1}, \alpha_l)^T$, $Y = (y_1, \dots, y_{l-1}, y_l)^T$, and $\Omega_{i,j} = \Phi(X_i)^T \Phi(X_j)$ ($i, j = 1, \dots, l - 1, l$).

Then, LS-SVM regression model is expressed as

$$f(X) = \sum_{i=1}^l \alpha_i \Phi(X_i)^T \Phi(X_j) + b. \tag{8}$$

The mapping function $\Phi(\cdot)$ can be paraphrased by a kernel function $K(\cdot, \cdot)$ because of the application of Mercer's theorem, which means that $K(\cdot, \cdot)$ ($i = 1, \dots, l - 1, l$) are any kernel functions satisfying the Mercer condition, and the Mercers condition has been applied:

$$K(X_i, X_j) = \Phi(X_i)^T \Phi(X_j) \quad (i, j = 1, \dots, l - 1, l). \tag{9}$$

This finally results in the following LS-SVM model for function regression:

$$f(X) = \sum_{i=1}^n \alpha_i K(X, X_i) + b. \tag{10}$$

As shown in Figure 1, the prediction model of phase space reconstruction and LS-SVM regression mainly has two steps. First, select the delay time (τ), embedding dimension (m), and LS-SVM parameters (γ and σ). The phase space reconstruction technique is used to determine the training sample pairs based on the parameters τ and m which are determined. Assuming the time series is $\{x_1, x_2, \dots, x_{N+1}\}$, the training sample set of attributes is as follows:

$$R = \begin{pmatrix} x_1 & x_{1+\tau} & \cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(m-1)\tau} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N-(m-1)\tau} & x_{N-(m-1)\tau+\tau} & \cdots & x_{N-(m-1)\tau+(m-1)\tau} \end{pmatrix}. \tag{11}$$

The training sample set of labels is $A = (x_{1+(m-1)\tau+1}, x_{2+(m-1)\tau+1}, \dots, x_{N+1})^T$. Second, predict future point x_i in the future. Select the attribute sample of the previous time as input in the phase space and use the trained LS-SVM model to obtain the predicted value of the moment.

2.2. Membrane Computing. Membrane computing (namely, p systems) arises as a new model of computation, inspired by the way that cells are structured into vesicles and abstracting the chemical reactions taking place inside them [44]. It is a branch of molecular computing that aims to develop models and paradigms that are biologically motivated. There has been a flurry of research activities in this area in recent years [45]. Because of the built-in nature of maximal parallelism inherent on the models, p systems have a great potential for implementing massively concurrent systems in an efficient way that would allow us to solve currently intractable problems.

A membrane system with degree d ($d > 0$) can be expressed as

$$\prod = (V, T, C, \mu, W_1, \dots, W_d, (R_1, \rho_1), \dots, (R_d, \rho_d)), \tag{12}$$

where V is an alphabet, whose elements are called objects, T denotes the output alphabet, C is a catalyst, which does not exhibit any change in the course of evolution, but some reaction must have its participation, μ is the membrane structure, which can be shown by $[\]$, W_i denotes multiple sets of objects in the membrane structure, and (R_i, ρ_i) are the set of rules, in which R_i and ρ_i denote rule and the priority of the rule, respectively.

In general, p system contains three core elements: membrane structure, object multiple sets, and evolution rules. A membrane system with given membrane structure, evolution rules, and decided objects will be performed in the form of nondeterministic and maximum parallel for the evolution rules. When all the objects are exhausted, the rules are no longer executed, the system downtime. A typical membrane system consists of cell-like membranes placed inside a unique "skin" membrane. Multisets of objects—usually strings of symbols—and a set of evolution rules are placed inside the regions delimited by the membranes. Each object can be transformed into other objects, can pass through a membrane, or can dissolve or create membranes. The evolution

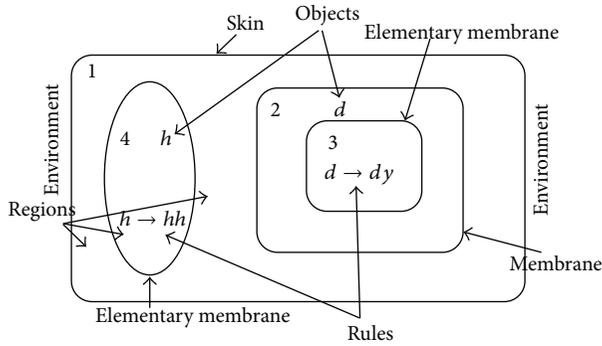


FIGURE 2: Simple membrane structure diagram.

between system configurations is done nondeterministically by applying the rules in parallel for all objects able to evolve [46]. As shown in Figure 2, a simple membrane structure diagram can be shown by $[[[[]_3]_2[[]_4]_1]_1]$. The skin membrane, which is the outermost membrane of this structure, separates the system from its environment. Several membranes, each of which defines a region, are placed inside the skin membrane. Elementary membranes do not contain any membrane. Each region forms a different compartment of the membrane structure and contains a multiset of objects or membranes. Where h and d denote objects, $h \rightarrow hh$ and $d \rightarrow dy$ are rules.

3. Parameters Joint Optimization Algorithm Based on Membrane Computing

The optimization algorithm based on cell-like membrane computing is an important branch of membrane computing. It is an intelligent optimization algorithm inspired by the mechanism and the function of biological cells and based on the existing framework of membrane computing. The steps generally are membrane structure establishment, the objects generation and evolution, and so forth. Shown in Figure 3 is the structure of P-LSSVM prediction model, with the initial objects as initial parameters of prediction model; these parameters are substituted into the phase space reconstruction and LS-SVM model. Then, parameters joint optimization algorithm based on membrane computing is used to decide the best combination of parameters. Algorithm specific process is as shown in Figure 3.

3.1. The Establishment of the Cellular Membrane Structure and the Generation of Objects. As shown in Figure 4, this paper adopts two layers structure for membrane, a skin contains B basic membrane, generate initial objects in each membrane. Generally, p system uses character or character string to encode, real number encoding are adopted in here, which can reduce the trouble of decode. For instance, $O = (o_1, o_2, o_3, o_4)$, where O is an object and o_1, o_2, o_3 , and o_4 denote τ, m, γ , and σ , respectively. We see each object as a solution of the optimization problem. Evolution of each membrane according to its own rules, all the membrane are executed in

parallel. The final optimal results are output through the skin, that is, the optimal solution.

3.2. Construct the Fitness Function. The goal of cell-like membrane computing optimization algorithm is to find the most suitable combination of parameters (τ, m, γ , and σ) in order to establish the optimal forecasting model. In this paper, we used the root mean square prediction error (RMSE) to construct the fitness function. That is, $f = 1/\text{RMSE}$, $\text{RMSE} = \sqrt{(1/Z) \sum_{i=1}^Z (y_i - \hat{y}_i)^2}$, where Z denotes the number of prediction points and y_i, \hat{y}_i represent the real values and predicted values, respectively.

3.3. Operation Rules. The basic rules of cellular membrane computing optimization method are selection, crossover, mutation, and communication [47]. The specific form is as follows.

(1) Selection rule: the rule of selection copies the objects to the next generation according to the size of the string. The size of the string is not the three-dimensional size of particles in biological cells but the value of the fitness function. Here, wheel disk method is used to select objects to the next generation.

(2) Crossover rule: for any two objects $O_i = (o_{i1}, o_{i2}, o_{i3}, o_{i4})$ and $O_j = (o_{j1}, o_{j2}, o_{j3}, o_{j4})$, use cross rule to obtain new object $O_k = (o_{k1}, o_{k2}, o_{k3}, o_{k4})$:

$$\begin{aligned} o_{k1} &= r \times o_{i1} + (1-r) \times o_{j1}, \\ o_{k2} &= r \times o_{i2} + (1-r) \times o_{j2}, \\ o_{k3} &= r \times o_{i3} + (1-r) \times o_{j3}, \\ o_{k4} &= r \times o_{i4} + (1-r) \times o_{j4}, \end{aligned} \quad (13)$$

where r is a random number in $(0, 1)$.

(3) Mutation rule: in evolution, according to a certain mutation probability, replace the worst t objects with randomly generated t objects. Mutating rule is described as follows:

$$\begin{aligned} R_{i,\text{mutation}} &: [q_{\min 1}, q_{\min 2}, \dots, q_{\min t}]_i \\ &\rightarrow [q_{\text{init}1}, q_{\text{init}2}, \dots, q_{\text{init}t}]_i, \end{aligned} \quad (14)$$

where $[]_i$ denotes membrane i , $q_{\min 1}, q_{\min 2}, \dots, q_{\min t}$ are t objects where fitness is the smallest in membrane i , and $q_{\text{init}1}, q_{\text{init}2}, \dots, q_{\text{init}t}$ are randomly generated t objects.

(4) Communication rule: each membrane i will transport the best L objects out of the membrane, while the best L objects of foreign membrane are brought into the membrane i [48]. This rule can be expressed as follows:

$$\begin{aligned} R_{i,\text{communication}} &= R_{i,\text{communication}1} \cup R_{i,\text{communication}2}, \\ R_{i,\text{communication}1} &: [q_{\max 1}, q_{\max 2}, \dots, q_{\max L}]_i \\ &\rightarrow []_i q_{\max 1}, q_{\max 2}, \dots, q_{\max L}, \end{aligned}$$

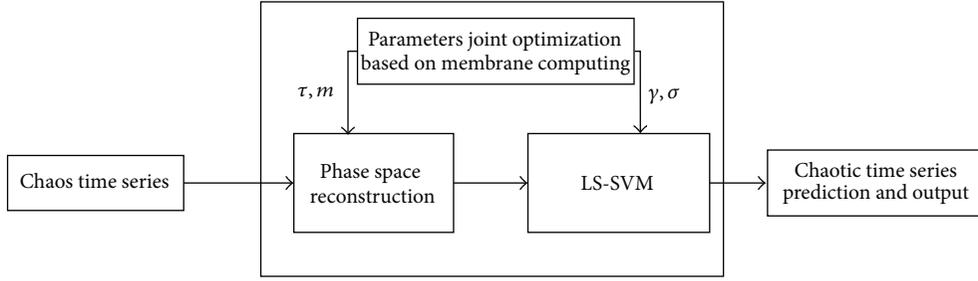


FIGURE 3: The structure of P-LSSVM prediction model.

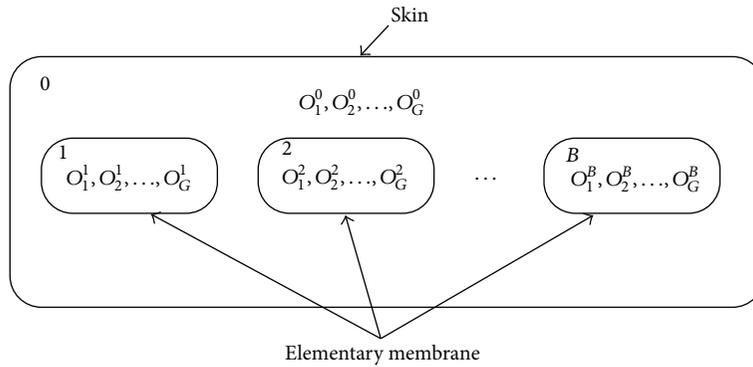


FIGURE 4: Membrane structure.

$$R_{i,communication2} : []_i q'_{max1}, q'_{max2}, \dots, q'_{maxL} \rightarrow [q'_{max1}, q'_{max2}, \dots, q'_{maxL}]_i, \quad (15)$$

where $[]_i$ denotes membrane i , $q_{max1}, q_{max2}, \dots, q_{maxL}$ are the best L objects in membrane i , and $q'_{max1}, q'_{max2}, \dots, q'_{maxL}$ are the best L objects out of membrane i .

3.4. Parameters Joint Optimization Algorithm Specific Steps in P-LSSVM Model. First of all, generate the initial objects as initial parameters of prediction model; then apply evolutionary rules to evolve until the stop conditions are met; all membranes are operating in parallel. Finally, output the fitness of the best object by the skin membrane, that is, the optimal solution. Specifically, consider the following.

Step 1. Initialize parameters and build cellular membrane structure.

(1) Initialization: the number of elementary membranes is B , the number of objects in each membrane is G , the largest number of iterations is $MaxT$, crossover probability is P_c , mutation probability is P_m , and the current iteration number is k , and so forth.

(2) Create membrane structure as shown in Figure 4, generating randomly G objects in each membrane; each object represents a set of parameters' combination, expressed in decimal coding.

Step 2. Optimize each membrane in turn.

(1) Every object in the membrane as a set of parameters (τ, m, γ , and σ) of P-LSSVM model; calculate the fitness of each object by training data and save the optimal object and its fitness.

(2) Use the reproduction, crossover, and mutation rules to evolve.

Step 3. Make use of communication rules; each membrane will transport the best L objects out of the membrane; at the same time, the best L objects outside the membrane will be shipped into the membrane.

Step 4. Determine whether the termination condition is satisfied, that is, whether it reaches the maximum number of iterations, when the number of iterations is less than the maximum number of iterations to continue iteration or stop iteration.

Step 5. The optimal object is output from the skin membrane.

4. Electromagnetic Environment Parameters Predictions Based on P-LSSVM Model

Electromagnetic spectrum is a fundamental strategic resource to support the national economy and national defense construction, along with the rapid development of information technology and it is widely used in various fields such as economic development, national defense construction, and social life [49]. Strategic value and basic role

increasingly highlight in the electromagnetic spectrum, with frequency contradictions increasingly prominent between countries, departments, and military and space businesses [50]. It is an important basis for spectrum management to control comprehensively the change trend of parameters in the electromagnetic environment of country or region [51]. It is the basis to master the frequency information for the frequency planning, frequency allocation, and sharing service frequency recovery work. The situation of electromagnetic environment can be reflected by the electromagnetic environment indicator parameters; these parameters mainly include band occupancy rate, channel occupancy rate, large-signal ratio, frequency offset, and the field strength. A large number of experiment shown that time series data with chaotic in the electromagnetic environment. Hence, we used the proposed prediction model to predict the indicator parameters of the electromagnetic environment. The experimental results show that the prediction model proposed in this paper is reasonable and effective.

Here, we chose the band occupancy rate to do the test. Band occupancy rate is calculated as follows: extracting all the signal points in the spectrum data, the signals point are merged with distance less than bandwidth by the below formula to calculate the band occupancy rate ($\text{Occupy}_{\text{Freband}}$):

$$\text{Occupy}_{\text{Freband}} = \frac{S_n * F_w}{F_{\text{end}} - F_{\text{begin}}}, \quad (16)$$

where S_n denotes the total number of signals judged, F_w is necessary bandwidth in this band for the type of specified business, F_{begin} is the start frequency point, and F_{end} is the cutoff frequency point.

4.1. Experimental Data Sources. In this paper, we adopt digital receiver EM100 which was provided by German Rohde & Schwarz Company and fixed radio monitoring station of Xihua University to collect data for the experiment. We collected data including frequency modulation (FM) broadcasting band and interphone band. As shown in Figures 5 and 6, in which the vertical axis denotes band occupancy rate, the horizontal axis represents the collection time, and left picture shows the data of band occupancy rate in FM broadcasting band, we collected for 680 hours, that is, obtaining 680 pieces of data. Right figure indicates acquisition data of band occupancy rate in interphone band; we continuously collected for 187 hours, that is, gaining 187 pieces of data. In order to facilitate narration, here we put the band data of FM broadcasting band and interphone band, denoted by “data set 1” and “data set 2,” respectively. Use the method of small amount of data to calculate the maximum Lyapunov index of two groups of data which are $\lambda_1 = 0.126$ and $\lambda_2 = 0.14$, respectively, which show the time series with chaos.

4.2. Data Preprocessing. This paper mainly uses the Grubbs criteria to deal with the abnormal data; the method is as follows: let $q(h, d)$ be the sequence of the collected data, with the time interval between two data collections $t = 1$ hour, where $h = 0, \dots, 23$ denote 24 hours of a day, $d = 1, \dots, H - 1$, H represents date code in total days of data

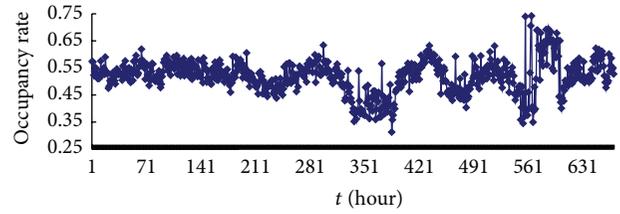


FIGURE 5: Band occupancy rate data of FM radio band.

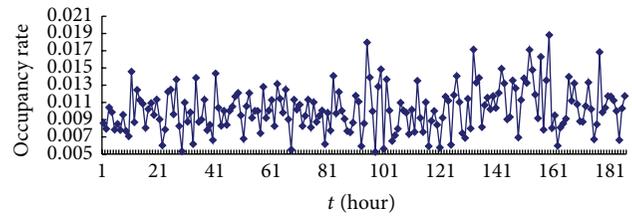


FIGURE 6: Band occupancy rate data of interphone band.

collection H , and q denotes the collected data. Using data set denoted by $Q = q_1, q_2, \dots, q_t, \dots$, for each time point h , we can get the expectation and variance of data sequence $q(h, d)$; the formula is as follows:

$$E(h) = \frac{1}{S} \sum_{k=1}^S q(h, d), \quad (17)$$

$$D(h) = \sigma_i^2 = \frac{1}{S} \sum_{k=1}^S [q(h, d) - E(h)]^2,$$

where S denotes the length of a unit.

According to the above two formulas, combined with Grubbs criteria, if the sample point meet to

$$|q(h, d) - E(h)| \geq G(n, \epsilon) \sigma_i. \quad (18)$$

The sample point should be removed, where $G(n, \epsilon)$ is the critical value of Grubbs criteria; it can be obtained by looking at Grubbs table; ϵ denotes the significance level; usually significance level $\epsilon = 0.05$.

The Grubbs criteria are used to deal with “data set 1” and “data set 2,” respectively. For the “data set 1” after processing with Grubbs criteria, the remaining 653 pieces of data, we use the front 600 pieces of data as the training data, determining the best parameters combination, and the surplus 53 pieces of data as test data, testing the prediction accuracy of the model. For the “data set 2” after processing with Grubbs criteria, the remaining 180 pieces of data, we use the front 150 pieces of data as the training data, determining the best parameters combination, and the surplus 30 pieces of data as test data, testing the prediction accuracy of the model.

4.3. Reference Model and Evaluation Criteria. In order to verify the validity of the model, this paper will compare the prediction model (P-LSSVM) proposed in this paper with conventional similar prediction model. The first reference model is the parameters joint optimization based

on genetic algorithm for chaos time series prediction (GA-LSSVM) [34]. The second reference model uses the mutual information method and Cao method to get the best delay time τ and embedding dimension m , respectively. And then use grid search method to obtain LS-SVM parameters (γ and σ) (denoted as M-C-LSSVM) [19]. The third reference model uses the mutual information method and false nearest neighbor method to calculate the optimal delay time τ and embedding dimension m , respectively. And then, use genetic algorithm to get the optimal combination parameters of LS-SVM (γ and σ) (denoted as M-F-LSSVM) [27]. The fourth reference model uses C-C method to seek simultaneously the best delay time τ and embedding dimension m . Then the optimal parameters of LS-SVM (γ and σ) by using genetic algorithm (denoted as C-C-LSSVM) [27, 52].

Meanwhile, this paper uses three evaluation criteria: normalized mean square error (NMSE), root mean square error (RMSE), and mean absolute percentage error (MAPE). NMSE, RMSE, and MAPE are defined, respectively, as follows:

$$\begin{aligned}
 \text{NMSE} &= \frac{1}{\sigma^2 Z} \sum_{i=1}^Z (y_i - \hat{y}_i)^2, & \sigma^2 &= \frac{1}{Z-1} \sum_{i=1}^Z (y_i - \bar{y})^2, \\
 \bar{y} &= \sum_{i=1}^Z y_i, \\
 \text{RMSE} &= \sqrt{\frac{1}{Z} \sum_{i=1}^Z (y_i - \hat{y}_i)^2}, \\
 \text{MAPE} &= \frac{1}{Z} \sum_{i=1}^Z \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\%,
 \end{aligned}
 \tag{19}$$

where Z is the number of prediction points, \bar{y} is the average value, and y_i and \hat{y}_i denote the real value and the predicted value of i th point, respectively.

4.4. Experimental Results. In this paper, the scope of parameters τ , m , γ , and σ is [1, 8], [3, 17], [1, 1000], and [0.1, 10000], respectively. In the process of evolution, the other parameters are set as follows: the number of elementary membranes $B = 20$, the number of objects in each membrane $G = 100$, evolution algebra $\text{Max } T = 1000$, crossover probability $P_c = 0.85$, and mutation probability $P_m = 0.05$. The optimal parameters combinations of each model are shown in Tables 1 and 2.

4.4.1. Single-Step Prediction. Selecting the first point as input to obtain first predicted value, then the real value of the first point is added to the historical data, predicting the next point. And so, obtain the predicted value of all points. Prediction results of five models are shown in Tables 3, 4, 5, 6, 7, and 8 and Figures 7, 8, 9, and 10.

4.4.2. Multistep Forecast. Selecting a point as input to obtain predicted value, then the prediction value of the first point is

TABLE 1: The optimal parameters combination of five models for FM broadcasting band.

Model	τ	σ	γ	σ
P-LSSVM	7	14	163.1	6647.5
GA-LSSVM	7	15	208.0	8250.4
M-C-LSSVM	3	15	744.1	6129.4
M-F-LSSVM	3	16	650.4	841.5
C-C-LSSVM	4	15	451.2	8082.0

TABLE 2: The optimal parameters combination of five models for interphone band.

Model	τ	σ	γ	σ
P-LSSVM	3	8	170.9	2162.1
GA-LSSVM	3	13	143.2	6532.1
M-C-LSSVM	2	14	614.3	1819.0
M-F-LSSVM	4	15	837.4	7497.8
C-C-LSSVM	2	12	798.8	6784.3

added to the historical data, predicting next point. And so, obtain the predicted value of all points. Predicted results of five models are shown in Tables 9, 10, 11, 12, 13, and 14 and Figures 11, 12, 13, and 14.

4.5. Analysis of Experimental Results. The optimal parameters combinations of five models for FM broadcasting band and interphone band are shown in Tables 1 and 2, respectively. As seen from experimental results, we can find that the parameters τ , m , γ , and σ are very sensitive to prediction accuracy; the optimal parameters combination is P-LSSVM model; FM broadcasting bands are 7, 14, 163.1, and 6647.5. Interphone bands are 3, 8, 170.9, and 2162.1. It can be seen from predicted results diagram (Figures 7 to 14) that whether single-step prediction or multistep prediction five models get very good results. However the P-LSSVM model predicts curve best fit to real data and other curves relative deviation from far away. For five prediction models, respectively, run 10 times, computing the maximum, minimum, mean, and variance of error. As can be seen from predicted results in Tables 3 to 14, three kinds of models evaluation standard are RMSE, NMSE, and MAPE; the model proposed in this paper is the minimum. This shows that not only is the P-LSSVM model reasonable and correct, but prediction accuracy is also enhanced.

Comparing single-step prediction with multistep prediction, it can be found that the error of multistep prediction is larger than the single-step prediction, indicating that the effect of single-step prediction is better than multistep prediction. The reason is that errors exist in every step, and the accumulation of error will lead to decline in the overall prediction accuracy.

5. Conclusion

Modeling and prediction of chaotic time series has become a hot spot in the research field of the chaotic signal processing.

TABLE 3: Five models predicted error based on RMSE for FM broadcasting band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	0.051	0.0486	0.0421	0.036	0.0217
run = 2	0.0501	0.0546	0.0464	0.0356	0.0252
run = 3	0.0489	0.0473	0.0425	0.0406	0.0261
run = 4	0.05	0.0545	0.0458	0.0347	0.0231
run = 5	0.0501	0.0588	0.0422	0.0407	0.023
run = 6	0.05	0.0583	0.0422	0.0365	0.0235
run = 7	0.0502	0.053	0.0439	0.0367	0.026
run = 8	0.0499	0.0545	0.0425	0.0379	0.028
run = 9	0.0495	0.0555	0.0423	0.0388	0.0282
run = 10	0.0484	0.0448	0.0474	0.0383	0.0281
Max.	0.051	0.0588	0.0474	0.0407	0.0282
Min.	0.0484	0.0448	0.0421	0.0347	0.0217
Ave.	0.0498	0.053	0.0437	0.0376	0.0253
Var.	5.00E - 07	2.00E - 05	4.00E - 06	4.00E - 06	6.00E - 06

TABLE 4: Five models predicted error based on NMSE for FM broadcasting band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	1.6941	1.5377	1.5652	1.3773	0.6544
run = 2	1.6302	1.9412	1.4676	1.3536	0.8085
run = 3	1.558	1.9554	1.5819	1.3795	0.8467
run = 4	1.6263	1.9338	1.4227	1.2984	0.7149
run = 5	1.6344	2.2537	1.4727	1.0777	0.7078
run = 6	1.6274	2.2121	1.4762	1.4095	0.7245
run = 7	1.6377	1.8296	1.4892	1.4172	0.8432
run = 8	1.6191	1.9315	1.5916	1.4934	0.9421
run = 9	1.5819	2.0028	1.4784	0.9811	0.9517
run = 10	1.5246	1.8079	1.4673	1.2086	0.9465
Max.	1.6941	2.2537	1.5819	1.4934	0.9517
Min.	1.5246	1.5377	1.4227	0.9811	0.6544
Ave.	1.6144	1.9046	1.5023	1.2996	0.814
Var.	0.0022	0.0409	0.0034	0.0264	0.0122

TABLE 5: Five models predicted error based on MAPE for FM broadcasting band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	0.0013	0.0015	0.001	9.58E - 04	8.19E - 04
run = 2	0.0013	0.0014	0.0013	9.05E - 04	4.94E - 004
run = 3	0.0015	0.0014	0.0013	9.79E - 04	4.57E - 04
run = 4	0.0013	0.0014	0.0014	8.39E - 04	4.87E - 04
run = 5	0.0015	0.0012	0.0012	8.08E - 04	8.54E - 04
run = 6	0.0015	0.0013	0.0013	8.92E - 04	6.12E - 04
run = 7	0.0015	0.0014	0.0013	8.78E - 04	7.84E - 04
run = 8	0.0013	0.0014	0.0013	9.00E - 04	5.94E - 04
run = 9	0.0015	0.0013	0.0012	8.30E - 04	7.01E - 04
run = 10	0.0015	0.0013	0.0012	8.40E - 04	6.32E - 04
Max.	0.0015	0.0015	0.0014	9.79E - 04	8.54E - 04
Min.	0.0013	0.0012	0.001	8.08E - 04	4.57E - 04
Ave.	0.0014	0.0014	0.0013	8.9E - 04	6.7E - 04
Var.	1.00E - 08	7.00E - 04	1.00E - 08	3.00E - 09	2.00E - 08

TABLE 6: Five models predicted error based on RMSE for interphone band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	0.0041	0.0044	0.0038	0.003	0.0023
run = 2	0.0042	0.0045	0.0037	0.0026	0.0023
run = 3	0.0039	0.0044	0.0038	0.0026	0.0024
run = 4	0.004	0.004	0.0036	0.0026	0.0024
run = 5	0.0042	0.0044	0.0038	0.0027	0.0023
run = 6	0.004	0.0041	0.0037	0.0026	0.0025
run = 7	0.004	0.0044	0.0037	0.0026	0.0025
run = 8	0.004	0.0045	0.0037	0.0031	0.0026
run = 9	0.0041	0.004	0.0038	0.0025	0.0025
run = 10	0.0037	0.0045	0.0037	0.0026	0.0024
Max.	0.0042	0.0045	0.0038	0.0031	0.0026
Min.	0.0037	0.004	0.0036	0.0025	0.0023
Ave.	0.004	0.0043	0.0037	0.0027	0.0024
Var.	2.00E - 08	4.00E - 08	5.00E - 09	4.00E - 08	1.00E - 08

TABLE 7: Five models predicted error based on NMSE for interphone band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	1.2458	1.7948	1.3449	1.0367	1.1577
run = 2	1.5297	1.8191	1.2826	1.0811	0.9321
run = 3	1.6352	1.7841	1.3119	1.0656	0.9817
run = 4	1.3912	1.4753	1.1894	1.0988	0.986
run = 5	1.4478	1.777	1.3191	1.2575	1.0121
run = 6	1.6357	1.5738	1.2857	1.2072	1.0362
run = 7	1.4753	1.7815	1.2601	1.0642	1.0053
run = 8	1.4478	1.8229	1.2627	1.0729	0.9984
run = 9	1.4534	1.4834	1.3027	1.079	1.0025
run = 10	1.5245	1.8605	1.412	1.023	0.975
Max.	1.6357	1.8605	1.4092	1.367	1.1577
Min.	1.2458	1.4753	1.1894	1.023	0.9321
Ave.	1.4786	1.7173	1.2971	1.0986	1.0087
Var.	0.0131	0.0216	0.0034	0.0056	0.0035

TABLE 8: Five models predicted error based on MASE for interphone band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	0.0094	0.0074	0.0044	0.0024	0.0017
run = 2	0.0093	0.0074	0.0038	0.0025	0.0015
run = 3	0.0078	0.0058	0.0032	0.0021	0.0015
run = 4	0.0086	0.0051	0.0048	0.0032	0.0011
run = 5	0.0077	0.0067	0.0029	0.0031	0.0014
run = 6	0.0045	0.0046	0.0041	0.0026	0.0014
run = 7	0.0046	0.006	0.003	0.0023	0.0015
run = 8	0.0085	0.0065	0.0032	0.0024	0.0014
run = 9	0.0084	0.0049	0.0042	0.0033	0.0016
run = 10	0.0071	0.0073	0.0043	0.0023	0.0021
Max.	0.0094	0.0074	0.0048	0.0033	0.0021
Min.	0.0045	0.0046	0.0029	0.0021	0.0011
Ave.	0.0076	0.0062	0.0038	0.0026	0.0015
Var.	3.00E - 06	1.00E - 06	4.00E - 07	2.00E - 07	7.00E - 08

TABLE 9: Five models predicted error based on RMSE for FM broadcasting band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	0.1996	0.1833	0.1498	0.1576	0.1162
run = 2	0.1957	0.192	0.1528	0.1311	0.1055
run = 3	0.1759	0.1776	0.1635	0.1426	0.0824
run = 4	0.1755	0.1617	0.1416	0.1366	0.1024
run = 5	0.1846	0.1756	0.1644	0.1397	0.1025
run = 6	0.1764	0.1534	0.1542	1.1458	0.0883
run = 7	0.1558	0.177	0.141	0.1468	0.1052
run = 8	0.1741	0.1716	0.154	0.1351	0.1076
run = 9	0.1777	0.1743	0.1639	0.1566	0.1066
run = 10	0.1768	0.1998	0.1641	0.1491	0.1065
Max.	0.1996	0.1998	0.1644	0.1576	0.1162
Min.	0.1558	0.1534	0.141	0.1311	0.0824
Ave.	0.1792	0.1766	0.1549	0.1441	0.1023
Var.	0.0001	0.0002	8.00E - 05	8.00E - 05	1.00E - 04

TABLE 10: Five models predicted error based on NMSE for FM broadcasting band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	9.3056	8.3541	6.8858	5.594	2.786
run = 2	8.7076	11.343	5.4844	4.8867	4.6361
run = 3	7.733	9.5406	7.592	6.1524	4.416
run = 4	8.6805	11.293	7.2655	5.3014	3.8288
run = 5	8.7599	13.787	8.068	4.1212	3.8322
run = 6	8.6594	13.388	6.04	5.5753	4.0734
run = 7	8.8136	10.493	7.9462	4.7008	4.944
run = 8	8.721	11.287	8.0008	5.74	4.4109
run = 9	7.9913	11.746	8.9826	5.9559	4.159
run = 10	7.549	10.485	7.7004	5.2602	3.417
Max.	9.3056	13.787	8.9826	6.1524	4.944
Min.	7.549	8.3541	5.4844	4.1212	2.786
Ave.	8.4291	11.171	7.3966	5.3288	4.0503
Var.	0.3017	2.6299	1.0599	0.3806	0.3905

TABLE 11: Five models predicted error based on MAPE for FM broadcasting band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	0.0061	0.0295	0.0038	0.0034	0.0016
run = 2	0.0085	0.0135	0.0037	0.0038	0.0025
run = 3	0.0067	0.0088	0.0045	0.0033	0.0016
run = 4	0.0097	0.0228	0.0036	0.0038	0.0025
run = 5	0.0053	0.0225	0.004	0.0033	0.002
run = 6	0.0084	0.0138	0.004	0.0033	0.002
run = 7	0.0085	0.0076	0.004	0.0041	0.002
run = 8	0.006	0.009	0.0037	0.0038	0.0022
run = 9	0.0723	0.0109	0.0043	0.0037	0.0021
run = 10	0.0058	0.0115	0.0046	0.0037	0.0026
Max.	0.0723	0.0295	0.0046	0.0041	0.0026
Min.	0.0053	0.0076	0.0036	0.0033	0.0016
Ave.	0.0137	0.015	0.004	0.0036	0.0021
Var.	0.0004	5.00E - 05	1.00E - 07	8.00E - 08	1.00E - 07

TABLE 12: Five models predicted error based on RMSE for interphone band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	0.0058	0.0063	0.0058	0.004	0.0026
run = 2	0.006	0.0064	0.0057	0.003	0.0028
run = 3	0.0056	0.0064	0.0057	0.0038	0.0025
run = 4	0.0057	0.0058	0.0056	0.0028	0.0024
run = 5	0.006	0.0063	0.0057	0.0029	0.0027
run = 6	0.0057	0.006	0.0057	0.0043	0.0023
run = 7	0.0057	0.0064	0.0057	0.0042	0.0022
run = 8	0.0057	0.0064	0.0056	0.0042	0.0027
run = 9	0.0058	0.0058	0.0057	0.005	0.0021
run = 10	0.0052	0.0064	0.0057	0.0037	0.0023
Max.	0.006	0.0064	0.0058	0.005	0.0028
Min.	0.0052	0.0058	0.0056	0.0038	0.0021
Ave.	0.0057	0.0062	0.0057	0.0038	0.0025
Var.	5.00E - 08	6.00E - 08	3.00E - 09	5.00E - 08	6.00E - 08

TABLE 13: Five models predicted error based on NMSE for interphone band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	3.0704	3.6665	3.0476	2.5014	1.1987
run = 2	3.2591	3.713	2.9508	1.6634	1.8601
run = 3	2.8179	3.7018	2.9877	2.1748	1.5472
run = 4	2.93	3.0356	2.917	1.5306	1.5779
run = 5	3.2554	3.605	3.004	1.6102	1.2004
run = 6	2.9786	3.2563	2.9549	2.8918	1.7923
run = 7	2.93	3.694	2.9264	2.7878	1.7732
run = 8	2.9325	3.7573	2.9167	2.7969	1.0517
run = 9	3.0542	3.0584	2.9812	3.852	1.0245
run = 10	2.4479	3.7791	2.9221	2.654	1.8802
Max.	3.2591	3.7791	3.0476	3.852	1.8801
Min.	2.4479	3.0356	2.9167	1.5306	1.0245
Ave.	2.9676	3.5267	2.9608	2.4463	1.4906
Var.	0.0536	0.0855	0.0019	0.5205	0.1166

TABLE 14: Five models predicted error based on MASE for interphone band.

Model	M-C-LSSVM	M-F-LSSVM	C-C-LSSVM	GA-LSSVM	P-LSSVM
run = 1	0.0209	0.015	0.0039	0.0026	0.0015
run = 2	0.0085	0.015	0.0037	0.0032	0.0012
run = 3	0.0243	0.0118	0.0038	0.0031	0.0012
run = 4	0.0225	0.0104	0.0034	0.0027	0.0021
run = 5	0.0112	0.0132	0.0033	0.0029	0.0015
run = 6	0.0039	0.0093	0.0031	0.0031	0.0022
run = 7	0.0059	0.0123	0.0034	0.0033	0.0026
run = 8	0.009	0.0132	0.0031	0.003	0.0013
run = 9	0.0125	0.0099	0.0037	0.0036	0.023
run = 10	0.005	0.0146	0.0038	0.0031	0.0026
Max.	0.0243	0.015	0.0039	0.0036	0.0026
Min.	0.0039	0.0093	0.0031	0.0026	0.0012
Ave.	0.0124	0.0125	0.0035	0.0031	0.0019
Var.	6.00E - 05	4.00E - 06	9.00E - 08	8.00E - 08	3.00E - 07

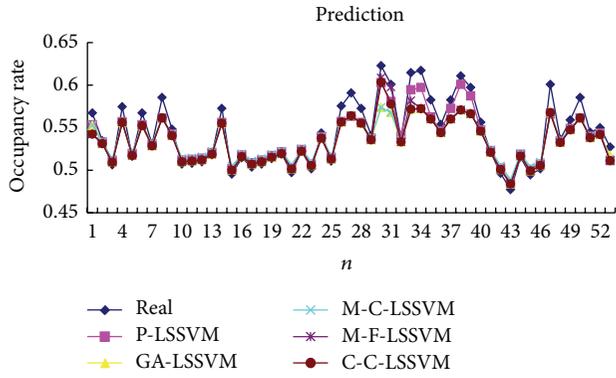


FIGURE 7: Five models predicted diagram for FM broadcasting band.

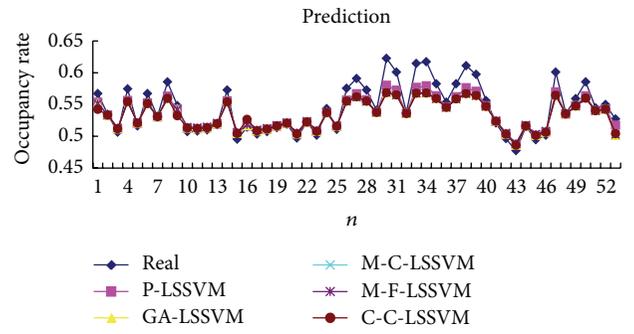


FIGURE 11: Five models predicted diagram for FM broadcasting band.

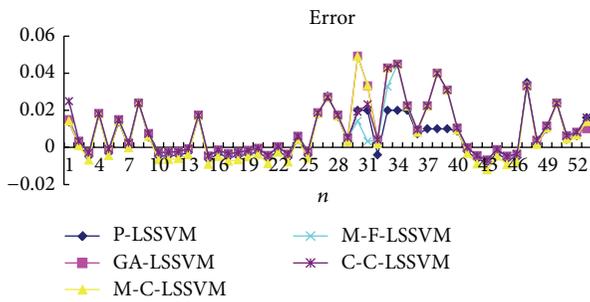


FIGURE 8: Five models predicted error diagram for FM broadcasting band.

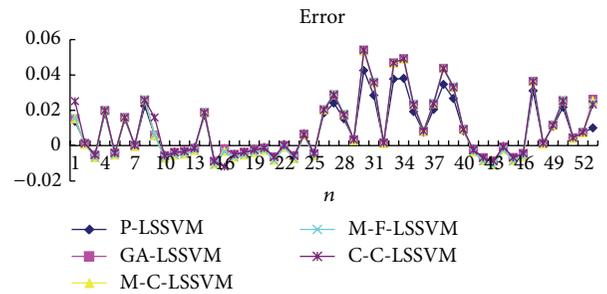


FIGURE 12: Five models predicted error diagram for FM broadcasting band.

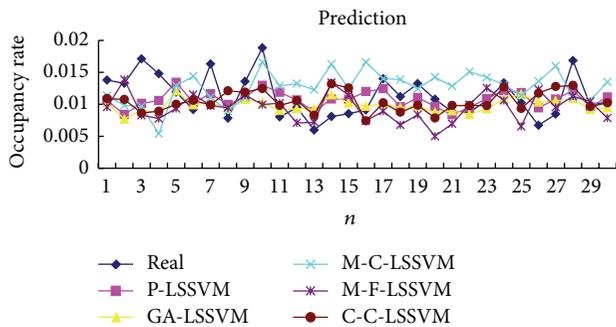


FIGURE 9: Five models predicted diagram for interphone band.

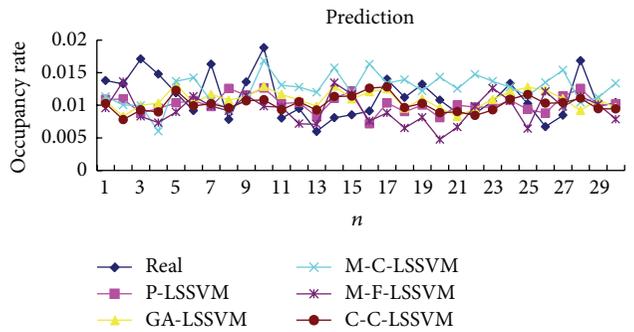


FIGURE 13: Five models predicted diagram for interphone band.

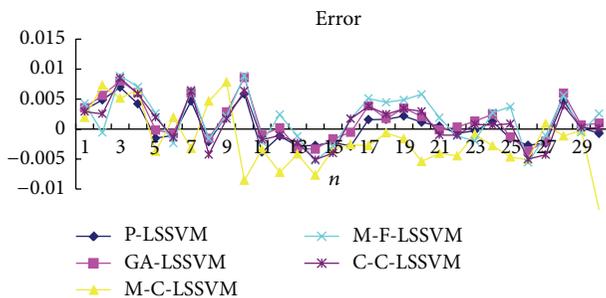


FIGURE 10: Five models predicted error diagram for interphone band.

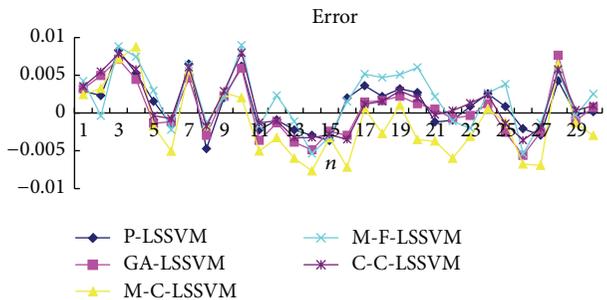


FIGURE 14: Five models predicted error diagram for interphone band.

In this paper, two defects were taken into consideration in the prediction model of LS-SVM for chaos time series prediction: on the one hand, ignoring the overall correlation of the parameters in prediction model and, on the other hand, considering the contact between the parameters, but the optimization methods have some limitations. For example, use genetic algorithm to solve the optimal parameter of prediction model, which itself has some limitations, such as falling into local optimum and iterative process complication. This paper puts forward a prediction model based on membrane computing optimization algorithm for chaos time series prediction; the model optimizes the parameters of phase space reconstruction and LS-SVM by using membrane computing optimization algorithm. Then, we used the model to forecast band occupancy rate of FM broadcasting band and interphone band. To show the applicability and superiority of the proposed model, this paper will compare the forecast model proposed in it with the traditional similar forecast model. The experimental results show that whether single-step prediction or multistep prediction, the proposed model performs best based on three error measures, namely, normalized mean square error (NMSE), root mean square error (RMSE), and mean absolute percentage error (MAPE). For deficiency in multistep prediction, in the next stage, we will further improve the prediction model or study other prediction models to improve the multistep prediction.

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 Nature Science Foundation of China (61372187), Sichuan Key Technology Research and Development Program (2012GZ0019, 2013GXZ0155), and Graduate Innovation Foundation of Xihua University (ycjj2014038).

References

- [1] R. Ren, X. J. Wang, and S.-H. Zhu, "Prediction of chaotic time sequence using least squares support vector domain," *Acta Physica Sinica*, vol. 55, no. 2, pp. 555–563, 2006.
- [2] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid Elman-NARX neural networks," *Neurocomputing*, vol. 73, no. 13–15, pp. 2540–2553, 2010.
- [3] H.-Y. Xing and T.-L. Jin, "Weak signal estimation in chaotic clutter using wavelet analysis and symmetric LS-SVM regression," *Acta Physica Sinica*, vol. 59, no. 1, pp. 140–146, 2010.
- [4] A. Kazem, E. Sharifi, F. K. Hussain, M. Saberi, and O. K. Hussain, "Support vector regression with chaos-based firefly algorithm for stock market price forecasting," *Applied Soft Computing Journal*, vol. 13, no. 2, pp. 947–958, 2013.
- [5] W.-Z. Cui, C.-C. Zhu, W.-X. Bao, and J.-H. Liu, "Prediction of the chaotic time series using support vector machines for fuzzy rule-based modeling," *Acta Physica Sinica*, vol. 54, no. 7, pp. 3009–3018, 2005.
- [6] P. Melin, J. Soto, O. Castillo, and J. Soria, "A new approach for time series prediction using ensembles of ANFIS models," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3494–3506, 2012.
- [7] P. Samui and D. P. Kothari, "Utilization of a least square support vector machine (LSSVM) for slope stability analysis," *Scientia Iranica*, vol. 18, no. 1, pp. 53–58, 2011.
- [8] Y. B. Sun, V. Babovic, and E. S. Chan, "Multi-step-ahead model error prediction using time-delay neural networks combined with chaos theory," *Journal of Hydrology*, vol. 395, no. 1–2, pp. 109–116, 2010.
- [9] V. Babovic, S. A. Sannasiraj, and E. S. Chan, "Error correction of a predictive ocean wave model using local model approximation," *Journal of Marine Systems*, vol. 53, no. 1–4, pp. 1–17, 2005.
- [10] F.-J. Chang, Y.-M. Chiang, and L.-C. Chang, "Multi-step-ahead neural networks for flood forecasting," *Hydrological Sciences Journal*, vol. 52, no. 1, pp. 114–130, 2007.
- [11] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [12] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.
- [13] L. J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1506–1518, 2003.
- [14] K. Chen and L. Liu, "Privacy preserving data classification with rotation perturbation," in *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM '05)*, pp. 589–592, November 2005.
- [15] J. W. Cai, S. S. Hu, and H. F. Tao, "Prediction of chaotic time series based on selective support vector machine ensemble," *Acta Physica Sinica*, vol. 56, no. 12, pp. 6820–6827, 2007.
- [16] J. A. K. Suykens, "Nonlinear modelling and support vector machines," in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, vol. 1, pp. 287–294, May 2001.
- [17] B. Jiang, H. Q. Wang, Y. Fu, X. Li, and G. Guo, "Based on the LS-SVM chaotic prediction of sea clutter," *Progress in Natural Science*, vol. 17, no. 3, pp. 415–421, 2007.
- [18] M. Shen, W.-N. Chen, J. Zhang, H. S.-H. Chung, and O. Kaynak, "Optimal selection of parameters for nonuniform embedding of chaotic time series using ant colony optimization," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 790–802, 2013.
- [19] Y. Q. Luo, J. B. Xia, and H. B. Wang, "Application of chaos-support vector machine regression in traffic prediction," *Computer Science*, vol. 36, no. 7, pp. 244–246, 2009.
- [20] Y. Benkler and H. Nissenbaum, "Commons-based peer production and virtue," *The Journal of Political Philosophy*, vol. 14, no. 4, pp. 394–419, 2006.
- [21] X. Y. Wang and M. Han, "Multivariate chaotic time series prediction based on hierarchic reservoirs," in *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 14–17, October 2012.
- [22] J. A. K. Suykens, J. Vandewalle, and B. de Moor, "Optimal control by least squares support vector machines," *Neural Networks*, vol. 14, no. 1, pp. 23–35, 2001.
- [23] H. S. He, J. Lu, L. F. Wu, and X. J. Qiu, "Time delay estimation via non-mutual information among multiple microphones," *Applied Acoustics*, vol. 74, no. 8, pp. 1033–1036, 2013.

- [24] H. Ma, X. Li, G. Wang, C. Han, J. Xu, and X. Zhu, "Selection of embedding dimension and delay time in phase space reconstruction," *Journal of Xian Jiaotong University*, vol. 38, no. 4, pp. 335–338, 2004.
- [25] R. Nath, "Modified generalized autocorrelation based estimator for time delays in multipath environment-A tradeoff in estimator performance and number of multipath," *Computers and Electrical Engineering*, vol. 37, no. 3, pp. 241–252, 2011.
- [26] Z. Q. Li, H. Zheng, and C. M. Pei, "A modified cao method with delay embedded," in *Proceedings of the 2nd International Conference on Signal Processing Systems (ICSPS '10)*, vol. 3, pp. V3458–V3460, July 2010.
- [27] S. H. Sun, H. Li, and Z. F. Zhang, "Oil price predicting based on unified solving by phase space reconstruction and parameters," *Computer Engineering and Applications*, vol. 49, no. 23, pp. 247–251, 2013.
- [28] I. M. Carrión, E. A. Arias Antúnez, M. M. A. Castillo, and J. J. M. Canals, "Parallel implementations of the false nearest neighbors method to study the behavior of dynamical models," *Mathematical and Computer Modelling*, vol. 52, no. 7-8, pp. 1237–1242, 2010.
- [29] C. L. Wu and K. W. Chau, "Data-driven models for monthly streamflow time series prediction," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 8, pp. 1350–1367, 2010.
- [30] C. Wei, M. Chen, C. S. Hui, and Y. S. Chang, "Study of basic method about WSD based on chaos," in *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '09)*, vol. 3, pp. 883–886, April 2009.
- [31] C. P. Liu, M. Y. Fan, G. W. Wang, and S. L. Ma, "Optimizing parameters of support vector machine based on gradient algorithm," *Control and Decision*, vol. 23, no. 11, pp. 1291–1296, 2008.
- [32] W. Wei, T. Hui, and X.-P. Ma, "Rock burst chaotic prediction on multivariate time series and LSSVR," in *Proceedings of the 25th Chinese Control and Decision Conference (CCDC '13)*, pp. 1376–1381, May 2013.
- [33] Z. Bo and A. Shi, "LSSVM and hybrid particle swarm optimization for ship motion prediction," in *Proceedings of the International Conference on Intelligent Control and Information Processing (ICICIP '10)*, pp. 183–186, August 2010.
- [34] C. Xiang, Z. Zhou, X. Yu, and L.-F. Zhang, "Study on chaotic time series prediction based on genetic algorithm," *Application Research of Computers*, vol. 28, no. 8, 2011.
- [35] G. Paun, G. Rozenberg, and A. Salomaa, *Handbook of Membrane Computing*, Oxford University Press, Oxford, UK, 2009.
- [36] G. Păun, *Membrane Computing: Main Ideas, Basic Results Application*, Idea Group Publishing, London, UK, 2004.
- [37] R. C. Muniyandi, A. M. Zin, and J. W. Sanders, "Converting differential-equation models of biological systems to membrane computing," *BioSystems*, vol. 114, no. 3, pp. 219–226, 2013.
- [38] J. Zhao and N. Wang, "A bio-inspired algorithm based on membrane computing and its application to gasoline blending scheduling," *Computers & Chemical Engineering*, vol. 35, no. 2, pp. 272–283, 2011.
- [39] G.-X. Zhang, C.-X. Liu, and H.-N. Rong, "Analyzing radar emitter signals with membrane algorithms," *Mathematical and Computer Modelling*, vol. 52, no. 11-12, pp. 1997–2010, 2010.
- [40] D. P. Daniel, P. C. Francisco, and M. A. Gutiérrez-Naranjo, "A parallel algorithm for skeletonizing images by using spiking neural P systems," *Neurocomputing*, vol. 115, pp. 81–91, 2013.
- [41] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," *Physical Review Letters*, vol. 45, no. 9, pp. 712–716, 1980.
- [42] F. Takens, "Detecting strange attractors in turbulence," *Lecture Notes in Mathematics*, vol. 898, pp. 361–381, 1981.
- [43] J. A. K. Suykens, J. de Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing*, vol. 48, no. 3, pp. 85–105, 2002.
- [44] A. Riscos-Núñez, "A Framework for Complexity Classes in Membrane Computing," *Electronic Notes in Theoretical Computer Science*, vol. 225, pp. 319–328, 2009.
- [45] O. H. Ibarra, "On membrane hierarchy in P systems," *Theoretical Computer Science*, vol. 334, no. 1–3, pp. 115–129, 2005.
- [46] C. Teuscher, "From membranes to systems: self-configuration and self-replication in membrane systems," *BioSystems*, vol. 87, no. 2-3, pp. 101–110, 2007.
- [47] L. Huang, I. H. Suh, and A. Abraham, "Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants," *Information Sciences*, vol. 181, no. 11, pp. 2370–2391, 2011.
- [48] T. M. Taher, R. B. Bacchus, K. J. Zdunek, and D. A. Roberston, "Long-term spectral occupancy findings in Chicago," in *Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN '11)*, pp. 100–107, May 2011.
- [49] H. Zhou, "Analysis and resolved strategy of complicated electromagnetic environment in battlefield," *Journal of the Academy of Equipment Command & Technology*, vol. 18, no. 16, 2007.
- [50] T. Shao, H. U. Yihua, S. H. Liang et al., "Methods for quantitative evaluation of battlefield electromagnetic environment complexity," *Electronics Optics & Control*, vol. 17, no. 1, 2010.
- [51] Z. Wang and S. Salous, *Spectrum occupancy analysis for cognitive radio [Ph.D. dissertation]*, Durham University, Durham, UK, 2009.
- [52] H. Yin and H. Wu, "Study on time series forecasting based on least squares support vector machine," *Computer Simulation*, vol. 2, p. 28, 2011.

Research Article

Composition of Web Services Using Markov Decision Processes and Dynamic Programming

Víctor Uc-Cetina, Francisco Moo-Mena, and Rafael Hernandez-Ucan

Facultad de Matemáticas, Universidad Autónoma de Yucatán, Anillo Periférico Norte, Tablaje Cat. 13615, Apartado Postal 192, Colonia Chuburná Hidalgo Inn, 97119 Mérida, YUC, Mexico

Correspondence should be addressed to Víctor Uc-Cetina; ucetina@uady.mx

Received 26 June 2014; Revised 17 September 2014; Accepted 14 October 2014

Academic Editor: Ahmad T. Azar

Copyright © 2015 Víctor Uc-Cetina et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a Markov decision process model for solving the Web service composition (WSC) problem. Iterative policy evaluation, value iteration, and policy iteration algorithms are used to experimentally validate our approach, with artificial and real data. The experimental results show the reliability of the model and the methods employed, with policy iteration being the best one in terms of the minimum number of iterations needed to estimate an optimal policy, with the highest Quality of Service attributes. Our experimental work shows how the solution of a WSC problem involving a set of 100,000 individual Web services and where a valid composition requiring the selection of 1,000 services from the available set can be computed in the worst case in less than 200 seconds, using an Intel Core i5 computer with 6 GB RAM. Moreover, a real WSC problem involving only 7 individual Web services requires less than 0.08 seconds, using the same computational power. Finally, a comparison with two popular reinforcement learning algorithms, sarsa and Q-learning, shows that these algorithms require one or two orders of magnitude and more time than policy iteration, iterative policy evaluation, and value iteration to handle WSC problems of the same complexity.

1. Introduction

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network, with an interface described in a machine-processable format called Web Services Description Language [1]. A Web service is typically modeled as a software component that implements a set of operations. The emergence of this type of software components has created unprecedented opportunities to establish more agile collaborations between organizations, and as a consequence, systems based on Web services are growing in importance for the development of distributed applications designed to be accessed via the Internet.

When a Web service is requested, all available Web services descriptions must be matched with the requested description, so that an appropriate service with the desired functionality can be found. However, since the number of available Web services is continuously growing year by year, finding the best match is not a trivial problem anymore, especially if we take into account that the matching criteria

must consider not only the desired functionality, but also other attributes such as execution cost, security, performance, and so forth.

If individual Web services are not able to meet complex requirements, they can be combined to create composite services [2]. A composite Web service has one initial task and one ending task, and between the initial and the ending tasks there can be $k = \{0, 1, 2, \dots, K\}$ individual tasks connected in sequential order. To create a composite Web service it is necessary to discover and select the most suitable services. The complexity of WSC involves three main factors: (1) the large number of dynamic Web Services instances with similar functionality that may be available to a complex service; (2) the different possibilities of integrating service instance components into a complex service process; (3) various performance requirements (e.g., end-to-end delay, service cost, and reliability) of a complex service.

1.1. Related Work. Some approaches to solve the WSC problem have focused on different graph-based algorithms [3–8].

Some others have proposed to use optimization methods specially designed for solving constraint satisfaction problems, such as integer programming [9], linear programming [10], or methods for solving the knapsack problem [11]. Artificial intelligence methods such as planning algorithms [12–14], ant colony optimization [15], fuzzy sets [2], and binary search trees [16] have been used too.

The use of methods based on Markov decision processes (MDPs) for the composition problem is certainly not new. In [17], the problem of workflow composition is modeled as a MDP and a Bayesian learning algorithm is used to estimate the true probability models involved in the MDP. In [18], the WSC is solved using QoS attributes in a MDP framework with two versions of the value iteration algorithm: one backward and recursive and one forward version. In [19], the authors proposed the use of what they call value of changed information. Their approach uses MDPs focusing on changes of the state transition function, in order to anticipate values of the service parameters that do not change the WSC. In [20], a combination of MDPs and HTN (Hierarchical Task Network) planning is proposed.

Solutions based on reinforcement learning are also relevant. For instance, in [21], reinforcement learning and preference logic were employed together to solve the WSC problem, obtaining some kind of qualitative solution. Authors argue that computing a qualitative solution has many advantages over a quantitative one. Other methods using Q-learning are given in [22–24]. It is important to remember that reinforcement learning methods [25] belong to a family of algorithms highly related to the MDPs. The main difference with these methods is that the state transition function is assumed to be unknown and therefore the agents need to explore their state and action spaces by executing different actions in different states and observe the numerical rewards obtained after each state transition.

1.2. Contributions of This Paper. The goal of automatic WSC is to determine a sequence of Web services that can be combined to satisfy a set of predefined QoS constraints. For problems where we need to find the sequence of actions maximizing an overall performance function, the MDPs are one of the most robust mathematical tools that we can use. Therefore, in this paper we propose an MDP model to solve the WSC problem. To show the reliability of our model, we conducted experiments with three of the most studied algorithms: policy iteration, iterative policy evaluation, and value iteration. Although all three algorithms provided good solutions, the policy iteration algorithm required the minimum number of iterations to converge to the optimal solutions. We also compared these three algorithms against sarsa and Q-learning, showing that the latter methods require one or two orders of magnitude and more time to solve composition problems of the same complexity.

This paper is structured as follows. Section 2 provides the basics of the MDPs framework and introduces the three algorithms that we tested. Section 3 introduces our MDP model for solving the WSC problem. Section 4 describes the experimental setup and presents the most relevant results. Section 5 presents comparative experiments with sarsa and

Q-learning algorithms. Finally, Section 6 concludes this paper by discussing the main findings and providing some advice for future research.

2. Markov Decision Processes

The WSC problem can be abstracted as the problem of selecting a sequence of actions, in such a way that we maximize an overall evaluation function. Such kind of sequential decision problems can be defined and solved in an MDP framework. An MDP is a tuple (S, A, P, γ, R) , where S is a set of states, A is a set of actions, $P(s_{t+1} | s_t, a_t)$ are the state transition probabilities for all states $s_t, s_{t+1} \in S$ and actions $a \in A$, $\gamma \in [0, 1)$ is a discount factor, and $R : S \times A \rightarrow \mathfrak{R}$ is the reward function.

The MDP dynamics is the following. An agent in state $s_t \in S$ performs an action a_t selected from the set of actions A . As a result of performing action a_t , the agent receives a reward with expected value $R(s_t, a_t)$ and the current state of the MDP transitions to some successor state s_{t+1} , according to the transition probability $P(s_{t+1} | s_t, a_t)$. Once in state s_{t+1} the agent chooses and executes an action a_{t+1} , receiving reward $R(s_{t+1}, a_{t+1})$ and moving to state s_{t+2} . The agent keeps choosing and executing actions, creating a path of visited states $s_t, s_{t+1}, s_{t+2}, \dots$

As the agent goes through states, s_0, s_1, s_2, \dots , it obtains the following rewards:

$$R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \dots \quad (1)$$

The reward at timestep t is discounted by a factor of γ^t . By doing so, the agent gives more importance to those rewards obtained sooner. In an MDP we try to maximize the sum of expected rewards obtained by the agent:

$$E \left[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \dots \right]. \quad (2)$$

A policy is defined as a function $\pi : S \rightarrow A$ mapping from the states to the actions. A value function for a policy π is the expected sum of discounted rewards, obtained by performing always the actions provided by π :

$$V^\pi(s) = E \left[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \dots \mid s_0 = s, \pi \right]. \quad (3)$$

V^π is the expected sum of discounted rewards that the agent would receive if it starts in state s and takes actions given by π . Given a fixed policy π , its value function V^π satisfies the Bellman equation:

$$V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s' \in S} P(s' | s, \pi(s)) V^\pi(s'). \quad (4)$$

The optimal value function is defined as

$$V^*(s) = \max_{\pi} V^\pi(s). \quad (5)$$

```

(1) foreach state do
(2)    $V(s) \leftarrow 0$ 
(3) end
(4) repeat
(5)   foreach state do
(6)      $V_{i+1}(s) \leftarrow \sum_{a \in A} \pi(s, a) \left[ R(s, a) + \gamma \sum_{s' \in S} P(s' | s, a) V_i(s') \right]$ 
(7)   end
(8) until convergence

```

ALGORITHM 1: Iterative policy evaluation.

```

(1) initialize  $\pi_0$  randomly
(2) repeat
(3)    $V_i \leftarrow R(s, \pi_i(s)) + \gamma \sum_{s' \in S} P(s' | s, \pi_i(s)) V_i(s')$ 
(4)   foreach state do
(5)      $\pi_{i+1}(s) \leftarrow \arg \max_{a \in A} \left[ R(s, a) + \gamma \sum_{s' \in S} P(s' | s, a) V_i(s') \right]$ 
(6)   end
(7) until convergence

```

ALGORITHM 2: Policy iteration algorithm.

This function gives the best possible expected sum of discounted rewards that can be obtained using any policy π . The Bellman equation for the optimal value function is

$$V^*(s) = \max_{a \in A} \left[R(s, a) + \gamma \sum_{s' \in S} P(s' | s, a) V^*(s') \right]. \quad (6)$$

The optimal value function is such that we have

$$V^*(s) = V^{\pi^*}(s) \geq V^\pi(s). \quad (7)$$

2.1. Dynamic Programming Algorithms for MDPs. When the state transition probabilities are known, dynamic programming can be used to solve (6). Next, we present three efficient algorithms for solving finite-state MDPs by means of dynamic programming. The first one is the iterative policy evaluation (given in Algorithm 1). The second one is the policy value iteration algorithm (given in Algorithm 2). This algorithm repeatedly computes the value function for the current policy and then updates the policy using the current value function. The third one, shown in Algorithm 3, called value function iteration, can be thought as an iterative update of the estimated value function using Bellman Equation (6).

The last two algorithms are known to converge usually faster than the first one. Moreover policy iteration and value iteration are standard algorithms for solving MDPs, and there is not currently universal agreement over which algorithm is better [26, 27].

3. Web Service Composition Model

In this section we define the MDP model used to represent and solve the Web service composition problem by means of dynamic programming algorithms.

We begin by describing the WSC problem in more details. Individual Web services can be categorized in classes by their functionality, input data, and output data. Given C different classes of individual Web services, the WSC problem consists in finding a sequence of length C of individual Web services $\langle w_1, w_2, \dots, w_C \rangle$, such that $w_i \in W_i$, for $i = 1, 2, \dots, C$, where W_i is the set of all available Web services of class i . Thus, we are making the assumption that a valid composite Web service needs a Web service from each of the existing classes. We are also making the assumptions that all available Web services have been previously categorized into C classes and that the ordering of the classes $W_1 < W_2 < \dots < W_C$ has been predefined. $W_i < W_j$ means that a Web service from set W_i must be executed before a Web service from set W_j to ensure the correct operation of the selected Web services. The correct operation depends basically on their functionality and input and output data. Therefore, the output of w_i must be fully compatible with the input of w_j .

Now, we are ready to introduce our model. We define a Web service composition problem as an MDP (S, A, P, γ, R) , where S is the set of states, A is the set of actions, P is the state transition probability function, γ is a discount factor such that $\gamma \in [0, 1)$, and R is the reward function. Elements S, A, P , and R are defined next.

```

(1) foreach state do
(2)    $V(s) \leftarrow 0$ 
(3) end
(4) repeat
(5)   foreach state do
(6)      $V_{i+1}(s) \leftarrow \max_{a \in A} \left[ R(s, a) + \gamma \sum_{s' \in S} P(s' | s, a) V_i(s') \right]$ 
(7)   end
(8) until convergence

```

ALGORITHM 3: Value iteration algorithm.

3.1. States. S is the set of states. Given a WSC problem with C classes, S consists of all compositions of length at most C . Thus, for $C = 1$, $S = \{\langle w_1 \rangle\}$, with $w_1 \in W_1$. A composition of length $l = 1$ is not really a composition; it is just a single Web service; however, we will relax the meaning of the word composition and will call it a composition of length $l = 1$. For $C = 2$, $S = \{\langle w_1 \rangle, \langle w_1, w_2 \rangle\}$, with $w_1 \in W_1$ and $w_2 \in W_2$. For $C = 3$, $S = \{\langle w_1 \rangle, \langle w_1, w_2 \rangle, \langle w_1, w_2, w_3 \rangle\}$, with $w_1 \in W_1$, $w_2 \in W_2$, and $w_3 \in W_3$. In general, for a WSC problem with C classes $S = \{\langle w_1 \rangle, \langle w_1, w_2 \rangle, \dots, \langle w_1, w_2, \dots, w_C \rangle\}$.

3.2. Actions. A is the set of all actions. Given a state s , the set of actions available from s is denoted by $A(s)$; thus $A = \{A(s)\}_{s \in S}$. An action consists of selecting a Web service to be included in the current composition. If the current composition is of length $l = i$, all the possibilities of selecting a Web service of class $c = i + 1$ will constitute the set of current available actions.

Formally, we say that $A = \{A(s_{l=0}), A(s_{l=1}), A(s_{l=2}), \dots, A(s_{l=C-1})\}$, where $A(s_{l=i})$ is read as the set of actions available from a state representing a composition of length $l = i$. Note that $A(s_{l=0})$ refers to set of actions available from a composition of length $l = 0$, which corresponds to the state where none of the Web services has been selected yet.

For example, if the current state represents the composition $\langle w_1, w_2 \rangle$ which is of length $l = 2$, then $A(s_{l=2})$ is given by all the possibilities of selecting a Web service of class $c = 3$. In other words, we are in a situation where we have already selected Web services from class $c = 1$ and class $c = 2$, and now we need to select a Web service from class $c = 3$.

3.3. Transition Probabilities. $P(s' | s, a)$ are the state transition probabilities for all states $s, s' \in S$ and actions $a \in A$, which are currently available from s and s' . Note that the probability of going from a state $s = \langle w_1 \rangle$ to the state $s' = \langle w_1, w_2 \rangle$ is 1. Meanwhile, the probability of going from the same state $s = \langle w_1 \rangle$ to a state $s' = \langle w_1, w_2, w_3 \rangle$ is 0. In other words, we can only go from a composition state of length $l = i$ to another composition state of length $l = i + 1$.

3.4. Reward Function. $R(s' | s, a)$ is the reward received when action a is executed and the environment makes a transition from s to s' . The reward function for our model is computed using three QoS attributes, as indicated in (8),

which was originally proposed in [22]. The QoS employed are availability, throughput, and execution time:

$$R(s) = \frac{av^s - av^{\min}}{av^{\max} - av^{\min}} - \frac{\text{time}^s - \text{time}^{\min}}{\text{time}^{\max} - \text{time}^{\min}} + \frac{tr^s - tr^{\min}}{tr^{\max} - tr^{\min}}, \quad (8)$$

where av^s , time^s , tr^s are the availability, average execution time, and throughput values for the last Web service added to the composition represented by state s . av^{\min} , time^{\min} , tr^{\min} and av^{\max} , time^{\max} , and tr^{\max} are the minimum and maximum values for all the Web services.

4. Experimental Evaluation

In this section we provide the results of our experimental comparison using two scenarios, one real and one artificial. The experiments that we present in this section were performed running policy iteration, iterative policy iteration, and value iteration algorithms, on an Intel Core i5 2.5 GHz processor, on Windows 8.1, 64 bits operating system, and 6 GB RAM.

4.1. Real Scenario. The WSC problem considered as our first experimental scenario consists of 2 classes of Web services. One class is about weather services that can be used to obtain the current temperature in a city. The other class is about Web services that can be used to convert temperatures from one metric unit to another, for example, from Fahrenheit to Celsius. In the class of weather services we considered 3 different Web services.

- (i) National Oceanic and Atmospheric Administration (NOAA) Web service, available at http://graphical.weather.gov/xml/SOAP_server/ndfdXMLserver.php.
- (ii) GlobalWeather Web service, available at <http://www.webservices.net/globalweather.asmx>.
- (iii) Weather channel Web service, available at <http://api.wunderground.com/>.

In the class of metric units conversion services we considered 4 different Web services.

- (i) A simple calculator Web service such as the one available at <http://www.dneonline.com/calculator.asmx>. Since

$$C = \frac{5 * (F - 32)}{9}, \tag{9}$$

we can use subtraction, multiplication, and division operations for the temperature conversion.

- (ii) ConvertTemperature Web service, available at <http://www.websvix.net/ConvertTemperature.asmx>.
- (iii) TemperatureConversions Web service, available at <http://webservices.daehosting.com/services/TemperatureConversions.wso>.
- (iv) TempConvert Web service, available at <http://www.w3schools.com/webservices/tempconvert.asmx>.

We obtained the QoS attribute values of all 7 Web services using a java program designed to get the attribute values with the following formulas:

$$\text{Availability} = \frac{C_S}{C_T}, \tag{10}$$

where C_S is the number of successful calls to the Web service and C_T are the total calls,

$$\text{Execution time} = \frac{T}{C_T}, \tag{11}$$

where T is the total execution time for all the C_T calls,

$$\text{Throughput} = \frac{C_S}{T}, \tag{12}$$

with $C_T = 50$.

In order to obtain representative QoS values for the Web services, we made many measurements, several days in different moments of the day. We obtained the values for each parameter and measurement, and then we calculated the average values for the QoS parameters.

Once we gathered the information of the QoS attributes we used all 3 dynamic programming algorithms to learn the best composite Web service. With 7 Web services belonging to 2 different classes, there are 12 possible compositions. All these possibilities are represented with the graph illustrated in Figure 1.

The graph of the real scenario illustrates each class of Web services as a layer. In this graph, each node represents an individual Web service. Node S represents the state where none of the Web services has been selected yet. Node G represents the state where a full composition of Web service has been accomplished. A path from S to G implies that a valid composite Web service has been generated.

Results with the real Web services scenario are plotted in Figure 2. All 3 algorithms found the solution for the Web service composition very quickly, in less than 0.07 seconds, with policy iteration being the winner.

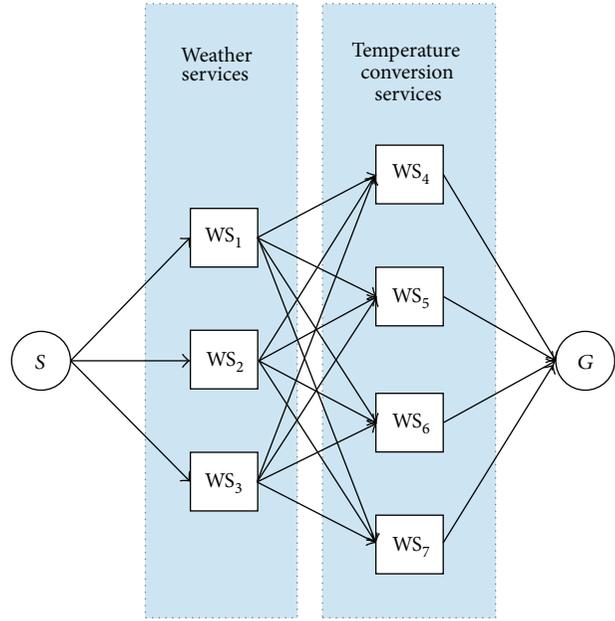


FIGURE 1: Graph for the real scenario with 2 classes of Web services. The first class contains 3 Web services and the second class contains 4 Web services. Each class is illustrated as a layer of nodes.

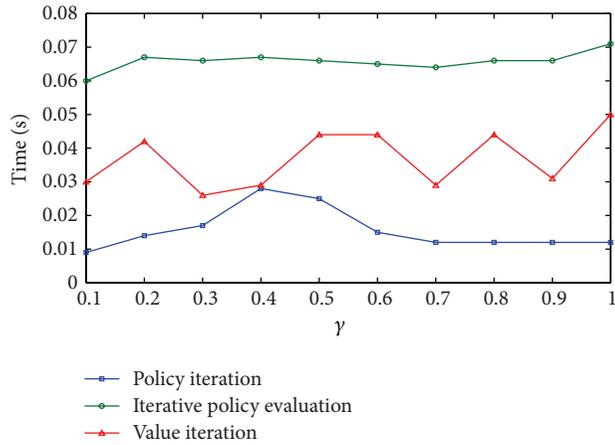


FIGURE 2: Learning times for the real scenario.

4.2. *Artificial Scenario.* As our second scenario to test all 3 dynamic programming algorithms, we simulated data for three QoS attributes: availability, execution time, and throughput. We created a maximum of 100,000 individual Web services, classified into 100 hypothetical classes of Web services. We assumed that every Web service in a class i can access all the Web services in class $i + 1$. Each of these classes is represented as a layer in Figure 3. Each layer contains 100 nodes or individual Web services.

As in the first scenario, node S is the initial state of the graph and represents a state where none of the Web services has been selected yet. Node G is reached when a valid composition has been accomplished. Nodes between S and G

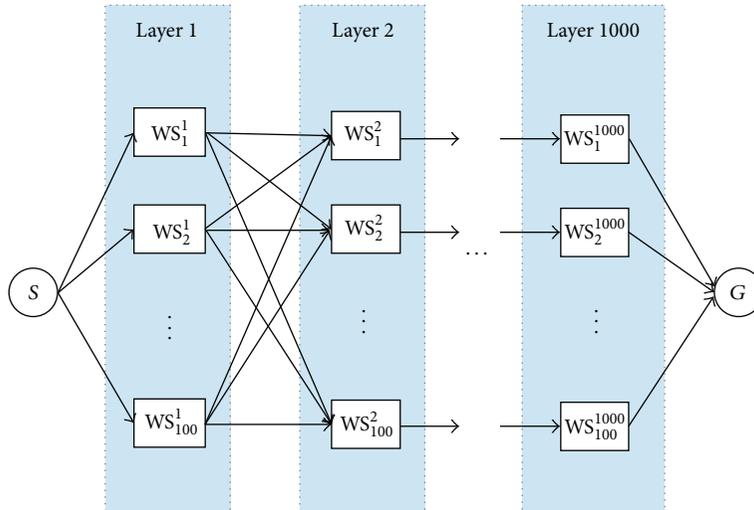


FIGURE 3: Graph for an artificially generated Web composition problem with a maximum of 1,000 selected nodes. Each node is selected out of 100 possible individual Web services belonging to the same class (layer).

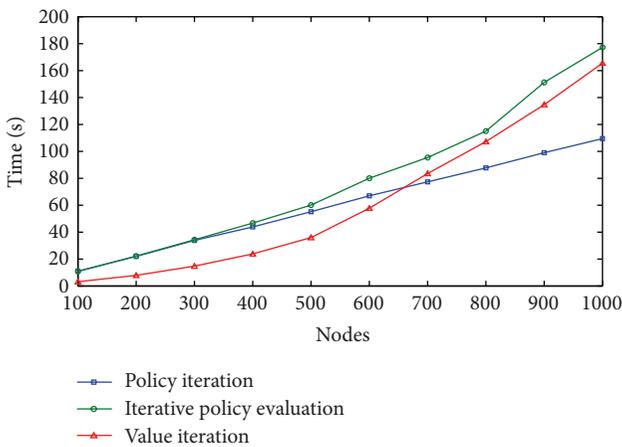


FIGURE 4: Learning times with $\gamma = 0.7$.

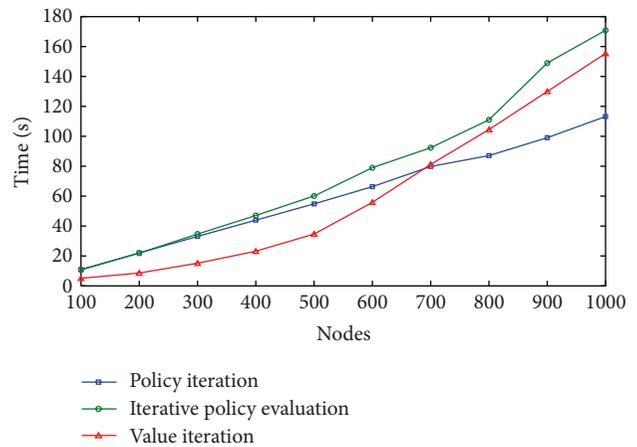


FIGURE 6: Learning times with $\gamma = 0.9$.

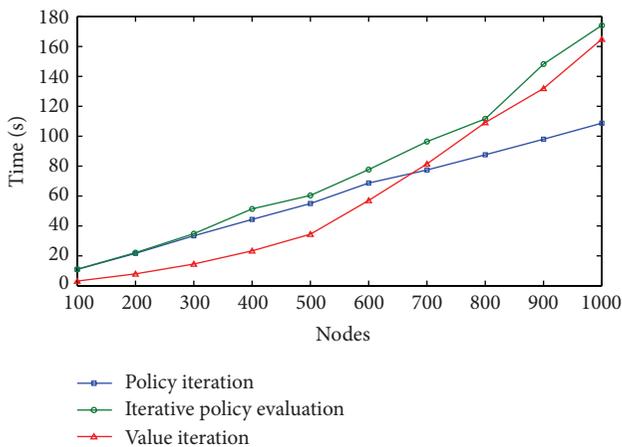


FIGURE 5: Learning times with $\gamma = 0.8$.

represent the available Web services. A route from S to G gives a possible composite Web service.

Results of this second set of experiments are shown in Figures 4, 5, and 6, for $\gamma = 0.7$, $\gamma = 0.8$, and $\gamma = 0.9$, respectively.

Each layer in the graph represents 100 Web services belonging to the same class. Therefore, when the number of nodes to be selected for a valid Web service composition is 1,000, we are really solving a problem with $100 \times 1,000 = 100,000$ Web services. We can see from the learning curves that the time needed to solve the MDP problem increases as the number of nodes is increased. Again, all 3 algorithms found the optimal solution, but policy iteration found it in less time. The best performances of the algorithms were obtained for $\gamma = 0.8$ and $\gamma = 0.9$, requiring less than 180 seconds to find the optimal composition using iterative policy evaluation and value iteration and less than 120 in the case of policy iteration.

```

(1) initialize  $Q(s, a)$  arbitrarily
(2) foreach training episode do
(3)   initialize  $s$ 
(4)   choose  $a$  from  $s$  using policy derived from  $Q$ 
(5)   repeat for each step of episode
(6)     take action  $a$ , observe  $r, s'$ 
(7)     choose  $a'$  from  $s'$  using policy derived from  $Q$ 
(8)      $Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma Q(s', a') - Q(s, a)]$ 
(9)      $s \leftarrow s'; a \leftarrow a'$ 
(10)  until  $s$  is terminal
(11) end

```

ALGORITHM 4: Sarsa algorithm.

```

(1) initialize  $Q(s, a)$  arbitrarily
(2) foreach training episode do
(3)   initialize  $s$ 
(4)   repeat for each step of episode
(5)     choose  $a$  from  $s$  using policy derived from  $Q$ 
(6)     take action  $a$ , observe  $r, s'$ 
(7)      $Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ 
(8)      $s \leftarrow s'$ ;
(9)     until  $s$  is terminal
(10) end

```

ALGORITHM 5: Q-learning algorithm.

5. Comparison with Sarsa and Q-Learning

In some related works [22–24], reinforcement learning algorithms were proposed to solve the Web service composition problem. In this section we compare the learning times required by sarsa and Q-learning against policy iteration, iterative policy evaluation, and value iteration.

5.1. Sarsa. Sarsa [25] is an on-policy temporal difference control algorithm which continually estimates the state-action value function Q^π for the behavior policy π and at the same time changes π toward greediness with respect to Q^π . Algorithm 4 presents the sarsa algorithm as taken from [25].

If the policy is such that each action is executed infinitely often in every state, every state is visited infinitely often, and it is greedy with respect to the current action-value function in the limit, then by decaying α , the algorithm converges to Q^* [28].

5.2. Q-Learning. Q-learning [29] is an off-policy temporal difference control algorithm which directly approximates the optimal action-value function, independently of the policy being followed. It is one of the most popular algorithms in reinforcement learning. Algorithm 5 reproduces the Q-learning algorithm as taken from [25].

If in the limit the action-values of all state-action pairs are updated infinitely often, with a decaying α , then the algorithm converges to Q^* with probability 1 [26, 30].

5.3. Learning Time Analysis. We have implemented sarsa and Q-learning algorithms to solve the real scenario problem defined previously in the experimental section. A comparison graph illustrating the time required by sarsa, Q-learning, policy iteration, iterative policy evaluation, and value iteration is given using a logarithmic scale in Figure 7. From this graph we can clearly see that sarsa and Q-learning required two orders of magnitude and more time to find the optimal composition.

Additionally, we ran experiments with a second artificially created scenario, with 3 layers of 20 Web services each. Once more, reinforcement learning methods required much more time than the dynamic programming algorithms. Logarithmic time curves given in Figure 8 show that sarsa and Q-learning required one order of magnitude and more time than dynamic programming algorithms. Furthermore, in some of the experiments, reinforcement learning algorithms failed to find the optimal solution, getting stuck in suboptimal compositions.

Dynamic programming methods converge faster than reinforcement learning methods simply because dynamic programming methods update every single state value at each iteration. Reinforcement learning methods only update the value of the states that happen to visit, giving its exploration policy, that is, epsilon greedy.

Furthermore, in terms of the deployment of an automatic Web service composition system, it is worth mentioning that the gathering of QoS information can be performed

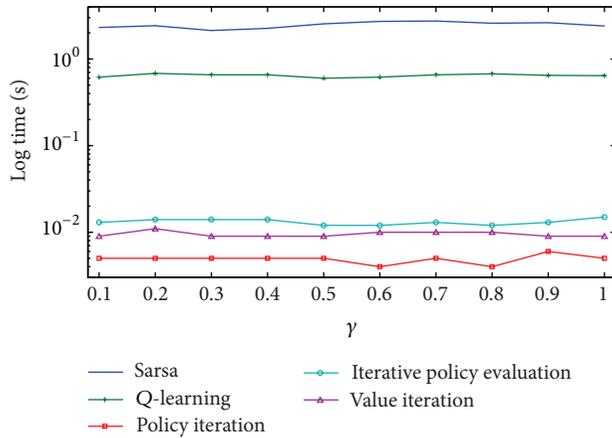


FIGURE 7: Learning times required for a real scenario of Web service composition, plotted in logarithmic scale. Reinforcement learning methods required two orders of magnitude and more time than dynamic programming methods.

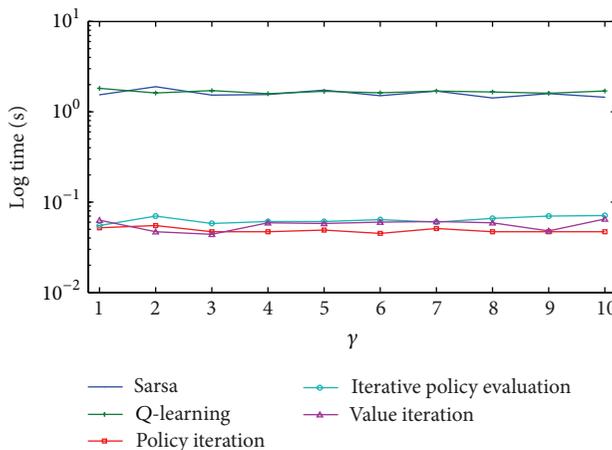


FIGURE 8: Learning times required for a simulated scenario with 3 layers of 20 Web services. Curves plotted in logarithmic scale show that reinforcement learning methods required ten times more time than dynamic programming algorithms to handle the same kind of problem.

at specific time intervals by a dedicated module of such system. Once we have gathered this information, which is fundamental for the evaluation of the reward function, there is no need to explore the state space of Web services as reinforcement learning methods do. We can simply run a dynamic programming algorithm to estimate the value function of the Web services and then compute the optimal composition of Web services.

6. Conclusion

In this paper we have proposed an MDP model to address the Web service composition problem. We used three dynamic programming algorithms, namely, iterative policy evaluation, value iteration, and policy iteration, to show the reliability

of our approach. Experiments were conducted with both artificially created data and a set of real data involving seven publicly available Web services.

Our experimental results show that policy iteration is the best one in terms of the minimum number of iterations needed to estimate an optimal policy. The optimal policy indicates the sequence of combined individual Web services making up a composite Web service with the highest evaluation of their QoS attributes.

Although some approaches using reinforcement learning have also been proposed, we argue that dynamic programming methods are better suited for the Web service composition problem than reinforcement learning methods. The reason is that reinforcement learning methods such as sarsa and Q-learning require a lot of exploration of the state space and consequently they need more iterations to make a good estimation of the optimal policy. To illustrate this, we compared sarsa and Q-learning against policy iteration, iterative policy evaluation, and value iteration. The result of this comparison is that sarsa and Q-learning required one or two orders of magnitude and more time than the dynamic programming methods to handle problems of the same complexity. Moreover, in some of the artificially created experiments, reinforcement learning algorithms got stuck in suboptimal Web services compositions.

None of the related works proposing the use of MDP-based methods to solve the Web service composition problem have provided a comparison study involving the five algorithms that we have analyzed in this work: iterative policy evaluation, value iteration, policy iteration, sarsa, and Q-learning. Moreover, we present experimental results using both a real scenario and a Web service composition scenario with artificially generated data. All other related works report experiments performed only with artificially created data.

Future research on this topic must address real Web services composition involving more nodes. Another interesting subject that deserves to be further investigated is the design of complex reward functions capable of handling an increasing number of QoS factors.

Conflict of Interests

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

Acknowledgment

The authors would also like to thank the Secretaria de Educacion de Mexico for the partial support through Grant PIFI-2013-31MSU0098J-14.

References

- [1] W3C Working Group, Web Services Architecture, 2004, <http://www.w3.org/TR/ws-arch/>.
- [2] V. X. Tran and H. Tsuji, "QoS based ranking for web Services: fuzzy approaches," in *Proceedings of the 4th International Conference on Next Generation Web Services Practices (NWeSP '08)*, pp. 77–82, Seoul, Republic of Korea, October 2008.

- [3] S.-Y. Hwang, E.-P. Lim, C.-H. Lee, and C.-H. Chen, "Dynamic Web service selection for reliable Web service composition," *IEEE Transactions on Services Computing*, vol. 1, no. 2, pp. 104–116, 2008.
- [4] D.-H. Shin, K.-H. Lee, and T. Suda, "Automated generation of composite web services based on functional semantics," *Journal of Web Semantics*, vol. 7, no. 4, pp. 332–343, 2009.
- [5] Y. Yan, P. Poizat, and L. Zhao, "Self-adaptive service composition through graphplan repair," in *Proceedings of the IEEE 8th International Conference on Web Services (ICWS '10)*, pp. 624–627, July 2010.
- [6] W. Jiang, S. Hu, D. Lee, S. Gong, and Z. Liu, "Continuous query for QoS-aware automatic service composition," in *Proceedings of the IEEE 19th International Conference on Web Services (ICWS '12)*, pp. 50–57, Honolulu, Hawaii, USA, June 2012.
- [7] Y. Feng, A. Veeramani, and R. Kanagasabai, "Automatic DAG-based service composition: a model checking approach," in *Proceedings of the IEEE 19th International Conference on Web Services (ICWS '12)*, June 2012.
- [8] Y. Yan, M. Chen, and Y. Yang, "Anytime QoS optimization over the PlanGraph for web service composition," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12)*, pp. 1968–1975, March 2012.
- [9] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for Web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, 2004.
- [10] D. Ardagna and B. Pernici, "Adaptive service composition in flexible processes," *IEEE Transactions on Software Engineering*, vol. 33, no. 6, pp. 369–384, 2007.
- [11] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for Web services selection with end-to-end QoS constraints," *ACM Transactions on the Web*, vol. 1, no. 1, article 6, 2007.
- [12] S.-C. Oh, D. Lee, and S. R. T. Kumara, "Effective Web service composition in diverse and large-scale service networks," *IEEE Transactions on Services Computing*, vol. 1, no. 1, pp. 15–32, 2008.
- [13] Y. Bo and Q. Zheng, "Semantic web service composition using graphplan," in *Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA '09)*, pp. 459–463, Xi'an, China, May 2009.
- [14] P. Rodriguez-Mier, M. Mucientes, and M. Lama, "Automatic web service composition with a heuristic-based search algorithm," in *Proceedings of the IEEE 9th International Conference on Web Services (ICWS '11)*, pp. 81–88, July 2011.
- [15] F. Qiqing, P. Xiaoming, L. Qinghua, and H. Yahui, "A global QoS optimizing web services selection algorithm based on MOACO for dynamic web service composition," in *Proceedings of the International Forum on Information Technology and Applications (IFITA '09)*, pp. 37–42, Chengdu, China, May 2009.
- [16] M. Oh, J. Baik, S. Kang, and H.-J. Choi, "An efficient approach for QoS-aware service selection based on a tree-based algorithm," in *Proceedings of the 17th IEEE/ACIS International Conference on Computer and Information Science (ICIS '08)*, pp. 605–610, IEEE, Portland, Ore, USA, May 2008.
- [17] P. Doshi, R. Goodwin, R. Akkiraju, and K. Verma, "Dynamic workflow composition using Markov decision processes," in *Proceedings of the IEEE International Conference on Web Services (ICWS '04)*, pp. 576–582, July 2004.
- [18] A. Gao, D. Yang, S. Tang, and M. Zhang, "Web service composition using Markov decision processes," in *Advances in Web-Age Information Management: Proceedings 6th International Conference, WAIM 2005, Hangzhou, China, October 11–13, 2005*, vol. 3739 of *Lecture Notes in Computer Science*, pp. 308–319, Springer, Berlin, Germany, 2005.
- [19] J. Harney and P. Doshi, "Selective querying for adapting web service compositions using the value of changed information," *IEEE Transactions on Services Computing*, vol. 1, no. 3, pp. 169–185, 2008.
- [20] K. Chen, J. Xu, and S. Reiff-Marganiec, "Markov-HTN planning approach to enhance flexibility of automatic web service composition," in *Proceedings of the IEEE International Conference on Web Services (ICWS '09)*, pp. 9–16, Los Angeles, Calif, USA, July 2009.
- [21] H. Wang, P. Tang, and P. Hung, "RLPLA: A reinforcement learning algorithm of web service composition with preference consideration," in *Proceedings of the IEEE Congress on Services Part II*, 2008.
- [22] H. Wang, X. Zhouy, X. Zhou, W. Liu, and W. Li, "Adaptive and dynamic service composition using Q-learning," in *Proceedings of the 22nd International Conference on Tools with Artificial Intelligence (ICTAI '10)*, pp. 145–152, Arras, France, October 2010.
- [23] V. Todica, M.-F. Vaida, and M. Cremene, "Formal verification in web services composition," in *Proceedings of the 18th IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR '12)*, pp. 195–200, May 2012.
- [24] L. Yu, W. Zhili, L. Meng, W. Jiang, and X.-S. Qiu, "Adaptive web services composition using Q-learning in cloud," in *Proceedings of the 9th IEEE World Congress on Services (SERVICES '13)*, pp. 393–396, Santa Clara, Calif, USA, July 2013.
- [25] R. S. Sutton and A. G. Barto, *Reinforcement Learning An Introduction*, The MIT Press, Cambridge, Mass, USA, 1998.
- [26] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, 1996.
- [27] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, Wiley-Interscience, 1994.
- [28] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 287–308, 2000.
- [29] C. Watkins, *Learning from delayed rewards [Ph.D. thesis]*, University of Cambridge, 1989.
- [30] T. Jaakkola, M. I. Jordan, and S. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Computation*, vol. 6, pp. 1185–1201, 1994.

Research Article

Integrating Reconfigurable Hardware-Based Grid for High Performance Computing

Julio Dondo Gazzano, Francisco Sanchez Molina, Fernando Rincon, and Juan Carlos López

Escuela Superior de Informatica, Universidad de Castilla-La Mancha, 13071 Ciudad Real, Spain

Correspondence should be addressed to Julio Dondo Gazzano; juliodaniel.dondo@uclm.es

Received 13 June 2014; Accepted 20 August 2014

Academic Editor: Shifei Ding

Copyright © 2015 Julio Dondo Gazzano et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

FPGAs have shown several characteristics that make them very attractive for high performance computing (HPC). The impressive speed-up factors that they are able to achieve, the reduced power consumption, and the easiness and flexibility of the design process with fast iterations between consecutive versions are examples of benefits obtained with their use. However, there are still some difficulties when using reconfigurable platforms as accelerator that need to be addressed: the need of an in-depth application study to identify potential acceleration, the lack of tools for the deployment of computational problems in distributed hardware platforms, and the low portability of components, among others. This work proposes a complete grid infrastructure for distributed high performance computing based on dynamically reconfigurable FPGAs. Besides, a set of services designed to facilitate the application deployment is described. An example application and a comparison with other hardware and software implementations are shown. Experimental results show that the proposed architecture offers encouraging advantages for deployment of high performance distributed applications simplifying development process.

1. Introduction

Scientific community is continuously increasing the demand for HPC [1]. Due to its nature, high performance applications are adapted to the existing computing capacity in order to achieve a good tradeoff between accuracy and processing time. Therefore, a system intended to provide a faster and accurate solution in HPC turns out of great interest [2].

FPGA-based scientific computation is one of the solutions for scientific community to improve the response time for numerically intensive computation [3]. Approaches for high performance reconfigurable computing (HPRC) integrate both processors and FPGAs into a parallel architecture. HPRC can achieve several orders of magnitude improvement in speed, size, and cost over conventional supercomputers [4, 5]. FPGAs offer high performance close to ASICs but, in contrast to these, FPGAs provide a high degree of flexibility similar to a general purpose computer. FPGA-based systems are faster than a pure software approach in terms of computational power [6]; therefore, an infrastructure allowing

for the exploitation of the high performance capability of a hardware implementation with the benefits of autonomous resource management is a promising alternative to be used in HPC.

However, the process needed to accelerate applications using reconfigurable hardware has required so far some expertise in hardware design in order to obtain maximum benefits from the hardware platform where the design is being implemented. People that use HPC mostly design their application by means of high-level languages such C, C++, but mapping an application onto hardware resources requires to describe it using hardware description languages (HDL). Although there are tools that help designers to obtain HDL code from C or C++ source code [7–9], to maximize the benefits of using hardware, it is necessary to have a good knowledge of hardware design. These benefits (speed, power, etc.) also depend on the target FPGA vendor. Furthermore, each FPGA vendor offers a set of features, such as primitives or dedicated hardware (DSP, processors, etc.), which help to obtain better results for specific architectures.

Besides the gap between Sw and Hw programming concepts that makes translation to hardware challenging, there are some aspects that have so far prevented the use of reconfigurable hardware-based platforms in HPC: the lack of tools for deployment of computational problems in a hardware platform usually leading to ad hoc structures, the lack of configurable or parameterizable hardware resources, and the lack of debugging methodologies and the low portability of components (unlike in software where the portability is high with the use of libraries), among others [10].

One way of bringing reconfigurable hardware-based platforms closer to HPC users is to develop a platform and a set of tools oriented to facilitate application partition and deployment and the integration and communication of heterogeneous components and to provide a complete platform management service that contemplate users and resources. In this regard, this work proposes the necessary infrastructure for the management and use of reconfigurable computational resources and the interfaces that are needed to obtain a system as generic as possible, increasing the possibilities of developers when creating compatible applications. Besides, a set of services provided to attain transparent user-application deployment has been developed. One of the key aspects is the use of the dynamic reconfiguration capabilities of FPGAs. Using dynamic reconfiguration allows for fine-grain modification of FPGA functionality, maintaining a set of services in the FPGA and changing only the area dedicated to user applications [11]. This work also provides the infrastructure for an efficient use or the partial reconfiguration capability of FPGAs.

This paper is structured as follows: related works are summarized in the next section; then a brief description of the architectural view of the platform is presented in Section 3. In Section 4, the infrastructure management system is presented where basic administration services are described. Next, in Section 5, the architecture computing nodes are defined, and partial reconfiguration issues are discussed. In Section 6, an example reconfigurable application design is detailed. Later, in Section 7, experimental results are presented and discussed. Finally, the last section presents conclusions.

2. Related Works

High performance computing clusters have evolved towards the inclusion of different kinds of hardware accelerators in order to overcome the limitations in the amount of parallelism achievable with commodity CPUs. Heterogeneous clustering is more efficient than homogeneous architectures, and tightly coupled accelerators help to reduce communication requirements by making use of data locality [12].

From the architectural point of view, accelerators can be included in the cluster using two different approaches: either as uniform node nonuniform system (UNNS) or as nonuniform node uniform systems (NNUS) [4]. In the first case, each node in the clusters includes only a single type of resources (i.e., only CPUs, GPUs, or FPGAs). Examples of this configuration are the SGI Altix servers [13], Netezza for

data warehouse applications [14], the Convey HCI and HC-lex hybrid computers [15], or the Cray XD1 [16], just to name a few of them. In some cases, FPGA technology is hidden behind an extended instruction where designated operations are accelerated in the hardware fabric. In other cases, FPGAs can only be accessed through a tightly coupled processor using a closed API. The main drawback of this approach is the communication overhead between processing elements (PE) and the need of special hardware for the integration of the accelerators with the communication backbone, which has a high impact in the final cost of the solution. In the NNUS approach, nodes are composed by a mixture of different kind of processing elements, but all of them include a similar set of characteristics, thus providing a homogeneous view of the cluster. One important advantage of the model is that it can take benefit of high-speed interconnection links between local PEs, providing much higher communication bandwidth without requiring special-purpose communications hardware. Also, the NNUS architecture is better suited for the single program multiple data (SPMD) paradigm, where a single copy of the program can be easily scaled to a multinode cluster.

Examples of NNUS nodes are the Axel [12] architecture or the commercial FPGA-based computer RYVIERA and COPACABAN platforms. SCIEngines [17] provides developers with a bare reconfigurable platform in which FPGAs resources are at the same level of processors. However, the development environment is not trivial for nonexpert hardware personnel.

The kind of systems described in the introductory paragraphs of this section falls into the category of the so-called heterogeneous computing platforms. Heterogeneous computing platforms are now in the heart of servers for HPC since the fall of single CPU clusters and the replacement of multicore-based systems. In the following, we disclose some of the most relevant works and existing technology for each one of the two main working areas where this work will contribute to the state of the art, namely, (1) FPGA integration techniques and (2) development model and tools.

2.1. Contributions to FPGA Integration Techniques. There are several reasons to integrate FPGAs also in HPC. The first and most obvious reason is performance. Secondly, and equally important, is the low power consumption. Finally, the last reason is the flexibility obtained when using FPGA in contrast to other hardware acceleration strategies used in HPC [18]. Most of the solutions in this concern place FPGAs as a simple coprocessor of a master entity (i.e., an on-board CPU) that typically runs a control program. FPGAs are, thus, surrogated to a lower level, behind the processor. This is the dominant role of FPGAs both in high performance embedded systems [19–23] and HPC servers.

Normally, this is done via FPGA PCI-Card solution, expanding the traditional computational node, and a producer-consumer computational model. This architectural solution allows easy technology adoption, but it has some drawbacks such as the communication overhead between hardware and software.

Examples of this configuration are [13–15] or [16], among others. In some cases, FPGA technology is hidden behind an extended instruction set where designated operations are accelerated in the hardware fabric. In other cases, FPGAs can only be accessed through a tightly coupled processor using a closed API.

Most of works using FPGA for HPC repeat the strategy of integrating an accelerator into applications to speed up the execution of the kernel of an algorithm [24, 25]. Nevertheless, this strategy is not intended to execute the whole application in hardware.

The approach presented in [26] represents an evolution with respect to the acceleration of a single algorithm. It offers an architecture where reprogrammable hardware resources can be used as if they were resources managed by the operating system, abstracting in this way user applications.

The proposed architecture is based on a card with partially reconfigurable FPGAs connected to the bus of a general purpose computer. This architecture loads those hardware components needed to accelerate an application, through a software layer that incorporates these FPGAs as if they were additional system resources. This work does not provide hardware communication transparency and replication services.

There are other proposals that use dynamic reconfiguration and on-the-fly bitstream relocation as the main features for the employed architecture. Into this group, one can find [27], where dynamically reconfigurable FPGAs are included. In this approach, reconfigurable areas are used to load application accelerators behaving as slaves. This fine grain reconfiguration allows a better use of resources and the possibility to launch more than one hardware core at the same time in the same FPGA. An inconvenient matter about this model is that communication from software to hardware provokes a reduction of performance when the hardware computation time is similar to the reconfiguration time. This work does not provide communication mechanisms between cards, limiting parallelization.

The Erlangen slot machine architecture [28] belongs to this group as well and proposes a system that exploits dynamic reconfiguration and relocation of bitstream to implement high performance AV stream applications. Their advantages are deployment transparency and communication inside the FPGA, but this architecture was not designed to scale to more than one device.

Another strategy used to accelerate an algorithm is the use of soft-core processors implemented in FPGAs, as those processors created using Mitrion C software [29, 30], which are customized and replicated automatically for each application. With this strategy, an increment of performance of the algorithm execution is obtained, but it is limited by Sw-Hw communication mechanisms. Its main advantage is related to application development due to the fact that developers use C-like software abstracting them from hardware details.

Most of these related works keep users tied to the physical architecture (Figure 12), so, the effort to take advantage of architecture specific resources needs to be done again when the architecture change (low portability). One solution is to use a logical architecture in order to provide designers

with the same architecture, abstracting them from physical architecture and providing also the infrastructure to perform mapping efficiently from logical to physical architectures. The main idea is to present an abstraction from computational resources to users, in order to keep them as far as possible from the underlying infrastructure, avoiding having a deep knowledge of the platform to perform an efficient application deployment.

2.2. Contributions to Development Models and Tools. One of the main problems with heterogeneous PEs integration is the very different nature of their architectures, which also affects the programming model typically used in each case. Regarding GPUs, considerable efforts have been done by vendors and the scientific community to provide high-level programming models and frameworks, such as CUDA [31] from Nvidia or, lately, the recent announcement of Altera [32] that uses OpenCL as the unique programming model for FPGAs, GPUs, and CPUs, and, as a result, they have been quickly adopted by mainstream software programmers. However, that is not exactly the case for FPGAs.

While FPGAs have proven to be very power efficient and good candidates for HPC, particularly for applications demanding fine-grain parallelism and nonstandard data sets, they have important drawbacks: they require highly specialized design skills, vendor toolchains are too focused on the optimization of the resulting circuits, but they provide little support for seamless integration of the solutions, and last but not least portability is still a challenge, even for different FPGA families of the same vendor. The research community has been working in high-level synthesis tools for more than a decade, in order to close the enormous productivity gap for FPGA design, and recently the release of several of these tools (such as Catapult-C [33] from Calypto, Vivado HLS [34] from Xilinx, or Symphony [35] from Synopsys) is beginning to show good expectative. However, these tools only address the problem of the hardware implementation of computational kernels, while there is still a long road to turn reconfigurable logic into a resource commodity.

Current state of the art in synthesis tools technology provides support for a big subset of the C language, excluding those aspects that cannot be easily mapped to a hardware implementation, such as dynamic memory management. Also the coding style and mapping rules have been simplified, broadening the community of users and not just targeting engineers with a hardware design background. It is only required to understand some basic concepts related to how compilers work. Most of these tools accept a plain C, C++, and SystemC description that can be shaped into a hardware implementation despite the definition of certain directives, such as the type of protocol to define the reception of the arguments, loop manipulation, or the identification of blocks that may benefit from a parallel implementation. The whole process of directive definition, synthesis, and results analysis takes no more than a few minutes, providing a mean to quickly explore the design space, which is in contrast to the several weeks that the same task would require following classic hardware design flows.

3. Model

The progress and expansion of distributed high performance application in software have been tied to computation platform development and their development models. These platforms have relied on (a) distributed systems technology that contributes to an interconnection middleware between clients and services, (b) object oriented modeling, providing design advantages through the data and functionality encapsulation, and (c) the movement toward clusters and grids facilitating the pooling and exploitation of computational resources.

An important aspect of each existing platform relies on its model. A model allows developers to take advantage of heterogeneous resources. The model will provide flexibility and tools, enabling the designer to develop applications. The model also gives information about the types of the existing computing elements, their interconnections, and their performance. The success of a platform is tied to the facility and the accuracy with which a real problem can be modeled or represented. Then, to facilitate the deployment of a distributed high performance computation problem, the adopted model must provide the advantages already provided by software approaches plus the benefits offered by reconfigurable hardware:

- (i) problem modeling facilities;
- (ii) automatic deployment;
- (iii) location transparency;
- (iv) communication transparency;
- (v) replication mechanisms.

Besides, the adopted model must specify several service levels depending on the expected application control degree, offering lower barriers and development facilities to both sides of HPC business: users and service providers. Furthermore, the adopted model should provide flexibility to dynamically compose available resources in order to optimally exploit the computational platform for a specific computational problem.

The platform model proposal presented in this paper, that we call reconfigurable grid (R-Grid), identifies a component model in the foundations of the solution to explore. This model will be supported by the physical platform and it will be the basement for application analysis, modelling, refactoring, and development.

The R-Grid platform model will ease the way application development is nowadays performed in FPGAs, particularly parallel applications, by means of this component model and domain specific libraries. The concept of component is technology independent. This makes the R-Grid proposal portable and not attached to the existing technology, allowing for the evolution of the concepts and techniques result of this research, to further and future technology generations.

In our approach, an application is constituted by a collection of components, whose nature depends on the adopted programming model and the chosen target hardware resource. The R-Grid platform model will facilitate and promote the use of the distributed reconfigurable computation

resources through domain specific primitives and components. In domains where artifacts as primitives and domain specific components already exist, they will be efficiently ported to the new specific hardware. In other cases, they will be inferred from the analysis of certain use cases and will be included in an available library.

R-Grid platform model supports different programming models such as those based on shared memory (openMP), remote method invocation (RMI), or messages passing interface (MPI).

4. Architecture Description

The infrastructure proposed in this paper gives the necessary support for transparent application deployment and automatic application management, in order to liberate users from architectural aspects during runtime.

This infrastructure was designed in three layers: the lowest layer is composed of reconfigurable resources on which user applications are instantiated; the second layer abstracts these resources providing a resource homogenization and facilitating application deployment; and the upper layer is used for resource and application management issues.

From a top-down perspective, the basic architecture of the platform comprises a central *management node* (upper layer) and a set of interconnected reconfigurable *computational nodes* with their corresponding communication adapters (second and lower layers). The management node is implemented in software and offers a set of basic services for resources and application management that will be described in detail later.

Computational nodes, on the other side, offer computational capacity (resources) and are composed of two main parts: one part consisting of a set of resources dedicated to accommodate user applications, and the other part includes a resource abstraction creating a homogeneous view to facilitate node integration.

These computational nodes can be either software nodes (processors) or hardware nodes (reconfigurable logic). Software computational nodes are built on a general purpose processor where applications are deployed in software, while hardware computational nodes are FPGAs with dynamically reconfigurable areas, where applications are implemented in hardware.

A scheme of the described infrastructure is depicted in Figure 1.

Both computational nodes and management nodes are interconnected through a high performance network. To obtain a homogeneous view of resources, a communication abstraction model based on the OOCE middleware [36] was used, where the communication between nodes is performed using remote method invocation (RMI). Through RMI, it is also possible to manage the deployment of applications as binary files, as well as to obtain information about the computational node type, the state of resources, and so forth. The OOCE middleware facilitates component adaptability and the communication between components using the client-server approach as it will be described later. For each

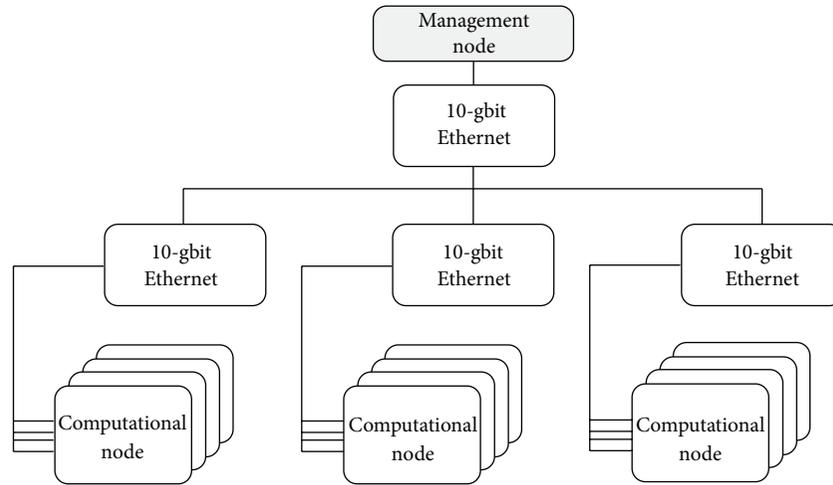


FIGURE 1: Infrastructure composed of several computational nodes managed by a management node.

client component, a proxy is added that represents the server component. If the client requires services from different servers a proxy for each one will be added to the client. Each proxy has the same interface as the correspondent server. Proxies translate the invocation sent to the servers into messages through the communication channel. In turn, a skeleton is added to each servant object in order to translate messages into invocations to the servant.

4.1. Management Node: The Infrastructure Management. The implementation of the management node is purely software. The technology used during development process was as follows: JAVA was used as a programming language, OOCE used as communication middleware, JDBC for database access, and SQLite as database.

Platform management aspects are covered by a set of basic services offered by the management node. These services are a key part of the system and are defined to facilitate the exploitation of computational resources in a simple and transparent way. These services are essential to facilitate both the deployment of applications and the use of the reconfigurable platform. The following subsections describe each one of these basic services.

4.1.1. Application Registry Service. The application registry service is the first service that a user employs.

This service allows the clients to manage their application repository. The repository is the only source of application supported by R-Grid so clients as a previous step to execution must include their application in the repository. The repository stores structural and binary information of applications.

Before delving into the application registry service, it is necessary to briefly describe the application model that better fits with the proposed infrastructure. This application model is intended to meet those requirements that can appear due to the use of not only software computational nodes but also dynamic reconfigurable hardware resources.

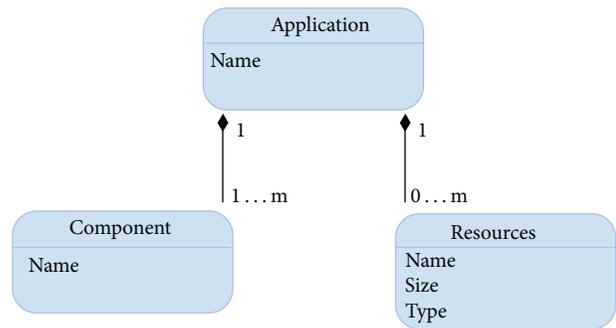


FIGURE 2: Application model.

Applications can be seen as a set of components interacting with each other to reach the solution of a certain problem using specific resources (Figure 2).

Each component is defined as a programmable unit to be instantiated in a computational node resource. Examples of application configurations are shown in Figure 3. Figure 3(a) describes an application composed by a controller interacting with several slave components. In Figure 3(b), an example of application composed by chained components is shown.

The application structural information contains data about the name of the application, its components, and the required resources defined in the application model. The set of all this information is called *application descriptor*.

The application registry service is in charge of collecting information about the application composition, that is, the application identifier, the components of the application, and the component binary files associated with a specific computational node model. To select the computational node model that better fits with the application to be deployed, the management node offers a list of computational nodes with different characteristics (i.e., memory bandwidth, area, maximum clock frequency, etc.) to the users. Users select the specific computational node and generate the corresponding

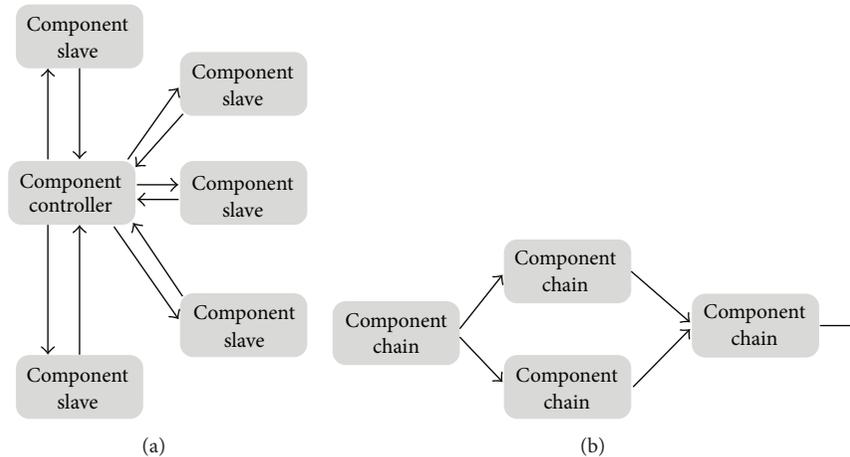


FIGURE 3: Applications schemes.

binary file for the chosen resource. The application identifier must be unique for each user. Likewise, each component has a unique component identifier in order to be reached by the remaining components of the application, independently of their locations.

4.1.2. Application Deployment Service. Once the application has been registered, the user can request the deployment of the application.

The deployment request triggers the loading of the binary file associated with the selected resource model contained in a computational node already registered in the previous step.

Besides the fact that the binary files have been created for a specific type of resource, the management node uses a set of metrics to select the proper computational node between all those that can hold the same bitstream for the given application constraints (bandwidth, memory, etc.). For example, in order to maximize the bandwidth, the objective of the management node is to deploy all components in the same computational node (i.e., in the same FPGA). Once the computational node has been selected, a binary file is transferred for resource configuration.

4.1.3. Application Location Service. As stated before, applications are defined as a set of collaborative components that can be instantiated in different distributed resources using several computational nodes. Then, to have a correct application execution, a transparent communication between local and remote application components is necessary.

To facilitate the access to local and/or remote components, the application location service provides location information for the application components. This is a hierarchical service composed of a global locator running in the management node and a local locator implemented in each FPGA. The global locator keeps location information about all components of an application that are deployed in the system, while local locator keeps information of only locally deployed components (in the same FPGA or processor). Assuming that the location of a component can vary during

the lifetime of the application depending on the priority of use of free resources, may occur that a component need to be moved either inside the same computational node or to another FPGA, in order to comply with some application restrictions (i.e., performance). Each time a component is deployed, the location service is updated. Then, if a component that has been moved needs to be reached, the request is redirected to the local locator first to obtain the new local address of the component and in case the component has been moved to a different FPGA the request is then redirected to the global locator which is provided with the new component address. This indirection process occurs only when a component changes its location. Once the component location is identified, the corresponding proxy is updated for further invocations. The location service uses a data structure that includes a component identifier as a primary key and the content associated with that key is the physical address of the component.

4.1.4. Data Structure Implementation. Data structure is a key part of management node. The management node will receive a plethora of concurrent request and has to maintain all information about system state in a safe and coherent way. The structure of data has been implemented through a relational database and a file system. The database will keep the dynamic information of system while the file system will store the application binary files.

The dynamic information of the system will include registered users, their applications, the state of each application, the pending actions to be performed (activation, stop), and the existent resources and their states. All this information follows the scheme represented in Figure 4.

4.2. Computational Node: The Resources Infrastructure. Computational nodes have two objectives. The first one is to offer their internal resources to the management node for application deployment. The second objective is to provide a homogeneous view independently of their nature (hardware or software), in order to allow the management node to have

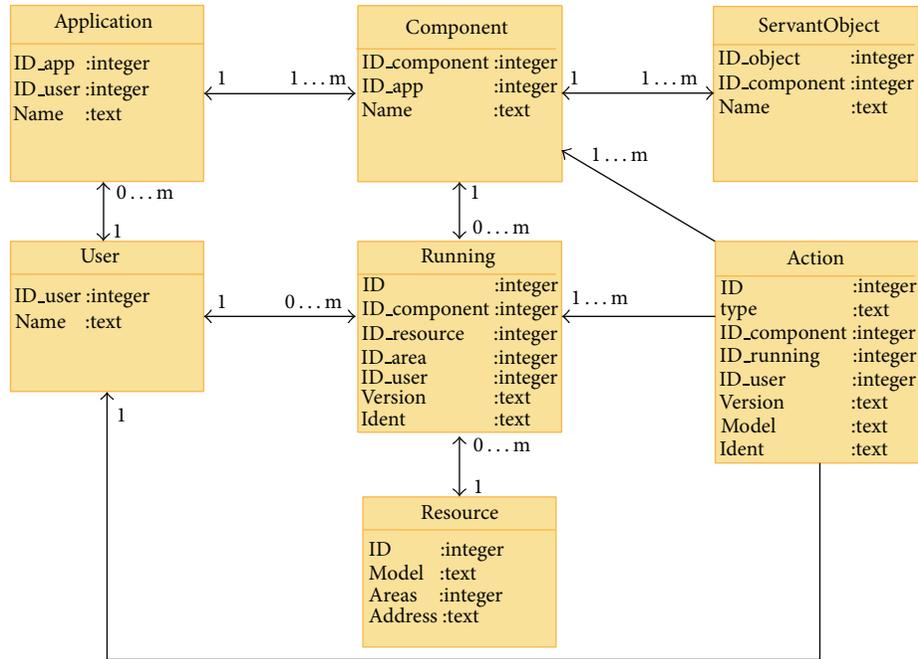


FIGURE 4: Database relational diagram.

a common treatment for all of them independently of their model. The only requirement is that all computational node models must offer the same interface.

When hardware, the computational node can be formed by one or by a set of reconfigurable areas, depending on the FPGA model. Each one of these areas, named dynamic area (Figure 5), is a resource that can be configured to contain a functional component of the user application. In case of dynamically reconfigurable FPGAs, all reconfigurable areas inside the same FPGA are defined of the same size and with the same amount of internal resources. This allows for the relocation of an instantiated component from one dynamic area to a different one in the same computational node or relocated in a different FPGA of similar characteristic. A scheme of the computational node is depicted in Figure 5. The amount and size of dynamic areas depend on the size and type of FPGAs; therefore, the architectural model, presented in this paper, combines computational resources of different characteristics in order to offer a wide range of reconfigurable resources for application deployment. Users can choose specific resources for their application selecting them from those offered by the management node as detailed in Section 4.1.

Dynamic areas have two interfaces, one dedicated to communicate either with the configuration kernel or with the rest of components belonging to the same application and the other one dedicated to provide access to local memory. The component named memory access interface allows the sharing of memory resources between all dynamic areas dividing space memory into spaces of local memory for each dynamic area separately.

The configuration of each dynamic area is performed by the configuration kernel component. This component provides three kinds of different services as follows:

- (a) reconfiguration service which is in charge of partial bitstream management during application deployment; this service is invoked only by the management node when a user requires application deployment;
- (b) location service that is part of the application location service provided by the management node as described in Section 4.1; this is an internal service whose main objective is to accelerate internal location consulting; this service is used to locate a component inside the computational node;
- (c) self-discovery service that allows FPGA to be discovered by the management node with the startup of the FPGA.

Self-discovery service is a key functionality for resource management. When connected, each FPGA will announce their own features (model, size), amount of resources, and type of areas to place components. This announcement is performed during the startup of the FPGA and the execution of the three described services is managed by the service manager component. It provides an interface offering methods such as *get_device_model*, *get_device_free_resources*, and *deploy_bitstream*, to configure resources. Through this interface, the management node can query each configuration kernel component of each FPGA about the computational node model or about the amount of free resources to launch a reconfiguration process in a specific dynamic area (Figure 16).

The network adapter component, in turn, translates the incoming messages into invocations to the configuration kernel or to the components placed in dynamic areas and composes internal invocations into messages to the outside.

4.2.1. Communication Model. One of the main characteristics of the grid systems is the possibility of task relocation according to several factors such as the amount of free resources, the policy employed in task distribution, or specific user requirements among others. To facilitate portability of components between different computational node models, a common communication interface (resource abstraction) is provided by the platform. As briefly introduced in Section 4, the communication model used in this work is based on the OOCE middleware. When using OOCE middleware, each component is the sum of both the functional core plus the corresponding adapters, proxy, or skeleton, depending on the type of functional core. In this way, a client will be formed by the client core plus the proxy of the server that will attend the clients invocation, and the server will be formed by the server core plus the skeleton. The proxy will translate the invocation to the communication channel protocol and, on the other side, the skeleton will translate from communication channel protocol to server invocations. Using this communication mechanism, the invocation between two remote components can be seen as a local invocation. This allows components to be adapted to the corresponding communication infrastructure isolating the functionality from the communication channel. The generation of these adapters to the corresponding communication channel is performed through the use of R-Grid tools from a description of the corresponding component interface.

This approach allows three degrees of transparency: (a) location transparency, because the client sees the server interface as if it was a local invocation, (b) access transparency because it is possible to reach the server independently of its implementation (hardware or software), and (c) communication transparency because this middleware can be used for any communication channel.

4.2.2. Network Protocol Requisites. Classical cluster system generally uses network protocols such as TCP/IP, but this is a stack that is not easy to implement in the hardware. For that reason, a specific protocol based on TCP/IP has been developed for the R-Grid infrastructure. This protocol should comply with a set of requisites in order to have a very easy hardware implementation and with a very low communication overhead.

- (i) Global system addressing: the first requisite in R-Grid is to have a transparent communication between components independently of the nature of the computational node they are placed.
- (ii) Orthogonal communication: the second requisite is to facilitate the access to either the component of an application or to the system services, using the same message format for both cases.

- (iii) Bidirectional communication: the third requisite is to have a bidirectional communication allowing the return of a value or a confirmation of completed operation or error messages.
- (iv) Safe communication: the protocol should also provide security communication issues in order to prevent the access to applications that do not belong to the corresponding user or the access of components from one application to another one.
- (v) Broadcast communication: the protocol should provide broadcast messages.

Starting from these requisites, the network protocol will be divided into a set of layers as depicted in Figure 6. These layers and the fields that each layer introduces into the message are described as follows.

Physical Layer. This layer makes upper layers independent of transmission media. Physical media used in R-Grid go from Ethernet till point to point communication RocketIO, or SATA. For Ethernet the physical layer has a field with source and destination MAC address plus a field to identify type of message.

Network Layer. This layer implements addressing functionality. It will be used to transport messages from a specific computational node to another inside the system.

DST Node. The *DST node* field indicates the destination computational node. It uses three special addresses that will not change in any R-Grid system:

- (i) address 0 indicating its own node;
- (ii) first assignable address for the management node;
- (iii) last assignable address for broadcasting messages.

SRC Node. Equivalent to *DST node* field, this field stores information about the source of the message.

Total Size. It indicates the size of the *payload* in this layer.

ID. It allows the identification of a message between different transactions. It is used to deal with message fragmentation.

Frag. This field contains information for fragmentation process control.

Transport Layer. This layer implements the multiplexing of functionalities inside computational node allowing hoosing a service or a deployed application component.

DST Area. The *DST area* field indicates which component in the computational node is the destination of the message. It uses two special addresses:

- (i) address (0) indicates services zone in the computational node;

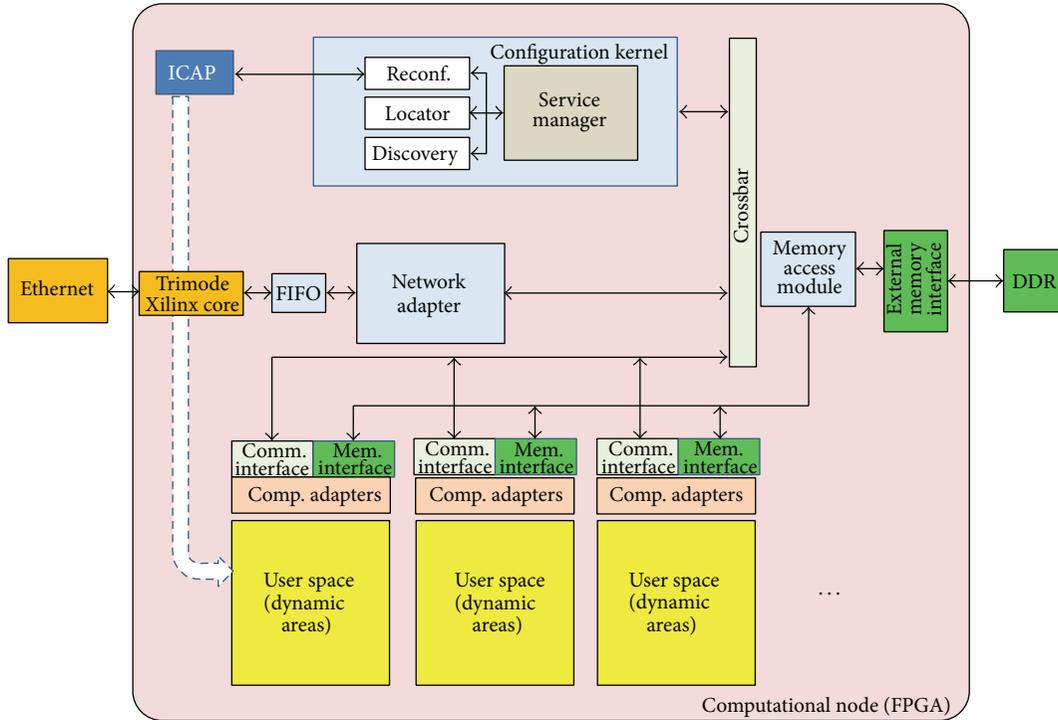


FIGURE 5: Computational node architecture.

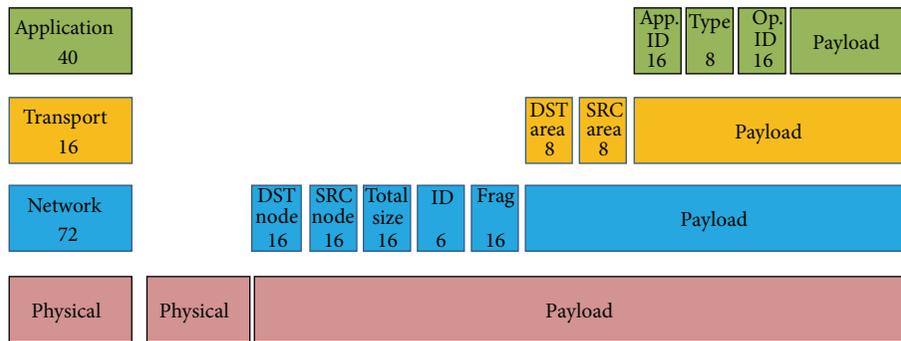


FIGURE 6: Network protocol layers.

(ii) the last assignable address indicates all deployed components.

SRC Area. The *DST area* field stores information about source component.

Op. ID. This field indicates the operation or functionality of the invoked component. If the functionality is a service, this field indicates the specific required service. The list of the operations corresponding to the platform services is described in Table 1.

Application Layer. This layer implements information about operations and type of message.

App. ID. This field provides the application ID. This value is unique for each deployed user’s application. All components of the same application will have the same ID. This field allows verification if a message received by a component belongs to the same application.

Type. It indicates if the message is a shipment, a reception, or an exception.

4.3. Partial Reconfiguration Infrastructure. The reconfiguration process starts when the management node sends an invocation to the configuration kernel to deploy a bitstream. The configuration kernel uses the internal configuration mechanism of FPGAs. Xilinx FPGAs have a component named internal configuration access port (ICAP) to perform the configuration of a partial region of the FPGA. The ICAP provides access to the FPGA configuration interface and the configuration registers performing a partial reconfiguration of the FPGA. The configuration kernel has been designed to

TABLE 1: Platform services and their corresponding op. ID.

OP. ID	Service description
0	Component registration service: it is the component registration in local locator
1	Location service: it gives the address of an application component
2	Deployment service: it deploys the binary file which is sent in the parameter field
3	Component start service: it sends a signal to start a component
4	Stop service: it stops a component that is running
5	Node ID update service: it changes the node ID at node network level
6	Discovery message service: when this service is selected, the resource announcement message is in the parameter field
7	Free resources available description service: it describes the number of node free resources

deal with the internal reconfiguration process of the ICAP. The configuration kernel acting as a dedicated DMA is an asynchronous component that reads the bitstream from memory and transfers it to the ICAP. This is done with burst transactions, allowing for fast partial reconfiguration of the FPGA without intervention of a processor. The ICAP receives the bitstream from the configuration kernel and loads it into the configuration register of the FPGA. This is the only component in the proposed infrastructure that is technologically dependent.

Besides, the configuration kernel updates the location information with valid component endpoints in the location service when necessary.

As dynamic areas in the same FPGA are defined with the same shape and size, the bitstream can be modified to be used in different areas [37]. The resources in terms of logic involved in the implementation of the configuration kernel and time overhead during partial bitstream reconfiguration are evaluated in Section 6.

5. Designing a Reconfigurable Application

The application design is determined mainly by its functionality, which is divided into small pieces and programmed in several components. Each one of these components is assigned to a *dynamic area*. Then, an application is composed of a set of partial bitstreams where each partial bitstream represents a component of the application.

A component consists of the functional core itself plus two interfaces. One interface is named communication interface and performs the communication to the rest of the application. The second interface is a proxy to memory, a memory interface.

The memory interface is a generic technology that is an independent description of the capabilities of a memory: read and write operations of a single word or a sequence of words. Such a description is a logical representation of real memories in the system. Any memory block in the system can

be modeled with the memory interface and used by means of its proxy, while, from the implementation point of view, we are simply reading and writing to a certain address computed from a base (the proxy reference) plus an offset, the address specified in the methods.

Components can have a passive role, and that means the component is waiting for an incoming invocation to process data and return a result or can have an active role invoking other components.

R-Grid offers to application developers a lot of freedom in modeling their application architecture. The two basic rules are the explicit intertask communication and the distributed memory. In R-Grid architecture hardware tasks can communicate each other without the action of a host; this avoids host-coprocessor architecture bottleneck existing in some HPRCs.

Another advantage is that developers can implement and deploy their code without restrictions because each task runs in a hardware sandbox. This freedom is compatible with a library of implemented common tasks. In this way, the developer can choose tasks from library or implement their own design.

In Figure 7, we can see briefly the application development workflow proposed by R-Grid.

It has five phases from the applications analysis to the application execution. The first phase corresponds to the selection of the programming model. Once the application has been designed according to the selected programming model, the next step is the partition of the application into a set of cores as subtasks. The next phase has two parts, one consisting in the generation of the corresponding stubs or adapters to form the component (core plus adapter) that will be implemented in hardware, plus the generation of the corresponding partial bitstream. These bitstreams will be generated according to the model of the selected FPGA. In this stage, the application descriptor is created. In Phase 4, the management node comes into the game when application and binary files are registered. Finally, the management node deploys the application in R-Grid architecture.

The synthesis process and the creation of the binaries files are performed using the tools provided by FPGA vendors. For computational nodes based on Xilinx FPGAs, we used Xilinx tools with support of dynamic reconfiguration. In order to simplify the synthesis process, R-Grid is provided with a project template containing constraint files, communication interfaces, and the element forming the static part of the FPGA, plus the initial partial bitstreams.

In order to choose an example that can reflect the needs of most scientific applications like those requiring solutions of several numerical linear algebra problems, such as eigenvalue problems, linear least square problems, and linear system of equations, among others [38], a matrix multiplier application is chosen as an example to be implemented. This example has two objectives: the first one is to illustrate how a high performance application is implemented in this infrastructure. The second objective is to test the platform.

This matrix multiplier is based on three functional cores: the data controller, the multiplier, and data storage. The *data controller* is in charge of matrix partitioning to perform

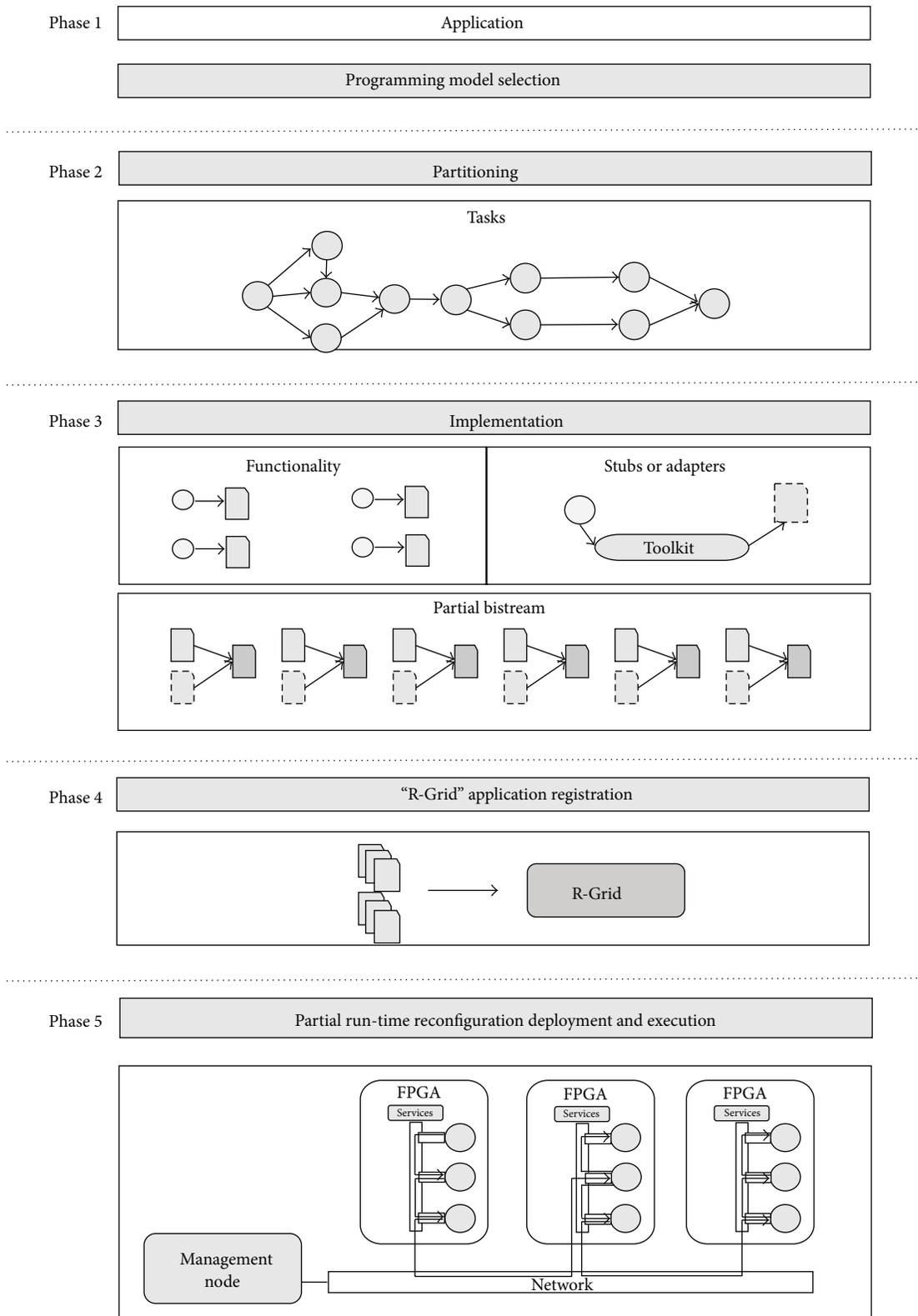


FIGURE 7: Application development flow.

calculus in parallel. The controller divides the multiplication process by the number of multiplier components involved. *Multipliers* are in charge of calculating partial results of the matrix partition assigned by the data controller. Finally,

the *data storage* functional core integrates partial results to create the result matrix.

Once the data controller, multiplier, and data storage functional cores have been developed, it is necessary to

TABLE 2: Resources used in communication mechanism in computational node.

Component	Slices	LUTs	FFs	Freq.
Network stack	1788	1380	2790	127.6 MHz
Adapters	624	643	981	186.4 MHz
Bus	284	210	335	220 MHz

add both of the network and memory interfaces to each functional core to create the corresponding component.

As stated before, interfaces (proxies and skeletons) to the communication channel and to the local memory are automatically generated using R-Grid tools.

To synthesize each component, a project template is provided that includes partial bitstream of target FPGA. The template is copied for each component of the application. In each copy of the template, the functionality source code and the autogenerated adapters code of the corresponding component are included forming each final component that will be instantiated in the computational node. Finally, a TOP design including designed components is included. Once components have been generated, they are synthesized and bitstreams are generated. Components are ready to be instantiated into dynamic areas.

The next step is the creation of the application descriptor in order to proceed with the application registry in R-Grid. The application descriptor is described in Algorithm 1 (Application Descriptor example).

Those bitstreams are loaded by the configuration kernel in the specified dynamic areas.

After application has been deployed, data controller assigns the corresponding data to be processed to the local memory associated with each multiplier deployed. After completion, the data controller sends an invocation to each multiplier component to start the multiplication process.

6. Experiments and Analysis

To test and validate the infrastructure and the described services presented in this work, several experiments have been developed.

6.1. Computational Node Implementation Analysis. The analysis of the implementation of a computational node has the following objectives: characterization of communication in terms of logical resources used in communication mechanism and available bandwidth; analysis of deployment time; and the characterization of location process in terms of involved resources and the time consumed in location process.

6.1.1. Characterization of Communication. Table 2 shows the amount of resources involved in the communication mechanism of the computational node in a Virtex 5 FX70T FPGA.

Regarding communication performance, the first analysis was the evaluation of the efficiency of R-Grid protocol. The

TABLE 3: Logical resources used by deployment service.

Slices	LUTs	FFs	Frequency
376	631	755	234.9 MHz

TABLE 4: Time consumed during location process.

Location process	Time
C to A	230 nsec
C to B	720 μ sec

results are depicted in Figure 8 with respect to the size of communication packets. In the graph, it can be observed that the header of the message has low overhead from 90 bytes, reaching an efficiency of 75%. Figure 9 shows the time consumed during different size packets transmission in each part of the communication mechanism of the computational node. The reception and transmission time (delivery time) represent the minimum time obtained if the network could satisfy the internal bandwidth. Processing time represents the time consumed in packet processing in a component performing echo function. Physical media time represents the transmission time for a 1 Gb Ethernet network. Total time includes reception, transmission, and processing time. As can be observed, the network communication time is a limiting factor to deliver data between remote processing elements.

6.1.2. Characterization of the Deployment Process. This test case has the objective of evaluating the reconfiguration time using R-Grid configuration kernel. In Figure 10, the involved blocks during deployment process and the time consumed in the reconfiguration of a 130 KB size bitstream, measured at each involved stage: request, bitstream transference from server to computational node, and computational node configuration, can be seen.

Table 3 shows the amount of resources involved in the deployment process.

6.1.3. Characterization of the Location Process. This test case was intended to evaluate the time consumed in the location process described in Section 4.1.3. The test consists in the location process of components A and B requested by component C. In this test, component A is placed in the same computational node of component C while component B is placed in a different node in order to evaluate local and global location process, respectively. Table 4 summarizes the time consumed in both cases. In the first case, the location service is provided by the local locator whereas the global locator is consulted in the second one.

6.2. Matrix Multiplication: SPMD and RMI Programming Model. The next experiment consists in the implementation of a matrix multiplier in the platform to analyze the viability of the proposal, evaluating whether the services are correctly implemented and the benefits are obtained in the complete application deployment process. This experiment was performed following the steps indicated next. First,

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE RGridApplication SYSTEM "RGridAPP.dtd">
<RGridApplication name="Matrix Multiplication">
  <component name="Data Controller">
  </component>
  <component name="Multiplier">
  </component>
  <component name="Data Storage">
  </component>
</RGridApplication>

```

ALGORITHM 1: Example of an Application Descriptor.

```

void stoAArray (int m, int p, byte [m x p] data);
void stoBArray (int p, int n, byte [p x n] data);
void Calculate (int ProcessingElements);

```

ALGORITHM 2: Data Controller Role Description.

a programming model for application development was chosen. Then, an analytical study about application needs and the necessary FPGA capabilities in terms of amount of logic, memory capacity, and bandwidth was done in order to check viability of the selected resources. The next step is to implement the application and the evaluation of resources. The last step consisted of evaluating whether the performance obtained is competitive compared with a traditional software solution.

The matrix multiplication example was developed following the SPMD (single process, multiple data) technique, using RMI as communication mechanism, and distributed memory.

A simple application architecture was proposed where three main roles were defined (see Figure 11).

- (i) Data controller role: this role is responsible for data reception. Once received, data is partitioned and sent to computing elements. In this experiment, the data controller role was implemented in the data controller component.
- (ii) Processor role: this role is responsible for data processing. This role can be played by several components placed in one or several nodes (multiplier components). Once data are processed, they are sent to the next player.
- (iii) Data storage role: it receives the processed data and it is responsible for combining the final data from the received partial data. This role is performed by the data storage component.

6.2.1. Data Controller Role. This role has several methods as indicated in Algorithm 2 (Data Controller Role description).

The first action is to store each operand matrix. This is performed through *stoAArray* and *stoBArray* methods, where *int m*, *int p*, and *int n* parameters indicate matrix dimension (rows and columns). Data partitioning is coordinated through *Calculate* method. This method is invoked indicating the amount of processing elements that will be used for matrix multiplication defined by the *ProcessingElements* parameter.

M, *P*, *N*, and *ProcessingElements* parameters are used to determine submatrix partitioning process.

6.2.2. Processor Role. Processing role was divided into two parts. One is devoted to store partial data received from data controller and the other is for multiplication process. The *stoSubMatrix* method stores partial data in local memory starting from the memory address indicated by *pos* parameter. The parameters *sizeXp* indicates the total amount of submatrix elements, and *Number_of_vectors* indicates the number of data vector sent to the processing element, whereas *data* parameter indicates the value of the corresponding matrix element (see Algorithm 3 Processor Role description).

To start multiplication process, data controller invokes *multSubMatrix* method where *posi* and *posj* parameters denote the position of each operand (already given by the *stoSubMatrix* method), the *size* denotes how many times the operation must be done, and the last two parameters *resultX* and *resultY* indicate the position, in the result matrix, where the result of the multiplication must be saved.

6.2.3. Storage Role. The storage role has methods for result matrix initialization, for obtaining data, and for data writing (see Algorithm 4 Storage Role description).

6.3. Physical Capacities Evaluation. As hardware devices two Xilinx FPGAs were selected: Virtex 5 VLX110T and VLX220T, with a DDR2 Micron MT8HTF12864HDZ-800 memory, 1 GB of capacity, and a peak transfer rate of 6400 MB/s, the model of the application corresponds to that shown in Figure 3(a), where the controller distributes matrix operand

```
void stoSubMatrix (int pos, int SizeXp, int number_of_vectors, byte[] data);
void multSubMatrix (int posi, int posj, int size, int resultX, int resultY);
```

ALGORITHM 3: Processor Role Description.

```
void setArray (int sizeX, int sizeY);
void putData (int x, int y, byte data);
byte[] getData (int x, int y, int size);
```

ALGORITHM 4: Storage Role Description.

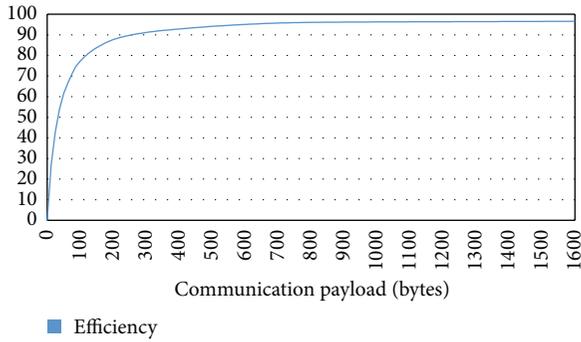


FIGURE 8: Efficiency of R-Grid protocol.

elements to computational kernels to perform the calculus in parallel. For that in each FPGA, four components were implemented.

6.3.1. Logic Requirements. Matrix multiplication is defined in (1), where A is a $M \times P$ matrix, B is a $P \times N$ matrix, and C is a $M \times N$ matrix:

$$A \times B = C, \quad (1)$$

$$C_{ij} = [AB]_{ij} = \sum_{k=1}^P A_{ik} * B_{kj}.$$

To define the size of reconfigurable area for matrix multiplier component implementation, we start developing a parameterizable matrix multiplier component. This component is a computational kernel that solves a $Q \times Q$ submatrix of the whole matrix of results. The computational kernel can solve $Q \times Q$ values per cycle. Input data are the Q rows and Q columns of the original matrix operands.

The computational kernel was implemented using 1×1 , 2×2 , 3×3 , and 4×4 submatrix sizes. These implementations were performed in two versions: one using DSP macros and the other without DSP macros.

Dynamic areas were defined using Xilinx PlanAhead tool. Each area occupies one clock region following vendor's advice. Both FPGA models have 16 different clock regions; therefore, each dynamic area offers 1/16 part of total resources.

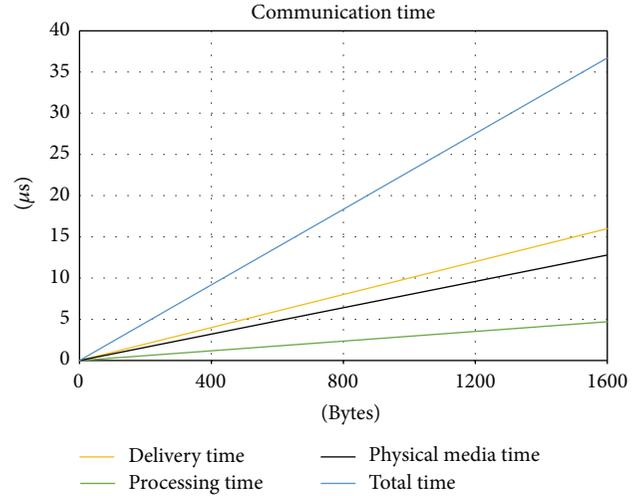


FIGURE 9: Communication time in each part of the communication process.

In this experiment, eight reconfigurable areas were defined to implement up to 8 components.

The synthesis data report is summarized in Figure 13, where $Total\ 110T$ and $Total\ 220T$ represent the total of available resources (normalized to 110t) per dynamic area defined in the *Virtex 5 VLX110T* and the *VLX220T* models, respectively. In this graphic, it can be observed that all computational kernel configurations can be implemented in defined dynamic areas, except 3×3 and 4×4 model using DSP in *Virtex 5 VLX110T*,

Now, the reconfiguration time is evaluated. The reconfiguration time depends on partial bitstream size and the reconfiguration latency.

The configuration kernel includes a hardware FIFO and it is optimized for burst reads and writes from the DDR memory to the ICAP controller. *Virtex 5* ICAP controller supports 32 bits/clock cycle bandwidth. As a result, the latency does not suffer from the memory *I/O* bottleneck and is completely delimited by the ICAP reprogramming latency; thus, it is near the technological limit of the device. In this experiment, the size of each partial bitstream for *Virtex 5 110T* FPGA model is of 233 KB and the reconfiguration time using configuration kernel is $596.5\ \mu\text{sec}$.

6.3.2. Memory Bandwidth. The second analysis carried out was to determine whether the available memory resources are sufficient for each computational kernel. For this, it is necessary to calculate the amount of memory and the memory bandwidth needed by each model of the computational kernel.

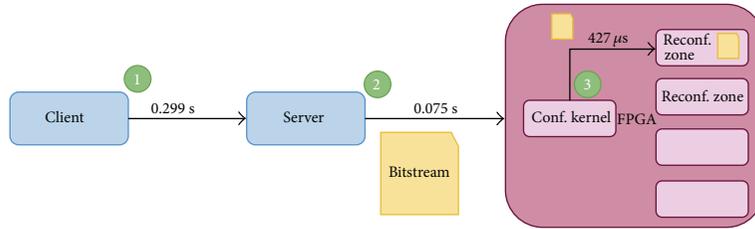


FIGURE 10: Reconfiguration time.

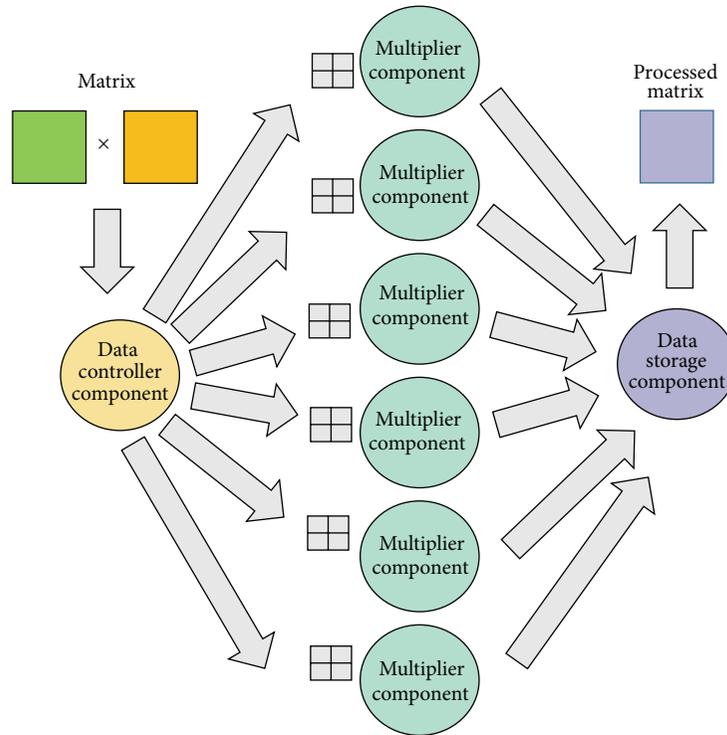


FIGURE 11: Logical architecture defined for the experiment. These three roles compose the logical architecture where the number of processing elements (multipliers) is variable.

Memory bandwidth is defined by the kernel size, and by the matrix element size, as it can be observed in the following equation:

$$\text{Bandwidth} = \text{Kernel}_{\text{size}} * \text{Data}_{\text{width}} * 2. \quad (2)$$

Figure 14 represents memory bandwidth requirements for 1×1 , 2×2 , 3×3 , and 4×4 kernel size, for two sizes of matrix element: 16 bits and 32 bits. As a reference, the total memory bandwidth supported by Virtex 5 was represented.

As it can be observed for a 16-bit-wide matrix element, the memory bandwidth is sufficient for all sizes of computational kernel, being optimum 4×4 computational kernel. For a 32-bit-wide of matrix elements, only 1×1 or 2×2 computational kernel can be used; otherwise, the memory bandwidth is not enough to feed the computational needs of the kernel.

6.3.3. *Memory Capacity.* The need of storage space for the resultant matrix depends on the size of matrix operands, the

width of matrix element, and the size of the kernel adopted. The memory space for the resultant matrix is defined by M_{total} while the memory space reserved for component operation is defined by $M_{\text{component}}$ just the way it is described in formula (3), where M , P , and N represent matrix dimensions:

$$M_{\text{total}} = (M * P + P * N + M * N) * \text{Data}_{\text{width}}, \quad (3)$$

$$M_{\text{component}} = (2 * P * \text{Kernel}_{\text{size}}) * \text{Data}_{\text{width}}.$$

Figure 15 represents the memory capacity requirements with respect to kernel size. For each component, 128 MB of memory space was reserved, allowing for the storage of the whole matrix.

6.4. *Performance Evaluation.* Once the memory space was defined, the next stage consists in performance evaluation for a 5000×5000 16-bit-width elements matrix multiplication, taking into account the computational kernel size and the

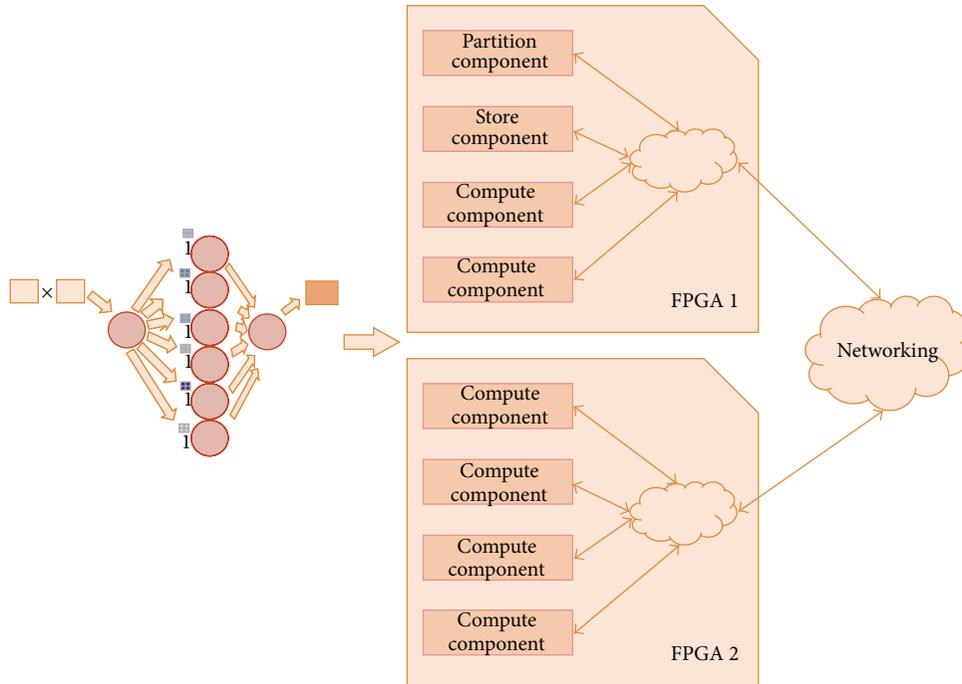


FIGURE 12: Physical architecture.

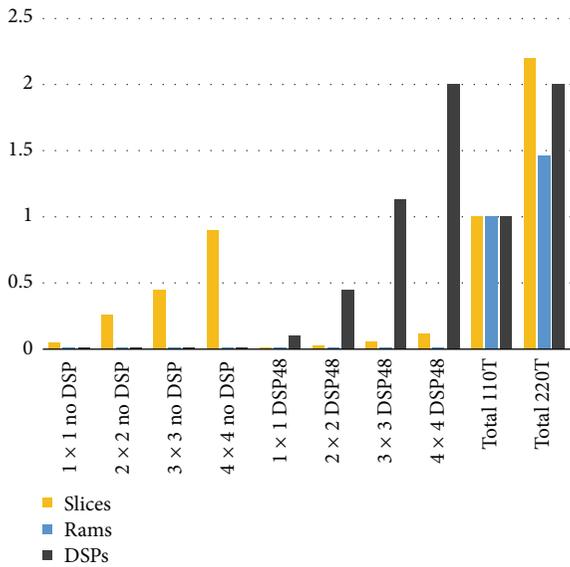


FIGURE 13: Logic resources requirement (normalized 110T).

amount of replicated computational units used. The result can be observed in Figure 17. This graphic shows time consumed (in seconds) during matrix multiplication using several hardware approaches (different kernel size) and a software solution.

Computation time needed by one kernel is determined by the size of the original matrix and the size of the kernel. Each 1 x 1 kernel can perform a multiplication of one column element and one row element and the result is added with the

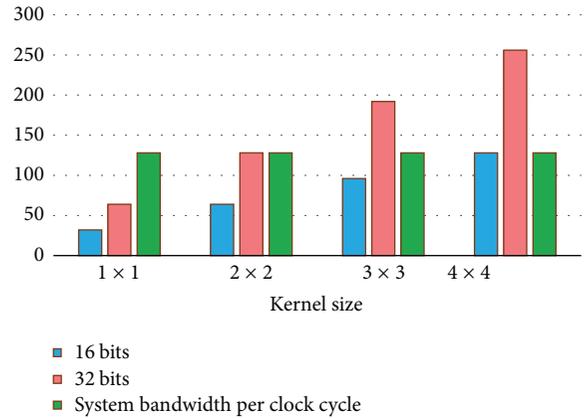


FIGURE 14: Memory bandwidth requirements per cycle.

accumulated value in the kernel in one clock cycle. Then, the total time, in clock cycles, consumed is equal to the number of columns of matrix *A* or the number of rows of matrix *B*. Once this time is known, the next step is to determine how many times this operation needs to be done, as it is defined in formula (4), where *M*, *P*, and *N* represent matrix dimensions:

$$\text{Time}_{\text{component}} = P,$$

$$\text{Time}_{\text{total}} = P * \frac{M * N}{\text{Kernel}_{\text{size}}^2}. \tag{4}$$

In order to attain an execution time very close to the computation time calculated above, it is necessary to hide the data transference time with the computation time of each

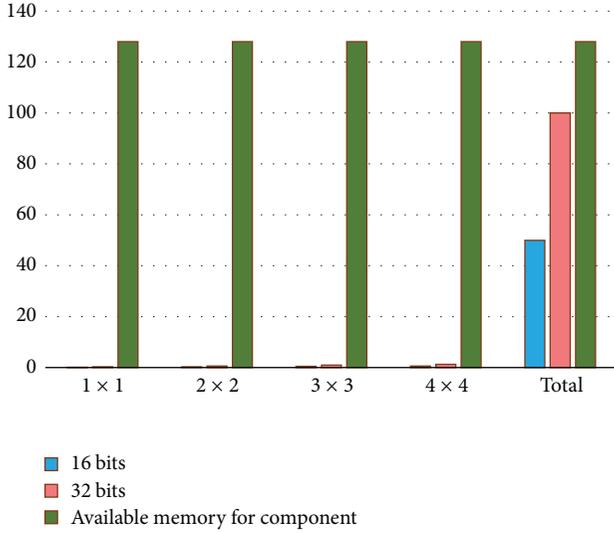


FIGURE 15: Storage requirements (matrix size 5000 x 5000).

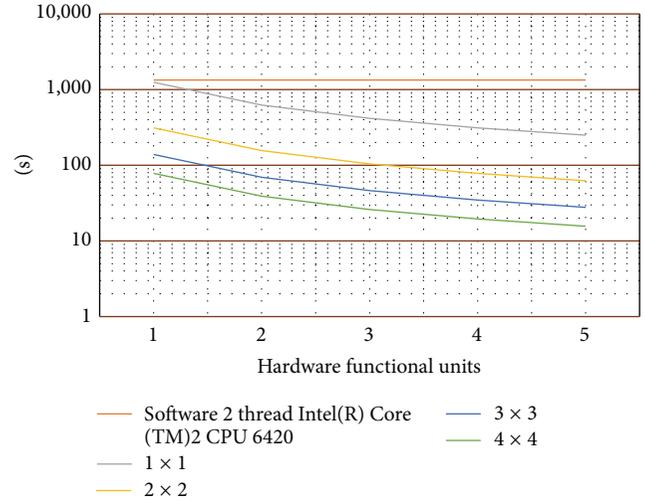


FIGURE 17: Time to compute (matrix 5000 x 5000 16 bits).

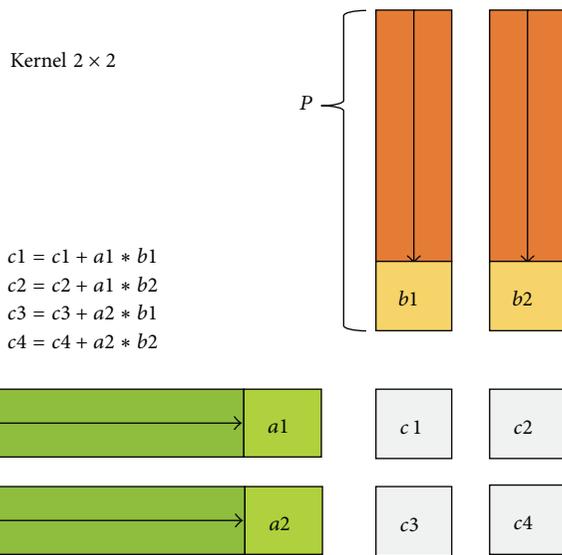


FIGURE 16: Kernel 2 x 2 composition.

computational kernel. In order to optimize the execution time and to avoid that the computational kernel remains inactive until new data is loaded in its local memory, it is necessary to determine the minimum amount of data to be loaded into local memory to ensure continuous processing time. It is necessary to reach a balance between the number of computational kernels and the amount of data needed by each one. This last value affects linearly the transference time and in a square mode the computation time. As shown in Figure 18, four nodes work in parallel. The transference time for each node is symbolized by TT Node n where n is the node number. If the amount of data loaded in each local memory allows an execution time long enough until new data is loaded, then the computing time shown in Figure 17 is actually equal to the total execution time. Time is calculated

from formula (5), where P represent the amount of rows or columns in $M \times P$ multiplied by $P \times N$ matrix multiplication.

Figure 19 represents the transference time for one computational kernel, the total transference time, and the computing time needed by each computational kernel, as a function of buffered data, for a 5000 x 5000 16-bit-wide element matrix multiplication, performed using 4 computational kernels of type 4 x 4. The buffered data indicates the amount of matrix elements of the matrix result that can be obtained.

It can be observed that the amount of data to be loaded in the local memory necessary to have a continuous execution time of each computational kernel is the necessary data to obtain 8 matrix elements of the result matrix:

$$\text{Time}_{\text{compute}} \geq \text{Time}_{\text{transference}} * N_{\text{kernels}}$$

$$N_{\text{cells}}^2 * P \geq \frac{\text{Data}_{\text{width}} * P * 2 * \text{Kernel}_{\text{size}} * N_{\text{cells}} * N_{\text{kernels}}}{\text{Bus}_{\text{width}}}$$

$$N_{\text{cells}} \geq \frac{2 * N_{\text{kernels}} * \text{Kernel}_{\text{size}} * \text{Data}_{\text{width}}}{\text{Bus}_{\text{width}}} \quad (5)$$

After all these considerations, the multiplication matrix was performed to measure the level of improvement that can be obtained using this approach. The computation time using this reconfigurable infrastructure with five 4 x 4 computational kernels was reduced from 1340 seconds using software until 16 seconds, which means an 83.75 times improvement.

7. Conclusion

In this paper, we introduce novel grid architecture and services for the integration of reconfigurable hardware (especially FPGAs) in the distributed high performance computing (HPC) world, so far dominated by multicore processors.

The principal contributions of this paper are on one side a computational model to integrate FPGAs at the same level

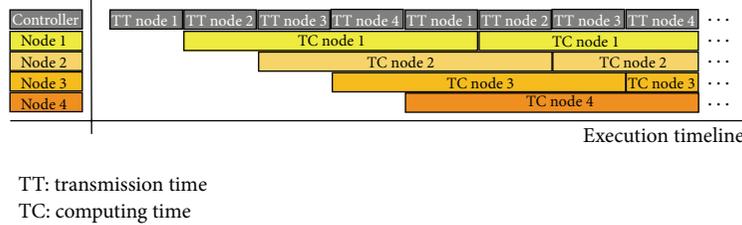


FIGURE 18: Timing diagram.

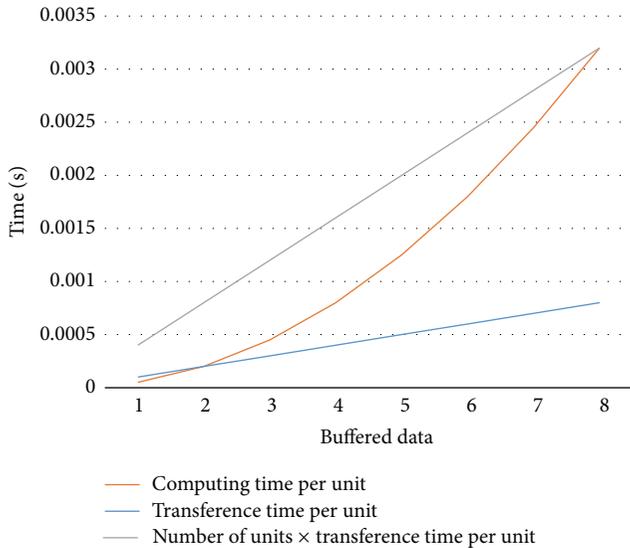


FIGURE 19: Computing versus transference time (5000 × 5000 4 × 4 16 bits).

of conventional processors, providing a transparent heterogeneous resources management and improving grid systems with the highest degree of efficiency that can be obtained for distributed computing applications using FPGAs and on the other side, a model for HPC applications development in order to exploit the benefits of this architecture. Besides, a set of services have been developed providing transparent application deployment including component location service and the mechanism for fast FPGA partial reconfiguration.

The infrastructure is made up of three layers: the lower layer is related to reconfigurable resources where user applications are instantiated, the second layer is for resource homogenization to facilitate application deployment, and the upper layer is for resource management issues. With this structure, resources are abstracted from the underlying platform in order to provide a homogeneous view for developers, facilitating migration between different computational node models.

This infrastructure was tested accelerating a 5000 × 5000 elements matrix multiplier application using two Virtex 5 FPGA with 8 reconfigurable areas in each one. Several analyses have been carried out to evaluate the viability of this work. Significant performance gains have been achieved and the obtained results encourage the use of this infrastructure

for HPC application deployment. There are still some aspects that need to be object of study and are out of the scope of this paper, such as the incorporation of scheduling mechanisms for application deployment including partial reconfiguration scheduling, a dynamic sizing of reconfigurable areas, and tools to facilitate resource analysis according to application requirements, just to name a few of open problems.

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 Spanish Government, Science and Innovation Department, under Project DREAMS (TEC2011-28666-C04-03).

References

- [1] M. C. Herbordt, T. VanCourt, Y. Gu et al., "Achieving high performance with FPGA-based computing," *Computer*, vol. 40, no. 3, pp. 50–57, 2007.
- [2] H. Li, Z.-H. Lu, and X.-B. Chi, "The applications and trends of high performance computing in finance," in *Proceedings of the 9th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES '10)*, pp. 193–197, Hong Kong, August 2010.
- [3] O. Lindtjorn, R. Clapp, O. Pell, H. Fu, M. Flynn, and H. Fu, "Beyond traditional microprocessors for geoscience high-performance computing applications," *IEEE Micro*, vol. 31, no. 2, pp. 41–49, 2011.
- [4] T. El-Ghazawi, E. El-Araby, M. Huang, K. Gaj, V. Kindratenko, and D. Buell, "The promise of high-performance reconfigurable computing," *Computer*, vol. 41, no. 2, pp. 69–76, 2008.
- [5] V. V. Klndratenko, C. P. Steffen, and R. J. Brunner, "Accelerating scientific applications with reconfigurable computing: getting started," *Computing in Science and Engineering*, vol. 9, no. 5, pp. 70–77, 2007.
- [6] J. Sun, G. D. Peterson, and O. O. Storaasli, "High-performance mixed-precision linear solver for FPGAs," *IEEE Transactions on Computers*, vol. 57, no. 12, pp. 1614–1623, 2008.
- [7] I. Page, "Constructing hardware-software systems from a single description," *The Journal of VLSI Signal Processing*, vol. 12, pp. 87–107, 1996.
- [8] D. Galloway, "The transmogripher C hardware description language and compiler for FPGAs," in *Proceedings of the IEEE*

- Symposium on FPGAs for Custom Computing Machines*, pp. 136–144, Napa Valley, Calif, USA, April 1995.
- [9] A. Takach, “Catapult c synthesis: creating parallel hardware from c++,” in *Proceedings of the International Symposium on Field-Programmable Gate Arrays Workshop*, February 2008.
 - [10] M. Araya-Polo, J. Cabezas, M. Hanzich et al., “Assessing accelerator-based HPC reverse time migration,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 147–162, 2011.
 - [11] K. Papadimitriou, A. Anyfantis, and A. Dollas, “An effective framework to evaluate dynamic partial reconfiguration in FPGA systems,” *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 6, pp. 1642–1651, 2010.
 - [12] K. H. Tsoi and W. Luk, “Axel: A heterogeneous cluster with FPGAs and GPUs,” in *Proceedings of the 18th ACM SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '10)*, pp. 115–124, ACM, New York, NY, USA, February 2010.
 - [13] Silicon Graphics International Corporation, 2012, <http://www.sgi.com/products/servers/altix/>.
 - [14] Netezza an IBM Company, 2012, <http://www.netezza.com/datawarehouse-appliance-products/index.aspx>.
 - [15] “Convey Computer Corporation,” 2012, <http://www.convey-computer.com>.
 - [16] Cray Inc, 2012, <http://www.cray.com/products/Legacy.aspx>.
 - [17] SciEngines Massively Parallel Computing, 2012, <http://www.sciengines.com>.
 - [18] R. Baxter, S. Booth, M. Bull et al., “High-performance reconfigurable computing—the view from Edinburgh,” in *Proceedings of the 2nd NASA/ESA Conference on Adaptive Hardware and Systems (AHS '07)*, pp. 273–279, August 2007.
 - [19] T. J. Callahan, J. R. Hauser, and J. Wawrzynek, “Garp architecture and C compiler,” *Computer*, vol. 33, no. 4, pp. 62–69, 2000.
 - [20] S. C. Goldstein, H. Schmit, M. Moe et al., “PipeRench: a coprocessor for streaming multimedia acceleration,” in *Proceedings of the 26th International Symposium on Computer Architecture (ISCA '99)*, pp. 28–39, May 1999.
 - [21] B. Mei, S. Vernalde, D. Verkest, and R. Lauwereins, “Design methodology for a tightly coupled VLIW/reconfigurable matrix architecture: A case study,” in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE '04)*, vol. 2, pp. 1224–1229, February 2004.
 - [22] M. B. Taylor, J. Kim, J. Miller et al., “The raw microprocessor: a computational fabric for software circuits and general-purpose programs,” *IEEE Micro*, vol. 22, no. 2, pp. 25–35, 2002.
 - [23] C. Ebeling, C. Fisher, G. Xing, M. Shen, and H. Liu, “Implementing an OFDM receiver on the RaPiD reconfigurable architecture,” *IEEE Transactions on Computers*, vol. 53, no. 11, pp. 1436–1448, 2004.
 - [24] X. Meng and V. Chaudhary, “A high-performance heterogeneous computing platform for biological sequence analysis,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1267–1280, 2010.
 - [25] Y.-T. Hwang, C.-C. Lin, and R.-T. Hung, “Lossless hyperspectral image compression system-based on HW/SW codesign,” *IEEE Embedded Systems Letters*, vol. 3, no. 1, pp. 20–23, 2011.
 - [26] C. H. Huang and P. A. Hsiung, “Hardware resource virtualization for dynamically partially reconfigurable systems,” *IEEE Embedded Systems Letters*, vol. 1, no. 1, pp. 19–23, 2009.
 - [27] X. Zhang, Y. Ding, Y. Huang, and X. Dong, “Design and implementation of a heterogeneous high-performance computing framework using dynamic and partial reconfigurable FPGAs,” in *Proceeding of the 10th IEEE International Conference on Computer and Information Technology (CIT '10)*, pp. 2329–2334, Bradford, UK, July 2010.
 - [28] M. Majer, J. Teich, A. Ahmadiania, and C. Bobda, “The Erlangen slot machine: a dynamically reconfigurable FPGA-based computer,” *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 47, no. 1, pp. 15–31, 2007.
 - [29] J. L. Tripp, M. B. Gokhale, and K. D. Peterson, “Trident: from high-level language to hardware circuitry,” *Computer*, vol. 40, no. 3, pp. 28–37, 2007.
 - [30] J. J. Koo, A. C. Evans, and W. J. Gross, “3-D brain MRI tissue classification on fpgas,” *IEEE Transactions on Image Processing*, vol. 18, no. 12, pp. 2735–2746, 2009.
 - [31] Nvidia, 2012, http://www.nvidia.com/object/cuda_home_new.html.
 - [32] Altera, 2012, <http://www.altera.com/b/opencl.html>.
 - [33] Calypto, 2014, <http://calypto.com/en/products/catapult/overview/>.
 - [34] Xilinx, 2014, <http://www.xilinx.com/products/design-tools/vivado/index.htm>.
 - [35] Synopsys, 2014, <http://www.synopsys.com/Systems/BlockDesign/HLS/Pages/default.aspx>.
 - [36] J. Barba, F. Rincón, F. Moya et al., “OOCE: Object-oriented communication engine for SoC design,” in *Proceeding of the 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD '07)*, pp. 296–302, Lubeck, Germany, August 2007.
 - [37] A. Sudarsanam, R. Kallam, and A. Dasu, “PRR-PRR dynamic relocation,” *IEEE Computer Architecture Letters*, vol. 8, no. 2, pp. 44–47, 2009.
 - [38] J. G. Siek and A. Lumsdaine, “The matrix template library: generic components for high-performance scientific computing,” *Computing in Science & Engineering*, vol. 1, no. 6, pp. 70–71, 1999.

Research Article

Performance Evaluation of the Machine Learning Algorithms Used in Inference Mechanism of a Medical Decision Support System

Mert Bal,¹ M. Fatih Amasyali,² Hayri Sever,³ Guven Kose,⁴ and Ayse Demirhan⁵

¹ Department of Mathematical Engineering, Yildiz Technical University, Davutpasa Campus A-220, Esenler, 34220 Istanbul, Turkey

² Department of Computer Engineering, Yildiz Technical University, Yildiz Campus, Besiktas, 34349 Istanbul, Turkey

³ Department of Computer Engineering, Hacettepe University, Beytepe Campus, Beytepe, 06530 Ankara, Turkey

⁴ Department of Information Management, Hacettepe University, Beytepe Campus, Beytepe, 06530 Ankara, Turkey

⁵ Department of Business Administration, Yildiz Technical University, Yildiz Campus, Besiktas, 34349 Istanbul, Turkey

Correspondence should be addressed to Mert Bal; mert.bal@gmail.com

Received 4 June 2014; Revised 7 August 2014; Accepted 20 August 2014; Published 11 September 2014

Academic Editor: Shifei Ding

Copyright © 2014 Mert Bal et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The importance of the decision support systems is increasingly supporting the decision making process in cases of uncertainty and the lack of information and they are widely used in various fields like engineering, finance, medicine, and so forth. Medical decision support systems help the healthcare personnel to select optimal method during the treatment of the patients. Decision support systems are intelligent software systems that support decision makers on their decisions. The design of decision support systems consists of four main subjects called inference mechanism, knowledge-base, explanation module, and active memory. Inference mechanism constitutes the basis of decision support systems. There are various methods that can be used in these mechanisms approaches. Some of these methods are decision trees, artificial neural networks, statistical methods, rule-based methods, and so forth. In decision support systems, those methods can be used separately or a hybrid system, and also combination of those methods. In this study, synthetic data with 10, 100, 1000, and 2000 records have been produced to reflect the probabilities on the ALARM network. The accuracy of 11 machine learning methods for the inference mechanism of medical decision support system is compared on various data sets.

1. Introduction

A decision support systems (DSSs) is a computer-based information system that supports organizational and business decision making activities. Medical decision support systems, which are variants of decision support systems, are intelligent software systems that are designed to improve clinical diagnosis system and to support the healthcare personnel in their decision. Intelligent decision support systems use artificial intelligence system techniques to support the healthcare personnel for selecting the best method for both diagnosis and also for treatment especially when the information about the treatment is incomplete or uncertain. These systems can work in both active and passive modes. When they are in passive mode, they will be used only when they are

required. When they are in active mode, they will be making recommendations as well. When we look at the approaches of the inference mechanisms, which constitute the most important part of the medical decision support systems, these approaches can be divided into two parts such as rule-based systems and data-driven systems. Rule-based systems are constructed on the knowledge base, which are formed by if-then structures. In this structure, the information base is formed by the rules. The operation logic of the system is to find relevant rules on basis of the available information, operate them, and continue to search for a rule until a result has been obtained.

Those rule-based systems have some strong features as well as some disadvantages. For example, the performance of the system decreases and the maintenance of the system

becomes difficult in case of the number of the rules being large enough. The examples of the medical decision support systems are MYCIN [1, 2], TRAUMAID [3], and RO²SE [4].

Data-driven systems, on the other hand, operate in large data stacks and support the decision making process using data mining methods. Several studies can be found on literature about data-driven systems. Some of these studies can be referred as Bayes networks [5], rough sets [6], and artificial neural networks [7] which are the examples of such studies. Data-driven systems are more flexible compared to the rule-based systems and they have the ability to learn by themselves.

In our previous study [8] ALARM network structure was used for the generated synthetic data on the same data set. When the results are examined in that study, it can be seen that the rule based method is more successful in the rate of 25% than the “Bayesian network based” method in all dimensions of the data sets. Besides, when both of these methods are combined and utilized together the success rate rises to 80%; that is, much higher rates are acquired in comparison to the values obtained by applying these methods individually.

In this study, the accuracy of II machine learning methods which can be used in the inference mechanism of the medical decision support systems is carried out on various data sets.

2. Decision Support Systems

Decision support systems (DSSs) are interactive computer-based systems or subsystems that are designed to help decision makers to decide and complete the decision process operations and also to determine and solve problems using communication technologies, information, documents, and models. They provide data storage and retrieval but enhance the traditional information access and retrieval functions with support for model building and model-based reasoning. They support framing, modeling, and problem solving. Typical application areas of DSSs are healthcare, management, and planning in business, the military, and any area in which management will encounter complex decision situations. DSSs are typically used for strategic and tactical decisions faced by upper-level management-decisions with a reasonably low frequency and high potential consequences, in which the time taken for thinking through and modeling the problem pays off generously in the long run [10].

Generally, decision support systems should include the following features.

- (i) DSSs are used to support the decision making process not to accomplish operational processes.
- (ii) DSSs should support each phase of the decision making process.
- (iii) DSSs support the half or full configured decision environments.
- (iv) DSSs support each management levels from bottom to top.
- (v) DSSs have interactive and user-friendly interfaces.
- (vi) DSSs use data and model as a basis.

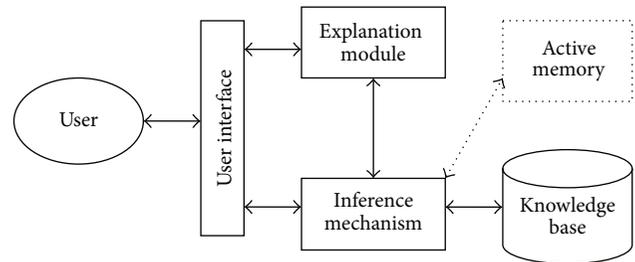


FIGURE 1: The main structure of decision support system.

Decision support systems and relevant operation methods can be divided into four main subjects. These subjects are called as inference mechanism, knowledge base, explanation module, and active memory. Inference mechanism constitutes the basis of decision support systems. In this part, the results are generated in consideration of the current information and/or the information that was entered to the system by the user. The generated results may be a decision or they may include guiding information. The second part is the knowledge base which holds the expert information used when the decision support system is making inference. The active memory part holds the information, which is supplied by the user and/or current inference processes. Also, explanation module, which may not be present on each decision support system, generates an accuracy validation and explanation in consideration of the results generated by the inference mechanism and knowledge base [11]. Those subjects and their relations are shown in Figure 1.

In rule-based systems, the knowledge base is formed by the rule group. The results are obtained for various circumstances on the problem relevant to the subject, using the generated rules. The rules forming the knowledge base are prepared by if-then structure. The content of an inference system, which is developed using rule-based methods, consists of the rules generated by if-then, the facts, and an interpreter that interprets the facts using the rules in the system [12].

There are two methods used to process the rules in the rule-based methods. These methods are forward chaining and backward chaining. In forward chaining method, the results are obtained using the preliminary facts with the help of the rules. In backward chaining method, it is started with a hypothesis (or target) and the rules, which will reach that hypothesis, are searched. The reached rules generate subrules and the process continues in this way.

In cases, which the result is estimated and this estimation should be verified, backward chaining method should be used instead of forward chaining method.

In order to generate the rule set in rule-based methods of inference systems, people who are experienced on the problem should contribute to the design of the system. This process usually proceeds with the help of experienced people in the rule development phase by determining the faults and defects in the estimations and using the planned system as a reference [13].

The designer usually develops simple interfaces for experts to contribute in the development phase. In the begin-

ning of the process, the experts start testing the systems as if they will use the system for operational purposes. The questions asked to the experts in the scope of the limited information of the systems are answered by the same experts.

The aim is to test the system in order to improve it. The expert who answered the questions evaluates the system by looking at the results generated by the system and then tries to correct the defined defects and faults by using the rule development tool. The rule set in the inference systems, which use rule-based methods, can be generated by the expert on the problem.

Data-driven systems examine large data pools in organizations. These systems usually work with the systems that collect data like data warehouse, and so forth. Data-driven systems take place in decision making process with online analytical processing (OLAP) and data mining methods. These systems work on very large datasets. The relations in these datasets are analyzed electronically and make predictions for future data relations. Data-driven systems use the bottom-up procedure to explain the characteristics of the data system [14].

3. Machine Learning Algorithms

Machine learning is about learning to make predictions from example of desired behavior or past observations. Learning methods have found numerous applications in performance modeling and evaluation [15]. The basic definitions of machine learning are given below.

3.1. Basic Definitions. Data points called *examples* are typically described by their values on some set of *features*. The space that examples live in is called the *feature space* and is typically denoted by X .

The *label* of an example will be predicted. The space of possible labels is denoted by Y .

A *learning problem* is some unknown data distribution D over $X \times Y$, coupled with a loss function $l(y, y')$ measuring the loss of predicting y' when the true label is y .

A *learning algorithm* takes a set of labeled training examples of the form $(x, y) \in X \times Y$ and produces a predictor $f : X \rightarrow Y$. The goal of the algorithm is to find f minimizing the expected loss $\vec{E}_{(x,y) \sim D} l(f(x), y)$.

There are two base learning problems, defined for any feature space X . In binary classification, examples are categorized into two categories [15].

Definition 1. A *binary classification* problem is defined by a distribution D over $X \times Y$, where $Y = \{0, 1\}$. The goal is to find a *classifier* $h : X \rightarrow Y$ minimizing the *error rate* on D :

$$e(h, D) = \vec{Pr}_{(x,y) \sim D} [h(x) \neq y]. \quad (1)$$

By fixing an unlabeled example $x \in X$, a *conditional distribution* $D | x$ over Y is found.

Regression is another basic learning problem, where the goal is to predict a real-valued label Y .

The loss function typically used in regression is the squared error loss between the predicted and actual labels.

Definition 2. A *regression problem* is defined by a distribution D over $X \times \mathcal{R}$. The goal is to find a function $f : X \rightarrow \mathcal{R}$ minimizing the *squared loss* [15]:

$$l(f, D) = \vec{E}_{(x,y) \sim D} (f(x) - y)^2. \quad (2)$$

The machine learning algorithms that are used in the study will be explained below.

3.2. C4.5 Decision Tree. A decision tree is basically a classifier that shows all possible outcomes and the paths leading to those outcomes in the form of a tree structure. Various algorithms for inducing a decision tree are described in existing literature, for example, CART (classification and regression trees) [16], OC1 [17], ID3, and C4.5 [18]. These algorithms build a decision tree recursively by partitioning the training data set into successively purer subsets [19].

C4.5 [18] is an algorithm used to generate a decision tree. C4.5 uses the fact that each attribute of the data can be used to make a decision that splits the data into smaller subsets. C4.5 examines the normalized information gain (difference in entropy) that results from choosing a feature for splitting the data [20]

$$\text{SplitInfo}_x T = - \sum_{i=1}^n \frac{T_i}{T} \log_2 \frac{T_i}{T}, \quad (3)$$

$$\text{Gain Ratio}_x (T) = \frac{\text{Gain}_x (T)}{\text{SplitInfo}_x T},$$

where $\text{SplitInfo}_x T$ represents the potential information provided by dividing dataset, T , into n partition corresponding to the outputs of attributes x , and $\text{Gain}_x (T)$ is how much gain would be achieved by branching on x .

3.3. Multilayer Perceptron (MLP). Multilayer perceptron (MLP) [21] also referred to as multilayer feed forward neural networks is the most used and popular neural network method. It belongs to the class of supervised neural network. The MLP topology consists of three sequential layers of processing nodes: an input layer, one or more hidden layers, and an output layer which produces the classification results.

A MLP structure is shown in Figure 2.

The principle of the network is that when data are presented at the input layer, the network nodes perform calculations in the successive layers until an output value is obtained at each of the output nodes. This output signal should be able to indicate the appropriate class for the input data. A node in MLP can be modeled as one or more artificial neurons, which computes the weighted sum of the inputs at the presence of the bias and passes this sum through

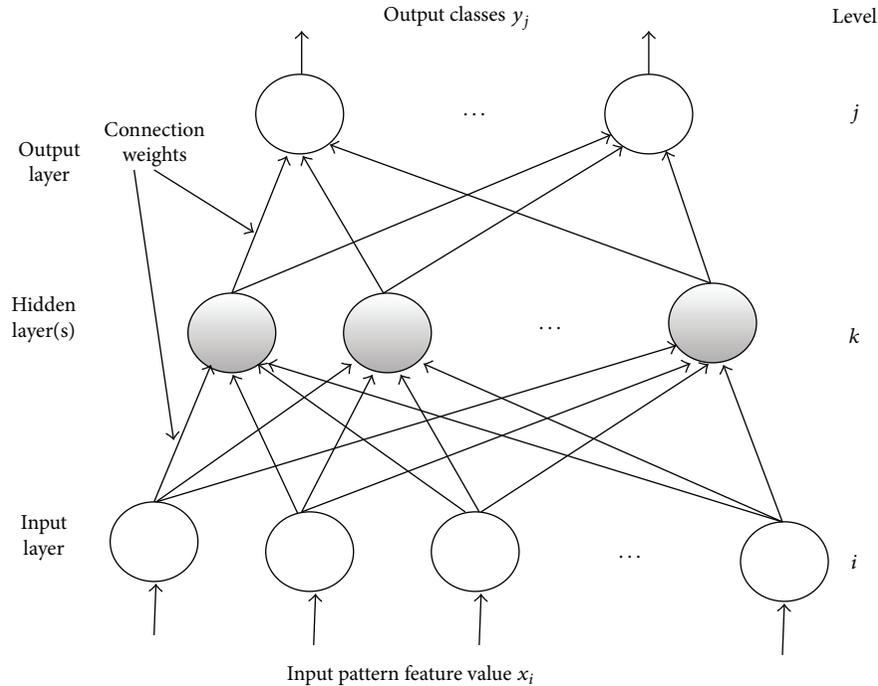


FIGURE 2: Structure of a multilayer perceptron [7].

the nonlinear activation function. This process is defined as follows [7]:

$$\mu_j = \sum_{i=1}^N w_{ji}x_i + \theta_j, \quad (4)$$

$$y_j = \varphi_j(\mu_j),$$

where μ_j is the linear combination of inputs x_1, x_2, \dots, x_N , θ_j is the bias (adjustable parameter), w_{ji} is the connection synaptic weight between the input x_i and the neuron j , and $\varphi(\cdot)$ is the activation function (usually nonlinear function) of the j th neuron, and y_j is the output. Here, hyperbolic tangent and logistic sigmoid function can be used for the nonlinear activation function. But, in most of the applications widely used logistic sigmoid function is applied as follows:

$$\varphi(\lambda) = \frac{1}{1 + e^{-\lambda}}, \quad (5)$$

where λ represents the slope of the sigmoid [22].

The bias term θ_j contributes to the left or right shift of the sigmoid activation function, depending on whether θ_j takes a positive or negative value.

3.3.1. Backpropagation Learning Algorithm. Learning in a MLP is an unconstrained optimization problem, which is subject to the minimization of a global error function depending on the synaptic weights of the network. For a given training data consisting of input-output patterns, values of synaptic weights in a MLP are iteratively updated by

a learning algorithm to approximate the desired value. This update process is usually performed by backpropagating the error signal layer by layer and adapting synaptic weights with respect to the magnitude of error signal [23].

The first backpropagation learning algorithm for use with MLP structures was presented by [21]. The backpropagation algorithm is one of the simplest and most general methods for the supervised training of MLP. This algorithm uses a gradient descent search method to minimize a mean square error between the desired output and the actual outputs. Backpropagation algorithm is defined as follows [7, 24].

- (i) Initialize all the connection weights w with small random values from a pseudorandom sequence generator.
- (ii) Repeat until convergence (either when the error J is below a preset value or until the gradient $\partial J/\partial w$ is smaller than a preset value).
 - (i) Compute the update using $\Delta w(m) = -\xi(\partial J(m)/\partial w)$,
 - (ii) Iterative algorithm requires taking a weight vector at iteration m and updating it as $w(m+1) = w(m) + \Delta w(m)$,
 - (iii) Compute the error $J(m+1)$,

where m is the iteration number, w represents all the weights in the network, and ξ is the learning rate and merely indicates the relative size of the change in weights. The error J can be chosen as the mean square error function between the actual

output y_j and the desired output d_j ; d and y are the desired and the network output vector of length N :

$$J(w) = \frac{1}{2} \sum_{j=1}^N (d_j - y_j)^2 = \frac{1}{2} (d - y)^2. \quad (6)$$

3.4. Support Vector Machines (SVMs). The support vector machines (SVMs) [25] is a type of learning machine based on statistical learning theory. SVMs are supervised learning methods that have been widely and successfully used for pattern recognition in different areas [26].

In particular in recent years SMVs with linear or non-linear kernels have become one of the most promising learning algorithms for classification as well as regression [27]. The problem that SVMs try to solve is to find an optimal hyperplane that correctly classifies data points by separating the points of two classes as much as possible [28].

Let x_i (for $1 \leq i \leq N_x$) be the input vectors in input space, with corresponding binary labels $y_i \in \{-1, 1\}$.

Let $\vec{X}_i = \Phi(x_i)$ be the corresponding vectors in feature space, where $\Phi(x_i)$ is the implicit kernel mapping, and let $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ be the kernel function, implying a dot product in the feature space [29].

$K(x, y)$ represents the desired notion of similarity between data x and y . $K(x, y)$ needs to satisfy a Mercer's condition in order for Φ to exist [28].

There are a number of kernel functions which have been found to provide good generalization capabilities [30].

The most commonly used kernel functions are as follows:

- Linear Kernel: $K(x_i, x_j) = x_i^T x_j$
- Polynomial Kernel: $K(x_i, x_j) = (\eta(x_i^T x_j) + r)^d$
- Gaussian Kernel: $K(x_i, x_j) = \exp(-\eta \|x_i - x_j\|^2)$
- Gaussian Radial Basis Function Kernel: $K(x_i, x_j) = \exp(-\eta \|x_i - x_j\|^2 / 2\sigma^2)$
- Sigmoid Kernel: $K(x_i, x_j) = \tanh(\eta(x_i x_j) + r)$

where $\eta > 0$ and r are kernel parameters, d is the degree of kernel and positive integer number, and σ is the standard deviation and positive real number.

The optimization problem for a soft-margin SVM is

$$\min_{\vec{w}, b} \left\{ \frac{1}{2} \|\vec{w}\|^2 + C \sum_i \xi_i \right\} \quad (7)$$

subject to the constraints $y_i(\vec{w}_i x + b) = 1 - \xi_i$ and $\xi_i \geq 0$, where \vec{w} is the normal vector of the separating hyperplane in feature space, and $C > 0$ is a regularization parameter controlling the penalty for misclassification. Equation (7) is referred to as the primal equation. From the Lagrangian form of (7), we derive the dual problem

$$\max_{\alpha} \left\{ \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \right\} \quad (8)$$

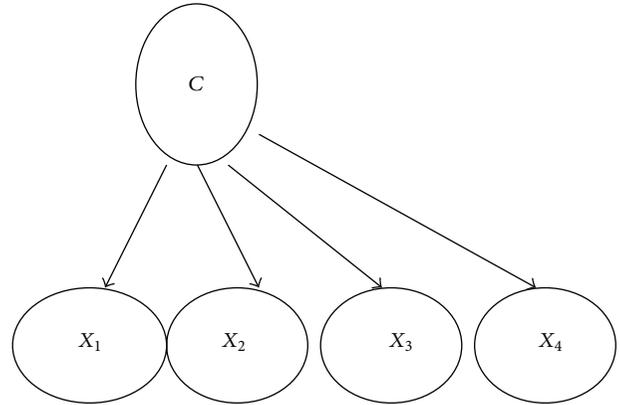


FIGURE 3: A simple Naïve-Bayes structure.

subject to $0 \leq \alpha_i \leq C$. This is a quadratic optimization problem that can be solved efficiently using algorithms such as sequential minimal optimization (SMO) [31].

Typically, many α_i go to zero during optimization, and the remaining x_i corresponding to those $\alpha_i > 0$ are called support vectors. To simplify notation, from here on we assume that all nonsupport-vectors have been removed, so that N_x is now the number of support vectors, and $\alpha_i > 0$ for all i . With this formulation, the normal vector of the separating plane \vec{w} is calculated as

$$\vec{w} = \sum_{i=1}^{N_x} \alpha_i y_i \vec{x}_i. \quad (9)$$

Note that because $\vec{X}_i = \Phi(x_i)$ is defined implicitly, \vec{w} exists only in feature space and cannot be computed directly. Instead, the classification $f(\vec{q})$ of a new query vector \vec{q} can only be determined by computing the kernel function of \vec{q} with every support vector:

$$f(\vec{q}) = \text{sign} \left(\sum_{i=1}^{N_x} \alpha_i y_i K(\vec{q}, x_i) + b \right), \quad (10)$$

where the bias term b is the offset of the hyperplane along its normal vector, determined during SVM training [29].

3.5. Naïve Bayes. Naïve-Bayes is one of the most efficient and effective inductive learning algorithms for machine learning and data mining [32].

A Naïve-Bayes Bayesian network is a simple structure that has the classification node as the parent node of all other nodes. This structure is shown in Figure 3.

No other connections are allowed in a Naïve-Bayes structure. Naïve-Bayes has been used as effective classifier for many years. It has two advantages over many other classifiers. First, it is easy to construct, as the structure is given *a priori* (and hence no structure learning procedure is required). Second, the classification process is very efficient. Both advantages are due to its assumption that all the features are independent of each other. Although this independence assumption is obviously problematic, Naïve-Bayes has surprisingly outperformed many sophisticated classifiers over

a large number of datasets, especially where the features are not strongly correlated [33].

The procedure of learning Naïve-Bayes (Figure 3) is as follows.

- (1) Let the classification node be the parent of all other nodes.
- (2) Learn the parameters (recall these are just the empirical frequency estimates) and output the Naïve-Bayes Bayesian network [34].

Typically, an example E is represented by a tuple of attribute values (x_1, x_2, \dots, x_n) , where x_i is the value of attribute X_i . Let C represent the classification variable, and let c be the value of C [32]. Naïve-Bayes classifier is defined as below:

$$\begin{aligned} &\text{classify}(x_1, x_2, \dots, x_n) \\ &= \underset{c}{\operatorname{argmax}} p(C = c) \prod_{i=1}^n p(X_i = x_i | C = c). \end{aligned} \quad (11)$$

3.6. Instance-Based Learning. Instance-based learning (IBL) [35] algorithms have several notable characteristics. They employ simple representations for concept descriptions, have low incremental learning costs, have small storage requirements, can produce concepts exemplars on demand, can learn continuous functions, and can learn nonlinearly separable categories; IBL algorithms have been successfully applied to many areas such as speech recognition, handwritten letter identification, and thyroid disease diagnosis.

All IBL algorithms consist of the following three components [36].

- (1) Similarity function: Given two normalized instances, this yields their numeric-valued similarity.
- (2) Classification function: Given an instance i to be classified and its similarity with each saved instance yields a classification for i .
- (3) Memory updating algorithm: Given the instance being classified and the results of the other two components updates the set of saved instances and their classification records.

The IB1 (one nearest neighbor) algorithm is the simplest instance-based learning algorithm. IB1 (one nearest neighbor) algorithm will be explained below.

3.6.1. IB1 (One Nearest Neighbor). IB1 [35] is an implementation of the simplest similarity based learner, known as nearest neighbor. IB1 simply finds the stored instance closest (according to Euclidean distance metric) to the instance to be classified. The new instance is assigned to the retrieved instance's class. Equation (12) shows the distance metric employed by IB1:

$$D(x, y) = \sqrt{\sum_{i=1}^n f(x_i, y_i)}. \quad (12)$$

Equation (10) gives the distance between two instances x and y ; x_i and y_i refer to the i th feature value of instance x and y , respectively.

For numeric valued attributes $f(x_i, y_i) = (x_i - y_i)^2$, for symbolic valued attributes $f(x, y) = 0$, if the feature values x_i and y_i are the same, and 1 if they differ [37].

3.7. Simple Logistic Regression. Logistic regressions are one of the most widely used techniques for solving binary classification problems. In the logistic regressions, the posterior probabilities p_i^* , $i \in \{1, 2\}$ are represented as in the following:

$$\Pi_1 = \frac{\exp(\eta)}{1 + \exp(\eta)}, \quad \Pi_2 = 1 - \Pi_1, \quad (13)$$

where η is a function of an input \vec{x}_0 . For example, η is a linear function of the input \vec{x}_0 , that is,

$$\eta = \vec{\alpha}^T \vec{x}_0 + \beta \quad (14)$$

and the parameters $\vec{\alpha}, \beta$ are estimated by the maximum likelihood method.

η is an arbitrary function of \vec{x}_0 . Note that if you choose an appropriate η , the model in (13) can represent some kinds of binary classification systems, such as neural networks and LogitBoost [38].

LogitBoost with simple regression functions as base learners is used for fitting the logistic models. The optimal number of LogitBoost iterations to perform is cross-validated, which leads to automatic attribute selection. This method is called "simple logistic" [39, 40]. LogitBoost algorithm is defined below.

3.7.1. LogitBoost Algorithm. The LogitBoost algorithm [41] is based on the observation that AdaBoost [42] is in essence fitting an additive logistic regression model to the training data. An additive model is an approximation to a function

$$F(x) = \sum_{i=1}^N c_i f_i(x), \quad (15)$$

where the c_i are constants to be determined and the f_i are basis functions. If it is assumed that $F(x)$ is the mapping that is looked for to fit as our strong aggregate hypothesis and the $f(x)$ are our weak hypothesis, then it can be shown that the two-class AdaBoost algorithm is fitting such a model by minimizing the criterion:

$$J(F) = E(\exp(-yF(x))), \quad (16)$$

where y is true class label in $\{-1, 1\}$. LogitBoost minimizes this criterion by using Newton-like steps to fit an additive logistic regression model to directly optimize the binomial log-likelihood $-\log(1 + \exp(-2yF(x)))$ [43].

3.8. Boosting. Boosting [44] is a meta-algorithm which can be viewed as a model averaging method. It is the most widely used ensemble method and one of the most powerful

learning ideas introduced in the last twenty years. Originally designed for classification, it can also be profitably extended to regression. One first creates a “weak” classifier; that is, it suffices that its accuracy on the training set is only slightly better than random guessing. A succession of models is built iteratively, each one being trained on a dataset in which points misclassified (or, with regression, those poorly predicted) by the previous model are given more weight. Finally, all of the successive models are weighted according to their success and then the outputs are combined using voting (for classification) or averaging (for regression), thus creating a final model. The original boosting algorithm combined three weak learners to generate a strong learner [45].

3.8.1. AdaBoost Algorithm. Let $\vec{X} = (\vec{x}_i, y_i), i = 1, 2, \dots, N$ be a training sample of observations, where $\vec{x}_i \in \mathfrak{R}^n$ is an n -dimensional vector of features, and y_i is a binary label: $y_i \in \{-1, +1\}$.

In a practical situation the label y_i may be hidden, and the task is to estimate it using the vector of features. Let us consider the most simple linear decision function

$$u_i = u(\vec{x}_i) = \sum_{j=0}^n w_j \cdot x_{ij}, \tag{17}$$

where x_{i0} is a constant term.

A decision rule can be defined as a function of decision function and threshold parameter

$$f_i = f(u_i, \Delta) = \begin{cases} 1, & \text{if } u_i \geq \Delta, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Let us consider minimizing the criterion

$$\sum_{i=1}^N \xi(\vec{x}_i, y_i) \exp(-y_i u(\vec{x}_i)), \tag{19}$$

where the weight function is given below:

$$\xi(\vec{x}_i, y_i) := \exp\{-y_i F(\vec{x}_i)\}. \tag{20}$$

It is assumed that the initial values of the ensemble decision function $F(\vec{x}_i)$ are set to zero.

Advantages of the exponential compared with squared loss function were discussed in [46]. Unfortunately, it is not possible to optimize the step-size in the case of exponential target function. It is essential to maintain low value of the step size in order to ensure stability of the gradient-based optimization algorithm. As a consequence, the whole optimization process may be very slow and time-consuming. The AdaBoost algorithm was introduced in [42] in order to facilitate optimization process. The following Taylor-approximation is valid under assumption that values of $u(\vec{x}_i)$ are small:

$$\exp\{-y_i u(\vec{x}_i)\} \approx \frac{1}{2} [(y_i - u(\vec{x}_i))^2 + 1]. \tag{21}$$

Therefore, quadratic-minimization (QM) model is applied in order to minimize (19).

Then, the value of the threshold parameters Δ for u_i is optimized and the corresponding decision rule $f_i \in \{-1, +1\}$ is found.

Next, we will return to (19),

$$\sum_{i=1}^N \xi(\vec{x}_i, y_i) \exp(-cy_i f(\vec{x}_i)), \tag{22}$$

where the optimal value of the parameter c may be easily found:

$$c = \frac{1}{2} \log \left\{ \frac{A}{B} \right\} \tag{23}$$

and where

$$A = \sum_{y_i=f(\vec{x}_i)} \xi(\vec{x}_i, y_i), \quad B = \sum_{y_i \neq f(\vec{x}_i)} \xi(\vec{x}_i, y_i). \tag{24}$$

Finally, for the current boosting iteration, we update the function F

$$F_{\text{new}}(\vec{x}_i) \leftarrow F(\vec{x}_i) + cf(\vec{x}_i) \tag{25}$$

and recomputed weight coefficients ξ according to (20) [47].

3.9. Bagging. Bagging [48] predictors is a method for generating multiple versions of a predictor and using these to get on aggregated predictor. The aggregation averages over the versions when predicting a numerical outcome and does a plurality vote when predicting a class. The multiple versions are formed by making bootstrap replicates of the learning set and using these as new learning sets. Tests on real and simulated data sets using classification and regression trees and subset selection in linear regression show that bagging can give substantial gains in accuracy. The vital element is the instability of the prediction method. If perturbing the learning set can cause significant changes in the predictor constructed, then bagging can improve accuracy [48].

3.10. Random Forest. Random forests [49] are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(\vec{x}, \Theta_k), k = 1, 2, \dots\}$, where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input \vec{x} .

3.11. Reduced Error Pruning Tree. Reduced error pruning (REP) was introduced by Quinlan [50], in the context of decision tree learning. It has subsequently been adapted to rule set learning as well [51]. REP produces an optimal pruning of a given tree, the smallest tree among those with minimal error with respect to a given set of *pruning examples*

[51, 52]. The REP algorithm works in two phases: first the set of pruning examples S is classified using the given tree T to be pruned. Counters that keep track of the number of examples of each class passing through each node are updated simultaneously. In the second phase—a bottom-up pruning phase—those parts of the tree that can be removed without increasing the error of the remaining hypothesis are pruned away [53]. The pruning decisions are based on the node statistics calculated in the top-down classification phase.

3.12. *ZeroR (Zero Rule)*. Zero rule (ZeroR, 0-R) is a trivial classifier, but it gives a lower bound on the performance of a given a dataset which should be significantly improved by more complex classifiers. As such it is a reasonable test on how well the class can be predicted without considering the other attributes [54].

4. ALARM Network Structure and Datasets

In order to compare the performances (in terms of accuracy) of machine learning methods in the scope of this study, the network structure, which is used in scientific studies and known as ALARM (a logical alarm reduction mechanism) network [5] in literature is used. ALARM network is a network structure that is prepared by using real patient information for many variables and shows the probabilities derived from the real life circumstances. ALARM network calculates the probabilities for different diagnosis based on the current evidences and recently it has been used for many researchers. Totally there are 37 nodes in ALARM network and the relationships and conditional probabilities among these have been defined. The medical information has been coded in a graphical structure with 46 arches, 16 findings, and 13 intermediate variables that relate the examination results to the diagnosis problems that represent 8 diagnosis problems. Two algorithms have been applied to this Bayes network; one of them is a message-passing algorithm, developed by Pearl [55] to update the probabilities in the various linked networks using conditioning methods and the second one is that the exact inference algorithm, developed by Lauritzen and Spiegelhalter [56] for local probability calculations in the graphical structure. There are three variables named diagnosis, measurements, and intermediate variables in the ALARM network.

- (1) *Diagnosis* and the qualitative information are on the top of the network. Those variables do not belong to any predecessors and they are deemed mutually independent from the predecessors. Each node is linked to the particular and detailed value sets that represent the severity and the presence of a certain disease.
- (2) *Measurements* represent any current quantitative information. All continuous variables are represented categorically with a discrete interval set that divides the value set.

- (3) *Intermediate variables* show the element that can not be measured directly. The probabilities in the Bayes network can represent both objective and subjective information. ALARM network includes statistical data, logical conditional probabilities, which are calculated from the equations relevant to the variables, and a certain number of subjective valuations and it is usually used to form the network structure over synthetically data.

In cases for all given different predecessor nodes, it is required to obtain a conditional probability for a node. The structure of ALARM network and defined variable are shown in Figure 4.

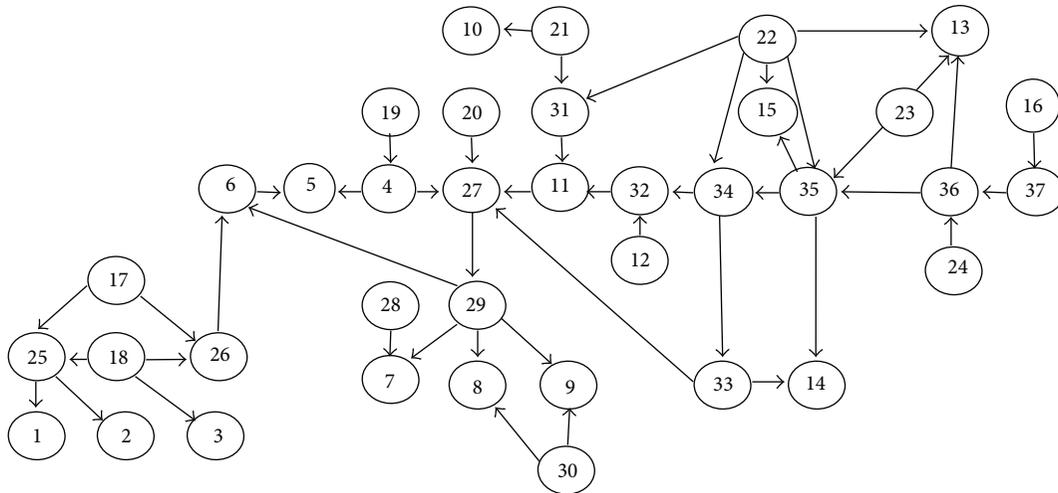
In order to compare the performances of algorithms mentioned in Section 3, synthetic test data with 10, 100, 1000, and 2000 records have been produced to reflect the possibilities on the ALARM network. For these operations, based on ALARM network structure, NETICA 3.18 [57] software has been used. Conditional probability diagram for ALARM network structure and a variable defined in the structure are shown in Figure 5. Some of the synthetic data has been taken as test data.

Each record on those generated data shows probable values for each of the 37 variables that were defined on this network. Each record consist of values for intermediate variable as well as 12 input and 11 output variables. The tests, which were carried out, send the input variable values on each record to the relevant module and keep the resulting list as a separate file. The accuracy of the results is decided by comparing the variable values on the relevant record on the test data. For each record, 11 probable results have been obtained.

The results that were obtained by using JavaBayes [58] open source software are applied to each of the generated synthetic data sets separately. 11 output variables for one record belonging 100 data sets are shown in Table 1. JavaBayes uses a generalized version of “*variable elimination*” method as an inference algorithm [59]. It has generated 110 output variables in 10 data sets, 1100 output variables in 100 data sets, 11000 output variables in 1000 data sets, and 22000 output variables in 2000 data sets.

In Table 1, for each data set only 11 output variables for one record are presented. In this table, first column shows the variable name (disease name) and the second column shows the accuracy and they are calculated by the software using Bayes theorem, third column shows the real situations in the ALARM network, fourth column shows the results, generated by the software, and fifth column shows the comparison between the real situation and the results generated by the software. In the fifth column, if the real situation and the results generated by the software are the same POSITIVE and if the real situation and the results generated by the software are not the same NEGATIVE result will be generated. POSITIVE values show correct diagnosis, and NEGATIVE values show incorrect diagnosis.

For example, in Table 1, the accuracy of the MinVol variable has been calculated as 0.9136 by the software. Because this value is not the same with the real situation,



- (1) Central venous pressure
- (2) Pulmonary capillary wedge pressure
- (3) History of left ventricular failure
- (4) Total peripheral resistance
- (5) Blood pressure
- (6) Cardiac output
- (7) Heart rate obtained from blood pressure monitor
- (8) Heart rate obtained from electrocardiogram
- (9) Heart rate obtained from oximeter
- (10) Pulmonary artery pressure
- (11) Arterial-blood oxygen saturation
- (12) Fraction of oxygen in inspired gas
- (13) Ventilation pressure
- (14) Carbon-dioxide content of expired gas
- (15) Minute volume, measured
- (16) Minute volume, calculated
- (17) Hypovolemia
- (18) Left-ventricular failure
- (19) Anaphylaxis
- (20) Insufficient anesthesia or analgesia
- (21) Pulmonary embolus
- (22) Intubation status
- (23) Kinked ventilation tube
- (24) Disconnected ventilation tube
- (25) Left-ventricular end-diastolic volume
- (26) Stroke volume
- (27) Catecholamine level
- (28) Error in heart rate reading due to low cardiac output
- (29) True heart rate
- (30) Error in heart rate reading due to electrocautery device
- (31) Shunt
- (32) Pulmonary-artery oxygen saturation
- (33) Arterial carbon-dioxide content
- (34) Alveolar ventilation
- (35) Pulmonary ventilation
- (36) Ventilation measured at endotracheal tube
- (37) Minute ventilation measured at the ventilator

FIGURE 4: ALARM network structure and the variables defined in the network [9].

the correct diagnosis has not been obtained. Similarly, for HREKG variable, the accuracy has been calculated as 0.8228 by the software. Because this value is the same with the real situation, the correct diagnosis has been obtained. Similar interpretations are also valid for other data sets. Each sample generated by ALARM network includes 12 independent and 11 depended variables. So we formed 11 classification datasets having 12 inputs and one output. The class labels for these 11 datasets are given at Table 2.

To see to effects of sample size, we generated several datasets having 10, 100, 1000, and 2000 samples for each of

11 classification datasets. At the end, we have 44 (= 11 * 4) classification datasets.

5. Experimental Design

We used 11 machine learning algorithms from WEKA library [60] for the classification of these 44 datasets. The algorithms are given in Table 3.

The default design parameters were selected for NB, MLP, SL, SMO, IBK, J48, and RT algorithms. For the

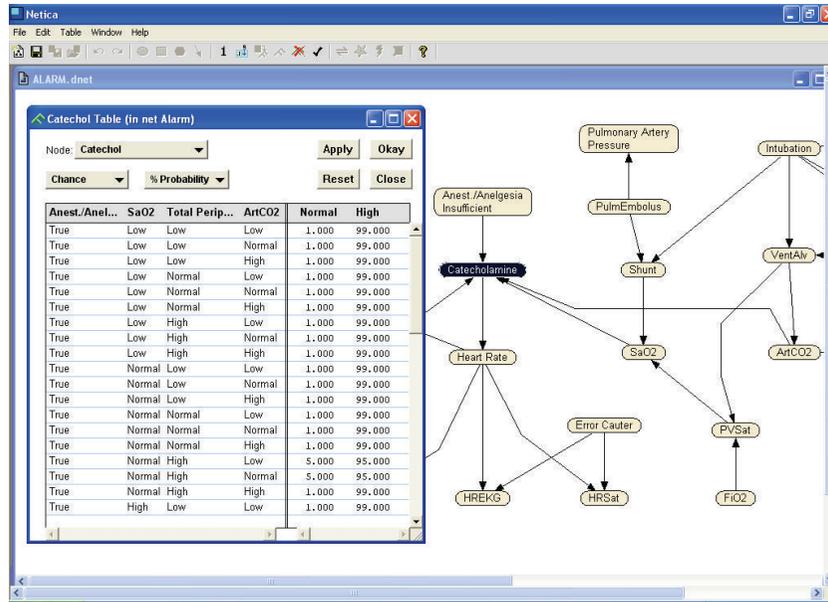


FIGURE 5: Conditional probability diagram for Alarm.dnet catechol variable.

TABLE 1: 11 Output variables for one record (100 datasets).

Variable name (disease)	Accuracy degree	Real situations	Results produced by the software	The comparison of the real situation and the result produced
History	0,9900	False	False	POSITIVE
Pres	0,9412	Normal	Zero	NEGATIVE
MinVol	0,9136	Normal	Zero	NEGATIVE
ExpCO2	0,9136	Normal	Zero	NEGATIVE
PAP	0,9000	Normal	Normal	POSITIVE
HRBP	0,8229	High	High	POSITIVE
HREKG	0,8229	High	High	POSITIVE
HRSat	0,8229	High	High	POSITIVE
CVP	0,7075	Normal	Normal	POSITIVE
PCWP	0,6970	Normal	Normal	POSITIVE
BP	0,4052	Low	Low	POSITIVE

TABLE 2: The class labels for 11 classification datasets.

Dependent variable (class)	Class labels
BP	Normal, low, high
CVP	Normal, low, high
ExpCO2	Normal, low, high, zero
History	False, true
HRBP	Normal, low, high
HREKG	Normal, low, high
HRSat	Normal, low, high
MinVol	Normal, low, high, zero
PAP	Normal, low, high
PCWP	Normal, low, high
Press	Normal, low, high, zero

TABLE 3: Used classification algorithms and abbreviations.

Algorithm name	Abbreviations
Zero rule	ZR
Naive-Bayes	NB
Multilayer perceptron	MLP
Simple logistic	SL
Support vector machines	SMO
One nearest neighbor	IBK
C4.5 decision tree	J48
Rep Tree	RT
Boosting	BS
Bagging	BG
Random forest	RF

TABLE 4: Classification accuracies with datasets having 10 samples (%).

	ZR	NB	MLP	SL	SMO	IBK	BS	BG	J48	RF	RT
BP	50.00	30.00	6.00	0.00	42.00	10.00	50.00	38.00	50.00	8.00	22.00
CVP	70.00	60.00	80.00	60.00	60.00	80.00	80.00	66.00	60.00	80.00	70.00
ExpCO2	50.00	50.00	60.00	50.00	52.00	50.00	50.00	46.00	50.00	54.00	20.00
History	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
HRBP	70.00	80.00	48.00	70.00	70.00	70.00	40.00	70.00	70.00	62.00	70.00
HREKG	60.00	70.00	42.00	48.00	60.00	40.00	30.00	58.00	30.00	44.00	60.00
HRSat	60.00	50.00	70.00	30.00	32.00	50.00	70.00	38.00	30.00	62.00	30.00
MinVol	70.00	70.00	80.00	70.00	72.00	70.00	80.00	70.00	70.00	72.00	70.00
PAP	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
PCWP	70.00	70.00	70.00	60.00	66.00	60.00	60.00	70.00	60.00	70.00	70.00
Press	70.00	70.00	80.00	70.00	70.00	70.00	80.00	70.00	70.00	70.00	70.00

TABLE 5: Classification accuracies with datasets having 100 samples (%).

	ZR	NB	MLP	SL	SMO	IBK	BS	BG	J48	RF	RT
BP	47.00	47.60	45.40	44.00	45.00	41.60	45.80	43.20	42.60	45.40	44.60
CVP	62.00	92.00	88.40	91.40	91.40	91.20	92.00	92.00	92.00	90.00	92.00
ExpCO2	65.00	70.80	73.20	74.00	69.20	69.00	65.00	67.60	69.20	71.60	66.20
History	98.00	98.00	100.00	98.00	98.00	98.00	100.00	98.00	98.00	98.40	98.00
HRBP	49.00	59.40	55.40	57.80	59.80	55.60	53.40	57.80	58.40	56.80	54.40
HREKG	48.00	55.60	56.80	54.80	56.80	54.20	54.00	50.80	53.20	54.60	50.20
HRSat	49.00	61.60	61.60	62.80	63.80	59.60	57.00	61.20	62.00	60.40	56.20
MinVol	79.00	84.00	86.20	84.60	85.20	82.00	79.00	79.00	82.00	83.40	79.00
PAP	88.00	88.00	86.80	88.00	88.00	88.00	88.00	88.00	88.00	87.40	88.00
PCWP	57.00	85.00	81.00	84.80	84.20	80.20	85.00	85.00	84.60	80.00	85.00
Press	80.00	85.00	89.80	87.20	85.20	84.00	80.40	82.20	85.00	85.60	81.40

TABLE 6: Classification accuracies with datasets having 1000 samples (%).

	ZR	NB	MLP	SL	SMO	IBK	BS	BG	J48	RF	RT
BP	45.00	46.32	44.52	46.44	46.58	44.58	45.00	47.18	46.52	44.84	46.60
CVP	67.70	87.64	87.00	87.72	87.08	87.14	85.60	87.70	87.70	87.06	87.68
ExpCO2	66.10	79.60	78.20	79.74	79.58	78.12	69.54	79.80	79.84	78.46	79.72
History	94.20	98.30	98.30	98.30	98.30	98.20	98.24	98.30	98.30	98.30	98.30
HRBP	48.00	67.12	66.18	67.20	67.56	66.48	59.20	67.06	67.04	66.62	66.88
HREKG	47.00	66.36	65.66	66.06	67.06	65.14	59.00	65.60	65.88	65.52	65.74
HRSat	47.40	65.68	65.02	65.88	66.84	64.60	59.00	65.14	65.20	64.68	65.10
MinVol	79.30	88.84	87.80	88.82	88.76	87.30	83.20	88.86	88.82	88.18	88.88
PAP	89.40	90.20	89.48	90.20	90.20	89.38	90.20	90.20	90.20	89.50	90.16
PCWP	65.20	86.80	86.40	86.90	86.72	86.18	83.80	86.90	86.90	86.50	86.90
Press	78.40	88.86	88.46	89.34	89.18	87.90	82.90	89.26	88.98	88.70	89.24

meta-algorithms (boosting, bagging, and random forest) the ensemble sizes were selected as 100 to be sure from maximum accuracy.

6. Experimental Results

The performance of each classification algorithm was evaluated using 5 runs of 10-fold cross validation. In each 10-fold cross validation, each dataset is randomly split into 10 equal size segments and results are averaged over 50 (5 * 10)

trials. The classification results are divided by 4 according to the dataset’s sample size. Tables 4, 5, 6, and 7 show the averaged classification accuracies with experiments having 10, 100, 1000, and 2000 samples, respectively.

Figure 6, shows the classification accuracies changes with the datasets’ sample size. J48 decision tree is used as classifier in Figure 6.

As can be seen at Tables 4–7 and Figure 6 when the sample size increases it gives more accurate results, as expected. Zero rule defines accuracy by chance. It selects the most existent

TABLE 7: Classification accuracies with datasets having 2000 samples (%).

	ZR	NB	MLP	SL	SMO	IBK	BS	BG	J48	RF	RT
BP	45.40	47.37	45.63	46.88	46.64	46.31	45.40	46.93	47.08	46.38	46.62
CVP	69.30	88.65	88.48	88.64	88.65	88.46	86.80	88.60	88.55	88.53	88.63
ExpCO2	66.65	80.15	79.19	80.25	80.23	78.96	70.24	79.98	80.03	79.42	79.99
History	94.15	98.45	98.40	98.43	98.45	98.12	98.42	98.45	98.45	98.43	98.45
HRBP	49.75	66.47	66.72	66.57	66.78	66.57	58.20	67.12	67.41	66.71	67.20
HREKG	48.80	65.26	63.92	65.00	64.85	64.79	57.25	64.91	64.60	65.10	64.52
HRSat	48.95	65.05	64.41	65.03	65.58	65.40	57.40	64.90	65.26	65.51	64.95
MinVol	77.70	88.02	87.44	87.99	87.93	86.91	82.00	88.04	88.01	87.59	88.03
PAP	88.95	89.90	89.43	89.88	89.86	89.38	89.90	89.90	89.90	89.53	89.90
PCWP	65.45	86.95	86.53	86.95	86.95	86.67	83.55	86.95	86.95	86.67	86.95
Press	78.20	89.68	89.62	90.03	90.07	89.27	82.80	90.11	90.00	89.73	90.00

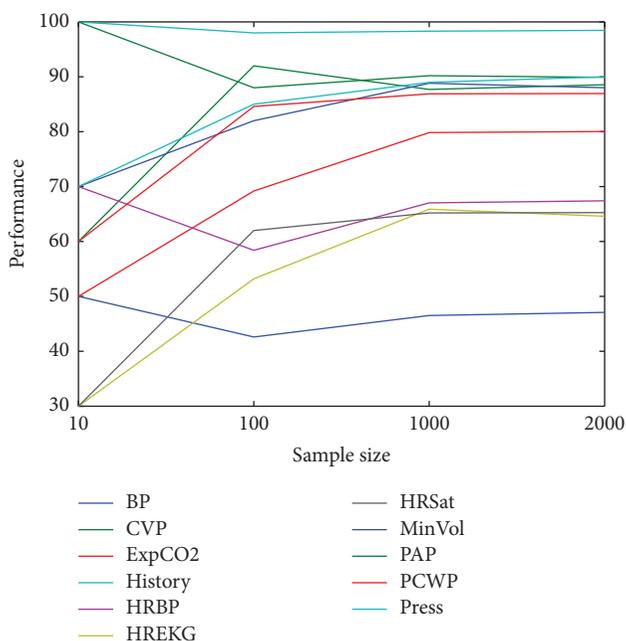


FIGURE 6: Classification accuracies changes with datasets' sample size.

class for all samples. In BP and PAP datasets, none of the algorithms won the zero rule. This means that the datasets can not be learned by any of the algorithms.

We compared the accuracies of all classification algorithms in a pairwise manner in Table 8. To compare two algorithms' performances, we employed the statistically significance difference test (paired *t*-test) with 0.05 significance level. The win/loss records in Table 8 are the number of wins and losses of the algorithm in the row over the method in the column. The number of ties is the sum of wins and losses subtracted from 11. For example, J48 won over MLP on 5 datasets and the algorithms have similar performances on other 6 datasets. For the comparison, the datasets having 2000 samples were only used.

In addition to statistical difference test, we also compared the classification algorithms according to their average ranks.

In the average rank comparison, for each of the datasets, the algorithms were ordered according to their performances. Then their ranks were averaged over 11 datasets. The average ranks and the sum of win and losses in Table 8 are given in Table 9.

According to Table 9, J48 (C4.5 decision tree) is the best ranked algorithm for our 11 datasets. The second one is bagging. According to the sum of wins, the best one is again J48.

To show the statistically meaningful difference between the average ranks we also applied the Nemenyi test [61]. According to is the Nemenyi test, the performance of two classifiers is significantly different if the corresponding average ranks differ by at least the critical difference (CD) calculated by

$$CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}} \tag{26}$$

In (26), *k* is the number of classifiers compared, *N* is the number of datasets, *q*_α is the critical value, and α is the significance level. In our experiments, the critical value (*q*_{0.05}) is 3.219 for 11 classifiers [62]. The critical difference (CD) is 3.129 * sqrt((11 * 12)/(6 * 11)) = 4.424. According to the Nemenyi test (at *P* < 0.05), there are no statistical differences between J48 and the algorithms having at most 4.424 + 3.55 = 7.974 average rank (NB, SL, SMO, BG, RF, and RT).

7. Conclusion

In cases of uncertainty and the lack of information, the most important part of the decision support systems which supports decision making process is the inference mechanism. There are data mining methods like SVM, MLP, decision trees, and so forth which are available in inference mechanism. Those methods can be used separately in an inference mechanism or also as a hybrid system, which consist of a combination of those methods.

In the study, for the generated synthetic data, ALARM network structure which is widely used in scientific studies has been used. This network structure is a structure that has been prepared using real patient information for many

TABLE 8: Pairwise comparison of accuracies (win/loss over 11 datasets) of all algorithms using 10 cv *t*-Test.

	ZR	NB	MLP	SL	SMO	IBK	BS	BG	J48	RF	RT
ZR	0/0	0/10	0/9	0/10	0/10	0/9	0/10	0/10	0/10	0/9	0/10
NB	10/0	0/0	3/0	0/0	0/0	5/0	8/0	0/0	0/1	3/0	0/0
MLP	9/0	0/3	0/0	0/3	0/3	2/0	8/1	0/4	0/5	0/0	0/4
SL	10/0	0/0	3/0	0/0	0/0	6/0	8/0	0/0	0/1	3/0	0/0
SMO	10/0	0/0	3/0	0/0	0/0	6/0	8/0	0/0	0/0	3/0	0/0
IBK	9/0	0/5	0/2	0/6	0/6	0/0	8/2	0/6	0/7	0/2	0/6
BS	10/0	0/8	1/8	0/8	0/8	2/8	0/0	0/8	0/8	1/8	0/8
BG	10/0	0/0	4/0	0/0	0/0	6/0	8/0	0/0	0/0	2/0	0/0
J48	10/0	1/0	5/0	1/0	0/0	7/0	8/0	0/0	0/0	4/0	0/0
RF	9/0	0/3	0/0	0/3	0/3	2/0	8/1	0/2	0/4	0/0	0/2
RT	10/0	0/0	4/0	0/0	0/0	6/0	8/0	0/0	0/0	2/0	0/0

TABLE 9: The average ranks of the algorithms over 11 datasets “and the sum of win/losses.”

Algorithm name	ZR	NB	MLP	SL	SMO	IBK	BS	BG	J48	RF	RT
Average rank	11	4.27	8.27	4.64	3.73	8	9.27	3.64	3.55	5.9	3.73
The number of wins/losses (over 110)	0/97	29/1	19/23	30/1	30/0	17/42	14/72	30/0	36/0	19/18	30/0

variables and shows the possibilities derived from the real life circumstances.

In this study, the performances of 11 machine learning algorithms (SVM, MLP, C4.5, etc.) are tested on 44 synthetic data sets (11 different dependent variables and 4 different dataset sizes). The comparison of algorithms we applied two different tests (statistically difference and average rank). C4.5 decision tree is the best algorithm according to the both of the tests for our 44 datasets. The datasets having more samples can be better predicted than having fewer samples.

In the future study, the comparison of the performances of the hybrid methods, which are combinations of the rule-based methods, and the data-driven methods and other machine learning systems will be carried out.

Conflict of Interests

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

References

- [1] E. H. Shortliffe and B. G. Buchanan, “A model of inexact reasoning in medicine,” *Mathematical Biosciences*, vol. 23, pp. 351–379, 1975.
- [2] E. H. Shortliffe, “Clinical decision based on physician-computer interactions: a symbolic reasoning approach,” in *Proceedings of the Annual Meeting Society for Computer Medicine*, Las Vegas, Nev, USA, 1977.
- [3] J. R. Clarke, D. P. Cebula, and B. L. Webber, “Artificial intelligence: a computerized decision aid for trauma,” *Journal of Trauma*, vol. 28, no. 8, pp. 1250–1254, 1988.
- [4] M. Zorman, P. Kokol, and G. Cerkvencik, “Decision trees and automatic learning in medical decision making,” in *Proceedings of the IAESTED International Conference on Intelligent Information Systems (IIS '97)*, p. 37, 1997.
- [5] I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper, “The ALARM monitoring system: a case study with two probabilistic inference techniques for belief networks,” in *Proceedings of the Second European Conference on Artificial Intelligence in Medical Care*, vol. 38, pp. 247–256, Springer, Berlin, Germany, 1989.
- [6] A. Wakulicz-Deja and P. Paszek, “Diagnose progressive encephalopathy applying the rough set theory,” *International Journal of Medical Informatics*, vol. 46, no. 2, pp. 119–127, 1997.
- [7] H. Yan, Y. Jiang, J. Zheng, C. Peng, and Q. Li, “A multilayer perceptron-based medical decision support system for heart disease diagnosis,” *Expert Systems with Applications*, vol. 30, no. 2, pp. 272–281, 2006.
- [8] G. Kose, H. Sever, M. Bal, and A. Ustundag, “Comparison of different inference algorithms for medical decision making,” *International Journal of Computational Intelligence Systems*, vol. 7, pp. 29–44, 2014.
- [9] J. Cheng, D. Bell, and W. Liu, “Learning Bayesian networks from data: an efficient approach based on information theory,” Tech. Rep., 1998, <http://webdocs.cs.ualberta.ca/~jcheng/Doc/report98.pdf>.
- [10] M. J. Druzdzal and R. R. Flynn, *Encyclopedia of Library and Information Science*, Taylor & Francis, New York, NY, USA, 3rd edition, 2010.
- [11] M. E. Corapcioglu, *Core of medical decision support system [M. Sc. thesis]*, Baskent University, Ankara, Turkey, 2006, (Turkish).
- [12] F. Hayes-Roth, “Rule-Based Systems,” *Communications of the ACM*, vol. 28, no. 9, pp. 921–932, 1985.
- [13] E. H. Shortliffe, S. G. Axline, B. G. Buchanan, T. C. Merigan, and S. N. Cohen, “An Artificial Intelligence program to advise physicians regarding antimicrobial therapy,” *Computers and Biomedical Research*, vol. 6, no. 6, pp. 544–560, 1973.
- [14] S. C. Yucebas, *HIPPOCRATES-I: medical diagnosis support system based on Bayesian network [M.S. thesis]*, Baskent University, Ankara, Turkey, 2006, (Turkish).

- [15] A. Beygelzimer, J. Langford, and B. Zadrozny, "Machine learning techniques-reductions between prediction quality metrics," in *Performance Modeling and Engineering*, Z. Liu and C. H. Xia, Eds., pp. 1–27, Springer, New York, NY, USA, 2008.
- [16] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth, Belmont, Mass, USA, 1984.
- [17] S. Murthy, S. Kasif, S. Salzberg, and R. Beigel, "OCI: randomized induction of oblique decision trees," in *Proceedings of the 11th National Conference on Artificial Intelligence*, pp. 322–327, MIT Press, Washington, DC, USA, July 1993.
- [18] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, Calif, USA, 1993.
- [19] G. Ssali and T. Marwala, "Computational intelligence and decision trees for missing data estimation," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 201–207, Hong Kong, June 2008.
- [20] A. Dehzeni, S. Phon-Amnualsuk, M. Manafi, and S. Safa, "Using rotation forest for protein fold prediction problem: an empirical study," in *Proceedings of the 8th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBIO '10)*, C. Pizzuti, M. D. Ritchie, and M. Giacobini, Eds., pp. 217–227, Springer, Berlin, Germany, 2010.
- [21] 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.
- [22] V. Havel, J. Martinovic, and V. Snasel, "Creating of conceptual lattices using multilayer perceptron," in *Proceedings of the International Workshop on Concept Lattices and Their Applications (CLA '05)*, R. Belohlavek and V. Snasel, Eds., pp. 149–157, Olomouc, Czech Republic, 2005.
- [23] A. Burak Goktepe, E. Agar, and A. Hilmi Lav, "Role of learning algorithm in neural network-based backcalculation of flexible pavements," *Journal of Computing in Civil Engineering*, vol. 20, no. 5, pp. 370–373, 2006.
- [24] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2nd edition, 2001.
- [25] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [26] B. Keshari and S. M. Watt, "Hybrid mathematical symbol recognition using support vector machines," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR '07)*, pp. 859–863, IEEE Computer Society, Curitiba, Brazil, September 2007.
- [27] C. Huang, Y. Lee, and D. K. J. Lin, "Model selection for support vector machines via uniform design," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 335–346, 2007.
- [28] E. Frias-Martinez, A. Sanchez, and J. Velez, "Support vector machines versus multi-layer perceptrons for efficient off-line signature recognition," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 6, pp. 693–704, 2006.
- [29] B. Tang and D. Mazzone, "Multiclass reduced-set support vector machines," in *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, W. W. Cohen and A. Moore, Eds., pp. 921–928, ACM, 2006.
- [30] J. N. S. Kwong and S. Gong, "Learning support vector machines for a multi-view face model," in *Proceedings of the British Machine Vision Conference (BMVC '99)*, T. P. Pridmore and D. Elliman, Eds., pp. 503–512, 1999.
- [31] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods-Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds., The MIT Press, Cambridge, Mass, USA, 1999.
- [32] H. Zhang, "The optimality of naïve bayes," in *Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference (FLAIRS '04)*, V. Barr and Z. Markov, Eds., pp. 17–19, AAAI Press, 2004.
- [33] P. Langley, W. Iba, and K. Thompson, "Analysis of Bayesian classifiers," in *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI '92)*, pp. 223–228, AAAI Press, San Jose, Calif, USA, July 1992.
- [34] J. Cheng and R. Greiner, "Comparing Bayesian network classifiers," in *Proceedings of the 15th International Conference on Uncertainty in Artificial Intelligence (UAI '99)*, K. Laskey and H. Prade, Eds., pp. 101–108, Morgan Kaufmann, San Francisco, Calif, USA, 1999.
- [35] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37–66, 1991.
- [36] D. W. Aha and D. Kibler, "Noise-tolerant instance-based learning algorithms," in *Proceedings of the 11th International Conference on Artificial Intelligence (IJCAI '89)*, pp. 794–799, Morgan Kaufmann, San Francisco, Calif, USA, 1989.
- [37] M. A. Hall, *Correlation-based feature selection for machine learning [Ph.D. thesis]*, The University of Waikato, Hamilton, New Zealand, 1999.
- [38] N. Yamaguchi, "Combining pairwise coupling classifiers using individual logistic regressions," in *Proceedings of the 13th International Conference on Neural Information Processing (ICONIP '06)*, I. King, J. Wang, L. Chan, and D. L. Wang, Eds., pp. 11–20, Springer, Berlin, Germany, 2006.
- [39] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Machine Learning*, vol. 59, no. 1-2, pp. 161–205, 2005.
- [40] M. Sumner, E. Frank, and M. Hall, "Speeding up logistic model tree induction," in *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD '05)*, A. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, Eds., pp. 675–683, Springer, Porto, Portugal, October 2005.
- [41] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.
- [42] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [43] R. A. Mc Donald, D. J. Hand, and I. A. Eckley, "An empirical comparison of three boosting algorithms on real data sets with artificial class noise," in *Proceedings of the 4th International Workshop on Multiple Classifier Systems (MCS '03)*, T. Windeat and F. Roli, Eds., pp. 35–44, Springer, Berlin, Germany, 2003.
- [44] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.
- [45] M. Sewell, *Ensemble Learning*, 2010, <http://machine-learning.martinsewell.com/ensembles/ensemble-learning.pdf>.
- [46] V. Nikulin, "Classification of imbalanced data with random sets and mean-variance filtering," *International Journal of Data Warehousing and Mining*, vol. 4, no. 2, pp. 63–78, 2008.
- [47] V. Nikulin, G. J. McLachlan, and S. K. Ng, "Ensemble approach for the classification of imbalanced data," in *Proceedings of the 22nd Australasian Joint Conference on Advances in Artificial Intelligence (AI '09)*, A. Nicholson and X. Li, Eds., pp. 291–300, Springer, Berlin, Germany, 2009.

- [48] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [49] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [50] J. R. Quinlan, "Simplifying decision trees," *International Journal of Man-Machine Studies*, vol. 27, no. 3, pp. 221–234, 1987.
- [51] T. Elomaa and M. Kääriäinen, "An analysis of reduced error pruning," *Journal of Artificial Intelligence Research*, vol. 15, pp. 163–187, 2001.
- [52] F. Esposito, D. Malerba, and G. Semeraro, "A comparative analysis of methods for pruning decision trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 476–491, 1997.
- [53] M. Kaariainen, T. Malinen, and T. Elomaa, "Selective Rademacher penalization and reduced error pruning of decision trees," *Journal of Machine Learning Research*, vol. 5, pp. 1107–1126, 2003/04.
- [54] Weka Manual for Version 3-6-3.
- [55] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Diego, Calif, USA, 1988.
- [56] S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Journal of the Royal Statistical Society B: Methodological*, vol. 50, no. 2, pp. 157–224, 1988.
- [57] Netica Software, http://www.norsys.com/netica_vb_api.htm.
- [58] Java Bayes Software, 2010, <http://www.cs.cmu.edu/~javabayes/Home>.
- [59] F. G. Cozman, "Generalizing variable elimination in bayesian network," in *Proceedings of the 7th Ibero-American Conference on Artificial Intelligence (IBERAMIA/SBIA '00)*, pp. 27–32, Sao Paulo, Brazil, 2000.
- [60] Weka Software, <http://www.cs.waikato.ac.nz/ml/weka>.
- [61] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
- [62] Critical values for the two-tailed Nemenyi test, http://www.cin.ufpe.br/~fatc/AM/Nemenyi_critval.pdf.

Research Article

A Hybrid Approach of Stepwise Regression, Logistic Regression, Support Vector Machine, and Decision Tree for Forecasting Fraudulent Financial Statements

Suduan Chen,¹ Yeong-Jia James Goo,² and Zone-De Shen²

¹ Department of Accounting Information, National Taipei University of Business, 321 Jinan Road, Section 1, Taipei 10051, Taiwan

² Department of Business Administration, National Taipei University, No. 67, Section 3, Ming-shen East Road, Taipei 10478, Taiwan

Correspondence should be addressed to Suduan Chen; suduanchen@yahoo.com.tw

Received 13 May 2014; Revised 22 August 2014; Accepted 23 August 2014; Published 11 September 2014

Academic Editor: Shifei Ding

Copyright © 2014 Suduan Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As the fraudulent financial statement of an enterprise is increasingly serious with each passing day, establishing a valid forecasting fraudulent financial statement model of an enterprise has become an important question for academic research and financial practice. After screening the important variables using the stepwise regression, the study also matches the logistic regression, support vector machine, and decision tree to construct the classification models to make a comparison. The study adopts financial and nonfinancial variables to assist in establishment of the forecasting fraudulent financial statement model. Research objects are the companies to which the fraudulent and nonfraudulent financial statement happened between years 1998 to 2012. The findings are that financial and nonfinancial information are effectively used to distinguish the fraudulent financial statement, and decision tree C5.0 has the best classification effect 85.71%.

1. Introduction

The financial statement is the main basis of decision-making by investors, creditors, and other accounting information demanders and concurrently also the concrete expression of management performance, financial condition, and possessing social responsibility of the listed and OTC companies, but the fraudulent financial statement (FFS) has the trend of becoming increasingly serious in recent years [1–8].

This behavior not only makes the investing public subject to vast amount of loss but also, more seriously, influences the capital market order. Because the fraudulent case is increasingly serious with each passing day, the United States Congress passed Sarbanes-Oxley Act in 2002 and mainly hope by which to improve the accuracy and reliability of the financial statement of a company and disclosure to make the auditors able to forecast the omen of the FFS before the FFS of an enterprise occurs. When one checks corporations' financial statements due to fraud which led to a significant

misstatement, there are fairly strict norms for audit staff in Taiwan [9].

The FFS can be regarded as a typical classification problem [10]. The classification problem carries out a computation mainly in light of the variable attribute numerical value of some given classification data to acquire the relevant classification rule of every classification and bring the unknown classification data into the rule to acquire the final classification result. Many authors apply the logistic regression to make a fraudulent classification and acquire the result in the FFS issue in the past [3, 6, 7, 11–13].

Data mining is an analytical tool used to handle a complicated data analysis. It discovers previously unknown information from mass data and aims for data to make an induction from the structured model as reference amount in making a decision with many different functions, such as classification, association, clustering, and forecasting [4, 5, 8, 14]. "Classification" function is used the most often therein, and its result can serve as the decision basis and prediction. However, whether every application of data mining in the

FFS is superior to the traditional classification model is controversial.

The purpose of this study is to expect that a better method of forecasting fraudulent financial statement can be presented to forecast the omen of the fraudulent financial statement and to reduce damage to the investors and auditors. The study will adopt the logistic regression and the support vector machine (SVM) as well as the decision tree (DT) C5.0 in data mining as the basis and match the stepwise regression to separately establish classification model to make a comparison. In conclusion, the study first aims at the “fraudulent financial statement” issue to make an arrangement for and carry out an exploration of relevant literature to ensure the research variable and sample adopted by the study. We then take the logistic regression, SVM, and DT C5.0 as the bases to establish the FFS classification model. Finally, we present the conclusions and suggestions of the study.

2. Literature Review

2.1. Fraudulent Definition. The FFS is a kind of intentional or illegal behavior, the result of which directly causes the seriously misleading financial statement or financial disclosure [2, 15]. Pursuant to the provision of SAS NO.99, a kind of fraudulent pattern is dishonest financial report, and it means a kind of intentional erroneous narration, neglecting amount or disclosure, which makes the misunderstood financial statement [6].

2.2. Research Method. The classification problem carries out a computation mainly in light of the variable attribute numerical value of some given classification data to acquire the relevant classification rule of every classification and bring the unknown classification data into the rule to acquire the final classification result. Many authors apply the logistic regression to make a fraudulent classification in the FFS issue in the past [3, 11, 12, 15–17]. However, the traditional statistic method has limitation of having to accord with specific assumption in data.

As a result, the machine learning way which does not require any statistic assumption about data portfolio rises abruptly. Many scholars recently try to adopt the machine learning way as the classification machine to conduct a research. The empirical result also points out that it possesses an excellent classification effect. Chen et al. [13] applied the neural network and SVM to forecast network invasion, and the research result indicates that the SVM has excellent classification ability. Huang et al. [18] applied the neural network and SVM to explore the classification model of credit evaluation. Shin et al. [19] conducted a relevant research of bankruptcy prediction. Yeh et al. [4] apply it in prediction of enterprise failure. On the other hand, Kotsiantis et al. [3] and Kirkos et al. [10] apply DT C5.0 in the relevant research to acquire the excellent classification result. Thus, the study will adopt the foresaid logistic regression, SVM, and DT C5.0 as the classifier construction classification model.

2.3. Variable Selection. As for variable selection via relevant literature exploration, some authors adopt the financial variable as the research variable [3, 10], others adopt the nonfinancial variable as the research variable [12, 16, 17], and still others adopt both the financial variable and nonfinancial variable as the research variable [15, 20].

Because financial statement data often have cheating suspicion, if we purely consider the financial variables, the possibility of erroneous classification may increase. Therefore, the study not only adopts the financial variable as the research variable, but also adds the nonfinancial variable to construct the fraudulent financial prediction model.

3. Methodology

The purpose of this study is to present a two-stage research model which integrates the financial variable and nonfinancial variable to establish the fraudulent early warning model of an enterprise. The procedure of the study is to aim at the data to make a stepwise regression analysis, to acquire the result of the important variable of the TTF after screening, and then to take such variable as the input variable of the logistic regression and SVM. Finally, the study makes a comparison and an analysis to acquire a better FFS classification result.

3.1. Stepwise Regression. The study selects a variable of the maximum classification ability in accordance with forward selection and incorporates the predictor into the model by stepwise increase. During each process, P value of the statistic test is used to screen the variables. If P value is less than or equal to 0.05, then the variable enters the regression model, and the selected variable is the independent variable of the regression model.

3.2. Logistic Regression. The logistic regression resembles the linear regression, while the response variable and explanatory variable of the general linear regression are usually the continuous variable, but the response variable explored by the logistic regression is the discrete variable; that is, it handles the qualitative variable of the two-dimensional independent variable problem (e.g., yes or no and success or failure). The model utilizes cumulative probability density function to convert real number value of the explanatory variable to probability value between 0 and 1. The elementary assumption is different from the analytic assumption of another multivariate analysis. The influence of the explanatory variable on the response variable is to fluctuate in the index form, which means that the logistic regression does not need to conform to the normal distribution assumption. In other words, it can handle the population of the nonnormal distribution and the problem of the nonlinear model and the nonmeasuring variable.

The general logistic regression model is as follows:

$$Y^* = \beta x + \varepsilon$$

$$Y = \begin{cases} 1: Y^* > 0 \\ 0: Y^* \leq 0 \end{cases}, \quad (1)$$

where Y : response variable of actual observation, $Y = 1$: a financial crisis event occurs, $Y = 0$: no financial crisis event occurs, Y^* : latent variable without observation, x : matrix of explanatory variable, β : matrix of explanatory variable parameter, and ε : error of explanatory variable.

3.3. Support Vector Machine (SVM). The operation model of the SVM projects the initial input vector to eigenspace of the high dimension with linear and nonlinear core function and utilizes the separating hyperplane to distinguish two or many materials of different classes. The SVM utilizes the hyperplane classifier to classify the materials.

3.3.1. Linear Divisibility. When the plain formed by the training sample data is linear, which consider the training vector: $x_i = (x_i^{(1)}, \dots, x_i^{(n)}) \in R^n$ belongs to two classes $y_i \in \{-1, +1\}$. In order to definitely distinguish the training vector class, it is necessary to find out the optimal partition hyperplane able to separate the materials.

If the hyperplane $w \cdot x + b$ can separate the training sample, it is shown as

$$w \cdot x_i + b > 0, \quad \text{if } y_i = 1, \tag{2}$$

$$w \cdot x_i + b < 0, \quad \text{if } y_i = -1. \tag{3}$$

Adjust w and b properly; (2) and (3) can be rewritten as

$$w \cdot x_i + b \geq 1, \quad \text{if } y_i = 1, \tag{4}$$

$$w \cdot x_i + b \leq -1, \quad \text{if } y_i = -1.$$

or as

$$y_i (w \cdot x_i + b) \geq 1, \quad \forall i \in \{1, \dots, n\}. \tag{5}$$

Pursuant to the statistics theory, the best interface not only separates two classes of samples correctly, but also maximizes the classification margin. The class margin of the interface $w \cdot x + b$ is shown as

$$d(w, b) = \min_{\{x_i|y_i=1\}} \frac{w \cdot x_i + b}{|w|} - \max_{\{x_i|y_i=-1\}} \frac{w \cdot x_i + b}{|w|}. \tag{6}$$

Equation (7) can be acquired from (4):

$$d(w, b) = \frac{1}{|w|} - \frac{-1}{|w|} = \frac{2}{|w|}. \tag{7}$$

So the problem of the maximization class margin $d(w, b)$ transforms to minimization $|w|^2/2$ under constraint condition (5). Pursuant to Lagrange relaxation, the foresaid problem must accord with the hypothesis of (8) and (9). In the foresaid condition, the minimization is shown as (10):

$$\alpha_i \geq 0, \tag{8}$$

$$\sum_i \alpha_i y_i = 0, \tag{9}$$

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j, \quad i = 1, \dots, n. \tag{10}$$

Every α_i corresponds to a training sample x_i , and the training sample of its corresponding $\alpha_i > 0$ is called the support vector. Classification function acquired finally is shown as

$$f(x) = \text{sgn}(w \cdot x_i + b) = \text{sgn}\left(\sum_{i=1}^{N_s} \alpha_i y_i x_i \cdot x + b\right), \tag{11}$$

where N_s is the number of the support vector.

3.3.2. Linear Indivisibility. If the training sample is linearly indivisible, (4) can be rewritten as

$$w \cdot x_i + b \geq 1 - \xi_i, \quad \text{if } y_i = 1 \tag{12}$$

$$w \cdot x_i + b \geq \xi_i - 1, \quad \text{if } y_i = -1.$$

where $\xi_i \geq 0, i = 1, \dots, n$.

If x_i is classified mistakenly, then $\xi_i > 1$. Thus, the mistaken classification is less than $\sum_i \xi_i$. Add a given parameter value in the objective function. Consider reasonably the maximum class margin and the minimum mistaken class sample; that is, seeking the minimum of $|w|^2/2 + C(\sum_i \xi_i)$ can acquire the SVM under linear indivisibility. Pursuant to Lagrange relaxation, the foresaid problem must accord with the hypothesis of (13) and (14). In the foresaid condition, the minimization is shown as (15):

$$0 \leq \alpha_i \leq C, \tag{13}$$

$$\sum_i \alpha_i y_i = 0, \tag{14}$$

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j, \quad i = 1, 2, 3, \dots, n. \tag{15}$$

3.4. Decision Tree (DT). The Decision Tree (DT) is the simplest in the inductive learning method [21]. It belongs to the data mining tool and can handle the continuous and noncontinuous variable. It establishes the tree structure diagram mainly by the given classification fact and induces some principles therein. The principles are mutually exclusive, and the DT generated can also make an out-of-sample prediction. The DT algorithms used most frequently include CART, CHAID, and C5.0 [22]. C5.0 [23] improves from ID3 [23]. Thanks to ID3 use limitation, it cannot handle the continuous numerical value materials; thus, Quinlan conducts a research for improvement, and C5.0 is developed to handle the continuous and the noncontinuous numerical value.

The DT C5.0 is mainly separated into two parts. The first part is classification criterion, which is calculated pursuant to the gain ratio. Construct the DT completely as shown in (2). Information gained in (16) is used to calculate the pretest and posttest gain of the data set and is defined as "pretest information" minus "postinformation" from (17). The entropy in (16) is used to calculate impurity, which is called randomness. In other words, it is used to calculate

TABLE 1: Results of stepwise regression variable screening.

Variable code	Variable classification	Variable description	Pr > ChiSq
X1	Financial	Accounts receivables/total assets	0.2401
X3	Financial	Inventory/current assets	0.0339
X10	Financial	Interest protection multiples	0.0694
X13	Financial	Debt ratio	0.0294
X15	Financial	Cash flow ratio	0.0025
X17	Financial	Accounts payable turnover	0.0295
X24	Financial	Operation profit/last year operation profit >1.1	0.0267
X29	Nonfinancial	Pledge ratio of shares of the directors and supervisors	0.0473

TABLE 2: Hit ratio of three models using the train datasets.

Research model	C5.0	Logistic	SVM
Hit ratio	93.94%	83.33%	78.79%

randomness of the data set. When randomness in the data set reaches the most disorderly state, the value will be 1.

Therefore, the less random the posttest data set is, the larger the information gain is calculated, and the more favorable it is for DT construction:

$$\text{Gain Ratio}(S, A) = \frac{\text{Information } n \text{ Gain}(S, A)}{\text{Entropy}(S, A)} \quad (16)$$

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v). \quad (17)$$

The second part is pruning criterion. Pursuant to the error based pruning (EBP), the DT is properly pruned to enhance the correct ratio of classification. EBP is evolved from the pessimistic error pruning (PEP), and such two pruning methods are presented by Quinlan. The main concept of the EBP is to make a judgment using the error ratio, calculate the error ratio of every node, and further judge the node which results in rise of the error ratio of the overall DT. Finally, this node is pruned properly to further enhance the correct ratio of the DT.

3.5. *Definition of Type I Error and Type II Error.* In order to establish the valid forecasting fraudulent financial statement, it is considerably important to measure type I type II errors of the study. Type I error is to mistakenly judge the normal financial statement company as the FFS company. This judgment does not cause investors' damage, but it carries out an erroneous audit opinion for being too conservative and further influences credit of the company audited. Type II error is that the FFS enterprise is mistaken for the normal enterprise. This classification error leads to auditing failure, auditors' investment loss, or investors' erroneous judgment.

4. Empirical Analysis

4.1. *Data Collection and Variables.* The research samples are the FFS enterprises from the years 1998 to 2012. 66 enterprises are selected from the listed and OTC companies of the Taiwan Economic Journal Data Bank (TEJ). The 1 by 1 pair way is adopted to match 66 normal enterprises, so there are 132 enterprises in total as research samples.

As for selection of the research variables, the study altogether selects 29 variables, including 24 financial variables and 5 nonfinancial variables (see appendix).

For consideration of the number of samples, to avoid having too few samples of the test group and to improve test accuracy, we propose to utilize 50% of the sample materials as the train sample to establish the regression classification model. The remaining 50% of the sample materials serve as the test sample to test validity of the classification model established.

In addition, to test the stability of the proposed research model, this study randomly selects three groups at a ratio of 80% from the test data as the test sample for cross-validation. The compartment and sampling of data in this research are shown in Figure 1.

4.2. *Model Development.* To begin with, the study aims for the financial and nonfinancial variable to screen using the stepwise regression screening method. The variables screened serve as the input variable of the logistic regression and SVM. Next, the study aims at every method to carry out the model training and test. Finally, the study compares the merit and demerit of the classification correct ratio and gives the relevant suggestions for the analytic result. The model construction is divided into three parts. The first part is the variable screening way; the second part is the classification way; the third part compares the test results of two kinds of classification models. The research process of the study is shown as Figure 2.

4.3. *Important Variable Screening.* While constructing the classification model, there may be quite many variables, but not every variable is important. Therefore, the variables of no account need to be eliminated to construct a simpler classification model. There are quite many variable screening

TABLE 3: C5.0 cross-validation results.

C5.0 model			Predict value		Hit ratio	Type I error	Type II error
			Non-FFS	FFS			
Actual value	CV1	Non-FFS	25	3	83.93%	10.71%	21.42%
		FFS	6	22			
	CV2	Non-FFS	25	3	87.50%	10.71%	14.28%
		FFS	4	24			
	CV3	Non-FFS	25	3	85.71%	10.71%	17.85%
		FFS	5	23			
	Average		25	3	85.71%	10.71%	17.85%
			5	23			

TABLE 4: Logistic regression cross-validation results.

Logistic regression model			Predict value		Hit ratio	Type I error	Type II error
			Non-FFS	FFS			
Actual value	CV1	Non-FFS	25	3	80.36%	10.71%	28.57%
		FFS	8	20			
	CV2	Non-FFS	26	2	82.14%	7.14%	28.57%
		FFS	8	20			
	CV3	Non-FFS	25	3	80.36%	10.71%	28.57%
		FFS	8	20			
	Average		25	3	80.95%	9.52%	28.57%
			8	20			

ways, among which the stepwise regression variable screening method is used most frequently [24].

Therefore, the study adopts the suggestions of Pudil et al. [24] to screen the variables using the stepwise regression by which to retain the research variables with more influence. The input variables of the study are screened via the stepwise regression to acquire the results as shown in Table 1, including 7 financial variables and 1 nonfinancial variable. Subsequently, the study takes these 8 variables as the new input variables to construct the classification model.

4.4. Classification Model. The prediction accuracy of the three types of models using the train datasets is displayed in Table 2.

As shown in Table 2, C5.0 has the best performance in the establishment of the prediction model and its accuracy rate is 93.94%. The traditional logistic model is the second best. The accuracy rate of the SVM model, at 78.79%, is the lowest of the three. The cross-validation results of the proposed three prediction models are shown in Tables 3 to 5.

4.4.1. Decision Tree (DT). The study constructs the DT C5.0 model, sets EBP at $\alpha = 5\%$, and adopt the binary partition principle to obtain the optimal spanning tree. The prediction results of the DT C5.0 classification model are shown as Table 3.

On average, 25 of the 28 non-FFS materials are correctly classified in the non-FFS, and three of them are incorrectly classified in the FFS. The type I error is 10.71%. On the other hand, 23 of the 28 FFS materials are correctly classified, and

the remaining five FFS materials are incorrectly classified in the non-FFS. The type II error is 17.85%.

4.4.2. Logistic Regression. Table 4 is the empirical results of the logistic classification model, which shows that 25 of 28 non-FFS materials are correctly classified and that three of them are incorrectly classified in the FFS. The overall type I error is 9.52%. In addition, 20 of the 28 FFS materials are correctly classified, and the remaining eight FFS materials are incorrectly classified in the non-FFS. The type II error is 28.57%.

4.4.3. Support Vector Machine (SVM). The operation core is set at RBF when the study constructs the SVM model. As for the parameter, the C search scope is set at 2^{-10} to 2^{10} , and γ is set at 0.1. The SVM classification results are shown as Table 5.

In this part, 26 of the 28 non-FFS materials are correctly classified, and two of them are incorrectly classified in the FFS. The type I error is 7.14%. In addition, 14 of the 28 FFS materials are correctly classified, and the remaining 14 FFS materials are incorrectly classified in the non-FFS. The type II error is 48.81%.

4.4.4. Comprehensive Comparison and Analysis. Kirkos et al. [10] pointed out that the merit and demerit of the evaluation model must also consider the type I error and type II error. The type I error means to classify the nonfraudulent companies into the fraudulent companies. Occurrence of these two type errors results from the auditing failure of the auditors. Type II error means that the auditors classify

TABLE 5: SVM cross-validation results.

	SVM model		Predict value		Hit ratio	Type I error	Type II error
			Non-FFS	FFS			
Actual value	CV1	Non-FFS	26	2	73.21%	7.14%	46.42%
		FFS	13	15			
	CV2	Non-FFS	26	2	71.43%	7.14%	50.00%
		FFS	14	14			
	CV3	Non-FFS	26	2	71.43%	7.14%	50.00%
		FFS	14	14			
Average	Non-FFS	26	2	72.02%	7.14%	48.81%	
	FFS	14	14				

TABLE 6: Summary of classification results.

Model	Type I error	Type II error	Hit ratio	Ranking
Logistic regression	9.52%	28.57%	80.95%	2
SVM	7.14%	48.81%	72.02%	3
DT C5.0	10.71%	17.85%	85.71%	1

TABLE 7: Paired-samples *t* test.

Model	<i>t</i> -value	DF	Significant (two-tailed)
C5.0—logistic	-5.201	2	0.35
Logistic—SVM	-16.958	2	0.03
SVM—C5.0	9.823	2	0.10

the fraudulent companies into the nonfraudulent companies. Both types of error would cause different loss costs, and the auditors must avoid occurrence of these two errors. Comparing the results of these three models, we conclude that the classification ability of the DT C5.0 is the best, the next is the logistic regression, and the last is the SVM. The classification correct ratios of three kinds of model are summarized as shown in Table 6.

The comparison shows that, although the logistic classification model performs the best for type I errors, the DT C5.0 possesses the best classification effect, both for type II errors and the hit ratio. The correct classification ratio is 85.71%, followed by 80.95% for the logistic model, and 72.02% for the SVM model.

Unlike general studies using type I errors to judge the performance of prediction models, FFS studies use type II errors to determine the performance of prediction models. For the sake of prudence, we conduct the statistical test of type II errors in the abovementioned cross-validation results to confirm whether the differences in between models are significantly other than 0. The analysis results are shown in Table 7, which shows that the *t*-values of the prediction model type II error differences are -5.201 (C5.0—Logistic); -16.958 (Logistic—SVM); and 9.823 (SVM—C5.0), respectively, and all of them reach the significance level.

5. Conclusion and Suggestion

As the fraudulent financial statement (FFS) increases on the trot in recent years, the auditing failure risk of the auditors also rises thereby. Therefore, many researches focus on developing a good classification model to reduce the relevant risk. In the past, the accuracy of forecasting FFS purely using regression analysis has been relatively low. Many scholars have pointed out that prediction by data mining can improve the accuracy rate. Thus, this study adopts stepwise regression to screen the important factors of financial and nonfinancial variables. Meanwhile, it combines the above with data mining techniques to establish a more accurate FFS forecast model.

A total of eight critical variables are screened via the stepwise regression analysis, including two parts: financial variables (accounts receivables/total assets, inventory/current assets, interest protection multiples, cash flow ratio, accounts payable turnover, operation profit/last year operation profit > 1.1) and nonfinancial variables (pledge ratio of shares of the directors and supervisors).

The financial variables include operating capabilities, profitability index, debt solvency ability index, and financial structure. The nonfinancial variables include relevant variables of stock rights and scale of an enterprise’s directors and supervisors. The results indicate that when auditors investigate FFS, they must focus on the alert provided by the nonfinancial information as well as the financial information.

In the classification model, the study adopts the logistic regression of the traditional classification method and the DT C5.0 and SVM of data mining to construct the classification model. The empirical result indicates that the SVM model performs the best in the type I error after comparison, and the DT C5.0 has the best classification performance in the type II error and overall classification correct ratio.

TABLE 8: Selection of the research variables.

Variable classification	Variable code	Variable description and computation
Financial variables	X1	Accounts receivables/total assets
	X2	Gross profit/total assets
	X3	Inventory/current assets
	X4	Inventory/total assets
	X5	Net profit after tax/total assets
	X6	Net profit after tax/fixed assets
	X7	Cash/total assets
	X8	Log total assets
	X9	Log total liabilities
	X10	Interest protection multiples (debt service coverage ratio, times interest earned)
	X11	Gross profit margin
	X12	Operating expense ratio
	X13	Debt ratio
	X14	Inventory turnover
	X15	Cash flow ratio
	X16	Net profit ratio before tax
	X17	Accounts payable turnover
	X18	Revenue growth rate
	X19	Debt/equity ratio
	X20	Earnings before interest, taxes, depreciation, and amortization
	X21	Current liabilities/total assets
	X22	Total assets turnover
	X23	Account receivable/last year accounts receivable >1.1
	X24	Operation profit/last year operation profit >1.1
Nonfinancial variables	X25	Shareholding ratio of the major shareholders
	X26	Shareholding ratio of directors and supervisors
	X27	Whether the chairman concurrently holds the position of CEO
	X28	Board size
	X29	Pledge ratio of shares of the directors and supervisors

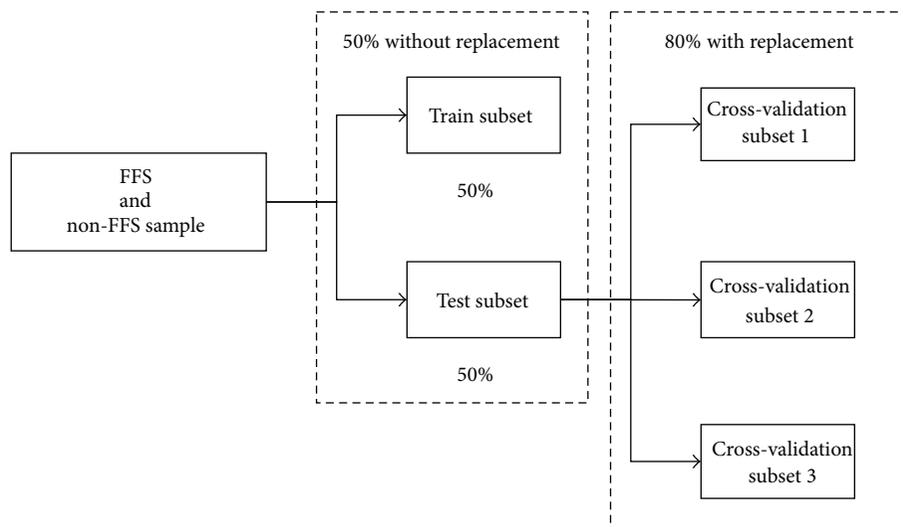


FIGURE 1: Train and test subsets design.

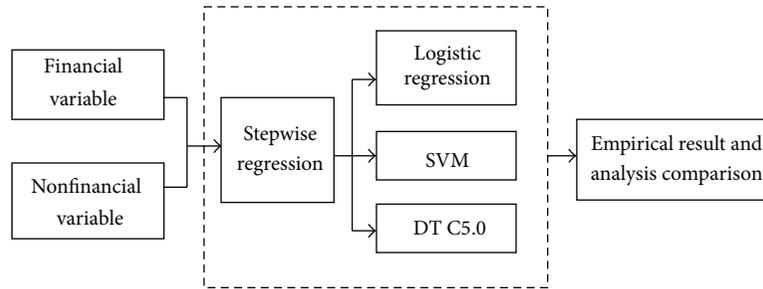


FIGURE 2: Research model.

One of the research purposes is to anticipate accommodating the auditors with another assistant auditing tool besides the traditional analysis method, but the research about the forecasting FFS is not sufficient. Therefore, the subsequent researchers can also adopt other methods to forecast the FFS to provide a better reference. In addition, future researchers can also try to adopt different variable screening methods to enhance the classification correct ratio of the method. As for the variable, some nonfinancial variables are difficult to measure, and material acquisition is difficult, so the study does not incorporate them. Finally, as for the sample, the study focuses on the FFS scope research, and a certain number of the FFSs may not be found. Therefore, the pair companies can also be the FFS companies in the coming year, which can influence the accuracy of the study. The findings of this study can provide a reference to auditors, certified public accountants (CPAs), securities analysts, company managers, and future academic studies.

Appendix

See Table 8.

Conflict of Interests

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

References

- [1] C. Spathis, M. Doumpos, and C. Zopounidis, "Detecting falsified financial statements: a comparative study using multi-criteria analysis and multivariate statistical techniques," *The European Accounting Review*, vol. 11, pp. 509–535, 2002.
- [2] Z. Rezaee, "Causes, consequences, and deterrence of financial statement fraud," *Critical Perspectives on Accounting*, vol. 16, no. 3, pp. 277–298, 2005.
- [3] S. Kotsiantis, E. Koumanakos, D. Tzelepis, and V. Tampakas, "Forecasting fraudulent financial statements using data mining," *Transactions on Engineering Computing and Technology*, vol. 12, pp. 283–288, 2006.
- [4] C.-C. Yeh, D.-J. Chi, and M.-F. Hsu, "A hybrid approach of DEA, rough set and support vector machines for business failure prediction," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1535–1541, 2010.
- [5] W. Zhou and G. Kapoor, "Detecting evolutionary financial statement fraud," *Decision Support Systems*, vol. 50, no. 3, pp. 570–575, 2011.
- [6] S. L. Humpherys, K. C. Moffitt, M. B. Burns, J. K. Burgoon, and W. F. Felix, "Identification of fraudulent financial statements using linguistic credibility analysis," *Decision Support Systems*, vol. 50, no. 3, pp. 585–594, 2011.
- [7] K. A. Kamarudin, W. A. W. Ismail, and W. A. H. W. Mustapha, "Aggressive financial reporting and corporate fraud," *Procedia-Social Behavioral Sciences*, vol. 65, pp. 638–643, 2012.
- [8] P.-F. Pai, M.-F. Hsu, and M.-C. Wang, "A support vector machine-based model for detecting top management fraud," *Knowledge-Based Systems*, vol. 24, no. 2, pp. 314–321, 2011.
- [9] Accounting Research and Development Foundation, *Audit the Financial Statements of the Considerations for Fraud*, Accounting Research and Development Foundation, Taipei, Taiwan, 2013.
- [10] S. Kirkos, C. Spathis, and Y. Manolopoulos, "Data mining techniques for the detection of fraudulent financial statements," *Expert Systems with Applications*, vol. 32, no. 4, pp. 995–1003, 2007.
- [11] T. B. Bell and J. V. Carcello, "A decision aid for assessing the likelihood of fraudulent financial reporting," *Auditing*, vol. 19, pp. 169–178, 2000.
- [12] V. D. Sharma, "Board of director characteristics, institutional ownership, and fraud: evidence from Australia," *Auditing*, vol. 23, no. 2, pp. 105–117, 2004.
- [13] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection," *Computers and Operations Research*, vol. 32, no. 10, pp. 2617–2634, 2005.
- [14] J. W. Seifert, "Data mining and the search for security: challenges for connecting the dots and databases," *Government Information Quarterly*, vol. 21, no. 4, pp. 461–480, 2004.
- [15] M. S. Beasley, "An empirical analysis of the relation between the board of director composition and financial statement fraud," *Accounting Review*, vol. 71, no. 4, pp. 443–465, 1996.
- [16] P. Dunn, "The impact of insider power on fraudulent financial reporting," *Journal of Management*, vol. 30, no. 3, pp. 397–412, 2004.
- [17] G. Chen, "Positive research on the financial statement fraud factors of listed companies in China," *Journal of Modern Accounting and Auditing*, vol. 2, pp. 25–34, 2006.
- [18] Z. Huang, H. Chen, C.-J. Hsu, W.-H. Chen, and S. Wu, "Credit rating analysis with support vector machines and neural networks: a market comparative study," *Decision Support Systems*, vol. 37, no. 4, pp. 543–558, 2004.

- [19] K.-S. Shin, T. S. Lee, and H.-J. Kim, "An application of support vector machines in bankruptcy prediction model," *Expert Systems with Applications*, vol. 28, no. 1, pp. 127–135, 2005.
- [20] S. L. Summers and J. T. Sweeney, "Fraudulently misstated financial statements and insider trading: an empirical analysis," *The Accounting Review*, vol. 73, no. 1, pp. 131–146, 1998.
- [21] G. Arminger, D. Enache, and T. Bonne, "Analyzing credit risk data: a comparison of logistic discrimination, classification tree analysis, and feedforward networks," *Computational Statistics*, vol. 12, no. 2, pp. 293–310, 1997.
- [22] S. Viaene, G. Dedene, and R. A. Derrig, "Auto claim fraud detection using Bayesian learning neural networks," *Expert Systems with Applications*, vol. 29, no. 3, pp. 653–666, 2005.
- [23] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.
- [24] P. Pudil, K. Fukuda, K. Beránek, and P. Dvůrák, "Potential of artificial intelligence based feature selection methods in regression models," in *Proceedings of the IEEE 3rd International Conference on Computational Intelligence and Multimedia Application*, pp. 159–163, 1999.

Research Article

SVM-RFE Based Feature Selection and Taguchi Parameters Optimization for Multiclass SVM Classifier

Mei-Ling Huang,¹ Yung-Hsiang Hung,¹ W. M. Lee,² R. K. Li,² and Bo-Ru Jiang¹

¹ Department of Industrial Engineering and Management, National Chin-Yi University of Technology, No. 57, Sec. 2, Zhong-Shan Road, Taiping District, Taichung 41170, Taiwan

² Department of Industrial Engineering & Management, National Chiao-Tung University, No. 1001, Ta-Hsueh Road, Hsinchu 300, Taiwan

Correspondence should be addressed to Mei-Ling Huang; huangml@ncut.edu.tw

Received 20 June 2014; Revised 5 August 2014; Accepted 5 August 2014; Published 10 September 2014

Academic Editor: Shifei Ding

Copyright © 2014 Mei-Ling Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, support vector machine (SVM) has excellent performance on classification and prediction and is widely used on disease diagnosis or medical assistance. However, SVM only functions well on two-group classification problems. This study combines feature selection and SVM recursive feature elimination (SVM-RFE) to investigate the classification accuracy of multiclass problems for Dermatology and Zoo databases. Dermatology dataset contains 33 feature variables, 1 class variable, and 366 testing instances; and the Zoo dataset contains 16 feature variables, 1 class variable, and 101 testing instances. The feature variables in the two datasets were sorted in descending order by explanatory power, and different feature sets were selected by SVM-RFE to explore classification accuracy. Meanwhile, Taguchi method was jointly combined with SVM classifier in order to optimize parameters C and γ to increase classification accuracy for multiclass classification. The experimental results show that the classification accuracy can be more than 95% after SVM-RFE feature selection and Taguchi parameter optimization for Dermatology and Zoo databases.

1. Introduction

The support vector machine (SVM) is one of the important tools of machine learning. The principle of SVM operation is as follows: a given group of classified data is trained by the algorithm to obtain a group of classification models, which can help predict the category of the new data [1, 2]. Its scope of application is widely used in various fields, such as disease or medical imaging diagnosis [3–5], financial crisis prediction [6], biomedical engineering, and bioinformatics classification [7, 8]. Although SVM is an efficient machine learning method, its classification accuracy requires further improvement in the case of multidimensional space classification and dataset for feature interaction variables [9]. Regarding such problems, in general, feature selection can be applied to reduce data structure complexity in order to identify important feature variables as a new set of testing instances [10]. By feature selection, inappropriate, redundant, and noise data of each problem can be filtered to reduce

the computational time of classification and improve classification accuracy. The common methods of feature selection include backward feature selection (BFS), forward feature selection (FFS), and ranker [11]. Another feature selection method, support vector machine recursive feature elimination (SVM-RFE), can filter relevant features and remove relatively insignificant feature variables in order to achieve higher classification performance [12]. The research findings of Harikrishna et al. have shown that computation is simpler and can more effectively improve classification accuracy in the case of datasets after SVM-REF selection [13–15].

As SVM basically applies on two-class data [16], many scholars have explored the expansion of SVM on multiclass data [17–19]. However, classification accuracy is not ideal. There are many studies on choosing kernel parameters for SVM [20–22]. Therefore, this study applies SVM-RFE to sort the 33 variables for Dermatology dataset and 16 variables for Zoo dataset by explanatory power in descending order and selects different feature sets before using the Taguchi

TABLE 1: Feature information for Dermatology and Zoo databases.

	Dermatology	Zoo
Dataset characteristics	Multivariate	Multivariate
Attribute characteristics	Categorical, integer	Categorical, integer
Associated tasks	Classification	Classification
Area	Life	Life
Number of instances	366	101
Number of attributes	33	16
Number of class	6	7

parameter design to optimize Multiclass SVM parameters C and γ to improve the classification accuracy for SVM multiclass classifier.

This study is organized as follows. Section 2 describes the research data; Section 3 introduces methods used through this paper; Section 4 discusses the experiment and results. Finally, Section 5 presents our conclusions.

2. Study Population

This study used the Dermatology dataset from University of California at Irvine (UCI) and the Zoo database from its College of Information Technology and Computers to conduct experimental tests, parameter optimization, and classification accuracy performance evaluation, using the SVM classifier.

In medicine, dermatological diseases are diseases of the skin that have a serious impact on health. As frequently occurring types of diseases, there are more than 1000 kinds of dermatological diseases, such as psoriasis, seborrheic dermatitis, lichen planus, pityriasis, chronic dermatitis, and pityriasis rubra pilaris. The Dermatology dataset was established by Nilsel in 1998 and contains 33 feature variables and 1 class variable (6-class).

The dermatology feature variables and data summary are as shown in Table 1. The Dermatology dataset has eight omissions. After removing the eight omissions, we retained 358 (instances) for this study. The instances of data of various categories are psoriasis (Class 1): 111 instances, seborrheic dermatitis (Class 2): 71 instances, lichen planus (Class 3): 60 instances, pityriasis (Class 4): 48 instances, chronic dermatitis (Class 5): 48 instances, and pityriasis rubra pilaris (Class 6): 20 instances. The Zoo dataset contains 17 Boolean-valued attributes and 101 instances. The instances of data of various categories are as follows: bear, and so forth (Class 1) 41 instances; chicken, and so forth (Class 2) 20 instances; seasnake, and so forth (Class 3) 5 instances; bass, and so forth (Class 4) 13 instances; (Class 5) 4 instances; frog, and so forth (Class 6) 8 instances; and honeybee, and so forth (Class 7) 10 instances.

Before feature selection, we conducted feature attribute coding. The feature attribute coding of Dermatology and Zoo databases is as shown in Tables 2 and 3.

TABLE 2: Attributes of Dermatology database.

ID	Attribute
V1	Erythema
V2	Scaling
V3	Definite borders
V4	Itching
V5	Koebner phenomenon
V6	Polygonal papules
V7	Follicular papules
V8	Oral mucosal involvement
V9	Knee and elbow involvement
V10	Scalp involvement
V11	Family history
V12	Melanin incontinence
V13	Eosinophils in the infiltrate
V14	PNL infiltrate
V15	Fibrosis of the papillary dermis
V16	Exocytosis
V17	Acanthosis
V18	Hyperkeratosis
V19	Parakeratosis
V20	Clubbing of the rete ridges
V21	Elongation of the rete ridges
V22	Thinning of the suprapapillary epidermis
V23	Spongiform pustule
V24	Munro microabscess
V25	Focal hypergranulosis
V26	Disappearance of the granular layer
V27	Vacuolisation and damage of basal layer
V28	Spongiosis
V29	Saw-tooth appearance of retes
V30	Follicular horn plug
V31	Perifollicular parakeratosis
V32	Inflammatory mononuclear infiltrate
V33	Band-like infiltrate
V34	Age

3. Methodology

3.1. *Research Framework.* The research framework of the study is shown in Figure 1. Steps are as follows.

- (1) Database preprocessing: delete the omissions and feature variable coding for Dermatology and Zoo datasets. And there are 358 and 101 instances left for Dermatology and Zoo databases for further experiment, respectively.
- (2) Feature selection: apply SVM-RFE ranking according to the order of importance of the features, and determine the feature set that contributes to the classification.
- (3) Parameter optimization: apply Taguchi parameter design in the parameters (C & γ) optimization of a Multiclass SVM Classifier in order to enhance the classification accuracy for the multiclass dataset.

TABLE 3: Attributes of Zoo database.

ID	Attribute
V1	Hair
V2	Feathers
V3	Eggs
V4	Milk
V5	Airborne
V6	Aquatic
V7	Predator
V8	Toothed
V9	Backbone
V10	Breathes
V11	Venomous
V12	Fins
V13	Legs
V14	Tail
V15	Domestic
V16	Cat-size

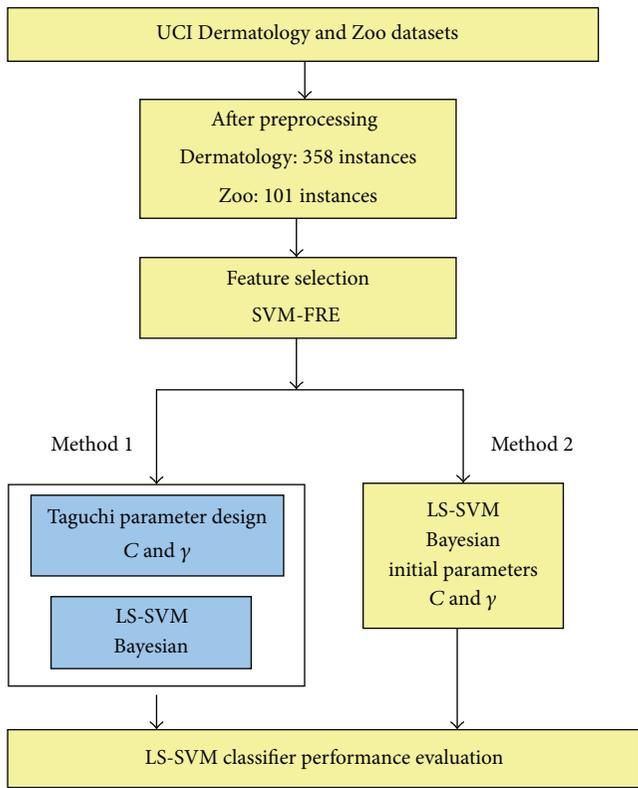


FIGURE 1: Research framework.

3.2. *Feature Selection.* Feature selection implies not only cardinality reduction, which means imposing an arbitrary or predefined cutoff on the number of attributes that can be considered when building a model, but also the choice of attributes, meaning that either the analyst or the modeling tool actively selects or discards attributes based on their usefulness for analysis. The feature selection method is a search strategy to select or remove some features of the

original feature set to generate various types of subsets to obtain the optimum feature subset. The subsets selected each time are compared and analyzed according to the formulated assessment function. If the subset selected in step $m + 1$ is better than the subset selected in step m , the subset selected in step $m + 1$ can be selected as the optimum subset.

3.3. *Linear Support Vector Machine (Linear SVM).* SVM is developed from statistical learning theory, as based on SRM (structural risk minimization). It can be applied on classification and nonlinear regression [6]. Generally speaking, SVM can be divided into linear SVM (linear SVM) and nonlinear SVM, described as follows.

(1) *Linear SVM.* The linear SVM encodes the training data of different types by classification with Class 1 as being “+1” and Class 2 as being “-1” and the mathematical symbol is $\{\{x_i, y_i\}_{i=1}^T, x_i \in \mathfrak{R}^m, y_i \in \{-1, +1\}\}$; the hyperplane is represented as follows:

$$w \cdot x + b = 0, \tag{1}$$

where w denotes weight vector, x denotes the input dataset, and b denotes a constant as a bias (displacement) in the hyperplane. The purpose of bias is to ensure that the hyperplane is in the correct position after horizontal movement. Therefore, bias is determined after training w . The parameters of the hyperplane include w and b . When SVM is applied on classification, the hyperplane is regarded as a decision function:

$$f(x) = \text{sign}(w \cdot x + b). \tag{2}$$

Generally speaking, the purpose of SVM is to obtain the hyperplane of the maximized marginal distance and improve the distinguishing function between the two categories of the dataset. The process of optimizing the distinguishing function of the hyperplane can be regarded as a quadratic programming problem:

$$\begin{aligned} \text{minimize} \quad & L_p = \frac{1}{2} \|w\|^2 \\ \text{subject to} \quad & y_i(x_i \cdot w + b) - 1 \geq 0, \quad i = 1, \dots, l. \end{aligned} \tag{3}$$

The original minimization problem is converted into a maximization problem by using the Lagrange Theory:

$$\begin{aligned} \text{max} \quad & L_D(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ \text{subject to} \quad & \sum_{i=1}^l \alpha_i y_i = 0, \quad i = 1, \dots, l \\ & \alpha_i \geq 0, \quad i = 1, \dots, l. \end{aligned} \tag{4}$$

Finally, the linear divisive decision making function is

$$f(x) = \text{sign} \left(\sum_{i=1}^n y_i \alpha_i^* (x \cdot x_i) + b^* \right). \tag{5}$$

If $f(x) > 0$, it means the sample is in the same category as samples marked with “+1”; otherwise, it is in the category of samples marked with “-1.” When the training data include noise, the linear hyperplane cannot accurately distinguish data points. By introducing slack variables ξ_i in the constraint, the original (3) can be modified into the following:

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|w\|^2 + C \left(\sum_{i=1}^l \xi_i \right) \\ & \text{subject to} \quad y_i (x_i \cdot w + b) - 1 + \xi_i \geq 0, \quad i = 1, \dots, l \\ & \quad \quad \quad \xi_i \geq 0, \quad i = 1, \dots, l, \end{aligned} \quad (6)$$

where ξ_i is the distance between the boundary and the classification point and penalty parameter C represents the cost of the classification error of training data during the learning process, as determined by the user. When C is greater, the margin will be smaller, indicating that the fault tolerance rate will be smaller when a fault occurs. Otherwise, when C is smaller, the fault tolerance rate will be greater. When $C \rightarrow \infty$, the linear inseparable problem will degenerate into a linear separable problem. In this case, the solution of the above mentioned optimization problem can be applied to obtain the various parameters and optimum solution of the target function using the Lagrangian coefficient; thus, the linear inseparable dual optimization problem is as follows:

$$\begin{aligned} & \text{Max} \quad L_D(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ & \text{Subject to} \quad \sum_{i=1}^l \alpha_i y_i = 0, \quad i = 1, \dots, l \\ & \quad \quad \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, l. \end{aligned} \quad (7)$$

Finally, the linear decision-making function is

$$f(x) = \text{sign} \left(\sum_{i=1}^n y_i \alpha_i^* (x \cdot x_i) + b^* \right). \quad (8)$$

(2) *Nonlinear Support Vector Machine (Nonlinear SVM)*. When input training samples cannot be separated using linear SVM, we can use conversion function φ to convert the original 2-dimensional data into a new high-dimensional feature space for linear separable problem. SVM can efficiently perform a nonlinear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. Presently, many different core functions have been proposed. Using different core functions regarding different data features can effectively improve the computational efficiency of SVM. The relatively common core functions include the following four types:

(1) linear kernel function:

$$K(x_i, y_i) = x_i^t \cdot y_j, \quad (9)$$

(2) polynomial kernel function:

$$K(x_i, y_j) = (\gamma x_i^t x_j + r)^m, \quad \gamma > 0, \quad (10)$$

(3) radial basis kernel function:

$$K(x_i, y_j) = \exp \left(\frac{-\|x_i - y_j\|^2}{2\sigma^2} \right), \quad \gamma > 0, \quad (11)$$

(4) sigmoid kernel function:

$$K(x_i, y_j) = \tanh(\gamma x_i^t \cdot y_j + r), \quad (12)$$

where the emissive core function is more frequently applied in high feature dimensional and nonlinear problems, and the parameters to be set are γ and C , which can slightly reduce SVM complexity and improve calculation efficiency; therefore, this study selects the emissive core function.

3.4. *Support Vector Machine Recursive Feature Elimination (SVM-RFE)*. A feature selection process can be used to remove terms in the training dataset that are statistically uncorrelated with the class labels, thus improving both efficiency and accuracy. Pal and Maiti (2010) provided a supervised dimensionality reduction method. The feature selection problem has been modeled as a mixed 0-1 integer program [23]. Multiclass Mahalanobis-Taguchi system (MMTS) is developed for simultaneous multiclass classification and feature selection. The important features are identified using the orthogonal arrays and the signal-to-noise ratio and are then used to construct a reduced model measurement scale [24]. SVM-RFE is an SVM-based feature selection algorithm created by [12]. Using SVM-RFE, Guyon et al. selected key and important feature sets. In addition to reducing classification computational time, it can improve the classification accuracy rate [12]. In recent years, many scholars improved the classification effect in medical diagnosis by taking advantage of this method [22, 25].

3.5. *Multiclass SVM Classifier*. SVM's basic classification principle is mainly based on dual categories. Presently, there are three main methods, one-against-all, one-against-one, and directed acyclic graph, to process multiclass problems [26], described as follows.

(1) *One-Against-All (OAA)*. Proposed by Bottou et al., (1994) the one-versus-rest converts the classification problem of k categories into k dual-category problems [27]. Scholars have also proposed subsequent effective classification methods [28]. In the training process, it must train k dual-category SVMs. When training the i th classifier, data in the i th category is regarded as “+1” and the data of the remaining categories is regarded as “-1” to complete the training of k dual-category SVM; during the testing process, each testing instance is tested by trained k dual-category SVMs. The classification results can be determined by comparing the outputs of SVM. Regarding unknown category x , the

decision function $\arg \max_{i=1,\dots,k} (w^i)^t \phi(x) + b^i$ can be applied to generate k decision-making values, and category x is the category of the maximum decision making value.

(2) *One-Against-One (OAO)*. When there are k categories, two categories can produce an SVM; thus, it can produce $k(k-1)/2$ classifiers and determine the category of the samples by a voting strategy [28]. For example, if there are three categories (1, 2, and 3) and a sample to be classified with an assumed category of 2, the sample will then be input into three SVMs. Each SVM will determine the category of the sample using decision making function $\text{sign}((w^{ij})^t \Phi(x) + b^{ij})$ and adds 1 to the votes of the category. Finally, the category with the most votes is the category of the sample.

(3) *Directed Acyclic Graph (DAG)*. Similar to OAO method, DAG is to disintegrate the classification problem k categories into a $k(k-1)/2$ dual-category classification problem [18]. During the training process, it selects any two categories from k categories as a group, which it combines into a dual-category classification SVM; during the testing process, it establishes a dual-category acyclic graph. The data of an unknown category is tested from the root nodes. In a problem with k classes, a rooted binary DAG has k leaves labeled by the classes where each of the $k(k-1)/2$ internal nodes is labeled with an element of a Boolean function [19].

4. Experiment and Results

4.1. *Feature Selection Based on SVM-RFE*. The main purpose of SVM-RFE is to compute the ranking weights for all features and sort the features according to weight vectors as the classification basis. SVM-RFE is an iteration process of the backward removal of features. Its steps for feature set selection are shown as follows.

- (1) Use the current dataset to train the classifier.
- (2) Compute the ranking weights for all features.
- (3) Delete the feature with the smallest weight.

Implement the iteration process until there is only one feature remaining in the dataset; the implementation result provides a list of features in the order of weight. The algorithm will remove the feature with smallest ranking weight, while retaining the feature variables of significant impact. Finally, the feature variables will be listed in the descending order of explanatory difference degree. SVM-RFE's selection of feature sets can be mainly divided into three steps, namely, (1) the input of the datasets to be classified, (2) calculation of weight of each feature, and (3) the deletion of the feature of minimum weight to obtain the ranking of features. The computational step is shown as follows [12].

(1) Input

Training sample: $X_0 = [x_1, x_2, \dots, x_m]^T$.

Category: $y = [y_1, y_2, \dots, y_m]^T$.

The current feature set: $s = [1, 2, \dots, n]$.

Feature sorted list: $r = []$.

(2) Feature Sorting

Repeat the following process until $s = []$.

To obtain the new training sample matrix according to the remaining features: $X = X_0(:, s)$.

Training classifier: $\alpha = \text{SVM-train}(X, y)$.

Calculation of weight: $w = \sum_k \alpha_k y_k x_k$.

Calculation of sorting standards: $c_i = (w_i)^2$.

Finding the features of the minimum weight: $f = \arg \min(c)$.

Updating feature sorted list: $r = [s(f), r]$.

Removing the features with minimum weight: $s = s(1 : -1, f + 1 : \text{length}(s))$.

(3) *Output: Feature Sorted List r* . In each loop, the feature with minimum $(w_i)^2$ will be removed. The SVM then retrains the remaining features to obtain the new feature sorting. SVM-RFE repeatedly implements the process until obtaining a feature sorted list. Through training SVM using the feature subsets of the sorted list and evaluating the subsets using the SVM prediction accuracy, we can obtain the optimum feature subsets.

4.2. SVM Parameters Optimization Based on Taguchi Method.

Taguchi Method rises from the engineering technological perspective and its major tools include the orthogonal array and SN ratio, where SN ratio and loss function are closely related. A higher SN ratio indicates fewer losses [29]. Parameter selection is an important step of the construction of the classification model using SVM. The differences in parameter settings can affect classification model stability and accuracy. Hsu and Yu (2012) combined Taguchi method and Staelin method to optimize the SVM-based e-mail spam filtering model and promote spam filtering accuracy [30]. Taguchi parameter design has many advantages. For one, the effect of robustness on quality is great. Robustness reduces variation in parts by reducing the effects of uncontrollable variation. More consistent parts are equal to better quality. Also, the Taguchi method allows for the analysis of many different parameters without a prohibitively high amount of experimentation. It provides the design engineer with a systematic and efficient method for determining near optimum design parameters for performance and cost. Therefore, by using the Taguchi quality parameter design, this study conducts the optimization design of parameters C and γ to enhance the accuracy of SVM classifier on the diagnosis of multiclass diseases.

This study uses the multiclass classification accuracy as the quality attribute of the Taguchi parameter design [21]. In general, when the classification accuracy is higher, it means the accuracy of the classification model is better; that is, the quality attribute is larger-the-better (LTB), and SN_{LTB} is defined as:

$$SN_{LTB} = -10 \log_{10} (MSD) = -10 \log_{10} \left[\frac{1}{n} \sum_{i=1}^n \frac{1}{y_i^2} \right]. \quad (13)$$

TABLE 4: Classification accuracy comparison.

Dermatology database					Zoo database				
C	γ				C	γ			
	1	3	10	12		0.1	5	10	12
1	52.57%	95.18%	94.08%	94.22%	1	71.18%	78.09%	62.36%	40.64%
10	52.57%	96.04%	97.94%	97.93%	10	71.18%	96.00%	91.00%	85.09%
50	52.57%	96.31%	96.86%	96.58%	50	71.18%	96.09%	96.00%	96.00%
100	52.57%	96.31%	96.32%	96.03%	100	71.18%	96.09%	96.09%	96.00%

TABLE 5: Factor level configuration of LS-SVM parameter design.

Dermatology database				Zoo database			
Control factor	Level			Control factor	Level		
	1	2	3		1	2	3
A(C)	10	50	100	A(C)	5	10	50
B(γ)	2.4	5	10	B(γ)	0.08	4	11

4.3. *Evaluation of Classification Accuracy.* Cross-validation measurement divides all the samples into a training set and a testing set. The training set is the learning data of the algorithm to establish the classification rules; the samples of the testing data are used as the testing data to measure the performance of the classification rules. All the samples are randomly divided into k -folds by category, and the data are mutually repelled. Each fold of the data is used as the testing data and the remaining $k - 1$ folds are used as the training set. The step is repeated k times, and each testing set validates the classification rules learnt from the corresponding training set to obtain an accuracy rate. The average of the accuracy rates of all k testing sets can be used as the final evaluation results. The method is known as k -fold cross-validation.

4.4. *Results and Discussion.* The ranking order of all features for Dermatology and Zoo databases, using RFE-SVM, is summarized as follows: Dermatology = {V1, V16, V32, V28, V19, V3, V17, V2, V15, V21, V26, V13, V14, V5, V18, V4, V23, V11, V8, V12, V27, V24, V6, V25, V30, V29, V10, V31, V22, V20, V33, V7, V9} and Zoo = {V13, V9, V14, V10, V16, V4, V8, V1, V11, V2, V12, V5, V6, V3, V15, V7}. According to the suggestions of scholars, the classification error rate of OAO is relatively lower when the number of testing instances is below 1000. Multiclass SVM parameter settings can affect the Multiclass SVM's classification accuracy. Arenas-García and Pérez-Cruz applied SVMs' parameters setting in the multiclass Zoo dataset [31]. They have carried out simulation, using Gaussian kernels, for all possible combinations of C and $Garmar$ from $C = [1, 3, 10, 30, 100]$ and $Garmar = \text{sqrt}(0.25d), \text{sqrt}(0.5d), \text{sqrt}(d), \text{sqrt}(2d), \text{and } \text{sqrt}(4d)$ with d being the dimension of the input data. In this study, we have executed wide ranges of the parameter settings for Dermatology and Zoo databases. Finally, the parameter settings are suggested as Dermatology $(C, \gamma) = \{C = 1, 10, 50, 100 \text{ and } \gamma = 1, 3, 10, 12\}$, Zoo $(C, \gamma) = \{C = 1, 10, 50, 100 \text{ and } \gamma = 0.1, 5, 10, 12\}$, and the testing accuracies are shown in Table 4.

As shown in Table 4, regarding parameter C , when $C = 10$ and $\gamma = \{5, 10, 12\}$, the accuracy of the experiment is higher than that of the experimental combination of $C = 1$

and $\gamma = \{5, 10, 12\}$; moreover, regarding parameter γ , the experimental accuracy rate in the case of $\gamma = 5$ and $C = \{1, 10, 50, 100\}$ is higher than that of the experimental combination of $\gamma = 0.1$ and $C = \{1, 10, 50, 100\}$. The near optimal value of C or γ may not be the same for different databases. Finding the appropriate parameter settings is important for the performance of classifiers. Practically, it is impossible to simulate every possible combination of parameter settings. And that is the reason why Taguchi methodology is applied to reduce the experimental combinations for SVM. The experimental step used in this study was first referred to the related study, ex, $C = [1, 3, 10, 30, 100]$, [31]; then set a possible range for both databases ($C = 1\sim 100, \gamma = 1\sim 12$). After that, we slightly adjusted the ranges to understand if there will be better results in Taguchi quality engineering parameter optimization for each database. According to our experimental result, the final parameter settings C and γ range 10~100 and 2.4~10, respectively, for Dermatology database; the parameters settings C and γ range 5~50 and 0.08~11, respectively, for Zoo databases. Within the range of Dermatology and Zoo databases parameters C and γ , we select three parameter levels and two control factors, A and B , to represent parameters C and γ , respectively. The Taguchi orthogonal array experiment selects $L_9(3^2)$ and the factor level configuration is as illustrated in Table 5.

After data preprocessing, Dermatology and Zoo databases include 358 and 101 testing instances, respectively. The various experiments of the orthogonal array are repeated five times ($n = 5$); the experimental combination and observations are summarized, as shown in Tables 6 and 7. According to (13), we can calculate the SN ratio for Taguchi experimental combination #1 as

$$\begin{aligned}
 SN_{LTB} &= -10 \log_{10} \left[\frac{1}{5} \times \left(\frac{1}{0.9631^2} + \frac{1}{0.9701^2} + \frac{1}{0.9697^2} \right. \right. \\
 &\quad \left. \left. + \frac{1}{0.9627^2} + \frac{1}{0.9614^2} \right) \right] \\
 &= -0.3060.
 \end{aligned}
 \tag{14}$$

TABLE 6: Summary of experiment data of Dermatology database.

Number	Control factor		Observation					Average	SN
	A	B	y_1	y_2	y_3	y_4	y_5		
1	1	1	0.9631	0.9701	0.9697	0.9627	0.9614	0.9654	-0.3060
2	1	2	0.9686	0.9749	0.9653	0.9621	0.9732	0.9688	-0.2755
3	1	3	0.9795	0.9847	0.9848	0.9838	0.9735	0.9813	-0.1647
4	2	1	0.9630	0.9615	0.9581	0.9599	0.9668	0.9619	-0.3379
5	2	2	0.9687	0.9721	0.9704	0.9707	0.9626	0.9689	-0.2746
6	2	3	0.9685	0.9748	0.9744	0.9712	0.9707	0.9719	-0.2475
7	3	1	0.9671	0.9689	0.9648	0.9668	0.9645	0.9664	-0.2967
8	3	2	0.9741	0.9704	0.9797	0.9799	0.9767	0.9762	-0.2098
9	3	3	0.9625	0.9633	0.9642	0.9678	0.9619	0.9639	-0.3191

($A_1 = 10, A_2 = 50, A_3 = 100; B_1 = 2.4, B_2 = 5, B_3 = 10$).

TABLE 7: Summary of experiment data of Zoo database.

Number	Control factor		Observation					Average	SN
	A	B	y_1	y_2	y_3	y_4	y_5		
1	1	1	0.9513	0.9673	0.9435	0.9567	0.9546	0.9547	-0.4037
2	1	2	0.9600	0.9616	0.9588	0.9611	0.9608	0.9605	-0.3504
3	1	3	0.7809	0.7833	0.7820	0.7679	0.7811	0.7790	-2.1694
4	2	1	0.7118	0.6766	0.7368	0.7256	0.7109	0.7123	-2.9571
5	2	2	0.9600	0.9612	0.9604	0.9519	0.9440	0.9555	-0.3960
6	2	3	0.8900	0.8947	0.9214	0.9050	0.9190	0.9060	-0.8598
7	3	1	0.7118	0.7398	0.7421	0.7495	0.7203	0.7327	-2.7064
8	3	2	0.9610	0.9735	0.9709	0.9752	0.9661	0.9693	-0.2709
9	3	3	0.9600	0.9723	0.9707	0.9509	0.9763	0.9660	-0.3013

($A_1 = 5, A_2 = 10, A_3 = 50; B_1 = 0.08, B_2 = 4, B_3 = 11$).

The calculation results of the SN ratios of the remaining eight experimental combinations are summarized, as in Table 6. The Zoo experimental results and SN ratio calculation are as shown in Table 7. According to the above results, we then calculate the average SN ratios of the various factor levels. With the experiment of Table 8 as an example, the average SN ratio \bar{A}_1 of Factor A at Level 1 is

$$\bar{A}_1 = \frac{1}{3} [-0.3060 + (-0.2755) + (-0.1647)] = -0.2487. \tag{15}$$

Similarly, we can calculate the average effects of \bar{A}_2 and \bar{A}_3 from Table 6. The difference analysis results of the various factor levels of Dermatology and Zoo databases are as shown in Table 8. The factor effect diagrams are as shown in Figures 2 and 3. As a greater SN ratio represents better quality, according to the factor level difference and factor effect diagrams, the Dermatology parameter level combination is A_1B_3 ; in other words, parameters $C = 10, \gamma = 10$, Zoo parameter level combination is A_1B_2 , and the parameter settings are $C = 5, \gamma = 4$.

When constructing the Multiclass SVM model using SVM-RFE, three different feature sets are selected according

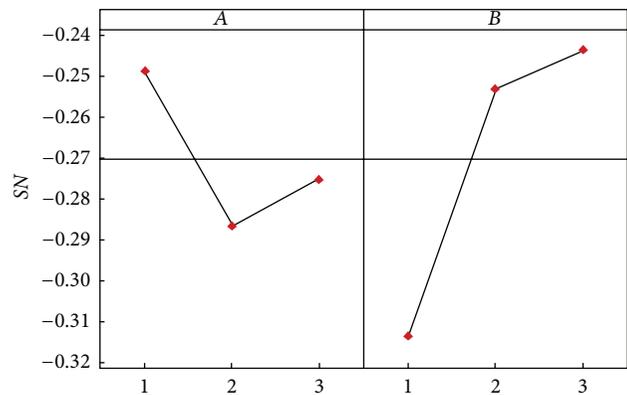


FIGURE 2: Main effect plots for SN ratio of Dermatology database.

to their significance. At the first stage, Taguchi quality engineering is applied to select the optimum values of parameters C and γ . At the second stage, it constructs the Multiclass SVM Classifier and compares the classification performance according to the above parameters. In the Dermatology experiment, Table 9 illustrates the two feature subsets containing 23 and 33 feature variables. The 33 feature

TABLE 8: Average of each factor at all levels.

Control factor	Dermatology				Control factor	Zoo			
	1	2	3	Difference		1	2	3	Difference
A(C)	-0.2487	-0.2867	-0.2752	0.0380	A(C)	-0.9745	-1.4043	-1.0929	0.4298
B(γ)	-0.3135	-0.2533	-0.2438	0.0697	B(γ)	-2.0224	-0.3391	-1.1102	1.6833

TABLE 9: Classification performance comparison of Dermatology database.

Methods	Dimensions	C	γ	Accuracy
SVM	33	100	5	95.10% \pm 0.0096
SVM-RFE	23	50	2.4	89.28% \pm 0.0139
SVM-RFE-Taguchi	23	10	10	95.38% \pm 0.0098

TABLE 10: Classification performance comparison of Zoo database.

Methods	Dimensions	C	γ	Accuracy
SVM	16	10	11	89% \pm 0.0314
SVM-RFE	6	50	0.08	92% \pm 0.0199
SVM-RFE-Taguchi	12	5	4	97% \pm 0.0396

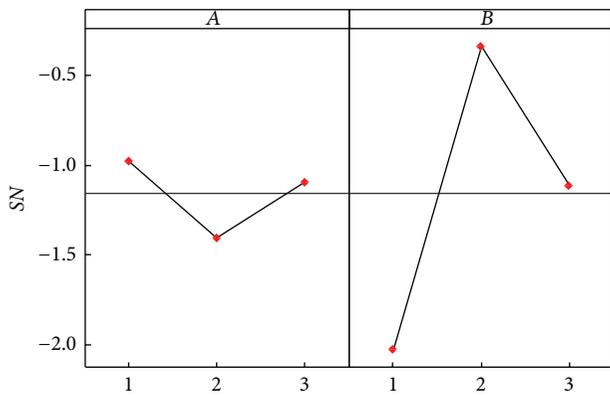


FIGURE 3: Main effect plots for SN ratio of Zoo database.

sets are tested by SVM and SVM, as based on Taguchi. The parameter settings and testing accuracy rate results are as shown in Table 9. The experimental results, as shown in Figure 4, show that the SVM ($C = 10, \gamma = 10$) testing accuracy rate of the 17-feature sets datasets can be higher than 90%, which is better than the accuracy rate of 20-feature sets dataset SVM ($C = 10, \gamma = 11$), up to 90%. Moreover, regardless of how many sets of feature variables are selected, the accuracy of SVM ($C = 50, \gamma = 2.4$) cannot be higher than 90%.

Regarding the Zoo experiment, Table 10 summarizes the experimental test results of sets containing 6, 12, and 16 feature variables using SVM and SVM based on Taguchi. As shown in Table 10, the experimental results show that the classification accuracy rate of the set of 12-feature variables in the classification experiment using SVM-RFE-Taguchi ($C = 10, \gamma = 10$) is the highest, up to 97% \pm 0.0396. As shown in Figure 5, the experimental results show that the classification

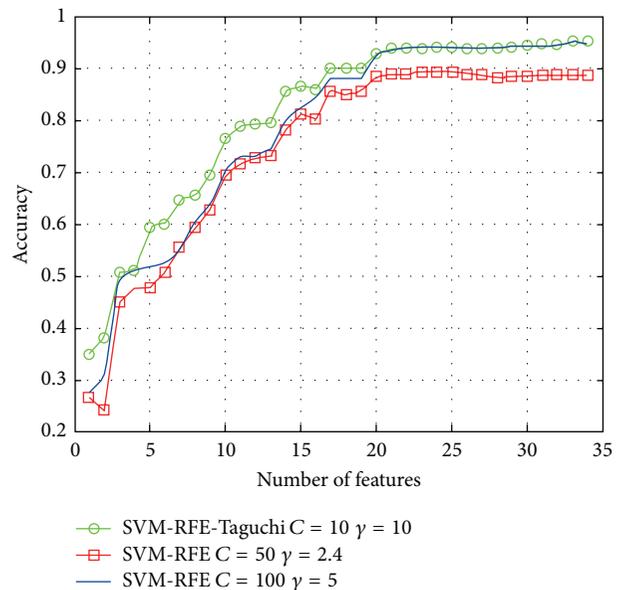


FIGURE 4: Classification performance comparison of Dermatology database.

accuracy rate of the dataset containing 7 feature variables by SVM-RFE-Taguchi ($C = 50, \gamma = 2.4$) can be higher than 90%, which can obtain relatively better prediction effects.

5. Conclusions

As the study on the impact of feature selection on the multiclass classification accuracy rate becomes increasingly attractive and significant, this study applies SVM-RFE and SVM in the construction of a multiclass classification method in order to establish the classification model. As RFE is a

TABLE 11: Comparison of classification accuracy in related literature.

Author	Method	Accuracy%
Dermatology database		
Xie et al. (2005) [16]	FOut_SVM	91.74%
Srinivasa et al. (2006) [32]	FCM_SVM	83.30%
Ren et al. (2006) [33]	LDA_SVM	72.09%
Our Method (2014)	SVM-RFE-Taguchi	95.38%
Zoo database		
Xie et al. (2005) [16]	FOut_SVM	88.24%
He (2006) [34]	NFPH_k-modes	92.08%
Golzari et al. (2009) [35]	Fuzzy_AIRS	94.96%
Our Method (2014)	SVM-RFE-Taguchi	97.00%

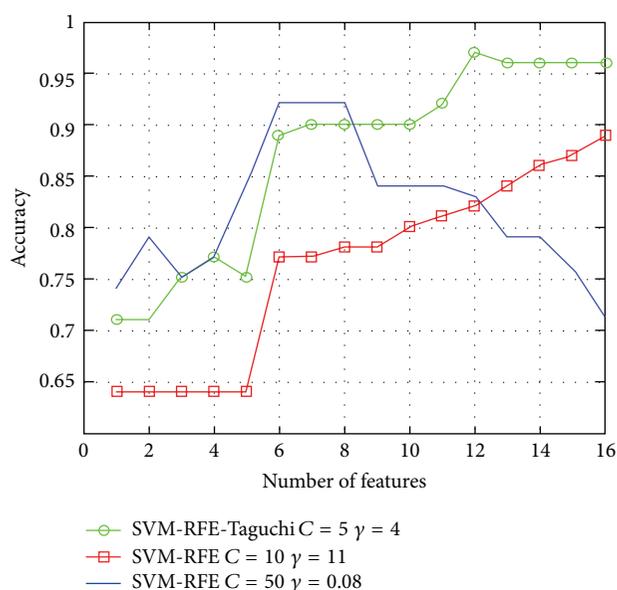


FIGURE 5: Classification performance comparison of Zoo database.

feature selection method of a wrapper model, it requires a previously defined classifier as the assessment rule of feature selection; therefore, SVM is used as the RFE assessment standard to help RFE in the selection of feature sets.

According to the experimental results of this study, with respect to parameter settings, the impact of parameter selection on the construction of SVM classification model is huge. Therefore, this study applies the Taguchi parameter design in determining the parameter range and selection of the optimum parameter combination for SVM classifier, as it is a key factor influencing the classification accuracy. This study also collected the experimental results of using different research methods in the case of Dermatology and Zoo databases [16, 32, 33], as shown in Table 11. By comparison, the proposed method can achieve higher classification accuracy.

Conflict of Interests

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

References

- [1] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, Cambridge, UK, 2000.
- [2] J. Luts, F. Ojeda, R. van de Plas Raf, B. de Moor, S. van Huffel, and J. A. K. Suykens, "A tutorial on support vector machine-based methods for classification problems in chemometrics," *Analytica Chimica Acta*, vol. 665, no. 2, pp. 129–145, 2010.
- [3] M. F. Akay, "Support vector machines combined with feature selection for breast cancer diagnosis," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3240–3247, 2009.
- [4] C.-Y. Chang, S.-J. Chen, and M.-F. Tsai, "Application of support-vector-machine-based method for feature selection and classification of thyroid nodules in ultrasound images," *Pattern Recognition*, vol. 43, no. 10, pp. 3494–3506, 2010.
- [5] H.-L. Chen, B. Yang, J. Liu, and D.-Y. Liu, "A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis," *Expert Systems with Applications*, vol. 38, no. 7, pp. 9014–9022, 2011.
- [6] P. Danenas and G. Garsva, "Credit risk evaluation modeling using evolutionary linear SVM classifiers and sliding window approach," *Procedia Computer Science*, vol. 9, pp. 1324–1333, 2012.
- [7] C. L. Huang, H. C. Liao, and M. C. Chen, "Prediction model building and feature selection with support vector machines in breast cancer diagnosis," *Expert Systems with Applications*, vol. 34, no. 1, pp. 578–587, 2008.
- [8] H. F. Liau and D. Isa, "Feature selection for support vector machine-based face-iris multimodal biometric system," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11105–11111, 2011.
- [9] Y. Zhang, Z. Chi, and Y. Sun, "A novel multi-class support vector machine based on fuzzy theories," in *Intelligent Computing: International Conference on Intelligent Computing, Part I (ICIC '06)*, D. S. Huang, K. Li, and G. W. Irwin, Eds., vol. 4113 of *Lecture Notes in Computer Science*, pp. 42–50, Springer, Berlin, Germany.
- [10] Y. Aksu, D. J. Miller, G. Kesidis, and Q. X. Yang, "Margin-maximizing feature elimination methods for linear and nonlinear kernel-based discriminant functions," *IEEE Transactions on Neural Networks*, vol. 21, no. 5, pp. 701–717, 2010.
- [11] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

- [12] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389-422, 2002.
- [13] S. Harikrishna, M. A. H. Farquad, and Shabana, "Credit scoring using support vector machine: a comparative analysis," in *Advanced Materials Research*, Trans Tech Publications, Zürich, Switzerland, 2012.
- [14] X. Lin, F. Yang, L. Zhou et al., "A support vector machine-recursive feature elimination feature selection method based on artificial contrast variables and mutual information," *Journal of Chromatography B: Analytical Technologies in the Biomedical and Life Sciences*, vol. 10, pp. 149-155, 2012.
- [15] R. Zhang and M. Jianwen, "Feature selection for hyperspectral data based on recursive support vector machines," *International Journal of Remote Sensing*, vol. 30, no. 14, pp. 3669-3677, 2009.
- [16] Z. X. Xie, Q. H. Hu, and D. R. Yu, "Fuzzy output support vector machines for classification," in *Advances in Natural Computation*, L. Wang, K. Chen, and Y. S. Ong, Eds., vol. 3612, pp. 1190-1197, Springer, Berlin, Germany.
- [17] Y. Liu, Z. You, and L. Cao, "A novel and quick SVM-based multi-class classifier," *Pattern Recognition*, vol. 39, no. 11, pp. 2258-2264, 2006.
- [18] J. Platt, N. C. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," in *Advances in Neural Information Processing Systems*, S. A. Solla, T. K. Leen, and K. R. Muller, Eds., vol. 12, pp. 547-553, 2000.
- [19] Y. Xu, S. Zomer, and R. G. Brereton, "Support vector machines: a recent method for classification in chemometrics," *Critical Reviews in Analytical Chemistry*, vol. 36, no. 3-4, pp. 177-188, 2006.
- [20] M. L. Huang, Y. H. Hung, and E. J. Lin, "Effects of SVM parameter optimization based on the parameter design of Taguchi method," *International Journal on Artificial Intelligence Tools*, vol. 20, no. 3, pp. 563-575, 2011.
- [21] H.-C. Lin, C.-T. Su, C.-C. Wang, B.-H. Chang, and R.-C. Juang, "Parameter optimization of continuous sputtering process based on Taguchi methods, neural networks, desirability function, and genetic algorithms," *Expert Systems with Applications*, vol. 39, no. 17, pp. 12918-12925, 2012.
- [22] Y. Mao, D. Pi, Y. Liu, and Y. Sun, "Accelerated recursive feature elimination based on support vector machine for key variable identification," *Chinese Journal of Chemical Engineering*, vol. 14, no. 1, pp. 65-72, 2006.
- [23] A. Pal and J. Maiti, "Development of a hybrid methodology for dimensionality reduction in Mahalanobis-Taguchi system using Mahalanobis distance and binary particle swarm optimization," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1286-1293, 2010.
- [24] C.-T. Su and Y.-H. Hsiao, "Multiclass MTS for simultaneous feature selection and classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 2, pp. 192-205, 2009.
- [25] X. Lin, F. Yang, L. Zhou et al., "A support vector machine-recursive feature elimination feature selection method based on artificial contrast variables and mutual information," *Journal of Chromatography B*, vol. 910, pp. 149-155, 2012.
- [26] E. Hüllermeier and S. Vanderlooy, "Combining predictions in pairwise classification: an optimal adaptive voting strategy and its relation to weighted voting," *Pattern Recognition*, vol. 43, no. 1, pp. 128-142, 2010.
- [27] L. Bottou, C. Cortes, J. Denker et al., "Comparison of classifier methods—a case study in handwritten digit recognition," in *Proceedings of the 12th Iaprr International Conference on Pattern Recognition*, vol. 2, pp. 77-82, IEEE Computer Society Press, Los Alamitos, Calif, USA, 1994.
- [28] J. Furnkranz, "Round robin rule learning," in *Proceedings of the 18th International Conference on Machine Learning (ICML '01)*, pp. 146-153, 2001.
- [29] M. R. Sohrabi, S. Jamshidi, and A. Esmaeilifar, "Cloud point extraction for determination of Diazinon: optimization of the effective parameters using Taguchi method," *Chemometrics and Intelligent Laboratory Systems*, vol. 110, no. 1, pp. 49-54, 2012.
- [30] W. C. Hsu and T. Y. Yu, "Support vector machines parameter selection based on combined taguchi method and staelin method for e-mail spam filtering," *International Journal of Engineering and Technology Innovation*, vol. 2, no. 2, pp. 113-125, 2012.
- [31] J. Arenas-García and F. Pérez-Cruz, "Multi-class support vector machines: A new approach," in *Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, vol. 2, pp. 781-784, April 2003.
- [32] K. G. Srinivasa, K. R. Venugopal, and L. M. Patnaik, "Feature extraction using fuzzy c-means clustering for data mining systems," *International Journal of Computer Science and Network Security*, vol. 6, no. 3A, pp. 230-236, 2006.
- [33] Y. Ren, H. Liu, C. Xue, X. Yao, M. Liu, and B. Fan, "Classification study of skin sensitizers based on support vector machine and linear discriminant analysis," *Analytica Chimica Acta*, vol. 572, no. 2, pp. 272-282, 2006.
- [34] Z. He, *Farthest-point heuristic based initialization methods for K-modes clustering [thesis]*, Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin, China, 2006.
- [35] S. Golzari, S. Doraisamy, M. N. Sulaiman, and N. I. Udzir, "Effect of fuzzy resource allocation method on AIRS classifier accuracy," *Journal of Theoretical and Applied Information Technology*, vol. 5, no. 1, pp. 18-24, 2009.

Research Article

Comparative Study on Interaction of Form and Motion Processing Streams by Applying Two Different Classifiers in Mechanism for Recognition of Biological Movement

Bardia Yousefi and Chu Kiong Loo

Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

Correspondence should be addressed to Chu Kiong Loo; ckloo.um@um.edu.my

Received 27 May 2014; Accepted 26 June 2014; Published 3 September 2014

Academic Editor: Shifei Ding

Copyright © 2014 B. Yousefi and C. K. Loo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Research on psychophysics, neurophysiology, and functional imaging shows particular representation of biological movements which contains two pathways. The visual perception of biological movements formed through the visual system called dorsal and ventral processing streams. Ventral processing stream is associated with the form information extraction; on the other hand, dorsal processing stream provides motion information. Active basic model (ABM) as hierarchical representation of the human object had revealed novelty in form pathway due to applying Gabor based supervised object recognition method. It creates more biological plausibility along with similarity with original model. Fuzzy inference system is used for motion pattern information in motion pathway creating more robustness in recognition process. Besides, interaction of these paths is intriguing and many studies in various fields considered it. Here, the interaction of the pathways to get more appropriated results has been investigated. Extreme learning machine (ELM) has been implied for classification unit of this model, due to having the main properties of artificial neural networks, but crosses from the difficulty of training time substantially diminished in it. Here, there will be a comparison between two different configurations, interactions using synergetic neural network and ELM, in terms of accuracy and compatibility.

1. Introduction

The recognition of human action is one of the interesting research field for decades in computer vision and machine learning areas. However, it has far more intriguing rout of intelligence systems and other relevant fields like psychophysical, neurophysiological, and theoretical neuroscience especially once it comes to biological movement mechanic which needs relevancy between biological and machine models. Studies in the area of physiologic and psychophysical have presented that there are several various processes for mechanism of biological motion analysis. It operates through detecting local energies in displacements of motion (see [1–3]). There are some spatial frequencies tuning considering inconsistency variations and contrast in luminance [1, 3]. In terms of motion analysis local or global motion, motion patterns have substantial influences. Temporal characteristic is considerable in perception of the movements too. Moreover,

synchronisation of object features bindingly [4] along with its motion and perceiving time is also proceeded in temporal processing [5] (with respect to visual system functionality of temporal limitations [6]). Besides the aforementioned points, recognition of biological movements in mammalian visual system is considered through two separated pathways. Each of these pathways is involving certain information, that is, motion representing information of dorsal processing stream and form pathway which involves data from ventral stream.

Two streams have used neural detectors for motion and form feature extraction and hierarchically allow the independency in size and style in both pathways and classification of generated features from both feed-forward pathways to categorize the biological movements. Corresponding results on the stationary biological motion recognition revealed that discrimination can be accomplished through particularly small latencies, constructing an important role of top-down unlikely signals [7]. The body shapes are determined by set of

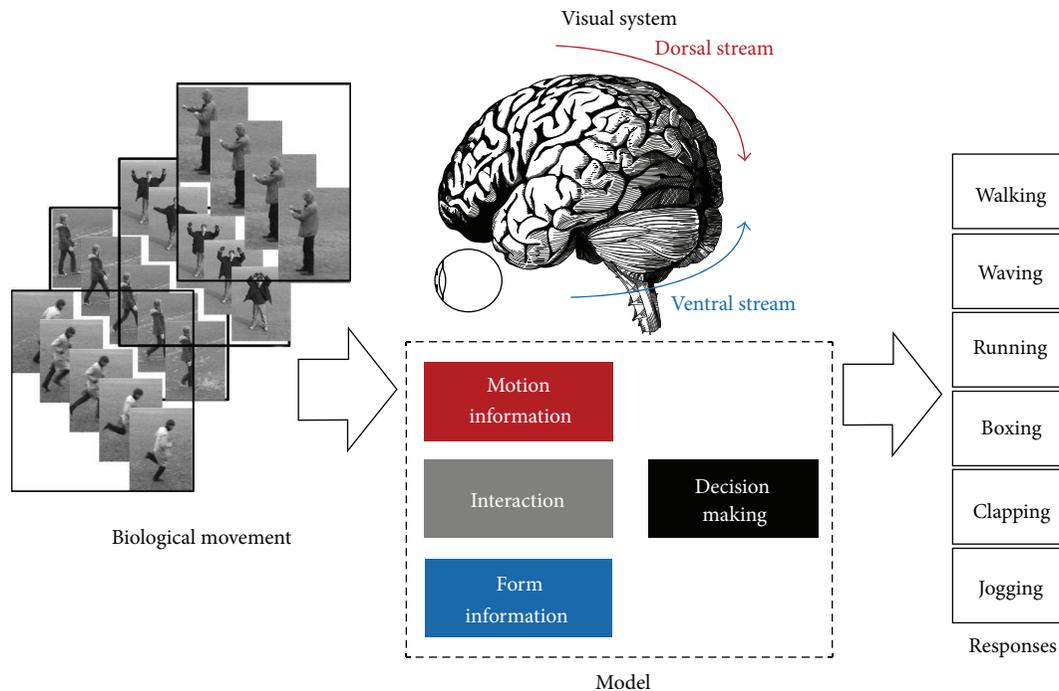


FIGURE 1: Figure reveals the analytical visual system model. The proposed approach suggests utilization of better interaction and classifier in model of the biological movement. To develop the computational models for recognition mechanism and characterize the recognition responses regarding various actions. The model is the perspective of the original model and consists of particular computations of motion and form feature data. The model operates for high-dimension of input streams and the outcome is a combination of the ventral and dorsal processing stream.

patterns like sequences of snapshots [8] which has constant feature within whole action episode. The presented method expands an earlier model used for stationary objects [8–13] recognition by adding and combining over the temporal information in pathways following the psychological evidences [14, 15]. It can be good relating to quantity tool for organizing, summarizing, and interpreting existent information based on the data provided by psychophysics, neurophysiology and functional imaging [8]. The approach quantitatively develops the original model for temporal analysis and even in computer simulations with respect to previous model architecture (see Figure 1).

Motion pathway involves information of optical flow which has fast natural temporal changes. It has consistency with neurophysiological data from neural detectors. Changing variation features due to its achievements within short changes between $\text{Frame}(t)$ and $\text{Frame}(t + 1)$ (t represents the time for each frame) creates instability in data attained from this pathway. Local detector of optical flow is connected with motion patterns and the model comprises population of four directed neurons in area of MT. However there is a connection between MT and V4 for motion and direction selection. Also, the motion edge selector cells (which have two opposite directions sensitivity) that it finds in areas of MT, MSTd, MSTl [16, 17], and many other parts of the dorsal streams and probably in the kinetic occipital area (KO) [8]. Also, motion selective edges can be like MT [16] and MSTl [17] in macaque monkey. Mild instability in the information of this pathway

can be a cause of disparity in the final decision. This problem has been properly solved by applying an inference system in this pathway which substantially decreased instability throughout the fast varying pathway.

Few models have been proposed for recognition of human body shape which is plausible and neurophysiologically uses for recognizing stationary form (for instance [9]). Our method follows an object recognition model [9] which is composed of form features detectors through utilization of ABM. It follows the data obtained from neurophysiological information concerning scale, position, and sizes invariance, in case of adaptive ABM, which needs further computational load along with the hierarchy. The methods which use Gabor like filters to model the detectors have good constancy by simple cells [18]. The complex-like cells in V1 area or in V2 and V4 are invariant in terms of position varying responses (see [8]) and size independency, typically in the area of V4. V2 and V4 are more selective for difficult form features, for example, junctions and corners while being not appropriate for motion recognition because of temporal dependency in these two pathways. Snapshots detectors is used to find the shape (form) pattern similar to the area of IT (inferotemporal cortex) of monkey where the view-tuned neurons located and complex shapes tune [16]. Snapshot neurons are like view-tuned neurons in area of IT and gives independent scale and position. Previous models used Gaussian radial basis functions for modelling and it adjusted in training which performed a key frame regarding training sequences.

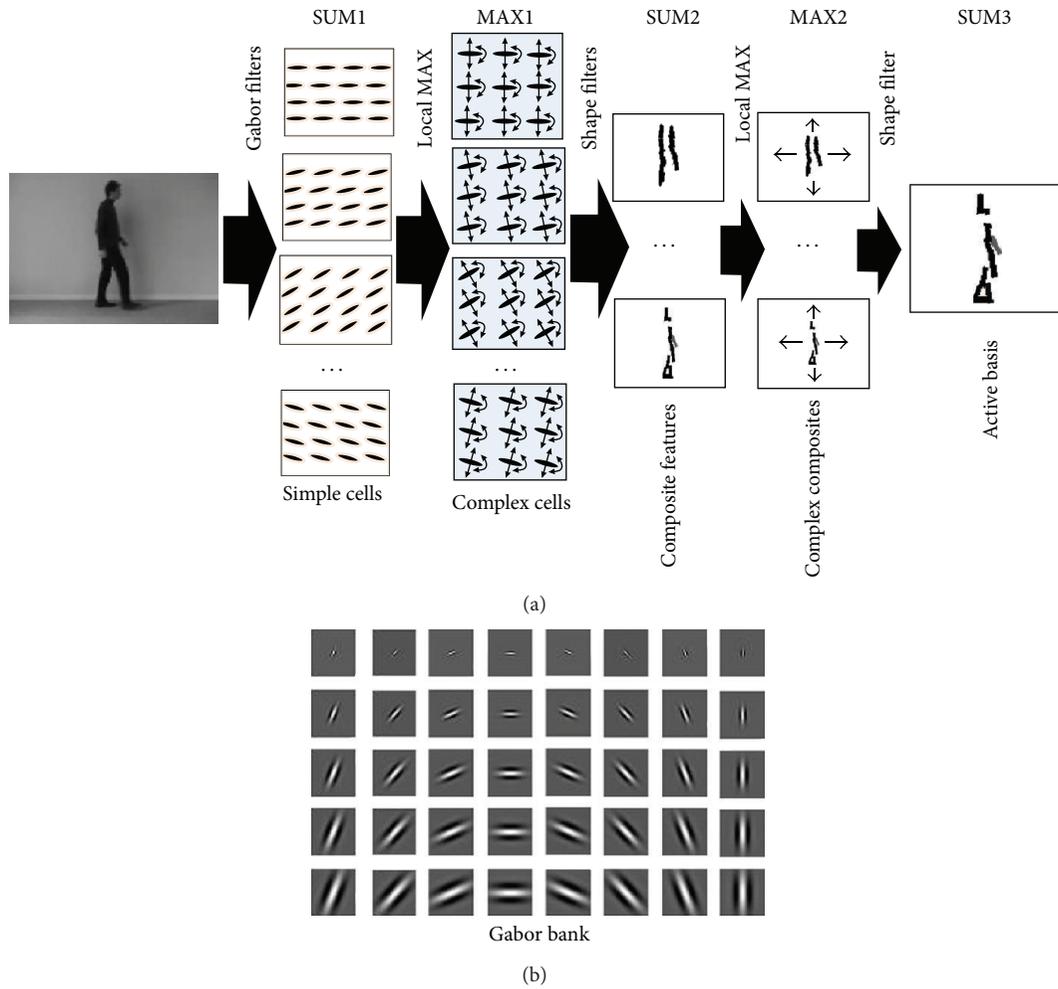


FIGURE 2: The explanation diagram of the ventral processing of the applying active basis model [24] which represents movements pattern and shape form of biological object within its movement episode. Active basis model is a Gabor based supervised object recognition method which can learn the object shape in the training stage and can be a utilized object recognizer within the action episode. (a) The processing diagram of the ABM process for finding human object presented. The similarity between the method and biological finding in different level has been mentioned in different stages. (b) It represents the Gabor bank filter in different scales and orientations. Overall, ABM has two stages, SUM and MAX, which make the hierarchy from simple cells to complex cells and at the end whole human object shape by active bases.

The final decision for recognition of biological movements is a combination of this information, the so-called interaction between two independent processing pathways. Interaction of these two streams is done at few levels in mammalian brains [19, 20] whereas many neurobiological, physiological, and psychological evidences show that the information coupling occur in many places for instance in STS level [21] and in different ways, that is, recurrent feedback loops [14]. Mutual links have suggested recurrent processing loops that permit interaction of top-down and bottom-up processing [14, 15, 22]. However, current neuroscience and psychophysics research specifies more extensive form signals influences on motion processing than previously assumed [15].

We introduce a comparison of two different perspective models which follow the original models utilizing ABM considering the interaction portion between these processing

pathways along with decision making segment. These interactions consider two different structural and inference models. Computational simulating along with testing the method is presented in the Results section. Finally, we conclude the recognition of biological movements model at the end; for examination of the proposed approach on a broader range of high-dimensional video streams, we measured responses to separated parallel pathways of visual system and overall results have been compared. Results for an instance patterns model in ventral path are revealed (Figure 2). The proposed model does a significant task of catching the constant pattern of ventral pathway responses to human movements (Figure 2, upper processing stream). The model considers the dorsal covering responses as almost half of the visual system decision The form pattern features in the model of visual system considers Gabor like stimuli in the form of hierarchical representation for object recognition task throughout the ventral

stream. ABM as Gabor based supervised method can boost the responses of the stream directive and can be excellent interpreted as providing the human object. Proposed model tries to increase performance by incorporating the form features with motion features form dorsal stream and using different classifiers for this aim in original model (see [8]).

2. System Overview

In this paper, a comparison investigation has been addressed between two classification methods in mechanism for recognition of biological movement model. It follows the original model and considers the psychological evidences regarding the model improvement. Furthermore, decision making portion in the model has been improved that it itself increases accuracy rate and complementary part of this comparison. For this aim, additional parts to the model have been presented and afterward in-depth comparison results will be presented.

2.1. Active Basis Model for Form Pathway. Gabor wavelet has been previously introduced to mammalian visual system model due to similarity with its stimuli in portion; however this kind of features has been widely used for human action recognition task (e.g., [23]) and similar tasks. Gabor wavelets (in dictionary elements) provide biologically deformable templates and have been widely used by active basis model (ABM) [24]. Shared sketch algorithm (SSA) tracks through AdaBoost. Within every repetition, matching pursuit followed by SSA selects a wavelet element. The objects numbers in different orientation, location, and scale are checked by this method. Choosing the minor dictionary elements for each image (sparse coding), there can be an image representation applying linear combination of mentioned elements by considering U a minor residual. SSA interacts with information of motion pathway and visually guides:

$$I = \sum_{i=1}^n c_i \beta_i + \epsilon. \quad (1)$$

Let $\beta = (\beta_i, i = 1, \dots, n)$ be Gabor wavelet set of sinusoid elements and components, $c_i = \langle I, \beta_i \rangle$, and ϵ an image coefficient which is kept unsolved [24]. Using wavelet sparse coding a large number of pixels reduce to small number of wavelet element. Training the natural image patches through sparse coding can be executed by dictionary elements of Gabor like wavelet which carries the simple cells in V1 [25]. Local shape extraction will be discretely done for entire frames similar to [24] filter responses in density and orientation for each pixels. ABM uses Gabor filter bank but in different form. A dictionary of Gabor wavelets contains n directions and m scales in the form of $GW_j(\theta, \omega)$, $j = 1, \dots, m \times n$, where $\theta \in \{k\pi/n, k = 0, \dots, n-1\}$ and $\omega = \{\sqrt{2}/i, i = 1, \dots, m\}$. Features of Gabor wavelet specify the posture, size, and location small variance of object form. In overall shape structure is considered to be maintained during the

recognition process. Every element response (convolution) offers the information of form with θ and ω . Consider

$$B = \langle GW, I \rangle \quad (2)$$

$$= \sum \sum GW(x_0 - x, y_0 - y : \omega_0, \theta_0) I(x, y),$$

where GW_j is a $[x_g, y_g]$, I is a $[x_i, y_i]$ matrices, and response of I to GW is a $[x_i + x_g, y_i + y_g]$. Consequently, earlier both matrices convolution must be expanded by adequate zeroes. Convolution consequence can be removed via the result gathering. An extra technique would be to shift back the frequencies centre (zero frequency) to the image center although it might be considered losing data reason. Training set of image shown by $\{I^m, m = 1, \dots, M\}$, SSA consecutively chooses B_i . The important opinion is to find B_i and thus the segments edges attained from I_m become maximum [24]. It requires to calculate $[I^m \cdot \beta] = \psi |\langle I^m \cdot \beta \rangle|^2$ for different i where $\beta \in$ Dictionary and ψ signifies sigmoid, whitening, and thresholding transformations. Then for maximizing $[I^m \cdot \beta]$ for all possible β will be computed, where $\beta = (\beta_i, i = 1, \dots, n)$ is the template, for every training image I^m scoring will be based on

$$M(I^m, \theta) = \sum_{i=1}^n \delta_i |I^m, \beta| - \log \Phi(\lambda \delta_i). \quad (3)$$

M is function of match scoring and δ_i attained from $\sum_{m=1}^M [I^m, \beta]$ concerning steps selection, and Φ is nonlinear function. The exponential model for logarithmic likelihood relation is attained from the template matching scores. The weight vectors are calculated by technique of maximum likelihood and are exposed by $\Delta = (\delta_i, i = 1, \dots, n)$ [24]. Consider

$$\text{Max}(x, y) = \max_{(x,y) \in D} M(I_m, \beta). \quad (4)$$

$\text{Max}(x, y)$ computes the maximum matching score previously obtained and D signifies I lattice. Here, there is no summation because of updating the size based on training system on frame $(t-1)$. Moreover, method tracks the object relating to motion feature to signify the moving object displacement. These displacements have been assisted to be detected better through guidance of motion information which is considered a substantial similarity with biological evidences [14, 15, 25].

2.2. Dorsal Pathway and Motion Information. The information of motion in recognition of biological movements is attained using optical flow. It finds out the movement pattern which has reliability by information of neurophysiological from neural detectors hierarchy. Areas of V1 and MT have some neurons for motion and direction selection in initial motion pathway level correspondingly. Visibility of every layer shows the principle dissimilar between previous and layerwise optical flow estimation. Shape of mask can perform while matching applies for the pixels which fall inside the mask (see [26]). Applied layerwise optical flow method (mentioned in [26]) has baseline optical flow algorithm of [27–29]. In overview, M_1 and M_2 are visible masks for the two frames $I_1(t)$ and $I_2(t-1)$ and the fields of flow from I_1

to I_2 and from I_2 to I_1 are denoted by (u_1, v_1) and (u_2, v_2) . Following terms will be deliberated utilizing the layerwise optical flow estimation. Objective function contains summing three parts and visible layer masks match these two images using Gaussian filter which called data terms matching $E_\gamma^{(i)}$, symmetric $E_\delta^{(i)}$, and smoothness $E_\mu^{(i)}$. Consider

$$E(u_1, v_1, u_2, v_2) = \sum_{i=1}^2 E_\gamma^{(i)} + \rho E_\delta^{(i)} + \xi E_\mu^{(i)}. \quad (5)$$

After objective function optimization and applying inner and outer fixed-point repetitions, coarse to fine search, bidirectional flows are attained and utilized for specifying the motion patterns. Compressed optic flow for all frames is calculated by straight template matching earlier frame applying the absolute difference summation (L1-norm). Though optic flow is principally noisy, no smoothing techniques have been done on it as the field of flow will be blurred in gaps and specially the places where motion information is significant. To get the proper optical flow response about its application in recommended model, optical flow will be used for adjusting active basis model and making it more efficient. To attain a reliable illustration through form pathway, optic flow estimates the velocity and flow direction. Response of the filter based on local matching velocity and direction will be maximal as these two parameters are constantly changing.

2.3. Fuzzy Inference in Dorsal Processing Stream. Fuzzy logic is a multivalued logic, that is, created from fuzzy set theory found by Zadeh (1965), and it deals with reasoning approximation [30]. It delivers great framework targeted at approximation reasoning which can suitably bring the imprecision and uncertainty together in model expert heuristics and linguistic semantics and handles necessary level organizing principles. A time dependent fuzzy system also uses many times regarding solution of control and classification and so forth, Chen and Liu (2005) present a delay-dependent robust fuzzy control for a class of nonlinear delay systems via state feedback [31].

Applying fuzzy inference system involves the interaction between both pathways. A fuzzy inference system to imply optical flow within motion pathway has been presented by considering the flow in every frame division and estimation of the membership value for every portion. The problem statement through initial assumptions for the human object velocity associates for both x and y directions. In general, $v_x, v_y \in \mathbb{R}^{m \times n}$ where m and n are sizes of image frame from input video stream.

Membership functions in triangular shapes for v_x and it will be the same for v_y velocity in x and y directions and signify quaternion correlator outputs in the enrolment stage belonging to motion pathways, respectively (i.e., $\mu_{v_x}^{C_{1,2}}(x)$, $\mu_{v_x}^{C_{2,4}}(x)$, $\mu_{v_x}^{C_{1,2}}(y)$, and $\mu_{v_x}^{C_{2,4}}(y)$) [32]. Maximum velocity in two coordinates has been considered for estimation of membership values which are related to each cell. Aggregation of these values will be considered and helps in overall judgement throughout the sequential frames within the path. The dependency regarding time variation for every frame of video

in this pathway is through definition of fuzzy membership scoring for every time division. Velocities information can be unstable due to many environmental situations, for example, camera shaking, dissimilar style of human object temporarily in front of camera, and the velocity amount being time dependent. Time definition in this context is based on the frame time per second and creates resistance for every frame with respect to previous score value of membership. It can be involved in mathematical parameter or even just additional programming algorithm.

Time dependent fuzzy optical flow division can be used for signifying an optical flow divisions class with fuzzy inference rules in time for every frame of video stream as unit of time defined here, as follows:

$$\begin{aligned} \tilde{\mu}_{v_i}^{C_i}(t) &= \tilde{\mu}_{v_i}^{C_i}(t - \tau) + \eta_{v_i}^{C_i}(t) (1 - \tilde{\mu}_{v_i}^{C_i}(t - \tau)) \\ t &\in [t_0, t_0 + k\tau], \quad k \in (0, 1, \dots, N), \end{aligned} \quad (6)$$

where τ is the frame time which is a parameter for camera and k is numbers of frames pasted from the cell changing (it means k will be reset after varying the cell membership). N is the maximum number of frame distance from present frame which does not unreasonably increase membership function value. We call $\eta_{v_i}^{C_i}(t)$ memory coefficient function and it can be just a mathematical parameter or programming algorithm to add the winner cell membership. t is the frame time where one division optical flow has the highest membership amount as compared with other divisions and it will be restarted by changing the division. At the end, aggregation of fuzzy inference scoring for flow in different body has been computed. Defuzzification has been done through IF-THEN rule and output belongs to highest scores among the actions classes and specifies the movement. The max. amount represents degree of belonging to each classes and at the end the decision will be based on "winner takes all" (selection of the maximum). For example, running, jogging, and walking involve the lower limb activities whereas boxing, clapping, and waving make flow in the upper limb of human object (for interested readers, please refer to [32]).

2.4. Extreme Learning Machine (ELM). Neural networks have been widely utilized in several research areas because of their ability to estimate difficult nonlinear mappings straight from the sample of input as well as offering models for a big class of artificial and natural phenomena that are problematic to hold via classical parametric techniques. Recently, Huang et al. [33–35] presented a novel algorithm for learning regarding single layer feed-forward neural network structural design named extreme learning machine (ELM) that solves the problems initiated through algorithms using gradient descent, for instance, backpropagation used in ANNs. ELM is able to considerably diminish the time quantity required to train neural network and has greatly enhanced faster learning and generalization performance. It needs lesser interventions of human and can run significantly faster than conventional techniques. It routinely concludes the parameters of network entirely, which evades unimportant external intervention by human and more effective in real-time applications. Some

advantages of extreme learning machine can be named: simplicity in usage, quicker speed of learning, greater generalization performance, appropriateness for several nonlinear kernel functions, and activation function [36]. Single hidden layer feed-forward neural network (SLFN) function with hidden nodes [37, 38] can be shown by mathematical explanation of SLFN integrating additive and sigmoid hidden nodes together in a joined method provided as follows:

$$f_L(x) = \sum_{i=1}^L \beta_i G(s_i, b_i, x), \quad x \in \mathfrak{R}^n, \quad a_i \in \mathfrak{R}^n. \quad (7)$$

Let a_i and b_i be the parameters of learning in hidden nodes and β_i represent the connecting weight of i^{th} for output node of hidden node. $G(s_i, b_i, x)$ is the output of i^{th} hidden node with respect to the input x . For additive hidden node with activation function $G(x) : \mathfrak{R} \rightarrow \mathfrak{R}$ (e.g., sigmoid and threshold), $G(s_i, b_i, x)$ is given by

$$G(a_i, b_i, x) = g(a_i, x + b_i), \quad b_i \in \mathfrak{R}. \quad (8)$$

Let a_i be the connecting weight vector of the input layer to i^{th} hidden node and b_i the i^{th} hidden node bias. For N , arbitrary different examples $(x_i, t_i) \in \mathfrak{R}^n \times \mathfrak{R}^m$. Now, x_i is a $n \times 1$ vector of contribution and t_i is a $m \times 1$ vector of target. If an SLFN by L hidden nodes can be estimated, these N samples have zero error. If then infers that there exist β_i, a_i , and b_i such that

$$f_L(x_j) = \sum_{i=1}^L \beta_i G(a_i, b_i, x), \quad j = 1, 2, \dots, N. \quad (9)$$

The equation above is mentioned in compacted way as follows:

$$H\beta = T, \quad (10)$$

where

$$H(\hat{a}, \hat{b}, \hat{x}) = \begin{bmatrix} G(a_1, b_1, x_1) & G(a_L, b_L, x_1) \\ G(a_1, b_1, x_N) & G(a_L, b_L, x_N) \end{bmatrix}_{N \times L}, \quad (11)$$

with $\hat{a} = a_1, \dots, a_L; \hat{b} = b_1, \dots, b_L; \hat{x} = x_1, \dots, x_N$. Consider

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_L^T \end{bmatrix}_{N \times m}. \quad (12)$$

Let H represent the hidden layer of SLFN output matrix with i^{th} column of H being i^{th} hidden nodes output with respect to inputs x_1, x_2, \dots, x_N . In terms of method application, the proposed approach seems to be a straight video processing task for machine. The rate of involving video frame is very much dependent on temporal order considering the information extraction in each pathway. ABM requires two frames by having two time unit differences and it is very similar with motion pathway. Considering that there will be an implementation of interaction between two independent processing streams which comprises the visual guidance from optical flow to SSA which it needs more frames, it means every frame

for being processed by ABM involves two frames for motion information. Furthermore, ABM itself requires two frames for operation so generally four frames are needed to operate whole system for one step. However, in the case of no internal additional interaction, there will be just two frames for each step.

3. Experimental Results

The approach has considered recognition task of biological movement in mammalian visual system. It followed the original model in this area whereas it has been scrutinizingly developed in many parts including process of object recognition in the form pathway and implying fuzzy inference in motion pathway. Yet, development in this model has suggested the implementation of interaction within both pathways processing streams. However, in the comparison part, both cases have been investigated. In addition, the influence of various classifiers for changing the decision making portion also has been analyzed. Two different classifiers have classified to examine the decision making effects in the model. Besides all these biologically inspired explanations, this machine perspective of the task is human action recognition. There must be many important cautions to be considered, including the biological point of view during entire steps of the task. For benchmarking of the method and following computer vision normality and estimation of the accuracy and performance, human action recognition datasets have been used. For general accuracy of system performance, KTH human action [39] has used and comparisons have been recorded and presented in the following sections. Moreover, Weizmann human action recognition robustness dataset [40] is also used to show the robust performance using the presented techniques. KTH action dataset as one of the principal single person human action datasets contains 598 action sequences and six different single person actions types, that is, boxing, jogging, clapping, walking, running, and waving. 25 people perform the actions in diverse conditions: outdoors with different clothes (s3), outdoors with scale variation (s2), outdoors (s1), and indoors with lighting variation (s4). Here, the sequences resolutions become 200×142 pixels through downsampling. For the approach, 5 random cases (subjects) have been used for training and making the form and motion predefined templates. As it is mentioned in the literature, KTH is a robust intrasubject difference with large set whereas the camera for taking the video throughout the preparation had some shacking and it creates many difficulties to use this database. Furthermore, it has four scenarios which are separately and independently tested and trained (i.e., four visually different databases, which share the same classes). Both alternatives have been run. For considering the human actions symmetry problem, there is a sequences mirror function along with vertical axis which can be obtainable for testing and training sets. Here all probable human actions intersection has been considered (e.g., one video has 24 and 32 action frames.)

3.1. Contribution between Motion and Form Features. Substantial contribution on the model development insipid of supervised Gabor wavelet based object recognition and additional inference fuzzy system in motion processing pathway is considered as interaction of two processing pathways along with utilization of different decision making part that can be done through changing the classifiers and analysis of its performance. Considering that there are many ways for combination of information obtained by these two processing pathways and much psychological, physiological, and neurophysiological evidence regarding the interaction of independent processing streams, this approach follows the mentioned valuable evidence to improve previously presented models (all follows the recognition mechanism of biological movement in original model). Importance of this interaction between these pathways is investigated via benchmarking performance within state-of-the-art methods (please see Table 1). Furthermore, the comparison is not only valuable in terms of information interaction decision making performance. It has very substantial result to represent the performance of modification in decision making parts. Here, we have shown a method for development of biologically inspired model of biological movement with respect to the original model and previous approaches. Feature extraction for pathways interaction and decision making between them has been considered which modified the feed-forward structure of these independent information.

3.2. Results. Implementing our method in terms of accuracy is considered as two stages for the general accuracy and stability test. The general accuracy is obtained for comparison study for interaction justification within the processing streams that have been done using KTH human action recognition dataset. For this aim comparison with state-of-the-art methods also considered the same dataset. Task of the proposed method has been implied by general human action recognition task. However, this task was also the same in the stability testing. Weizmann human action robustness dataset is used concerning the cluttered background benchmarking of robustness. Using ABM is one of the strength points of proposed development in the model. Furthermore, optical flow involvement and information combination between two processing pathways can be a very good reason for this. Fuzzy inference system in the motion pathway is a good point for increasing the robustness. It is very obvious due to eliminating very quick changes of flow within optical flow outcomes. Fast variations of flow in motion pathway usually can be a cause of disparity within the decision making processes. This can diminish accuracy rate for the model and it is not realistic in the actual environmental situations because every second of the video including many frame images and changing the action in fraction of second and within the frames seems far from reality. It must be considered because the model is the implementation of mammalian visual system. An overview to attain the action prototype way and its discussion is considered in this portion. The comparison of development in the approach in the aspect of interaction along with decision making expansion is illustrated and discussed in this section.

TABLE 1: The proposed comparison method recognition results revealed among previous human action recognition method accuracies (bio- or non-bioinspired) on KTH human action dataset.

Methods	Accuracy (%)	Years
Schüldt et al., [39]	71.72	2004
Niebles et al., [45]	83.33	2008
Jhuang et al., [42]	91.79	2007
Schindler and Van Gool [11]	92.79	2008
Wang and Huang [34]	91.29	2005
Zhang and Tao [43]	U-SFA: 86.67	2012
	S-SFA: 86.40	
	D-SFA: 89.33	
	SD-SFA: 93.87	
Yousefi and Loo [32]	SNN: 86.46	2014
Proposed method	ELM: 96.5	

3.3. Overview on Action Prototypes. As it is used and presented [32, 41], every human action has certain form similarity and specific structural configurations. These mentioned shapes can be a substantial abstract of every human action during time process in video. We divide every human action movement in its sequences to five primitive basic movements which is not necessarily common among various movements. These primarily action abstracts are called action prototypes and can mostly reconstruct every human action applying them. They also can be very good representative of the action in many environmental situations and style invariance property in the actions. It is motivated by the training map of human objects within the actions or any other tasks. These action prototypes have been computed through two-time utilization of synergetic neural network melting for every different action which gives action abstracts. For this aim five different action episodes are randomly chosen and considered as training map of the proposed approach and excluded from the testing dataset. Deliberate prototype images seeing style invariance signify one action in five different snapshots (for more information please refer to [32, 41]). The outcomes of melting process in synergetic neural network does very much look like abstracting a set of human object actions using eigendecomposition which gives eigenimages within a set. The action prototype has a very significant and essential role in the form processing information in the ventral stream.

3.4. Experimental Results. The benchmarks are mentioned in this section and the approach follows the implementation of fuzzy inference using optical flow division presented in [13] and further interactions are scrutinizingly investigated. Moreover, modification in decision making section explores in-depth. The task in this section considered more look-alike computer vision task regarding human action recognition. Confusion matrices are obtained in the similar experimental conditions as [13]. The tables and confusion substantially represent the better result presentation within modification of the classifier and decision making block of the algorithm in biologically inspired model. Recognition accuracy comparison has been demonstrated in Figures 6 and 7. Comparison

performed by depiction of the accuracy in terms of comparison with state-of-the-art methods and similar methods which are more biologically inspired. Similarity of the presented model has been deliberated in the assessment. KTH human action dataset is used for benchmarking and the evaluation assessment compared with state-of-the-art methods in the same dataset for consistency in the experimental results (see Figure 4) [11, 42–46]. Also it is noticeable, as previously mentioned, that the training map and action prototypes obtained from the random selection of the human action set in four different scenario videos from KTH and excluded from the testing set have no overlap between these two sets. Utilization of the training map within the performance estimation is shown by simple comparison in current videos frames snippets with the action prototype which is merely template matching. It gives a score of the matching for every human action prototype. It comprises the information of form representing the ventral processing streams outcomes and needs to involve the information motion pathways. Figure 6 depicts the confusion matrix and Figure 7 shows some results of the proposed expansion in the recognition mechanism of biological movements. Confusion matrix rows denote the results of corresponding classification, although, respectively, columns signify the examples to be classified. As it is shown through these figures and corresponding results, the highest confusion happens among running, jogging, and walking. To distinguish these actions is difficult as the actions performance by some subjects has resemblance. Correspondingly, another misclassification happens mainly between alike classes, similar earlier confusion, or hand-clapping and hand-waving (please see confusion matrices in Figures 5 and 6). Following the mentioned parts regarding the action prototypes computed by twofold synergetic neural network melting within whole action frames. These action prototypes perform as action abstract within the recognition mechanism. It can be used for recognition and categorization of action in the form pathway. Besides, there was an adjustment in this pathway which involved the motion information into form path via analysis of the type of action whether it occurs in lower or upper limb; the relevant membership function organized this task. There can be a discussion for this performance; this approach implementation can be done through a simple programming rather than complex mathematical computation. The method gives very good time delay memory which is totally time dependent and it provides robustness within quick changes of optical flow and motion information plus dramatically diminishing the disparity rate.

3.5. Relation to Existing Methods and Discussion. The presented method is utilized for mechanism of biological movement and main focus of this approach concentrates on interaction of two visual processing streams and decision making within these paths. Here, general difference and similarity between existing methods and this approach are shortly investigated. The method is totally in direction of psychological and physiological evidences. It particularly follows the original model of biological movement recognition [8, 11, 42, 47] considering psychological evidences [14, 15]. The obvious

change in this area can be considered applying a supervised Gabor wavelet based object recognition method in ventral stream which is presented [41]. Applying ABM increases the focus of form pathway in information of form and structure of human object and provides more stability in the recognition task. Moreover, it follows the characteristic of simple and complex cells to attain the shape of object in form pathway and gives reliability and robustness in form pathway particularly in the clutter background. On the other hand, information of motion is considered through utilization of optical flow in dorsal pathway. Optical flow can substantially give motion information within the video frames and object movements can be shown by simple silhouette representing the flow of human object within the considered frames. Optical flow is successfully used by the original model many times but it can reveal instability due to fast variation of the input video streams. The fuzzy optical flow division has been introduced for this pathway and increases the rate of stability and more reliability via delivering the time memory and time delay, in the processing of quick variation input [32]. It gives good combination of fast variation of motion information and this delay gave more robustness in the recognition mechanism. The interaction of these two parallel independent processing streams has been investigated for many years in different areas especially psychology and physiology. In visual system, Gabor like filters mainly have a representation role for simple and complex cells. ABM is an appropriate characteristic for this part, particularly concerning its contribution in object recognition task. It could follow the concerning encoded object shape [15]. The shape of object concerns in form pathway and ventral processing stream has been properly deliberated based on training phase and explanation for its reliability is done human prototypes. ABM is somehow contemplate Gabor action inducement for pin-down form processing at two-level local information around limb angle from orientations and global body structure of Gabor signaled by Gabor paths spatial arrangement. On the contrary, optical flow used for motion information extraction has tracked the second characteristic and contains filtering through direction selection sensors and its incorporation for resolving the well-known aperture problem. Motion information shows both motion signals local velocity categories and motion trajectories joint utilizes signals in form path by guiding SSA in ABM [48] as a good representation of crossconnection between V4 and MT [14, 15], that is, a very substantial interaction effect within these two processing pathways [32, 41]. However, the interaction in both processing pathways is not limited to this interaction and will occur in different regions of visual processing stream. Form and motion processing principal view in human visual system, it is assumed that these two traits are controlled by self-determination and distinct modules ([8, 11, 13, 23]). It has been identified that form signal information can influence motion processing more broadly than earlier believed (see [15]) and the proposed approach reflects direct motion information effect on form processing. Visual system connectivity is categorized by crossconnections with respect to feed-forward of parallel connection ([14, 49, 50]). Optical flow division method delivers bottom-up and top-down

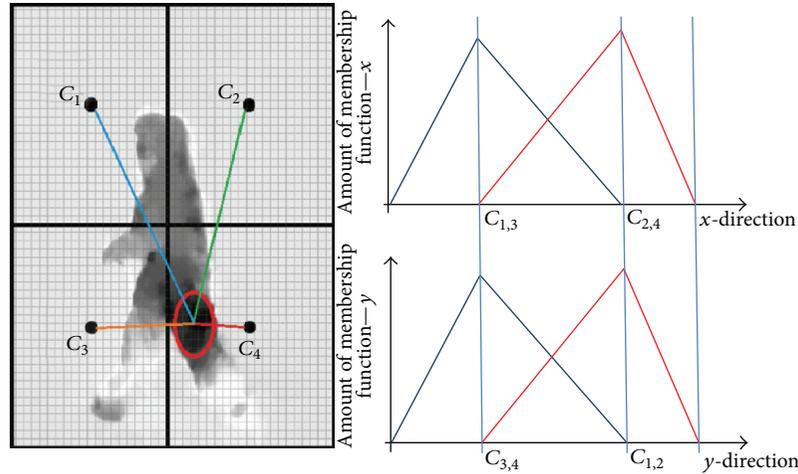


FIGURE 3: The result of dorsal processing stream applying optical flow [26] and the optical flow division into the fuzzification has been depicted. The resolution of divisions is designed for categorization of actions group to have additional interference of dorsal and ventral processing streams. It can be a good representative of the interaction on MT, middle temporal of dorsal stream, and V4, ventral stream (for shape and orientation), or the MST area with inferior temporal (IT) (see more details in [14, 15]). The membership function of the action will be estimated from the position of maximum flow in the flow image. Membership values are aggregated through the proposed technique to increase the robustness. The input image of action mentioned in the figures is obtained from KTH human action recognition dataset [39].



FIGURE 4: The figure depicts KTH human action dataset. To test the recognition of biological movements one of the well-known human action recognition datasets has been utilized in its performance. Here, the set represents KTH human action dataset. It is noticeable to mention that KTH dataset is one of the largest human action datasets having six various human actions in four different scenarios.

processing interaction and connection among brain regions within dual computational streams and can be decent descriptive connection between dorsal and ventral streams (i.e., V4 and MT; see Figure 3) [14, 15]. Dorsal stream is correspondingly supposed to preform spatial computation correspondences (where) and ventral stream regarding object

recognition task (what) in the cortical areas of V4, V2, V1, and IT(inferotemporal cortex) accompanied by existing conflict evidence to a whole separation of “what” and “where” in macaque brain information (see [50, 51]) demonstrating about information for position and size of objects are similarly signified in macaques inferotemporal cortex. However, proposed method is an initial spatial configuration and distinctiveness isolation into distributed processing pathways requires weighty hardware computation. However having optical flow low resolution divisions (four alienated portions) could be a worthy factor aimed at the computational load diminishing. The precise classified sequences are described as highest results existing in the field literature. To place suggested technique in this context, we have mentioned it with the state-of-the-art methods. Our method is a frame-based which tracks for all frames inaction sequences. The individual labels formerly attained from training map basically associate with a label sequence done majority voting (like [11, 13]). The interacted approach comparison by state-of-the-art methods has been performed and it is shown in Table 1. Its accuracy just represents the concerns in comparison with other similar methods indicating relative compatibility and significant performance for proposed approach.

4. Conclusion

The presented approach has addressed a very substantial interrelevant comparison of the interaction of two processing streams of mammalian brain visual system. The developments in decision making portion along with a significant comparison within these pathways have been scrutinizingly investigated. Generally, the interaction of motion information to form processing pathways has shown a very good and reasonable effect in the recognition model and it can

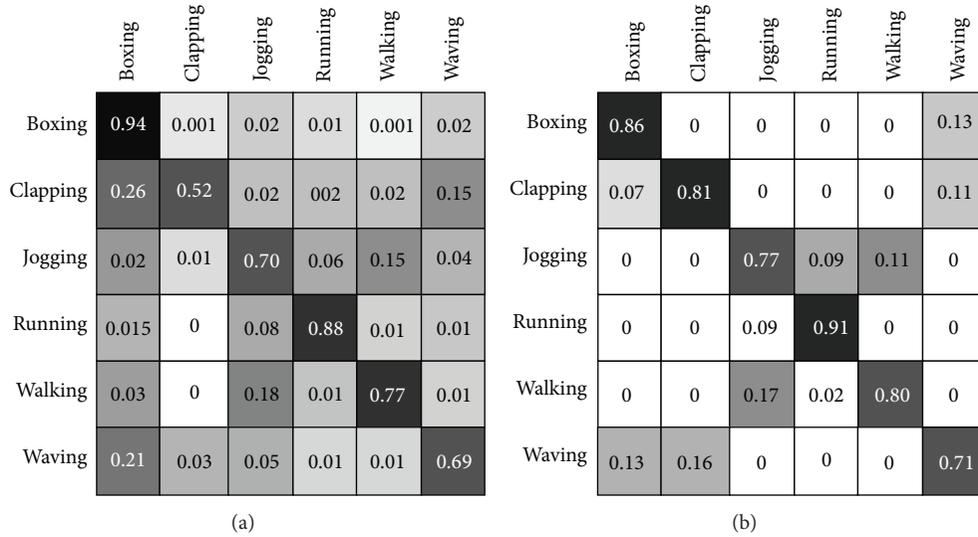


FIGURE 5: Confusion matrices SNN classifying KTH dataset obtained by adapted active basis model as combination of form and motion pathways. Confusion matrices of the proposed approach have been presented for the case without fuzzy interference system, left matrix, and, after it, right matrix which are achieved from human action movements of KTH dataset [39]. The robustness of the method after adding the fuzzy interference stabilizer is considerably increased. The wrong recognitions in the left confusion matrix have been decreased especially in case of some actions, that is, clapping. Moreover, soar of robustness helps increase the overall accuracy and gives better results in classification of biological movement. The accuracy of categorizations using unbalanced SNN is reached at 86.46%.

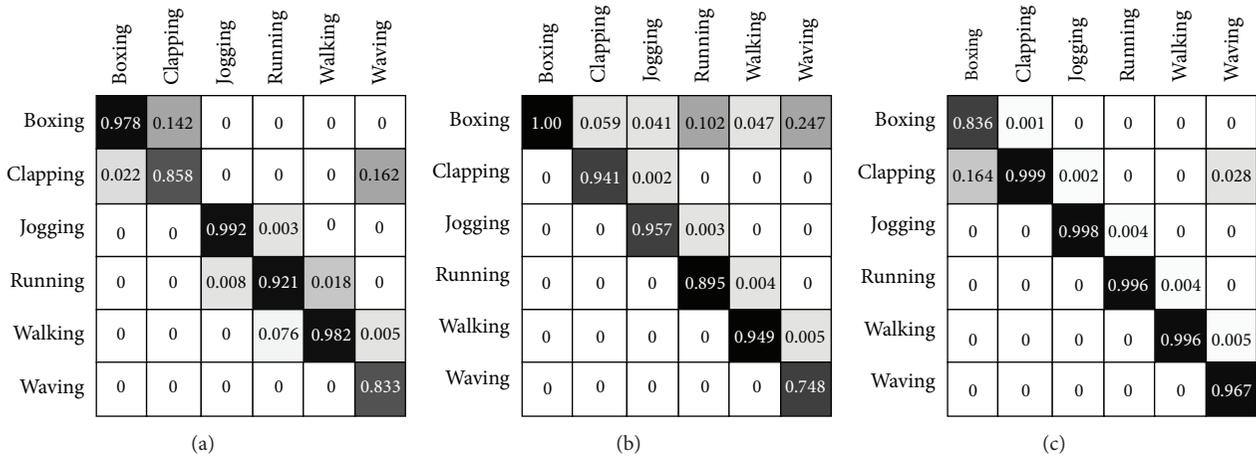


FIGURE 6: Confusion matrices ELM classifying KTH dataset attained by adapted active basis model as combination of form and motion pathways. Confusion matrices of the proposed approach have been presented which is obtained from human action movements of KTH dataset [39]. There are three different kernels which have been used in classifying using ELM algorithm [33–38] in the decision making and categorization of the biological movement. From left to right, RBF kernel-ELM, wavelet kernel ELM, and sigmoid-ELM confusion matrices have been depicted where sigmoid kernel-ELM has better results in classification of biological movement. The accuracy of categorizations is ELM-Wav = 91.5%, ELM-RBF = 92.7%, and ELM-Sig = 96.5%.

represent crossconnection of V4 and MT in brain [23]. The human action prototype outcomes using twofold synergetic neural network melting have been reviewed and considered for recognition of form information in form processing pathway. For benchmarking, the task has been converted to a computer vision and human action recognition and two datasets have been used regarding evaluation and recognition performance with the state-of-the-art methods. The cross-connection in feed-forward biologically inspired method

also has been presented accordingly. Correspondingly it had respectable performance in dissimilar datasets along with reasonable computational cost. As a limitation, it currently has no mechanisms for invariance alongside rotation and variations in viewpoint although it can be considered to put mechanism regarding multiscale. ABM is a delicate algorithm and requires further attention though its training still can be more developed to be a powerful tool for form pathway that is far from this approach purposes.

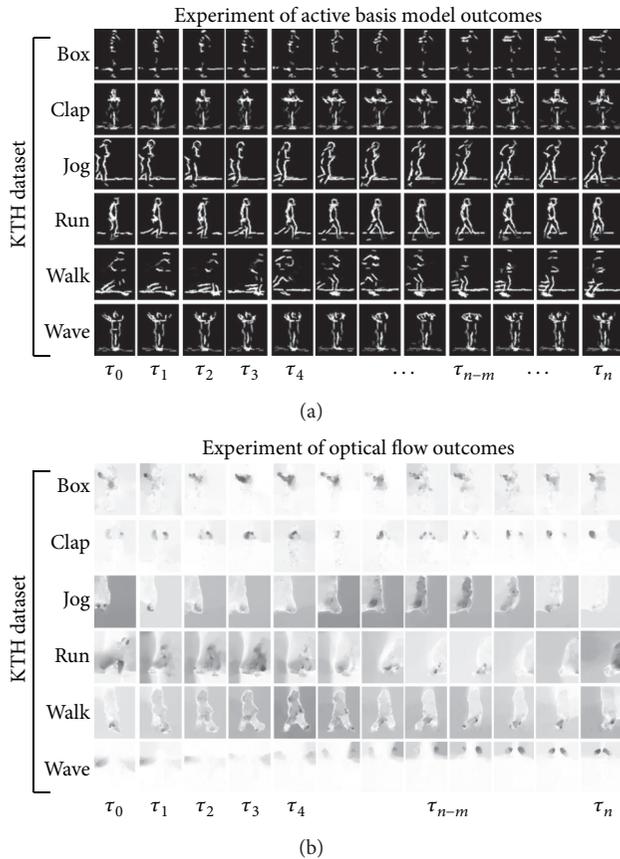


FIGURE 7: Simulation results for simple biological movement paradigm based on ABM [24] in the ventral processing stream and optical flow [26] in dorsal stream are shown. Each row within the panel reveals the response of ABM during the episode as well as flow generated for every different action. The set of biological movements belongs to the biological movements which is from KTH dataset [39]. (a) The simulation results of the different actions of KTH dataset. (b) Optical flow simulation results.

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 Ce Liu for providing the code for layerwise optical flow [26] as well as Ying Nian Wu for active basis model code [24]. They are very grateful to Professor Guang-Bin Huang and his student Zhou Hongming in Nanyang Technological University for their guidance in using ELM [32–38, 41]. They acknowledge Naoki Masuyama contributions in this regard by providing useful comments. This research was sponsored by Grants from Contract no. UM.C/HIR/MOHE/FCSIT/10, High Impact Research (HIR) Foundation in University Malaya (UM), Malaysia.

References

- [1] E. H. Adelson and J. R. Bergen, “Spa-tiotemporal energy models for the perception of motion,” *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 2, no. 2, pp. 284–299, 1985.
- [2] S. Shioiri and P. Cavanagh, “ISI produces reverse apparent motion,” *Vision Research*, vol. 30, no. 5, pp. 757–768, 1990.
- [3] S. Shioiri and K. Matsumiya, “Motion mechanisms with different spa- tiotemporal characteristics identified by an MAE technique with superimposed gratings,” *Journal of Vision*, vol. 9, no. 5, article 30, 2009.
- [4] K. Moutoussis and S. Zeki, “A direct demonstration of perceptual asynchrony in vision,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 264, no. 1380, pp. 393–399, 1997.
- [5] D. Whitney and I. Murakami, “Latency difference, not spatial extrapolation,” *Nature Neuroscience*, vol. 1, pp. 656–657, 1998.
- [6] A. O. Holcombe, “Seeing slow and seeing fast: two limits on perception,” *Trends in Cognitive Sciences*, vol. 13, no. 5, pp. 216–221, 2009.
- [7] S. Thorpe, D. Fize, and C. Marlot, “Speed of processing in the human visual system,” *Nature*, vol. 381, no. 6582, pp. 520–522, 1996.
- [8] M. A. Giese and T. Poggio, “Neural mechanisms for the recognition of biological movements,” *Nature Reviews Neuroscience*, vol. 4, no. 3, pp. 179–192, 2003.
- [9] M. Riesenhuber and T. Poggio, “Hierarchical models of object recognition in cortex,” *Nature Neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999.
- [10] M. Riesenhuber and T. Poggio, “Neural mechanisms of object recognition,” *Current Opinion in Neurobiology*, vol. 12, no. 2, pp. 162–168, 2002.
- [11] K. Schindler and L. Van Gool, “Action Snippets: how many frames does human action recognition require?” in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, vol. 1–12, pp. 3025–3032, Anchorage, Alaska, USA, June 2008.
- [12] S. Danafar, A. Giusti, and J. Schmidhuber, “Novel kernel-based recognizers of human actions,” *EURASIP Journal on Advances in Signal Processing*, vol. 2010, Article ID 202768, 2010.
- [13] S. Danafar, A. Gretton, and J. Schmidhuber, “Characteristic kernels on structured domains excel in robotics and human action recognition,” in *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pp. 264–279, Springer, 2010.
- [14] L. L. Cloutman, “Interaction between dorsal and ventral processing streams: where, when and how?” *Brain and Language*, vol. 127, no. 2, pp. 251–263, 2012.
- [15] G. Mather, A. Pavan, R. Bellacosa Marotti, G. Campana, and C. Casco, “Interactions between motion and form processing in the human visual system,” *Frontiers in Computational Neuroscience*, 2013.
- [16] B. M. Dow, A. Z. Snyder, R. G. Vautin, and R. Bauer, “Magnification factor and receptive field size in foveal striate cortex of the monkey,” *Experimental Brain Research*, vol. 44, no. 2, pp. 213–228, 1981.
- [17] S. Eifuku and R. H. Wurtz, “Response to motion in extrastriate area MSTl: center-surround interactions,” *Journal of Neurophysiology*, vol. 80, no. 1, pp. 282–296, 1998.
- [18] J. P. Jones and L. A. Palmer, “An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat

- striate cortex," *Journal of Neurophysiology*, vol. 58, no. 6, pp. 1233–1258, 1987.
- [19] Z. Kourtzi and N. Kanwisher, "Activation in human MT/MST by static images with implied motion," *Journal of Cognitive Neuroscience*, vol. 12, no. 1, pp. 48–55, 2000.
- [20] K. S. Saleem, W. Suzuki, K. Tanaka, and T. Hashikawa, "Connections between anterior inferotemporal cortex and superior temporal sulcus regions in the macaque monkey," *Journal of Neuroscience*, vol. 20, no. 13, pp. 5083–5101, 2000.
- [21] M. A. Giese and L. M. Vaina, "Pathways in the analysis of biological motion: computational model and fMRI results," *Perception*, vol. 30, p. 119, 2001.
- [22] R. Laycock, S. G. Crewther, and D. P. Crewther, "A role for the "magnocellular advantage" in visual impairments in neurodevelopmental and psychiatric disorders," *Neuroscience and Biobehavioral Reviews*, vol. 31, no. 3, pp. 363–376, 2007.
- [23] B. Wang, Y. Liu, W. Wang, W. Xu, and M. Zhang, "Multi-scale locality-constrained spatiotemporal coding for local feature based human action recognition," *The Scientific World Journal*, vol. 2013, Article ID 405645, 11 pages, 2013.
- [24] Y. N. Wu, Z. Si, H. Gong, and S. Zhu, "Learning active basis model for object detection and recognition," *International Journal of Computer Vision*, vol. 90, no. 2, pp. 198–235, 2010.
- [25] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.
- [26] C. Liu, *Beyond pixels: exploring new representations and applications for motion analysis [Ph.D. thesis]*, Massachusetts Institute of Technology, 2009.
- [27] T. Brox, A. Bruhn, N. Papenberger, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Computer Vision—ECCV 2004*, vol. 3024 of *Lecture Notes in Computer Science*, pp. 25–36, Springer, Berlin, Germany, 2004.
- [28] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.
- [29] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sanchez, "Symmetrical dense optical flow estimation with occlusions detection," in *Proceedings of the Computer Vision (Eccv '02)*, vol. 2350, pp. 721–735, 2002.
- [30] L. A. Zadeh, "Fuzzy sets," *Information and Computation*, vol. 8, pp. 338–353, 1965.
- [31] B. Chen and X. Liu, "Delay-dependent robust H_∞ control for T-S fuzzy systems with time delay," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 4, pp. 544–556, 2005.
- [32] B. Yousefi and C. K. Loo, "Development of biological movement recognition by interaction between active basis model and fuzzy optical flow division," *The Scientific World Journal*, vol. 2014, Article ID 238234, 14 pages, 2014.
- [33] G. Huang, Q. Zhu, and C. 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, July 2004.
- [34] D. Wang and G.-B. Huang, "Protein sequence classification using extreme learning machine," in *Proceeding of the International Joint Conference on Neural Networks (IJCNN '05)*, vol. 3, pp. 1406–1411, can, July 2005.
- [35] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [36] R. Rajesh and J. S. Prakash, "Extreme learning machines—a review and state-of-the-art," *International Journal of Wisdom Based Computing*, vol. 1, no. 1, pp. 35–49, 2011.
- [37] N. Liang, G. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.
- [38] G. Huang, L. Chen, and C. 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.
- [39] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, pp. 32–36, Cambridge, UK, August 2004.
- [40] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.
- [41] B. Yousefi, C. K. Loo, and A. Memariani, "Biological inspired human action recognition," in *Proceedings of the IEEE Workshop on Robotic Intelligence in Informationally Structured Space (RiiSS '13)*, pp. 58–65, IEEE, 2013.
- [42] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, pp. 1253–1260, Rio de Janeiro, Brazil, October 2007.
- [43] Z. Zhang and D. Tao, "Slow feature analysis for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 436–450, 2012.
- [44] N. Naveen, V. Ravi, C. R. Rao, and N. Chauhan, "Differential evolution trained radial basis function network: application to bankruptcy prediction in banks," *International Journal of Bio-Inspired Computation*, vol. 2, no. 3-4, pp. 222–232, 2010.
- [45] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 299–318, 2008.
- [46] Y. Wang and G. Mori, "Human action recognition by semilantent topic models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1762–1774, 2009.
- [47] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 726–733, Nice, France, October 2003.
- [48] S. M. Thurman and H. Lu, "Complex interactions between spatial, orientation, and motion cues for biological motion perception across visual space," *Journal of Vision*, vol. 13, no. 2, article 8, 2013.
- [49] C. Distler, D. Boussaoud, R. Desimone, and L. G. Ungerleider, "Cortical connections of inferior temporal area TEO in macaque monkeys," *Journal of Comparative Neurology*, vol. 334, no. 1, pp. 125–150, 1993.
- [50] D. J. Felleman and D. C. Van Essen, "Distributed hierarchical processing in the primate cerebral cortex," *Cerebral Cortex*, vol. 1, no. 1, pp. 1–47, 1991.
- [51] S. R. Lehky, X. Peng, C. J. McAdams, and A. B. Sereno, "Spatial modulation of primate inferotemporal responses by eye position," *PLoS ONE*, vol. 3, no. 10, Article ID e3492, 2008.

Research Article

Efficiently Hiding Sensitive Itemsets with Transaction Deletion Based on Genetic Algorithms

Chun-Wei Lin,^{1,2} Binbin Zhang,³ Kuo-Tung Yang,⁴ and Tzung-Pei Hong^{4,5}

¹ Innovative Information Industry Research Center (IIIRC), School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China

² Shenzhen Key Laboratory of Internet Information Collaboration, School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China

³ Medical School, Shenzhen University, Shenzhen 518060, China

⁴ Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan

⁵ Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan

Correspondence should be addressed to Binbin Zhang; binbinsherry.zhang@gmail.com

Received 28 May 2014; Revised 7 August 2014; Accepted 8 August 2014; Published 1 September 2014

Academic Editor: Shifei Ding

Copyright © 2014 Chun-Wei Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Data mining is used to mine meaningful and useful information or knowledge from a very large database. Some secure or private information can be discovered by data mining techniques, thus resulting in an inherent risk of threats to privacy. Privacy-preserving data mining (PPDM) has thus arisen in recent years to sanitize the original database for hiding sensitive information, which can be concerned as an NP-hard problem in sanitization process. In this paper, a compact prelarge GA-based (cpGA2DT) algorithm to delete transactions for hiding sensitive itemsets is thus proposed. It solves the limitations of the evolutionary process by adopting both the compact GA-based (cGA) mechanism and the prelarge concept. A flexible fitness function with three adjustable weights is thus designed to find the appropriate transactions to be deleted in order to hide sensitive itemsets with minimal side effects of hiding failure, missing cost, and artificial cost. Experiments are conducted to show the performance of the proposed cpGA2DT algorithm compared to the simple GA-based (sGA2DT) algorithm and the greedy approach in terms of execution time and three side effects.

1. Introduction

With the rapid growth of data mining technologies in recent years, useful and meaningful information can thus be easily discovered for the purpose of decision making in different domains. The discovered information can be mostly classified into association rules [1–5], sequential patterns [6–9], classification [10–12], clustering [13, 14], and utility mining [15–18], among others. Among them, mining association rules method is the most common way to find the potential relationships between the purchased items or goods in a very large database. Some applications require protection against the disclosure of private, confidential, or secure data. For example, social security numbers, address information, credit card numbers, and purchasing behaviors of customers

can be considered as the confidential, private, or privacy information.

Instead of personal information, privacy issue can be extended to business. Based on business purposes, shared information among companies may be extracted and analyzed by other partners, thus causing the security threats. Privacy-preserving data mining (PPDM) [19–22] was proposed to reduce privacy threats by hiding sensitive information while allowing required information to be discovered from databases. Such data may implicitly contain confidential information that will lead to privacy threats if it is misused. Heuristic methods [20, 21, 23–26] have been proposed to choose the appropriate data for sanitization in order to hide the sensitive information. During the procedure to hide the sensitive information, side effects of missing cost and

artificial cost are thus generated and should be concerned in PPDM. The optimal way to select the sensitive information to be hidden is, however, concerned as the NP-hard problem in sanitization process [22, 27]. Genetic algorithms (GAs) [28] are able to find optimal solutions using the principles of natural evolution. The amount of chromosomes is thus required to process the several operations in evaluation process of simple GAs.

To solve the limitations of traditional GA-based algorithms with high requirements of memory and computations at each evolutionary process, the compact GA (cGA) mechanism [29] and the prelarge concept [30] are adopted in the proposed cpGA2DT algorithm. Based on the cGA mechanism, only two chromosomes are competed to each other at each iteration. The probabilities of transactions to be selected are increased along with the winner chromosome. The probabilities of transactions to be selected are, however, decreased along with the loser chromosome. Since only two chromosomes are generated for the competition, the memory requirements of populations can be greatly reduced. In addition, a flexible fitness function is designed to evaluate three side effects at each evolutionary process. This procedure causes the computations of multiple database rescans. The prelarge concept is adopted in the proposed cpGA2DT algorithm to find the prelarge itemsets [30, 31] in advance, thus reducing the computations of multiple database rescans at each evolution. To the best of our knowledge, this is the first approach to solve the limitations by considering both the time and the space complexities with transaction deletion for hiding sensitive itemsets. A straightforward approach (greedy) and a simple GA-based algorithm are also designed as a benchmark to evaluate the performance of the proposed cpGA2DT in regard to the execution time and the number of three side effects in the experiments. Contributions of this paper can be illustrated as follows.

- (1) Most past approaches applied heuristic ways to sanitize the original database for the purpose of hiding sensitive itemsets by deleting partial items. In this paper, a GA-based approach is thus proposed to optimize the selected transactions to be deleted, thus minimizing the side effects in PPDM.
- (2) It requires the amount of memory in evaluation process based on traditional GA approach. In this proposed approach, cGA is applied to reduce the population size based on probability distribution to select the appropriate transactions to be deleted.
- (3) The prelarge concept is used in the proposed algorithm to reduce the execution time for database rescan in chromosome evaluation.
- (4) An evaluation function with three adjustable weights is designed in the evaluation process to minimize the side effects of PPDM.

The remainder parts of this paper are organized as follows. Related works are described in Section 2; preliminary of PPDM is mentioned in Section 3. The proposed approach is illustrated in Section 4. An example is given in Section 5.

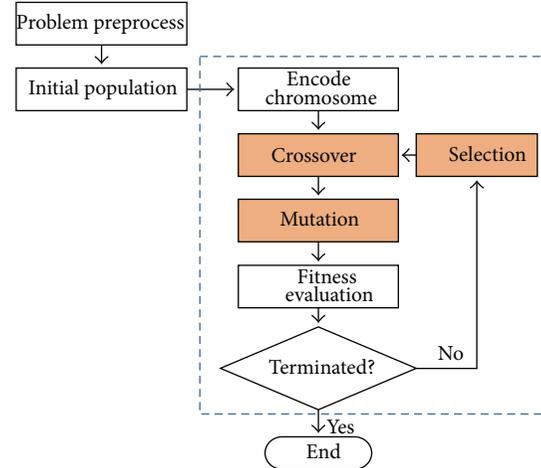


FIGURE 1: Flowchart of GAs.

Experiments are conducted in Section 6. Conclusion is given in Section 7.

2. Review of Related Works

Related works of genetic algorithms, data sanitization, and prelarge concept are briefly reviewed in this section.

2.1. Genetic Algorithms. Holland applied the natural selection and the survival of the fittest of Darwin theory and proposed the evolutionary computation of genetic algorithms (GAs) [28]. GAs are the search techniques, which are designed and developed to find a set of feasible solutions in a limited amount of time [32, 33]. According to the principle of survival of the fittest, GAs generate the next population by various operations with each individual in the population representing a set of possible solutions. Three basic operations including crossover, mutation, and selection are performed on chromosomes for the next generations. Each chromosome is then evaluated by the designed fitness function. This procedure is recursively processed until the predefined termination criteria are achieved. Flowchart of GAs is shown in Figure 1.

Traditional GAs have to generate the size of populations for the purpose of performing crossover, mutation, and selection operations for the next generations, thus causing memory lack problem. Compact genetic algorithm (cGA) was thus proposed to simulate traditional GAs with only the probability vector for selection operation and population size without the crossover and mutation operations in order to generate two individuals (or chromosomes) at competition [29]. The probability of the i th vector in the winner chromosome is increased, but the loser probability is decreased. A cGA algorithm can reduce the memory requirements without the crossover and mutation operations but still can approximately mimic the behaviors of traditional GAs.

2.2. *Data Sanitization.* Data mining [1, 34–37] is progressively developed to extract useful and meaningful information or rules from a very large database. The misuse of data mining techniques may, however, lead to security threats and privacy concerns. Privacy-preserving data mining (PPDM) [19, 23, 24, 38] was thus proposed to hide the confidential, private, or secure information before it is published in public or shared among alliances. Most approaches were proposed to perturb the original database for the purpose of hiding sensitive information in PPDM. Agrawal and Srikant introduced a quantitative measure to evaluate the utility of PPDM methods [19]. Lindell and Pinkas stated hiding confidential information on the union of shared databases among two parties without revealing any unnecessary information [20]. Oliveira and Zaiane, respectively, designed the multiple-rule hiding MinFIA, MaxFIA, and IGA algorithms to efficiently hide sensitive itemsets and introduced the performance measures for three side effects [39]. Dasseni et al. then proposed a hiding approach based on the hamming-distance approach to decrease the confidence or support values of association rules for hiding sensitive information [40]. Three heuristic algorithms are designed, respectively, to increase the supports of antecedent parts, to decrease the supports of consequent parts, and to decrease the support of either the antecedent or the consequent parts until the supports or confidences of association rules below the threshold values. Amiri then proposed aggregate, disaggregate, and hybrid approaches to hide multiple sensitive rules [23]. The designed aggregate approach computes the union of the supporting transactions for all sensitive itemsets. The transactions with the most sensitive and the least sensitive itemsets are thus removed to hide the sensitive information. The disaggregate approach aims to remove individual items from transactions and then remove whole transactions, thus reducing side effects of PPDM. Hybrid one is to combine the previous designed algorithms to firstly identify sensitive transactions and secondly to delete items from those of transactions until the sensitive information has been hidden. Many heuristic approaches are still being developed in progress for the purpose of hiding different types of knowledge in PPDM [21, 26, 41].

The optimal sanitization of databases is regarded to be an NP-hard problem [22, 27]. Genetic algorithms (GAs) were usually used to find optimal solutions in the least amount of time [28]. Fewer studies have adopted GAs to find optimal solutions to hide sensitive information. Han and Ng proposed secure protocols for rule discovery based on private arbitrarily partitioned data among two parties without compromising their data privacy using GAs [42]. It uses the true positive rate multiplied by the true negative rate to define the fitness function for evaluating the goodness of each decision rule. Dehkordi et al. designed three multiobjective methods to partially remove the items from the original database [43]. Only the number of modified transactions is considered in the fitness function for evaluation. The other side effects of missing cost and artificial cost thus arose in the evaluation process. In this paper, three side effects are concerned in the designed fitness function for hiding sensitive itemsets with transaction deletion based on cGA algorithm.

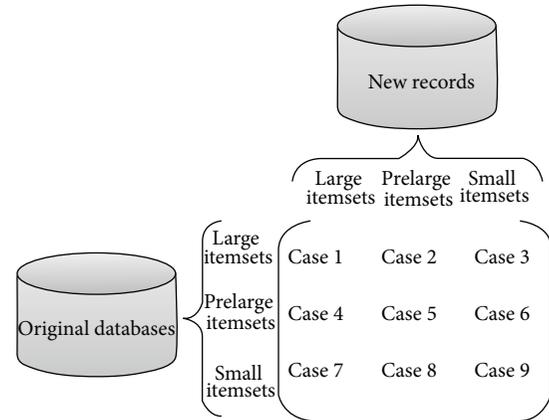


FIGURE 2: Nine cases arise as a result of transaction deletion.

2.3. *Prelarge Concept.* Data mining techniques are used to discover useful and meaningful information or rules to aid managers in making efficient decisions in many different domains. Most data mining techniques handle, however, the static database to extract the required information. Cheung et al., respectively, designed FUP [44] and FUP2 [45] concepts to maintain and update the discovered information in dynamic databases. The original database is still, however, required to be rescanned based on the FUP and FUP2 concepts in the updating process. Hong et al. proposed prelarge concepts [30, 31] for the purpose of efficiently updating the discovered information without rescanning the original database each time. Prelarge itemset is not large itemset but has high potential to be large in the future through the data insertion or deletion process. Upper (the same as the minimum support threshold in conventional mining algorithms) and lower support thresholds are used to define the large and prelarge itemsets. Prelarge itemsets are used as a buffer to reduce the movement of an itemset directly from large to small and vice versa. For transaction deletion based on prelarge concept [30], nine cases thus arose and are shown in Figure 2.

From Figure 2, cases 2, 3, 4, 7, and 8 do not affect the final frequent itemsets of association rules. Case 1 may remove some discovered frequent itemsets of association rules. Cases 5, 6, and 9 may produce new frequent itemsets of association rules. If all frequent or prelarge itemsets are prestored from the original database, cases 1, 5, and 6 can be easily maintained and updated. An itemset in Case 9 cannot possibly be a large itemset in the updated database as long as the number of deleted transactions is a considerably small proportion of the original databases, which can be defined as [30]

$$f \leq \frac{(S_u - S_l) \times |D|}{S_u}, \quad (1)$$

where S_l is a lower support threshold, S_u is an upper support threshold, and $|D|$ is the number of transactions in databases. If the number of deleted transactions satisfies the above condition, which is smaller than the safety bound f , an itemset in Case 9 is absolutely not large in the updated

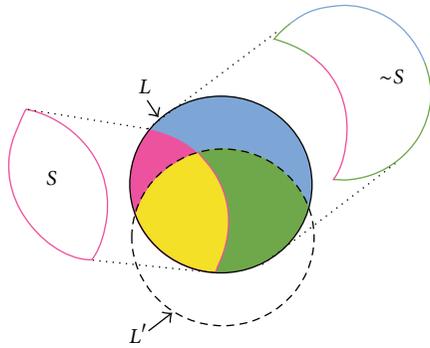


FIGURE 3: The relationship of itemsets before and after the PPDM process.

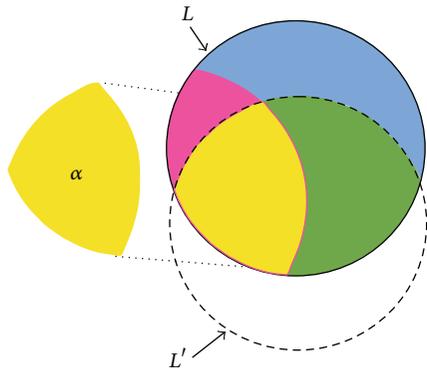


FIGURE 4: The set of sensitive itemsets that fail to be hidden.

databases. It is thus unnecessary to rescan the original databases. In the proposed cpGA2DT, the prelarge concepts are adopted to reduce the database rescan in the evaluation process, thus speeding up computations.

3. Preliminaries

Before sanitization process to hide the sensitive itemsets, frequent itemsets can be discovered by data mining techniques. Let $I \in \{i_1, i_2, \dots, i_n\}$ be the set of items in the database D ; a database D consists of several transactions as $D \in \{t_1, t_2, \dots, t_m\}$, in which each transaction is a set of items. A minimum support threshold is set at σ . Denote a support of an item (itemset) by $\text{sup}(i_j)$. An item (itemset) is denoted by $\text{freq}(i_j)$ if it is considered as a large or frequent item (itemset) as $\text{freq}(i_j) = \text{sup}(i_j)/|D| \geq \sigma$.

In PPDM, it is required not only to hide sensitive itemsets but also to minimize the side effects. The relationship of itemsets before and after the PPDM process can be seen in Figure 3, where L represents the large itemsets of D , S represents the sensitive itemsets defined by users that are large, $\sim S$ represents the nonsensitive itemsets that are large, and L' is the large itemsets after some transactions are deleted.

Let α be the number of sensitive itemsets that fail to be hidden. Thus, the number of sensitive itemsets should ideally be zero after the database is sanitized. The set of sensitive itemsets is shown in Figure 4, in which α part is the interaction of S and L' .

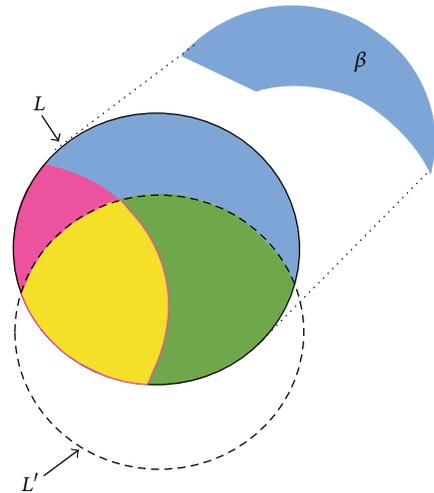


FIGURE 5: The set of sensitive itemsets that fail to be hidden.

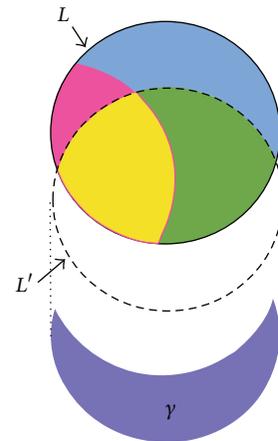


FIGURE 6: The set of artificial itemsets.

Definition 1. The hiding failure of the sensitive itemsets in PPDM is defined as α , in which $\alpha = S \cap L'$.

Another evaluation criterion is the number of missing itemsets, which is denoted by β . A missing itemset is a nonsensitive large itemset in the original database but is not extracted from the sanitized database. This side effect is shown in Figure 5, in which the β part is the difference of $\sim S$ and L' .

Definition 2. The missing itemsets in PPDM are defined by β , in which $\beta = \sim S - L' = (L - S) - L'$.

The last evaluation criterion is the number of artificial itemsets, which is denoted by γ . It represents the set of large itemsets appearing in the sanitized database but not belonging to the large itemset in the original database. This side effect is shown in Figure 6, in which the γ part is the difference of L' and L .

Definition 3. The artificial itemsets in PPDM are defined as γ , in which $\gamma = L' - L$.

Hiding sensitive itemsets or information is not only one purpose of PPDM but also minimizing the above side effects for data sanitization.

4. Proposed Compact Prelarge Genetic Algorithm to Delete Transactions (cpGA2DT)

In this paper, a cpGA2DT approach is thus proposed to find the appropriate transactions to be deleted for hiding sensitive itemsets. The sensitive itemsets to be hidden can be defined below.

Definition 4. Suppose that a set of HS consist of the amounts of sensitive itemsets to be hidden; thus $HS = \{s_1, s_2, \dots, s_k\}$.

In the proposed cpGA2DT for hiding the sensitive itemsets through transaction deletion, the support count of a sensitive itemset must be below the minimum support threshold, in which each transaction to be deleted must contain any of the sensitive itemsets in HS.

Definition 5. Suppose an original database $D = \{T_1, T_2, \dots, T_n\}$; a database D' is thus projected from D , in which each T_j in D' must consist of any of the sensitive itemsets in HS.

In GAs, a chromosome corresponds to a possible solution. Suppose that m is appropriate transactions from D' to be deleted for hiding the sensitive itemsets. A chromosome with m genes is thus designed. Each gene represents a possible transaction to be deleted as a positive integer of transaction ID (TID) value or *null*.

Definition 6. Suppose a projected database $D' = \{T_1, T_2, \dots, T_n\}$, in which each T_j represents a transaction ID. Suppose that m is appropriate transactions to be deleted; a chromosome c_i is a set of m gens. Each m in c_i is represented as a transaction T_j or *null*.

In GAs, a flexible fitness function with three adjustable weights to evaluate the goodness of chromosomes is thus designed.

Definition 7. A fitness function to evaluate the goodness of a chromosome c_i is defined as

$$\text{fitness}(c_i) = w_1 \times \alpha + w_2 \times \beta + w_3 \times \gamma, \quad (2)$$

where w_1 , w_2 , and w_3 are the weighting parameters. The α , β , and γ are the hiding failure, missing cost, and artificial cost. Details of the notations and the proposed cpGA2DT algorithm are described in Algorithm 1.

4.1. Proposed cpGA2DT Algorithm. The designed cpGA2DT algorithm is described in Algorithm 1.

TABLE 1: Original database.

TID	Item
1	a, b, c
2	b, c, e
3	a, b, c, e
4	a, b, e
5	a, b, e
6	a, c, d
7	b, c, d, e
8	b, c, e
9	c
10	a, b

For the proposed cpGA2DT, it adopts both the compact GA and prelarge concepts to reduce not only the computations of database rescan but also the population size at each evaluation. Prelarge itemsets (PL) act like buffers and are used to reduce the movement of itemsets directly from large to small and vice versa when transactions are deleted (in steps (1) and (2)). In competition process, only two individuals are used for competition (in step (8)). This approach can reduce the population size to speed up the evaluation process. When the termination condition is not satisfied, two chromosomes are then generated again, respectively, to increase the probability of selected transactions in the winner chromosome but decrease the probability of selected transactions in the loser chromosome.

5. An Illustrated Example

In this section, an example is given to demonstrate the proposed cpGA2DT for privacy-preserving data mining. Assume that an original database contains 10 transactions shown in Table 1.

Also assume that the set of sensitive itemsets is defined as $\{be, bce\}$ to be hidden. The minimum support threshold is set at 40%. The proposed algorithm is then processed as follows. The transactions with any of the sensitive itemsets in Table 1 are then projected. In this example, transactions 2, 3, 4, 5, 7, and 8 are then projected to form another projected database. The initial probabilities of those five transactions are initially set at 0.5. The lower support threshold for deriving the prelarge itemsets in this example is calculated as $S_l = S_u \times (1 - m/|D|) = 0.4 \times (1 - 4/10) = 0.24$. The database is scanned to find the large and prelarge itemsets. The results are, respectively, shown in Tables 2 and 3.

Two chromosomes (individuals) are then generated randomly according to the probability vector with 4 genes. The results are then shown in Table 4.

The chromosomes in Table 4 are then competed by the designed fitness function. In this example, the weights for three factors are, respectively, set as 0.5, 0.3, and 0.2. Take C_A as an example to illustrate the evolutionary process. The number of hiding failures for C_A is 0 since all sensitive itemsets (be, bce) are completely hidden; the number of missing itemsets of C_A is 3 (itemsets e, bc , and ce are missing),

Input: D, HS, m, S_u, S_l .
Output: A sanitized database D^* .
Termination condition: The $fitness := 0$ or the number of generation $:= N$.

(1) set $S_l = S_u \times \left(1 - \frac{m}{|D|}\right)$.
(2) scan D to get L and PL respectively by S_u and S_l .
(3) **for** ($j \leftarrow 1, n; a \leftarrow 1, k$) **do**
 if ($si_a \subseteq T_j$) **then**
 project T_j from D to form D' .
 end if
end for
// initialize the probability vector for each transaction T_j in D' .
(4) **for** ($i \leftarrow 1, |D'|$) **do**
 $p[i] := 0.5$.
end for
// generate two individuals with m genes from D' by $p[i]$.
(5) $c_A[a] := \{T_j \text{ or } 0, T_j \subseteq D', 1 \leq a \leq m\}$.
(6) $c_B[a] := \{T_j \text{ or } 0, T_j \subseteq D', 1 \leq a \leq m\}$.
// compete c_A and c_B .
(7) winner, loser := compete(c_A, c_B) by fitness.
// update the probability vector towards to the better chromosome.
(8) **for** ($i \leftarrow 1, |D'|$) **do**
 $p[i] := p[i] + 1/|D'|$ for the T_j of winner.
 $p[i] := p[i] - 1/|D'|$ for the T_j of loser.
end for
(9) **if** terminated condition is not satisfied **then**
 perform Steps 5 to 8.
else
 terminate.
end if

ALGORITHM 1: cpGA2DT algorithm.

TABLE 2: Large itemsets.

1-itemset	Count	2-itemset	Count	3-itemset	Count
a	6	ab	5	bce	4
b	8	bc	5		
c	7	be	6		
e	6	ce	4		

TABLE 3: Prelarge itemsets.

Prelarge 1-itemset	Count	Prelarge 2-itemset	Count	Prelarge 3-itemset	Count
d	2	ac	3	abc	2
		ae	3	abe	3
		cd	2		

TABLE 4: Two individuals.

C_A	2	7	8	5
C_B	3	2	4	7

and the number of artificial itemsets of C_A is 1 (itemset ac arose). The fitness value of C_A is calculated as $fitness(C_A) = 0.5 \times 0 + 0.3 \times 3 + 0.2 \times 1 (=1.1)$. The C_B is processed in

TABLE 5: Probability vector.

TID	2	3	4	5	7	8
Probability	0.5	0.667	0.667	0.33	0.33	0.5

the same way, and $fitness(C_B) = 0.5 \times 0 + 0.3 \times 3 + 0.2 \times 0 (=0.9)$. In the competition process, the C_B is better than C_A ; the probabilities of transactions 2, 3, 4, and 7 are then, respectively, increased and updated in the probability vector by $0.5 + 1/6 (=0.667)$; the probabilities of transactions 2, 5, 7, and 8 are then, respectively, decreased and updated in the probability vector by $0.5 - 1/6 (=0.33)$. After that, the probability vector is updated and shown in Table 5.

Steps (5) to (8) are then, recursively, processed until the termination condition is satisfied. In this example, three criteria are used as the termination conditions. The criteria are as follows. The fitness function value of the best chromosome is 0; or a predefined number of generations is achieved; or the probability vector is converged. After the evolutionary process, the top-4 transactions with high probabilities in the probability vector are then selected as the transactions to be deleted in the sanitization process.

TABLE 6: Three databases.

Database	Transactions	Items	Avg. of transactions
Mushroom	8,124	119	23
BMSWebview-1	59,602	497	2.5
BMSWebview-2	77,512	3,340	5

6. Experimental Results

Experiments are conducted to show the performance of the proposed cpGA2DT, which was performed on a Pentium IV processor at 2 GHz and 512 M of RAM running on the Mandriva platform. A greedy approach and a simple GA-based algorithm [46] are also designed as a benchmark to be compared with the proposed algorithm. For the greedy approach, it scans the transactions from top to down to directly delete the transactions with sensitive itemsets. The termination of the greedy algorithm is the number of the deleted transactions, which is predefined by users. A simple GA-based approach uses simple GAs to hide the sensitive information. Three real databases mushroom [47], BMS-WebView1 [48], and BMS-WebView2 [48] are used to evaluate the performance of the proposed cpGA2DT in terms of the execution time and the number of three side effects. The weights for three side effects α , β , and γ are set at 0.5, 0.25, and 0.25, which can be adjusted by users. Details of the three databases used in the experiments are shown in Table 6.

6.1. Execution Time. Execution times obtained the proposed cpGA2DT; greedy and simple GA-based algorithms are then compared at various sensitivity percentages of the sensitive itemsets for three databases. Results are shown in Figures 7, 8, and 9. The S_u is initially set at 1.5%. According to predefined number of transactions to be deleted (the size of chromosome) in the original database, the S_i is easily retrieved for deriving the prelarge itemsets, thus speeding up the execution time without computations of database rescan.

From Figures 7 to 9, it is obvious to see that the straightforward greedy approach has the best performance in execution time since it does not consider any side effects but directly delete the transactions for the purpose of hiding sensitive itemsets. The proposed cpGA2DT can greatly reduce the execution time compared to the simple GA-based algorithm since for cpGA2DT it is unnecessary to rescan the original database for evaluating fitness at each iteration. Experiments are then conducted to show the execution times for three algorithms at various minimum support thresholds. The results are then shown in Figures 10, 11, and 12.

Form Figures 10 and 12, it is obvious to see that the greedy approach has the best performance of execution time at various minimum support thresholds. The proposed cpGA2DT has the best performance in BMSWebview-1 database. The simple GA-based algorithm still has the worst performance in execution time since it requires to rescan the original database to evaluate the goodness of fitness at each iteration. The side effects of hiding failure, missing cost, and the artificial cost are also evaluated to show the performance of the proposed cpGA2DT. The descriptions are given as follows.

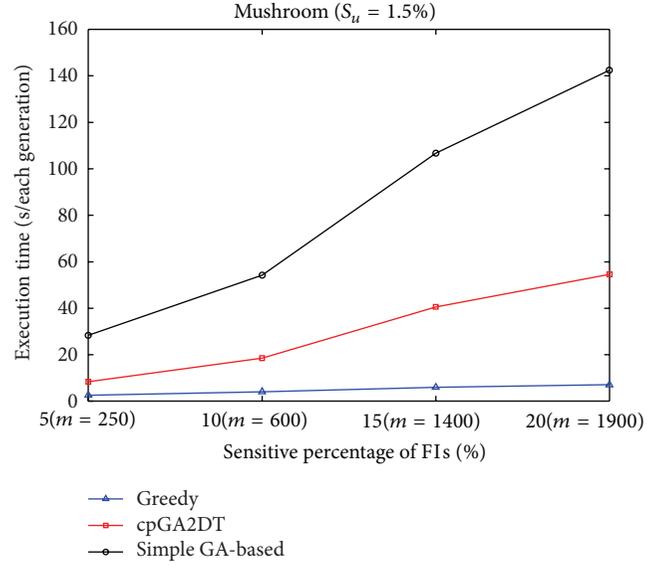


FIGURE 7: Comparisons of execution time at various sensitivity percentages for mushroom database.

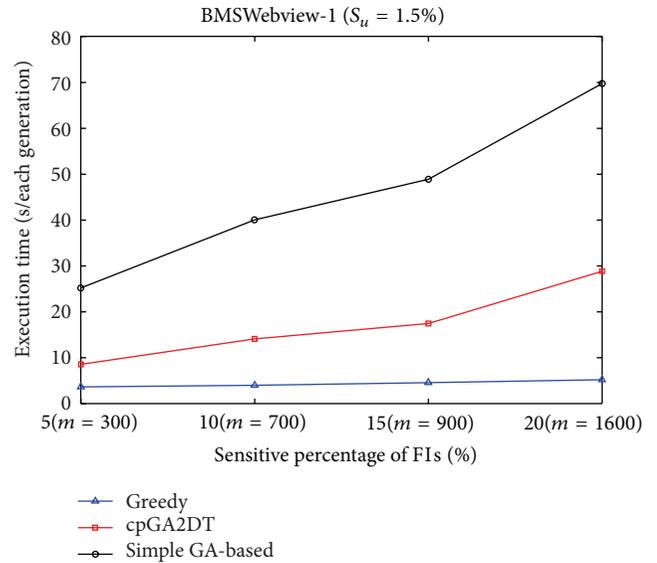


FIGURE 8: Comparisons of execution time at various sensitivity percentages for BMSWebview-1 database.

6.2. Hiding Failure (HF). The hiding failure is one of the side effects to evaluate whether the sensitive information has been successfully hidden before and after sanitization process, which can be calculated as

$$HF = \frac{|HS(D^*)|}{|HS(D)|}, \tag{3}$$

where $|HS(D^*)|$ is the number of sensitive itemsets after sanitization process and the $|HS(D)|$ is the number of sensitive itemsets before sanitization process. The hiding failure obtained three algorithms at various sensitivity percentages

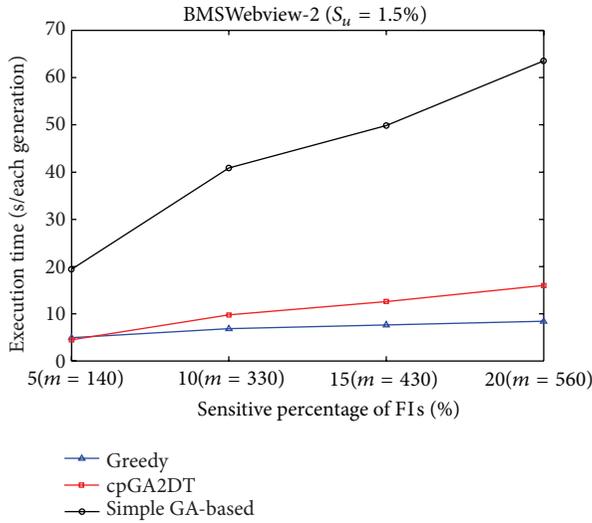


FIGURE 9: Comparisons of execution time at various sensitivity percentages for BMSWebview-2 database.

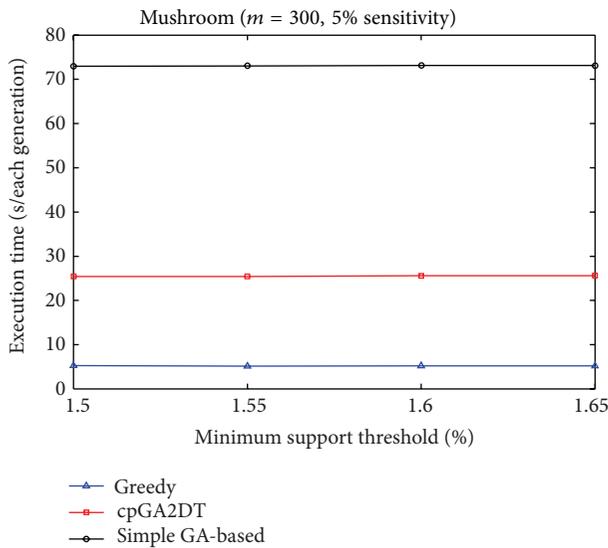


FIGURE 10: Comparisons of execution time at various minimum support thresholds for mushroom database.

of the sensitive itemsets for three databases with $S_u (= 1.5\%)$. The results are then shown in Figures 13, 14, and 15.

From Figures 13 to 15, it is obvious to see that the greedy approach has the worst performance for hiding the sensitive itemsets in three databases. The proposed cpGA2DT generally has the best performance for hiding the sensitive itemsets in three databases except when the sensitive percentage is set at 10% of frequent itemsets in BMSWebview-2 database. Experiments are then conducted to show that the performance of hiding failure obtained three algorithms at various minimum support thresholds. The results are then shown in Figures 16, 17, and 18.

From Figures 16 to 18, it is easily found that the proposed cpGA2DT generally has the best performance of hiding

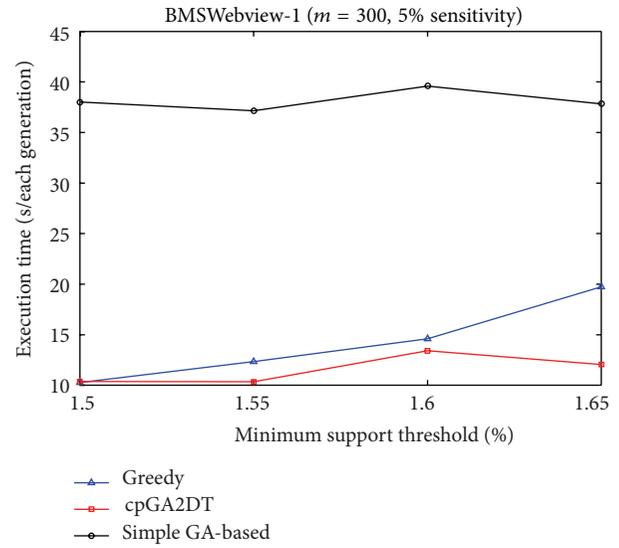


FIGURE 11: Comparisons of execution time at various minimum support thresholds for BMSWebview-1 database.

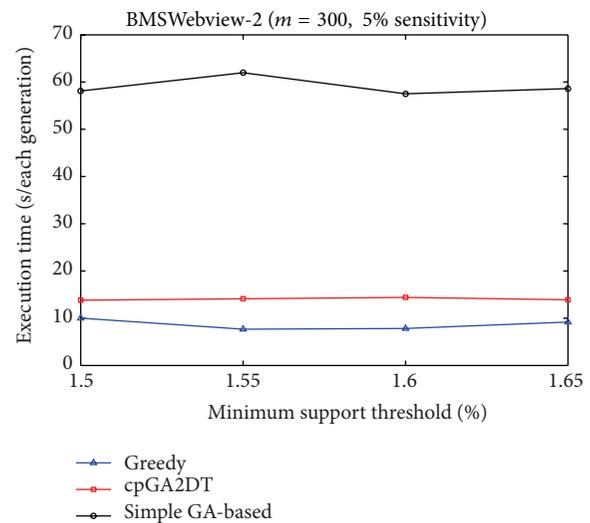


FIGURE 12: Comparisons of execution time at various minimum support thresholds for BMSWebview-2 database.

failure at various minimum support thresholds for three databases and is better than the greedy and the simple GA-based algorithms in most cases at various minimum support thresholds for three databases.

6.3. *Missing Cost (MC)*. The side effects of missing cost are also evaluated to show the performance of the proposed cpGA2DT, which is calculated as

$$MC = \frac{|FIs(D)| - |FIs(D^*)|}{|FIs(D)|}, \quad (4)$$

where $|FIs(D)|$ is the number of frequent itemsets before data sanitization and $|FIs(D^*)|$ is the number of frequent itemsets after data sanitization. Note that even sensitive

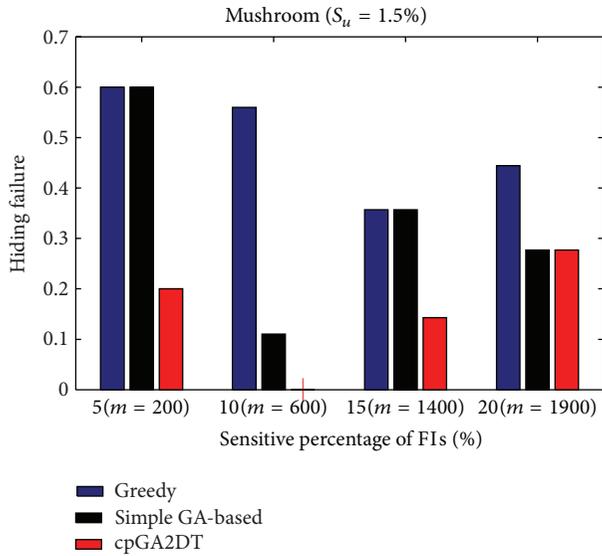


FIGURE 13: Comparisons of hiding failure at various sensitivity percentages of the frequent itemsets for mushroom database.

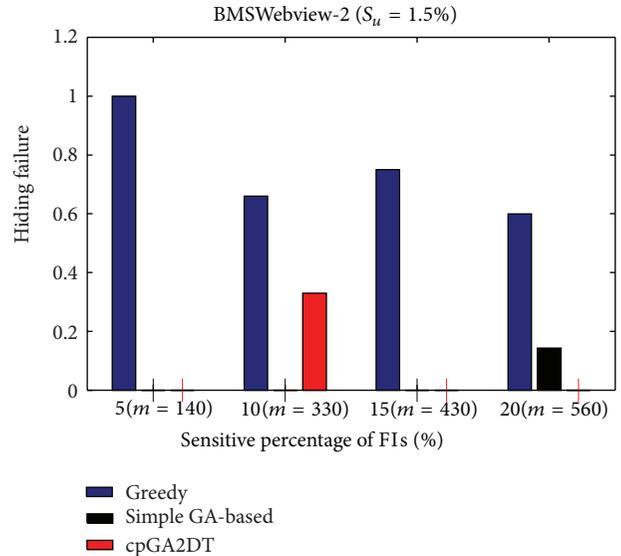


FIGURE 15: Comparisons of hiding failure at various sensitivity percentages of the frequent itemsets for BMSWebview-2 database.

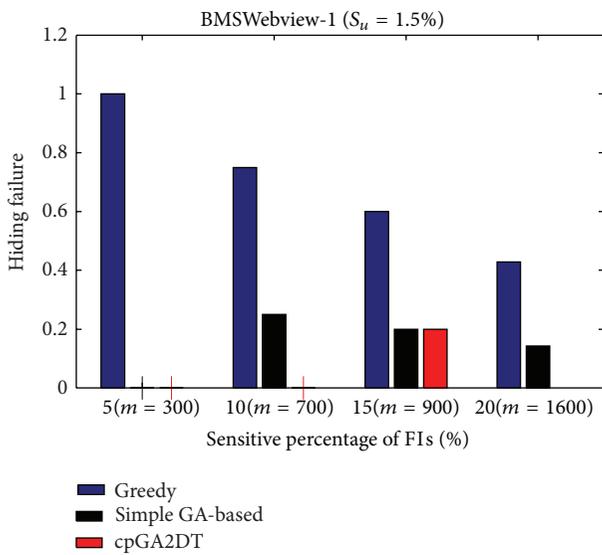


FIGURE 14: Comparisons of hiding failure at various sensitivity percentages of the frequent itemsets for BMSWebview-1 database.

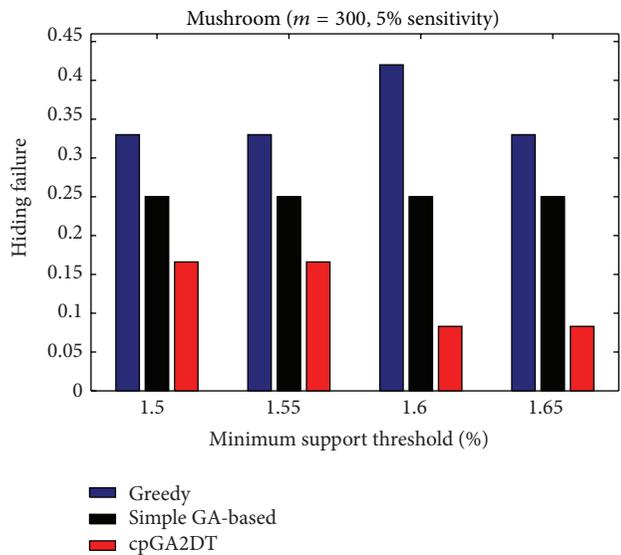


FIGURE 16: Comparisons of hiding failure at various minimum support thresholds for mushroom database.

itemsets are the frequent itemsets but not considered here to calculate the missing cost. The missing cost obtained three algorithms which are then compared at various sensitivity percentages of the sensitive itemsets for three databases with $S_u (= 1.5\%)$. The missing cost that obtained three algorithms has, however, zero for the mushroom database since the mushroom database is too small for data sanitization. All sensitive itemsets can thus be successfully hidden without any missing cost in mushroom database. The results for the other two databases are then shown in Figures 19 to 20.

In the experiments of the proposed cpGA2DT, the weight of hiding failure is set at 0.5, which is higher than the missing cost and artificial cost. From Figure 19, the proposed

cpGA2DT has generated some missing costs at 15% and 20% sensitive percentages of frequent itemsets. The proposed cpGA2DT has not any missing cost in BMSWebview-2 database. Experiments are then conducted to show that the performance of missing cost obtained three algorithms at various minimum support thresholds for three databases. Again, the missing cost is zero for the obtained three algorithms for mushroom database. The results for the other two databases are then shown in Figures 21 to 22.

From Figure 21, the proposed cpGA2DT algorithm has no missing cost for the BMSWebview-1 database. The greedy approach slightly outperforms better than the proposed cpGA2DT in the BMSWebview-2 but the proposed

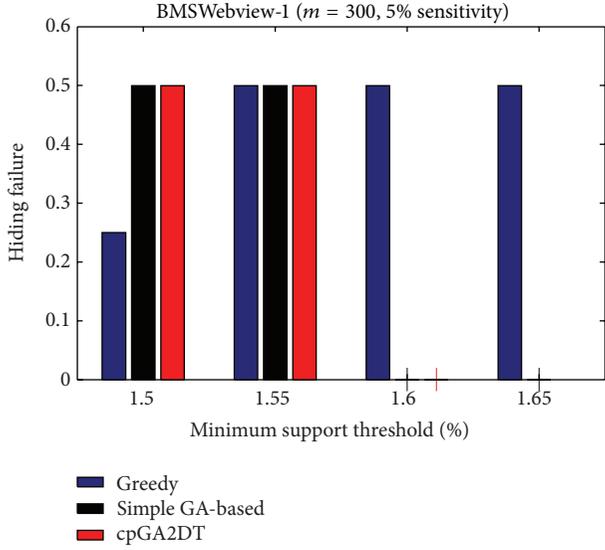


FIGURE 17: Comparisons of hiding failure at various minimum support thresholds for BMSWebview-1 database.

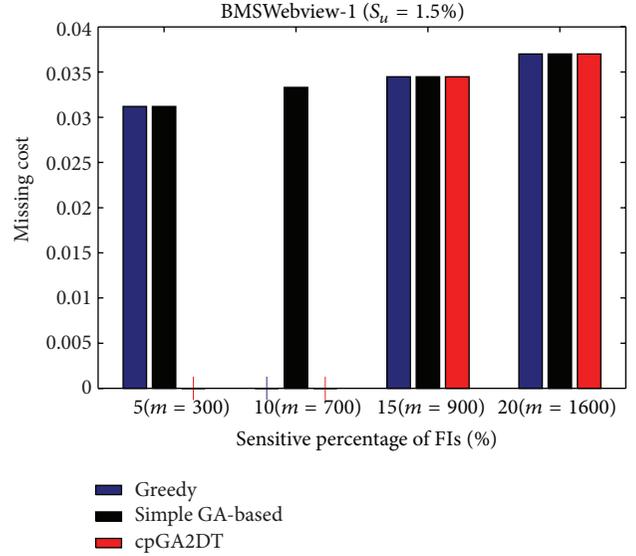


FIGURE 19: Comparisons of missing cost at various sensitivity percentages of frequent itemsets for BMSWebview-1 database.

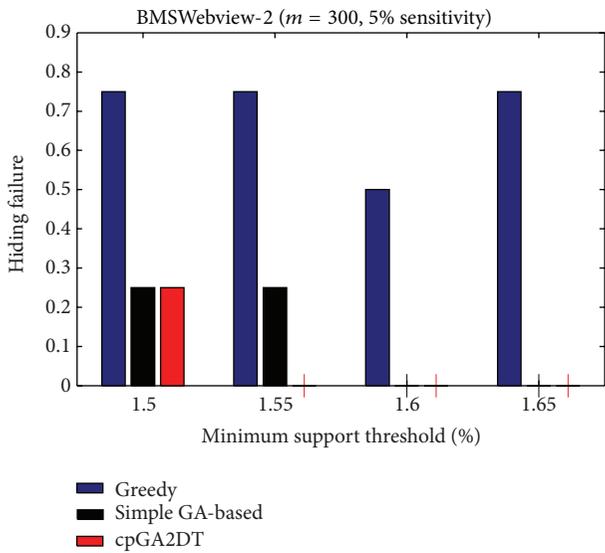


FIGURE 18: Comparisons of hiding failure at various minimum support thresholds for BMSWebview-2 database.

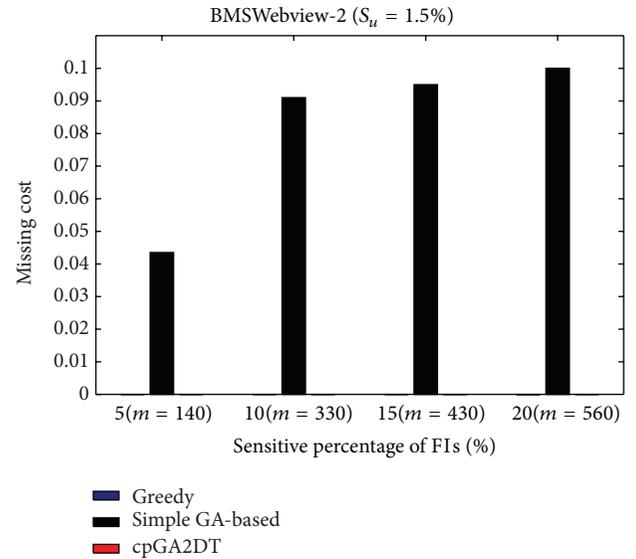


FIGURE 20: Comparisons of missing cost at various sensitivity percentages of frequent itemsets for BMSWebview-2 database.

cpGA2DT still achieves good performance at the 1.5% and 1.6% minimum support thresholds with zero missing cost. In the experimental process, we have also found that the greedy approach is executed to delete transactions from top transactions to down ones, and the deleted transactions of the greedy approach in BMSWebview-2 have fewer numbers of items within it. Thus, the missing cost of the greedy approach is a little bit better than the proposed algorithm at 1.65% minimum support threshold.

6.4. Artificial Cost (AC). The side effects of artificial cost are also evaluated to show the performance of the proposed

cpGA2DT, which is calculated as

$$AC = \frac{|FIs(D^*)| - |FIs(D^*) \cap FIs(D)|}{|FIs(D^*)|} \quad (5)$$

In three databases that obtained three algorithms in various sensitivity percentages of the frequent itemsets and various minimum support thresholds, there are not any side effects of artificial cost. For the greedy approach in the experiments, the deleted transactions have short length with lower support items; thus the artificial cost is not shown. For the proposed cpGA2DT, instead of the above reason of the greedy approach, the artificial cost is also considered as a factor in the

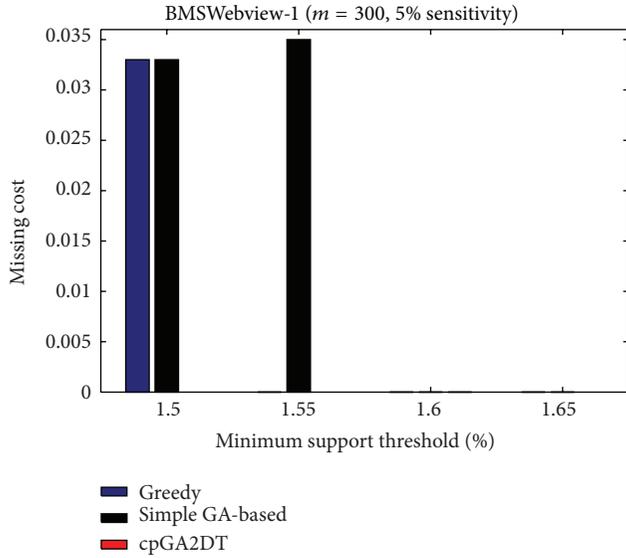


FIGURE 21: Comparisons of missing cost at various minimum support thresholds for BMSWebview-1 database.

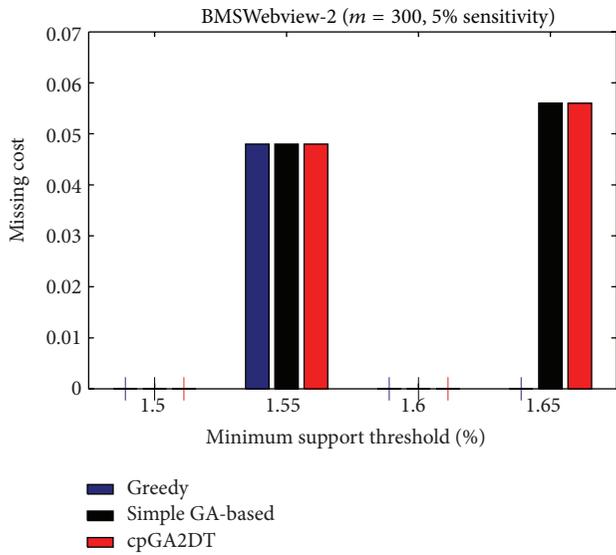


FIGURE 22: Comparisons of missing cost at various minimum support thresholds for BMSWebview-2 database.

evaluation process, thus avoiding the side effects of artificial cost.

7. Conclusion

In this paper, a compact GA-based cpGA2DT algorithm is thus proposed to hide the sensitive itemsets through transaction deletion. A flexible fitness function with three adjustable weights is also designed to consider the general side effects of hiding failure, missing cost, and the artificial cost to determine the goodness of the chromosomes. The prelarge concept is adopted in the proposed algorithm to reduce the computations of database rescan. The size of the

populations is also reduced by the compact GA approach, thus reducing the memory lack problems of traditional GAs. Experiments are conducted to show that the proposed cpGA2DT algorithm outperforms better than the greedy and simple GA-based algorithms considering all criteria of side effects but the execution time.

Notations

- D : Original database to be sanitized
- $|D|$: Number of transactions in D
- D' : Projected database from D in which each transaction in D' contains any sensitive itemsets s_i in HS
- D^* : Sanitized database after the designed algorithm
- HS: A set of sensitive itemsets to be hidden, $HS = \{s_1, s_2, \dots, s_k\}$
- m : Number of transactions to be deleted for hiding sensitive itemsets
- S_u : Upper support threshold
- S_l : Lower support threshold, $S_u > S_l$
- L : A set of large itemsets in which the count of each itemset is larger than or equal to $|D| \times S_u$
- PL: A set of prelarge itemsets in which the count of each itemset lies between $|D| \times S_u$ and $|D| \times S_l$
- p : Probability vector of transactions in D'
- c_A, c_B : Two competition chromosomes.

Conflict of Interests

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

Acknowledgments

This research was partially supported by the Shenzhen Peacock Project, China, under Grant KQC201109020055A, by the Natural Scientific Research Innovation Foundation, Harbin Institute of Technology, under Grant HIT.NSRIF.2014100, and by the Shenzhen Strategic Emerging Industries Program under Grant ZDSY20120613125016389.

References

- [1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the International Conference on Very Large Data Bases*, pp. 487–499, 1994.
- [2] T. F. Gharib, H. Nassar, M. Taha, and A. Abraham, "An efficient algorithm for incremental mining of temporal association rules," *Data and Knowledge Engineering*, vol. 69, no. 8, pp. 800–815, 2010.
- [3] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: a frequent-pattern tree approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53–87, 2004.

- [4] T. Hong, C. Lin, and Y. Wu, "Incrementally fast updated frequent pattern trees," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2424–2435, 2008.
- [5] C.-W. Lin, T.-P. Hong, and W.-H. Lu, "The Pre-FUFP algorithm for incremental mining," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9498–9505, 2009.
- [6] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the IEEE 11th International Conference on Data Engineering*, pp. 3–14, San Jose, Calif, USA, March 1995.
- [7] C. Kim, J. Lim, R. T. Ng, and K. Shim, "SQUIRE: sequential pattern mining with quantities," *Journal of Systems and Software*, vol. 80, no. 10, pp. 1726–1745, 2007.
- [8] J. Pei, J. Han, B. Mortazavi-Asl et al., "Mining sequential patterns by pattern-growth: The prefixspan approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1424–1440, 2004.
- [9] R. Srikant and R. Agrawal, "Mining sequential patterns: generalizations and performance improvements," in *Proceedings of the International Conference on Extending Database Technology: Advances in Database Technology*, pp. 3–17, 1996.
- [10] S. B. Kotsiantis, "Supervised machine learning: a review of classification techniques," in *Proceedings of the Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pp. 3–24, 2007.
- [11] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [12] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
- [13] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, pp. 25–71, 2006.
- [14] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Transactions on Computers*, vol. 22, no. 11, pp. 1025–1034, 1973.
- [15] R. Chan, Q. Yang, and Y.-D. Shen, "Mining high utility itemsets," in *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM '03)*, pp. 19–26, November 2003.
- [16] C. W. Lin, G. C. Lan, and T. P. Hong, "An incremental mining algorithm for high utility itemsets," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7173–7180, 2012.
- [17] Y. Liu, W. K. Liao, and A. Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets," in *Advances in Knowledge Discovery and Data Mining*, pp. 689–695, 2005.
- [18] U. Yuna, H. Ryanga, and K. H. Ryub, "High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates," *Expert Systems with Applications*, vol. 41, pp. 3861–3878, 2014.
- [19] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *SIGMOD Record*, vol. 29, no. 2, pp. 439–450, 2000.
- [20] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Advances in Cryptology—CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20–24, 2000*, vol. 1880 of *Lecture Notes in Computer Science*, pp. 36–54, 2000.
- [21] S. M. Oliveira, O. Zaïane, and Y. Saygin, "Secure association rule sharing," *Advances in Knowledge Discovery and Data Mining*, vol. 3056, pp. 74–85, 2004.
- [22] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, "State-of-the-art in privacy preserving data mining," *SIGMOD Record*, vol. 33, no. 1, pp. 50–57, 2004.
- [23] A. Amiri, "Dare to share: protecting sensitive knowledge with data sanitization," *Decision Support Systems*, vol. 43, no. 1, pp. 181–191, 2007.
- [24] M. Atallah, A. Elmagarmid, M. Ibrahim, E. Bertino, and V. Verykios, "Disclosure limitation of sensitive rules," in *Proceedings of the Workshop on Knowledge and Data Engineering Exchange (KDEX '99)*, pp. 45–52, Chicago, Ill, USA, November 1999.
- [25] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, pp. 217–228, July 2002.
- [26] Y. Wu, C. Chiang, and A. L. P. Chen, "Hiding sensitive association rules with limited side effects," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 29–42, 2007.
- [27] C. C. Aggarwal, J. Pei, and B. Zhang, "On privacy preservation against adversarial data mining," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 510–516, August 2006.
- [28] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1992.
- [29] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 287–297, 1999.
- [30] T. P. Hong and C. Y. Wang, "Maintenance of association rules using pre-large itemsets," in *Intelligent Databases: Technologies and Applications*, pp. 44–60, 2007.
- [31] T. P. Hong, C. Y. Wang, and Y. H. Tao, "A new incremental data mining algorithm using pre-large itemsets," *Intelligent Data Analysis*, vol. 5, pp. 111–129, 2001.
- [32] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman, 1989.
- [33] X. Wang, Q. He, D. Chen, and D. Yeung, "A genetic algorithm for solving the inverse problem of support vector machines," *Neurocomputing*, vol. 68, no. 1–4, pp. 225–238, 2005.
- [34] M. Chen, J. Han, and P. S. Yu, "Data mining: an overview from a database perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 866–883, 1996.
- [35] M. Mohamed and M. Darwieesh, "Efficient mining frequent itemsets algorithms," *International Journal of Machine Learning and Cybernetics*, 2013.
- [36] B. Nath, D. K. Bhattacharyya, and A. Ghosh, "Incremental association rule mining: a survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 3, no. 3, pp. 157–169, 2013.
- [37] B. Vo, T. Le, F. Coenen, and T.-P. Hong, "Mining frequent itemsets using the n-list and subsume concepts," *International Journal of Machine Learning and Cybernetics*, 2014.
- [38] E. Bertino, D. Lin, and W. Jiang, "A survey of quantification of privacy preserving data mining algorithms," in *Privacy-Preserving Data Mining*, C. Aggarwal and P. Yu, Eds., vol. 34, pp. 183–205, Springer, New York, NY, USA, 2008.
- [39] S. R. M. Oliveira and O. R. Zaïane, "Privacy preserving frequent itemset mining," in *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining*, pp. 43–54, 2002.
- [40] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, "Hiding association rules by using confidence and support," in *Proceedings of the International Workshop on Information Hiding*, pp. 369–383, 2001.
- [41] T. P. Hong, C. W. Lin, K. T. Yang, and S. L. Wang, "Using TF-IDF to hide sensitive itemsets," *Applied Intelligence*, vol. 38, no. 4, pp. 502–510, 2013.

- [42] S. Han and W. Ng, "Privacy-preserving genetic algorithms for rule discovery," in *Data Warehousing and Knowledge Discovery*, I. Song, J. Eder, and T. Nguyen, Eds., vol. 4654, pp. 407–417, Springer, Berlin, Germany, 2007.
- [43] M. N. Dehkordi, K. Badie, and A. K. Zadeh, "A novel method for privacy preserving in association rule mining based on genetic algorithms," *Journal of Software*, vol. 4, no. 6, pp. 555–562, 2009.
- [44] D. W. Cheung, J. Han, V. T. Ng, and C. Y. Wong, "Maintenance of discovered association rules in large databases: an incremental updating technique," in *Proceedings of the IEEE 12th International Conference on Data Engineering*, pp. 106–114, March 1996.
- [45] D. W. L. Cheung, S. D. Lee, and B. Kao, "A general incremental technique for maintaining discovered association rules," in *Proceedings of the International Conference on Database Systems for Advanced Applications*, pp. 185–194, 1997.
- [46] T. Hong, I. Yang, C. Lin, and S. Wang, "Evolutionary privacy-preserving data mining," in *Proceedings of the World Automation Congress (WAC '10)*, pp. 1–7, September 2010.
- [47] Frequent itemset mining dataset repository, 2012, <http://fimi.ua.ac.be/data/>.
- [48] Z. Zheng, R. Kohavi, and L. Mason, "Real world performance of association rule algorithms," in *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)*, pp. 401–406, San Francisco, Calif, USA, August 2001.

Research Article

Color Image Segmentation Based on Different Color Space Models Using Automatic GrabCut

Dina Khattab,¹ Hala Mousher Ebied,¹ Ashraf Saad Hussein,² and Mohamed Fahmy Tolba¹

¹ Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt

² Faculty of Computer Studies, Arab Open University-Headquarters, 13033 Al-Safat, Kuwait

Correspondence should be addressed to Dina Khattab; dina.reda.khattab@gmail.com

Received 1 June 2014; Revised 16 August 2014; Accepted 16 August 2014; Published 31 August 2014

Academic Editor: Shifei Ding

Copyright © 2014 Dina Khattab et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a comparative study using different color spaces to evaluate the performance of color image segmentation using the automatic GrabCut technique. GrabCut is considered as one of the semiautomatic image segmentation techniques, since it requires user interaction for the initialization of the segmentation process. The automation of the GrabCut technique is proposed as a modification of the original semiautomatic one in order to eliminate the user interaction. The automatic GrabCut utilizes the unsupervised Orchard and Bouman clustering technique for the initialization phase. Comparisons with the original GrabCut show the efficiency of the proposed automatic technique in terms of segmentation, quality, and accuracy. As no explicit color space is recommended for every segmentation problem, automatic GrabCut is applied with *RGB*, *HSV*, *CMY*, *XYZ*, and *YUV* color spaces. The comparative study and experimental results using different color images show that *RGB* color space is the best color space representation for the set of the images used.

1. Introduction

The process of partitioning a digital image into multiple segments is defined as image segmentation. Segmentation aims to divide an image into regions that can be more representative and easier to analyze. Such regions may correspond to individual surfaces, objects, or natural parts of objects. Typically image segmentation is the process used to locate objects and boundaries (e.g., lines or curves) in images [1]. Furthermore, it can be defined as the process of labeling every pixel in an image, where all pixels having the same label share certain visual characteristics [2]. Usually segmentation uses local information in the digital image to compute the best segmentation, such as color information used to create histograms or information indicating edges, boundaries, or texture information [3].

Color image segmentation that is based on the color feature of image pixels assumes that homogeneous colors in the image correspond to separate clusters and hence meaningful objects in the image. In other words, each cluster defines a class of pixels that share similar color properties.

As the segmentation results depend on the used color space, there is no single color space that can provide acceptable results for all kinds of images. For this reason, many authors tried to determine the color space that will suit their specific color image segmentation problem [4]. In this work, a segmentation of color images is tested with different classical color spaces, *RGB*, *CMY*, *XYZ*, *YUV*, and *HSV*, to select the best color space for the considered kind of images.

The segmentation process is based on the GrabCut segmentation technique [5], which is considered as one of the powerful state-of-the-art techniques for the problem of color image segmentation. The iterative energy minimization scheme of the GrabCut is based on the powerful optimization of the Graph Cut technique [6] which allows for the generation of the global optimal segmentation. In addition, Graph Cut can be easily well extended to the problem of N-D images. Furthermore, the cost energy function of the Graph Cut minimization process allows it to be defined in terms of different image features such as color, region, boundary, or any mixture of image features. This flexibility provides wide potential for the use of GrabCut in different applications.

On the other hand, GrabCut is considered as a bilabel segmentation technique, where images can be segmented into two background and foreground regions only. Initial user intervention is required in order to specify an object of interest to be segmented out of the image, considering all the remaining image pixels as one background region. This classifies the GrabCut as a semiautomatic segmentation technique and turns the quality of the initialization and hence the segmentation performance, sensitive to the user selection. In other words, poor GrabCut initialization may lead to bad final segmentation accuracy which might require extra user interactions with the segmentation results for fine tuning [5].

In this work, a modified GrabCut is proposed as an automatic segmentation technique, which can segment the image into its natural objects without any need for the initial user intervention. Automation of GrabCut is applied using Orchard and Bouman clustering [7] as an unsupervised clustering technique. The selection of the Orchard and Bouman clustering is based on the empirical comparison results carried out in the work of [8]. The paper exploits the use of some evaluation criteria to evaluate the discriminating power of the automatic GrabCut with the different color spaces. The remainder of the paper is organized as follows. Section 2 provides a basic background on segmentation based-color space models, image segmentation using GrabCut, and unsupervised clustering techniques. Section 3 explains the different color space models. Section 4 illustrates the Orchard and Bouman clustering. The original GrabCut technique and details of its modification are explained in Section 5. Experimental results are presented in Section 6, while the conclusion and future work are presented in Section 7.

2. Related Work

As no common opinion has emerged about which is the best choice for color space based image segmentation, some research work tried to identify the best color space for a specific task. Several works [9, 10] show that different color spaces are useful for the problem of color image segmentation. Jurio et al. [11] have carried out a comparative study between different color spaces in cluster based image segmentation using two similar clustering algorithms. Their study involved the test of four color spaces, *RGB*, *HSV*, *CMY*, and *YUV*, in order to identify the best color representation. They obtained their best results in most cases using *CMY* color space, while *HSV* also provided good results. Busin et al. [4] proposed a method to automatically select a specific color space among classical color spaces. This selection was done according to an evaluation criterion based on a spectral color analysis. This criterion evaluates the quality of the segmentation in each space and selects the best one, which preserves its own specific properties. A study of the ten most common color spaces for skin color detection was presented in [12]. They concluded that *HSV* is the best color space to detect skin in an image. Another study that was applied for the classification of pizza toppings [13] proved that the polynomial SVM classifier combined with *HSV* color space is the best approach among five different color spaces. Based on a comparative study between the *RGB* and *HSV* models,

Ruiz-Ruiz et al. [14] declared that the best accuracy was achieved with *HSV* representation in order to achieve real time processing in real farm fields for crop segmentation.

GrabCut is considered one of the powerful techniques used for color image segmentation. It has been applied to different segmentation problems such as human body segmentation [15–17], video segmentation [18], semantic segmentation [19], and volume segmentation [20]. In [17], an automatic extraction of the human body from color images was developed by Hu. The iterated GrabCut technique was used to dynamically update a trimap contour, which was initialized from the results of a scanning detector used for detecting faces from images. The research has some drawbacks, as the process goes through many steps and iterations, in addition to being constrained to human poses with frontal side faces. A fully automatic Spatio-Temporal GrabCut human segmentation methodology was proposed by Hernandez et al. [16]. They developed methodology that takes the benefits of the combination of tracking and segmentation. Instead of the initial user intervention to initialize the GrabCut algorithm, a set of seeds defined by face detection and a skin color model are used for initialization. Another approach to segment humans from cluttered images was proposed by Gulshan et al. in [15]. They utilized the local color model based GrabCut for automatic segmentation. This GrabCut local color model was used to refine the crude human segmentations they obtained. In video segmentation, Corrigan et al. [18] extended GrabCut for more robust video object segmentation. They extended the Gaussian mixture model (GMM) of the GrabCut algorithm, so that the color space was complemented with the derivative in time of the pixel's intensities in order to include temporal information in the segmentation optimization process. Göring et al. [19] integrated GrabCut into a semantic segmentation framework by labeling objects in a given image. Most recently, Ramírez et al. [20] proposed a fully parallelized scheme using GrabCut for 3D segmentation that has been adopted to run on GPU. The scheme aims at producing efficient segmentation results for the case of volume meshes, in addition to reducing the computational time.

Clustering [21], the unsupervised classification of patterns into groups, is one of the most important tasks in exploratory data analysis [22]. It has a long and rich history in a variety of scientific disciplines including anthropology, biology, medicine, psychology, statistics, mathematics, engineering, and computer science. Clustering in image segmentations [2, 23, 24] is defined as the process of identifying groups of similar image primitives. Unsupervised clustering techniques [25] are content based clustering, where content refers to shapes, textures, or any other information that can be inherited from the image itself.

In the cases of bilabel segmentation, good separation between foreground and background is required. This can be implemented through finding clusters with a low variance, since this makes the cluster easier to separate from the others. The selection of the Orchard and Bouman clustering technique [7] is guided by Ruzon and Tomasi [26] and Chung et al. [27] in order to get tight and well separated clusters. They have worked on solving the problem of image matting

that is required for image compositing. In their approach, Orchard and Bouman binary split algorithm has been used for partitioning the unknown region colors into several clusters, in order to generate a color distribution for the unknown region to be estimated. According to a comparative study in [8], the Orchard and Bouman clustering outperformed other unsupervised clustering techniques including self-organizing maps (SOFM) and fuzzy C-means (FCM) for the automation of the GrabCut in terms of improving the segmentation accuracy.

3. Color Space Models

The most widely used color space is the *RGB* color space, where a color point in the space is characterized by three color components of the corresponding pixel which are red (*R*), green (*G*), and blue (*B*). However since there exist a lot of color spaces, it is useful to classify them into fewer categories with respect to their definitions and properties. Vandenbroucke [28] proposed the classification of the color spaces into the following categories.

- (i) The primary spaces which are based on the theory that assumes it is possible to match any color by mixing an appropriate amount of the three primary colors: the primary spaces are the real *RGB*, the subtractive *CMY*, and the imaginary *XYZ* primary spaces. The conversion from *RGB* to *CMY* is

$$\begin{aligned}
 C' &= 1 - R & C &= \min(1, \max(0, C' - K')) \\
 M' &= 1 - G & M &= \min(1, \max(0, M' - K')) \\
 Y' &= 1 - B & Y &= \min(1, \max(0, Y' - K')) \\
 & & K' &= \min(C', M', Y')
 \end{aligned} \tag{1}$$

and the conversion from *RGB* to *XYZ* is

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{2}$$

- (i) The luminance-chrominance spaces, which are computed of one color component that represents the luminance and two color components that represent the chrominance: the *YUV* color space is an example of the luminance-chrominance spaces. The conversion from *RGB* to *YUV* is

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.2989 & 0.5866 & 0.1145 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{3}$$

- (ii) The perceptual spaces that try to quantify the subjective human color perception by means of three measures, intensity, hue, and saturation: the *HSV*

is an example of the perceptual color space. The conversion from *RGB* to *HSV* is

$$H = \begin{cases} 0, & \text{if Max} = \text{Min} \\ \left(60^\circ \times \frac{G - B}{\text{Max} - \text{Min}} + 360^\circ\right) & \\ \times \text{mod } 360^\circ, & \text{if Max} = R \\ 60^\circ \times \frac{B - R}{\text{Max} - \text{Min}} + 120^\circ, & \text{if Max} = G \\ 60^\circ \times \frac{R - G}{\text{Max} - \text{Min}} + 240^\circ, & \text{if Max} = B \end{cases} \tag{4}$$

$$S = \begin{cases} 0, & \text{if max} = 0 \\ \frac{\text{Max} - \text{Min}}{\text{Max}} & \text{otherwise} \end{cases}$$

$$V = \text{Max}.$$

4. Orchard and Bouman Clustering Technique

Orchard and Bouman [7] is a color quantization clustering technique that uses the eigenvector of the color covariance matrix to determine good cluster splits. The algorithm starts with all the pixels in a single cluster. The cluster is then split into two using a function of eigenvector of the covariance matrix as the split point. Then it uses the eigenvalues of the covariance matrices to choose which of the resulting clusters is candidate for the next splitting. This procedure is repeated until the desired number of clusters is achieved. It is an optimal solution for large clusters with Gaussian distributions.

For example, consider C_1 as a set of pixels, in order to divide it into K clusters:

- (1) calculate μ_1 , the mean of C_1 , and Σ_1 , the covariance matrix of C_1 ,
- (2) for $i = 2$ to K do the following:
 - (i) find the set C_n which has the largest eigenvalue and store the associated eigenvector e_n ,
 - (ii) split C_n into two sets, $C_i = \{x \in C_n : e_n^T z_n \leq e_n^T \mu_n\}$ and $C_n^* = C_n - C_i$,
 - (iii) compute μ_n^* , Σ_n^* , μ_i , and Σ_i .

This results in K pixel clusters.

5. Image Segmentation Using GrabCut

Image segmentation is simply the process of separating an image into foreground and background parts. Graph Cut technique [6] was considered as an effective way for the segmentation of monochrome images, which is based on the Min-Cut/Max-Flow algorithm [29]. GrabCut [5] is a powerful extension of the Graph Cut algorithm to segment color images iteratively and to simplify the user interaction needed for a given quality of the segmentation results. Section 5.1



FIGURE 1: Example of GrabCut segmentation. (a) GrabCut allows the user to drag a rectangle around the object of interest to be segmented. (b) The segmented object.

explains the original semiautomatic GrabCut algorithm as developed by Rother et al. in [5], while its modification for automatic segmentation is presented in Section 5.2.

5.1. Original Semiautomatic GrabCut. The GrabCut algorithm learns the color distributions of the foreground and background by giving each pixel a probability to belong to a cluster of other pixels. It can be explained as follows: given a color image I , let us consider the $z = (z_1, \dots, z_n, \dots, z_N)$ of N pixels, where $z_i = (C_{1i}, C_{2i}, C_{3i})$, $i \in [1, \dots, N]$, and C_j is the j th color component in the used color space. The segmentation is defined as an array $\alpha = (\alpha_1, \dots, \alpha_N)$, $\alpha_i \in \{0, 1\}$, assigning a label to each pixel of the image, indicating if it belongs to the background or the foreground. The GrabCut algorithm consists mainly of two basic steps: initialization and iterative minimization. The details of both steps are explained in the following subsections.

5.1.1. GrabCut Initialization. The novelty of the GrabCut technique is in the “incomplete labeling” which allows a reduced degree of user interaction. The user interaction consists simply of specifying only the background pixels by dragging a rectangle around the desired foreground object (Figure 1). The process of GrabCut initialization works as follows.

Step 1. A trimap $T = \{TB, TU, TF\}$ is initialized in a semiautomatic way. The two regions TB and TU contain the initial background and uncertain pixels, respectively, while $TF = \emptyset$. The initial TB is determined as the pixels around the outside of the marked rectangle. Pixels belonging to TB are considered as a fixed background, whereas those belonging to TU will be labeled by the algorithm.

Step 2. An initial image segmentation $\alpha = (\alpha_1, \dots, \alpha_i, \dots, \alpha_N)$, $\alpha_i \in \{0, 1\}$, is created, where all unknown pixels are tentatively placed in the foreground class ($\alpha_i = 1$ for $i \in TU$) and all known background pixels are placed in the background class ($\alpha_i = 0$ for $i \in TB$).

Step 3. Two full covariance Gaussian mixture models (GMMs) are defined, each consisting of $K = 5$ components, one for

background pixels ($\alpha_i = 0$) and the other one for foreground (initially unknown) pixels ($\alpha_i = 1$). The K components of both GMMs are initialized from the foreground and background classes using the Orchard and Bouman clustering technique.

5.1.2. GrabCut Iterative Energy Minimization. The final segmentation is performed using the iterative minimization algorithm of the Graph Cut [6] in the following steps.

Step 4. Each pixel in the foreground class is assigned to the most likely Gaussian component in the foreground GMM. Similarly, each pixel in the background is assigned to the most likely background Gaussian component.

Step 5. The GMMs are thrown away and new GMMs are learned from the pixel sets created in the previous set.

Step 6. A graph is built and Graph Cut is run to find a new foreground and background classification of pixels.

Step 7. Steps 4–6 are repeated until the classification converges.

This has the advantage of allowing the automatic refinement of the opacities α , as newly labeled pixels from the TU region of the initial trimap are used to refine the color of the GMM.

5.2. Proposed Automatic GrabCut. Although the incomplete user labeling of GrabCut reduces the user interaction substantially, it is still a requirement in order to initiate the segmentation process. This identifies GrabCut as a semiautomatic/supervised segmentation algorithm. In order to allow the image to be segmented into proper segments without any user guidance, this requires replacing the semiautomatic/supervised step of GrabCut initialization with a totally automatic/unsupervised one.

In this paper, the Orchard and Bouman [7] is proposed to be used as an image clustering technique to automatically set the initial trimap T and the initial segmentation (Section 5.1, Steps 1 and 2). The distinction between the trimap and

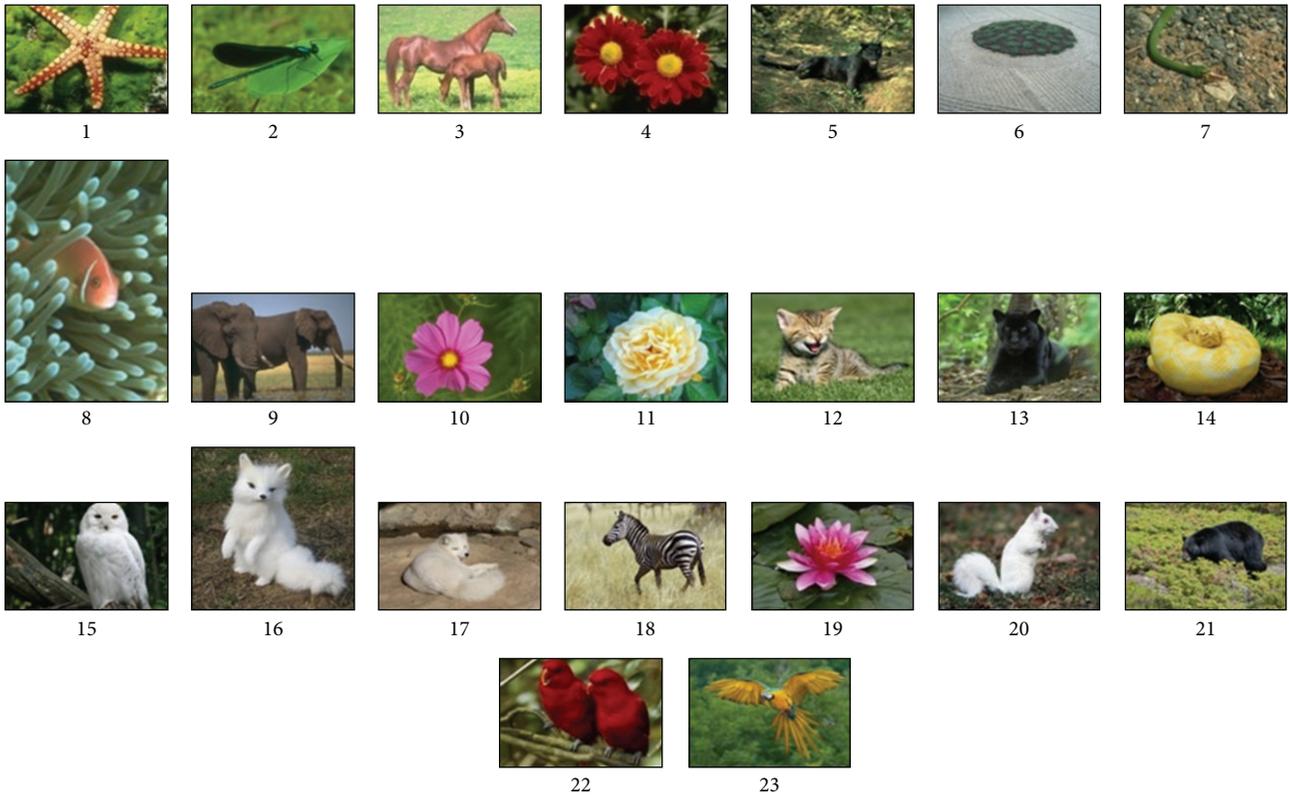


FIGURE 2: The dataset of images.

the segmentation formalizes the separation between the region of interest to be segmented and the final segmentation derived by the GrabCut algorithm. In the automatic technique, Steps 1 and 2 of the GrabCut initialization process will be modified as follows.

Step 1. While the original GrabCut constructs a trimap T of two regions, TB and TU, as a fixed background and unknown regions, respectively, the proposed automatic technique considers the whole image as one unknown region TU, where $TU = \{z_i \in \{z_1, \dots, z_n, \dots, z_N\}, i \in [1, \dots, N]\}$. This means that no fixed foreground or background regions are known and all image pixels will be involved in the minimization process to be labeled by the algorithm.

Step 2. The image is initially separated into two foreground TF and background TB regions, using the Orchard and Bouman clustering technique. During this step, a new GMM is introduced, which consists of only two components ($K = 2$): one component for the background pixels ($\alpha_i = 0$) and the other for the foreground pixels ($\alpha_i = 1$). The Orchard and Bouman clustering technique is then applied and repeated until reaching the number of components ($K = 2$) in the GMM, resulting in separating the image exactly into two clusters.

Step 3. The colors of image pixels belonging to each cluster (foreground and background clusters) generated from

the previous step are then used to initialize another two full covariance Gaussian mixture models (GMMs) with ($K = 5$).

Steps 4–7. The learning portion of the algorithm runs exactly as the original GrabCut (Section 5.1, Steps 4–7).

6. Results and Discussions

The automatic GrabCut technique was experimentally tested using a dataset of 23 different images, as shown in Figure 2. According to literature, many recent works in the fields of cluster based image segmentation and automatic image segmentation are conducting their experiments on fewer numbers of images such as [9, 11, 30, 31]. They are using a dataset of 8, 4, 4, and 15 images, respectively. In this work using a dataset of 23 images can be considered a reasonable number of test cases. This dataset is collected partially from the Berkeley segmentation dataset [32] and from publically available images [33] in a way that matches certain criteria. These criteria consider a special fitting into two class segmentations, including having mainly one object (as a foreground) and a well separation between the foreground and background color regions.

For evaluation, it was noticed that no binary segmentations exist as part of the human segmentations included in the Berkeley segmentation dataset [32]. For this reason, the ground truth data for our selected dataset is manually

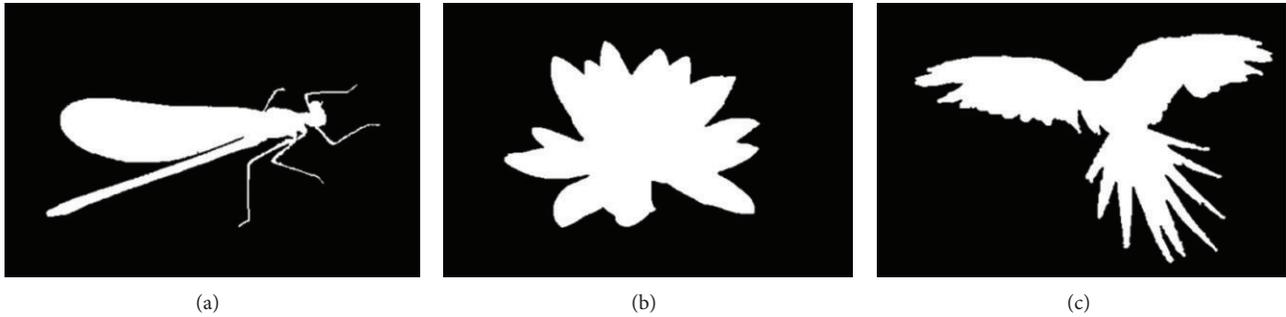


FIGURE 3: Samples of the manual binary ground truths generated.

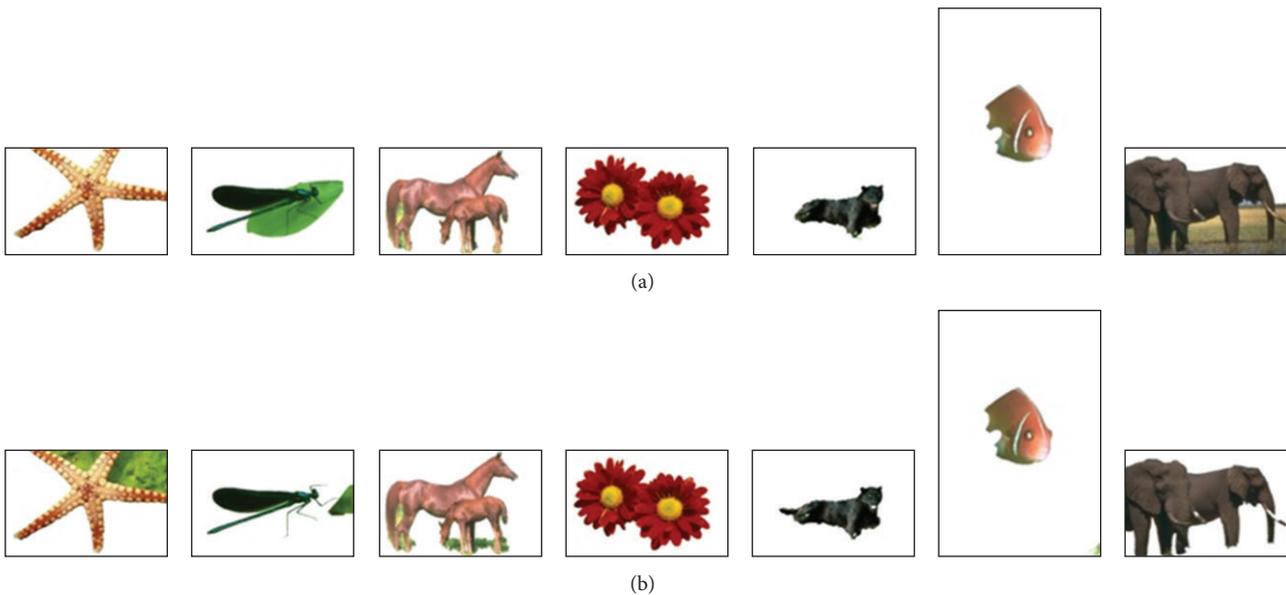


FIGURE 4: Visual comparison of the segmentation results of (a) original semiautomatic GrabCut and (b) automatic GrabCut initialized using Orchard and Bouman.

generated using standard image processing tools (Adobe Photoshop). Figure 3 displays samples of the manual binary ground truths generated. The error rate and the overlap score rate are used as two evaluation metrics. The error rate is calculated as the fraction of pixels with wrong segmentations (compared to ground truth) divided by the total number of pixels in the image. The overlap score rate is given by $y_1 \cap y_2 / y_1 \cup y_2$, where y_1 and y_2 are any two binary segmentations.

In the first experiment, automatic GrabCut, which is initialized using Orchard and Bouman, is applied and compared to the original GrabCut algorithm. Figure 4 shows sample visual results for the segmentation using ($K = 5$) components for GMMs as recommended by Rother et al. [5]. Table 1 shows the quantitative comparison between the original and modified GrabCut for the whole dataset as presented in Figure 2. As shown in Table 1, the automatic GrabCut using Orchard and Bouman clustering outperforms the original one in terms of minimizing the error and improving the segmentation accuracy. The average error rate is 3.64% for the automatic GrabCut compared to 4.28% for the original

GrabCut technique. The overall performance looks better in terms of the standard deviation (SD) which exhibits 3.61% for the automatic GrabCut compared to 5.5% for the original GrabCut.

Some cases with bad segmentation error using the original GrabCut can be noticed in Table 1 (images 1 and 9). This explains one main drawback of the original GrabCut initialization, which makes the segmentation results sensitive to the user selection of the area of interest to be segmented. This occurs when other objects, which are out of interest, may be considered as part of the foreground by being located within the area of the dragged rectangular boundary around the object of interest. The segmentation results of these two images are visually illustrated in Figure 4(a). It can be noticed how a large portion of the leaf appears in the final segmentation of the insect image. The same problem occurred when considering the land as part of the foreground area with the elephant image. The quantitative comparisons of the error rates generated for these two images in Table 1 and visual comparisons in Figure 4 illustrate the efficiency

TABLE 1: Comparisons between the original and automatic GrabCut.

Image	Error rate %		Overlap score rate %	
	Original semiautomatic GrabCut	Automatic GrabCut using Orchard and Bouman	Original semiautomatic GrabCut	Automatic GrabCut using Orchard and Bouman
1	3.05	18.70	95.91	58.57
2	15.48	5.74	43.96	75.69
3	4.79	7.31	91.51	85.48
4	3.07	3.09	97.06	97.02
5	4.16	3.75	82.42	85.18
6	0.86	0.86	97.17	97.17
7	2.40	2.40	69.00	69.01
8	0.87	1.08	92.76	90.81
9	25.81	2.17	67.66	97.31
10	2.82	2.81	96.32	96.35
11	2.05	2.05	97.04	97.04
12	4.99	4.93	88.45	89.32
13	2.28	2.30	95.71	95.64
14	2.36	2.56	96.85	96.41
15	2.78	3.50	94.14	91.98
16	3.16	3.02	93.38	93.80
17	2.08	2.10	95.19	95.11
18	3.88	3.86	90.95	91.06
19	2.88	2.92	93.44	93.30
20	1.43	1.44	96.47	96.43
21	1.27	1.27	94.62	94.64
22	3.30	3.16	93.37	93.73
23	2.57	2.58	93.75	93.74
Avg.	4.28	3.64	89.44	90.21
SD	5.50	3.61	12.76	9.87

of the automatic GrabCut in handling such a problem. The efficiency of the automatic GrabCut is provoked by preventing any hard constraints to be specified during initialization either for foreground or background (Section 5.2, Step 1).

In the second experiment, the automatic GrabCut, which is initialized using Orchard and Bouman, is applied with different color space models, including *RGB*, *XYZ*, *CMY*, *YUV*, and *HSV*. The features that identify each image pixel are only the values of its three components in the selected color space. The final segmentation results are obtained for all used images. For a quantitative comparison, Table 2 shows the error rate and the overlap score rate for the whole dataset. The results in Table 2 are ordered in ascending order from left to right in terms of the total number of good image segmentation results and the average error rates. We can see that the *RGB* space is the one that obtains better results for most of the images in terms of the average error rate. *YUV* and *XYZ* follow with very little increase in the average error rate. They exhibit almost the same average error and overlap score rates, which are 5.49% for the error rate and 95.35% for the overlap score rate and 5.63% for the error rate and 95.79% for the overlap score rate, respectively. Figure 5 shows visual

segmentation results for some images, while Figure 6 shows graph plots of the average segmentation error rate and the overlap score rate for all different color spaces.

7. Conclusions and Future Work

In this paper, a modification of GrabCut is presented to eliminate the need of initial user interaction for guiding segmentation and hence converting GrabCut into an automatic segmentation technique. The modification includes using Orchard and Bouman as an unsupervised clustering technique to initialize the GrabCut segmentation process. Based on a dataset of 23 images, the experiments revealed that automatic GrabCut using Orchard and Bouman clustering outperforms the original GrabCut. It reduces the need for user intervention while segmentation and adds extra advantage for the GrabCut via automation. Furthermore, it provides robust and accurate segmentation with average error rates of 3.64% compared to the results of 4.28% average error rate that is achieved by the original GrabCut. In addition, the performance of the automatic GrabCut is evaluated using five different color spaces, *RGB*, *YUV*, *XYZ*, *HSV*, and

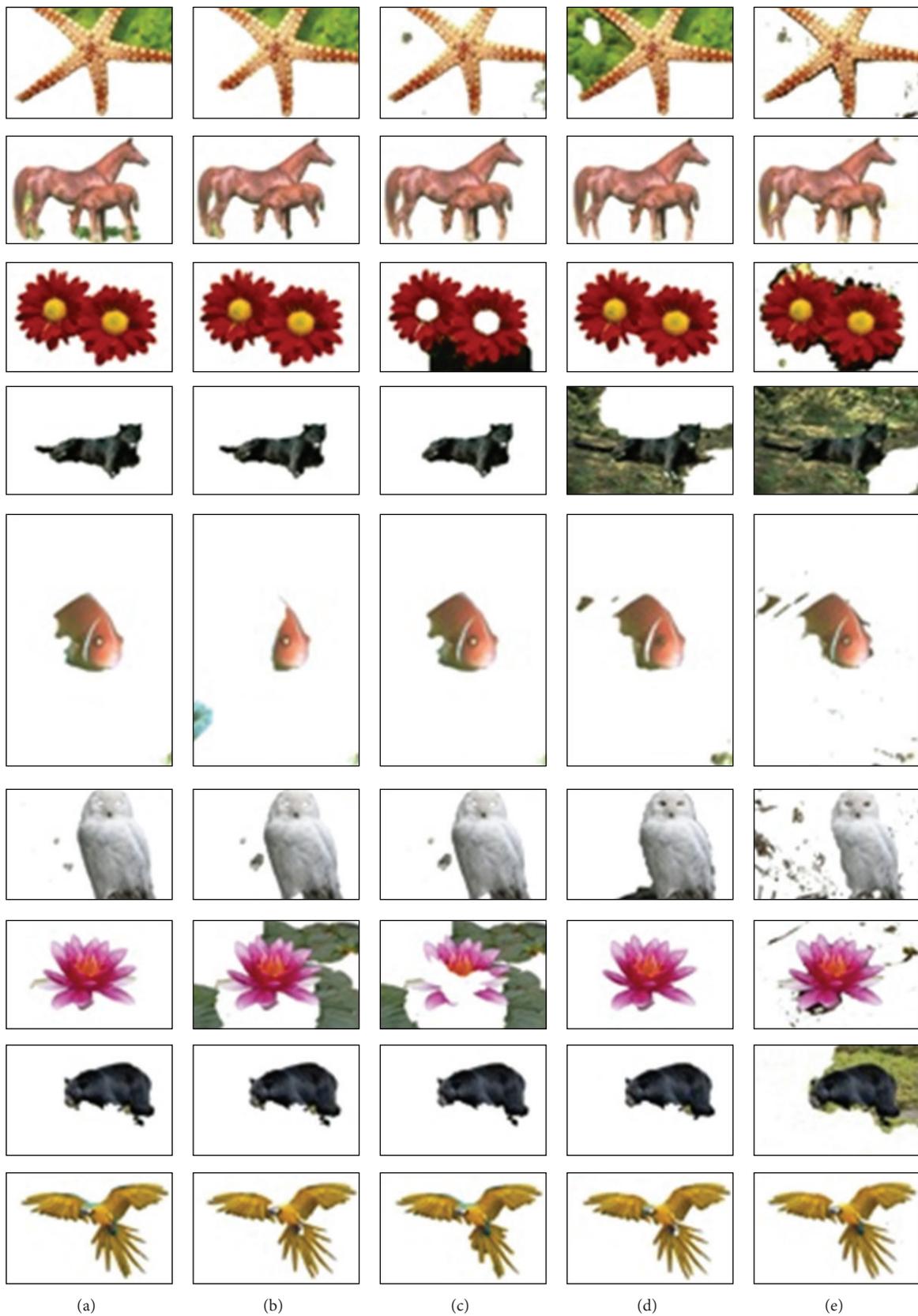


FIGURE 5: Visual comparison of segmentation results for automatic GrabCut applied in the (a) RGB, (b) YUV, (c) XYZ, (d) CMY, and (e) HSV color spaces.

TABLE 2: Experimental segmentation results on different color spaces using automatic GrabCut.

Image	Error rate %					Overlap score rate %				
	RGB	YUV	XYZ	CMY	HSV	RGB	YUV	XYZ	CMY	HSV
1	18.70	20.09	5.45	36.31	5.21	58.57	92.99	98.47	98.05	99.24
2	5.74	2.79	2.92	2.91	19.19	75.69	98.85	98.63	97.95	98.86
3	7.31	5.51	3.90	3.76	5.37	85.48	95.38	99.21	99.31	98.22
4	3.09	3.07	18.95	3.08	12.67	97.02	99.19	88.13	99.11	100
5	3.75	3.76	4.20	42.25	74.38	85.18	89.33	85.70	99.19	96.56
6	0.86	0.86	0.89	30.82	29.90	97.17	99.51	99.56	74.05	98.20
7	2.40	2.33	2.28	1.20	36.11	69.01	99.85	99.76	93.21	98.43
8	1.08	6.94	1.08	2.68	2.68	90.81	44.91	97.27	88.69	88.74
9	2.17	2.16	2.17	31.43	28.80	97.31	99.22	99.24	82.93	85.93
10	2.81	4.19	4.23	4.81	5.69	96.35	99.90	99.92	97.39	100
11	2.05	2.07	2.18	14.99	15.93	97.04	99.91	99.59	63.79	61.16
12	4.93	5.22	4.97	4.44	8.29	89.32	92.48	93.30	96.81	81.42
13	2.30	2.39	2.42	42.31	27.58	95.64	98.73	99.03	99.91	97.09
14	2.56	2.81	3.06	4.15	4.28	96.41	98.37	97.97	94.79	94.55
15	3.50	3.68	3.68	5.27	12.71	91.98	99.28	99.28	99.22	88.40
16	3.02	2.94	2.98	8.78	49.66	93.80	98.89	98.94	99.56	99.79
17	2.10	2.20	2.12	35.89	6.09	95.11	99.07	99.00	99.65	81.17
18	3.86	6.71	6.76	35.85	80.23	91.06	98.95	98.71	96.36	100
19	2.92	37.29	45.78	5.06	6.40	93.30	99.88	64.12	90.29	99.84
20	1.44	1.43	1.47	8.48	27.59	96.43	98.95	99.10	98.04	98.82
21	1.27	1.27	1.04	1.55	22.74	94.64	99.70	98.99	98.18	99.92
22	3.16	3.39	3.39	5.21	5.29	93.73	94.65	94.36	89.97	89.85
23	2.58	3.21	3.56	3.25	3.28	93.74	95.13	94.91	94.95	95.58
Avg.	3.64	5.49	5.63	14.54	21.31	90.21	95.35	95.79	93.54	93.55
SD	3.61	7.92	9.47	15.24	21.64	9.87	11.37	7.84	9.01	9.29

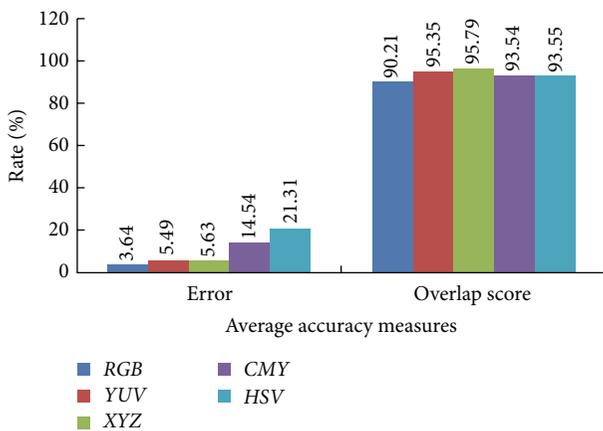


FIGURE 6: Comparison of average accuracy measures for applying automatic GrabCut segmentation on different color spaces.

CMY. The experimental results show that the segmentation results depending on the RGB color space provided the best segmentation results compared to other color spaces for the considered set of images.

This study can be improved by enlarging the dataset and including different kinds of images. On the other hand, future work might include modifying the energy minimization procedure of the automatic GrabCut to allow for multilabel optimization and segmentation.

Conflict of Interests

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

References

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice-Hall, 3rd edition, 2006.
- [2] M. Lalitha, M. Kiruthiga, and C. Loganathan, "A survey on image segmentation through clustering algorithm," *International Journal of Science and Research*, vol. 2, no. 2, pp. 348–358, 2013.
- [3] N. Sharma, M. Mishra, and M. Shrivastava, "Colour image segmentation techniques and issues: an approach," *International Journal of Scientific & Technology Research*, vol. 1, no. 4, pp. 9–12, 2012.
- [4] L. Busin, N. Vandenbroucke, and L. Macaire, "Color spaces and image segmentation," *Advances in Imaging and Electron Physics*, vol. 151, pp. 65–168, 2008.

- [5] . Rother C, V. Kolmogorov, and A. Blake, “GrabCut”: interactive foreground extraction using iterated graph cuts,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.
- [6] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images,” in *Proceedings of the 8th International Conference on Computer Vision (ICCV '01)*, vol. 1, pp. 105–112, IEEE, Vancouver, Canada, July 2001.
- [7] M. T. Orchard and C. A. Bouman, “Color quantization of images,” *IEEE Transactions on Signal Processing*, vol. 39, no. 12, pp. 2677–2690, 1991.
- [8] D. Khattab, H. M. Ebied, A. S. Hussien, and M. F. Tolba, “Automatic GrabCut based on unsupervised clustering for image segmentation,” *Journal of Computer Science and Technology*. Submitted.
- [9] O. Alata and L. Quintard, “Is there a best color space for color image characterization or representation based on multivariate gaussian mixture model?” *Computer Vision and Image Understanding*, vol. 113, no. 8, pp. 867–877, 2009.
- [10] M. Pagola, R. Ortiz, I. Irigoyen et al., “New method to assess barley nitrogen nutrition status based on image colour analysis: comparison with SPAD-502,” *Computers and Electronics in Agriculture*, vol. 65, no. 2, pp. 213–218, 2009.
- [11] A. Jurio, M. Pagola, M. Galar, C. Lopez-Molina, and D. Paternain, “A comparison study of different color spaces in clustering based image segmentation,” in *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications*, vol. 81 of *Communications in Computer and Information Science*, pp. 532–541, Springer, 2010.
- [12] J. M. Chaves-González, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez, “Detecting skin in face recognition systems: a colour spaces study,” *Digital Signal Processing*, vol. 20, no. 3, pp. 806–823, 2010.
- [13] C.-J. Du and D.-W. Sun, “Comparison of three methods for classification of pizza topping using different colour space transformations,” *Journal of Food Engineering*, vol. 68, no. 3, pp. 277–287, 2005.
- [14] G. Ruiz-Ruiz, J. Gómez-Gil, and L. M. Navas-Gracia, “Testing different color spaces based on hue for the environmentally adaptive segmentation algorithm (EASA),” *Computers and Electronics in Agriculture*, vol. 68, no. 1, pp. 88–96, 2009.
- [15] V. Gulshan, V. Lempitsky, and A. Zisserman, “Humanising GrabCut: Learning to segment humans using the Kinect,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops '11)*, pp. 1127–1133, Barcelona, Spain, November 2011.
- [16] A. Hernandez, M. Reyes, S. Escalera, and P. Radeva, “Spatio-Temporal GrabCutt human segmentation for face and pose recovery,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '10)*, San Francisco, Calif, USA, June 2010.
- [17] Y. Hu, *Human Body Region Extraction from Photos*, MVA, 2007.
- [18] D. Corrigan, S. Robinson, and A. Kokaram, “Video matting using motion extended GrabCut,” in *Proceedings of the IET European Conference on Visual Media Production (CVMP '08)*, London, UK, 2008.
- [19] C. Göring, B. Fröhlich, and J. Denzler, “Semantic segmentation using GrabCut,” in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP '12)*, pp. 597–602, February 2012.
- [20] J. Ramírez, P. Temoche, and R. Carmona, “A volume segmentation approach based on GrabCut,” *CLEI Electronic Journal*, vol. 16, no. 2, 2013.
- [21] R. Kaur and G. S. Bhathal, “A survey of clustering techniques,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 5, pp. 153–157, 2013.
- [22] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Computing Surveys*, vol. 31, no. 3, pp. 316–323, 1999.
- [23] A. Gulhane, P. L. Paikrao, and D. S. Chaudhari, “A review of image data clustering techniques,” *International Journal of Soft Computing and Engineering*, vol. 2, no. 1, pp. 212–215, 2012.
- [24] S. Naz, H. Majeed, and H. Irshad, “Image segmentation using fuzzy clustering: a survey,” in *Proceedings of the 6th International Conference on Emerging Technologies (ICET '10)*, pp. 181–186, October 2010.
- [25] N. Grira, M. Crucianu, and N. Boujemaa, “Unsupervised and semisupervised clustering: a brief survey,” in *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2005.
- [26] M. A. Ruzon and C. Tomasi, “Alpha estimation in natural images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 18–25, IEEE, 2000.
- [27] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, “A Bayesian approach to digital matting,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, pp. II264–II271, December 2001.
- [28] N. Vandenbroucke, L. Macaire, and J.-G. Postaire, “Color image segmentation by pixel classification in an adapted hybrid color space. Application to soccer image analysis,” *Computer Vision and Image Understanding*, vol. 90, no. 2, pp. 190–216, 2003.
- [29] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.
- [30] H. S. Kumar, K. Raja, K. Venugopal, and L. Patnaik, “Automatic image segmentation using wavelets,” *International Journal of Computer Science and Network Security*, vol. 9, no. 2, pp. 305–313, 2009.
- [31] C. V. Narayana, E. S. Reddy, and M. S. Prasad, “Automatic image segmentation using ultra fuzziness,” *International Journal of Computer Applications*, vol. 49, pp. 6–13, 2012.
- [32] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV '01)*, 2001.
- [33] <http://www.google.com/>.

Research Article

Ephedrine QoS: An Antidote to Slow, Congested, Bufferless NoCs

Juan Fang,¹ Zhicheng Yao,^{1,2} Xiufeng Sui,² and Yungang Bao²

¹ College of Computer Science, Beijing University of Technology, Beijing 100124, China

² Institute of Computing Technology, Chinese Academy of Science, Beijing 100190, China

Correspondence should be addressed to Xiufeng Sui; suixiufeng@ict.ac.cn

Received 24 June 2014; Accepted 30 July 2014; Published 28 August 2014

Academic Editor: Shifei Ding

Copyright © 2014 Juan Fang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Datacenters consolidate diverse applications to improve utilization. However when multiple applications are colocated on such platforms, contention for shared resources like networks-on-chip (NoCs) can degrade the performance of latency-critical online services (high-priority applications). Recently proposed bufferless NoCs (Nychis et al.) have the advantages of requiring less area and power, but they pose challenges in quality-of-service (QoS) support, which usually relies on buffer-based virtual channels (VCs). We propose QBLESS, a QoS-aware bufferless NoC scheme for datacenters. QBLESS consists of two components: a routing mechanism (QBLESS-R) that can substantially reduce flit deflection for high-priority applications and a congestion-control mechanism (QBLESS-CC) that guarantees performance for high-priority applications and improves overall system throughput. We use trace-driven simulation to model a 64-core system, finding that, when compared to BLESS, a previous state-of-the-art bufferless NoC design, QBLESS, improves performance of high-priority applications by an average of 33.2% and reduces network-hops by an average of 42.8%.

1. Introduction

Web service companies such as Google, Yahoo, Amazon, and Microsoft deploy datacenters with hundreds to thousands of machines to host millions of users, all of whom may be running large, data-intensive applications [1]. Latency-critical interactive applications must provide quality of service that is predictable and often strictly defined. To satisfy variable daily demands and to avoid contention for shared memory and network resources, datacenter operators overprovision resources, which results in poor resource utilization. Finding better ways to deliver the required QoS is thus essential for improving datacenter efficiency and managing costs.

Networks-on-chip are important shared resources in the manycore devices that will likely be used to build future datacenters. The NoCs in such chips are responsible for conveying operands between cores, accessing main memory, managing coherence, and performing I/O [2–4].

Traditional NoCs use router buffers to reduce the number of dropped or deflected (misrouted) packets. These buffers,

however, improve effective bandwidth at the expense of design complexity, chip area, and power consumption [4, 5]. Furthermore, these costs increase with the number of cores, making bufferless NoCs attractive for large-scale manycore chips [5, 6].

In contrast, bufferless NoCs eliminate on-chip router buffers so that when a flit arrives, a router must immediately select an appropriate output port to forward it. Although previous studies show that bufferless NoCs can reduce router area by 60% and save power consumption by 40% [5], the difficulty in providing QoS in such designs has prevented their use in datacenter environments with latency-critical applications. The NoCs used in datacenters generally rely on buffers to create virtual channels for different levels of service [7].

To address the problem, we propose QBLESS, a QoS-aware bufferless NoC scheme targeting datacenters. Instead of using the prevalent VC-based QoS mechanisms, QBLESS tags flits with priority bits and leverages this information in its deflection routing and congestion-control mechanisms.

The flits of latency-critical applications are assigned a high priority, making them privileged with respect to routing in the QBLESS NoC.

QBLESS routers implement two arbitration mechanisms based on priority information. First, the routing mechanism always allocates appropriate output ports to privileged flits and deflects flits of low-priority applications. To avoid live-lock, the high-priority flits undergo loss of privilege after N hops, where N is a system parameter influenced by factors such as application memory access characteristics and network size (for more details, see Section 3.1). Second, QBLESS adopts a dynamic source-throttling mechanism to control network congestion according to two rules: (1) privileged sources will never be throttled and (2) the throttling rates of nonprivileged sources are proportional to their IPFs (instructions-per-flit), a measure that indicates memory access intensity.

To enable the QBLESS NoC scheme, we add a QoS-register to each core and design a router architecture that can be programmed for various applications demands. We study QBLESS in simulator, experimental results which show that QBLESS effectively improves QoS and performance. Compared to BLESS [5], a current state-of-the-art bufferless NoC, QBLESS improves the performance of latency-critical applications by up to 55.1% (60.0%) in a 64-core (100-core) system with an 8×8 (10×10) mesh NoC. Average improvement is 33.2% (38.2%). Somewhat counterintuitively, QBLESS does not hurt low-priority applications but improves their performance by 1.7%, on average, over BLESS.

2. Background and Related Work

Datacenters are built from high-end chip multiprocessor (CMP) servers. CMPs rely on efficient networks-on-chip to synchronize cores and to coordinate access to shared memory and I/O resources. Here we present background specific to datacenter NoCs and briefly survey the most relevant prior work.

2.1. QoS and Utilization in Datacenters. In datacenters using CMPs with tens of cores, more and more workloads are deployed on a single server, and thus they must share resources. Kambadur et al. [8] point out that in Google datacenters, an average of 14 hyperthreads from heterogeneous applications run simultaneously on one server. For instance, on a single machine, there may be five to even twenty unique applications running together. Such mixed workloads degrade application performance. In particular, they can influence the QoS of interactive online services, which is strongly related to user experience and is a key factor in the revenue of the Internet companies. Datacenter operators thus overprovision resources to guarantee QoS to these latency-critical applications, even if doing so lowers resource utilization. For instance, Google [9] reports that CPU utilization in a typical 20,000-server datacenter for online services averaged about 30% during January through March, 2013. In contrast, batch-workload datacenters averaged 75% CPU utilization during the same period.

Modern datacenters sacrifice server utilization to guarantee the QoS of online services by separating them from batch workloads. Previous efforts to increase utilization while keeping a high level of QoS have colocated the two incompatible kinds of workloads on the same node to eliminate interference. Tang et al. explore the impact of the shared memory subsystem (including the last level cache (LLC) and front side bus (FSB)) on Google datacenter applications [10]. They propose ReQoS [1] to monitor the QoS of latency-sensitive applications and adaptively reduce the memory demands of low-priority applications. They also study the negative effects that nonuniform memory access (NUMA) [11] brings to Google's important web services like the Gmail backend and web-search frontend.

Previous work on guaranteeing datacenter QoS mainly focuses on the on-chip and off-chip memory subsystems. However, just as the security level is defined by the weakest component, QoS is dictated by the least robust participant: this means that all shared resources must be QoS-aware if any are to meet service-level agreements (SLAs). Improving NoC QoS technology for interactive datacenter applications is thus one promising direction for achieving higher throughput and greater energy-efficiency.

2.2. QoS-Aware Buffered NoCs. Dally and Towles [7] show that adding buffers to create virtual channels not only prevents deadlock but also makes it possible to provide different levels of service.

Many buffer-based QoS approaches have thus been proposed for NoCs. For example, MANGO [12] guarantees QoS by prioritizing virtual circuits and partitioning virtual channels (VCs) with different priorities. Instead of prioritizing VCs, Bolotin et al. [13] propose prioritizing control packets over data packets. Das et al. [14] propose application-aware prioritization policies to improve overall application throughput and ensure fairness in NoCs. Grot et al. [15] propose a preemptive virtual clock (PVC) scheme to reduce dedicated VCs for QoS-support. Ouyang and Xie [16] design LOFT, a scheme that leverages a local frame-based scheduling mechanism and a flow-control mechanism to guarantee QoS. Grot et al. propose Kilo-NOC [17], a topology-aware QoS NoC architecture, that can substantially reduce buffer overhead.

2.3. Bufferless NoCs. Some recent work focuses on alternative designs that are tradeoff power consumption, die area, and performance. One promising direction is bufferless routing [5], which temporarily misroutes or drops and retransmits packets to effectively resolve output port contention. Moscibroda and Mutlu [5] propose the BLESS routing algorithm which consists of a set of rules for routers to select flits and output ports. Fallin et al. [18] propose the CHIPPER router architecture to reduce the complexity of BLESS control logic.

Bufferless routing yields significant network power savings with minimal performance loss when the network load is low-to-medium. In such bufferless NoCs, router area is reduced by 40–75% and power consumption is reduced by 20–40% [5, 6, 18]. However, for network-intensive

workloads, bufferless routing behaves much worse than traditional buffered NoCs due to high deflections rates and bandwidth saturation.

To bridge the performance gap between the buffered and bufferless NoCs at high network load, one possible approach is to directly improve the efficiency of bufferless deflection routing. Previous work [6, 19–21] uses source-throttling or constraining applications' network request rates to reduce deflection-rates and improve overall system throughput. Nychis et al. [6] propose the BLESS-throttling (BLESS-T) algorithm to mitigate congestion by limiting traffic from NoC-insensitive applications. Ausavarungnirun et al. [19] propose an application-aware mechanism, adaptive cluster throttling (ACT), to improve throughput and fairness by throttling cluster of application. Kim et al. [20] propose clumsy flow-control (CFC) to degrade network congestion by implementing credit based flow-control in bufferless NoCs.

Another approach is to make a hybrid network that can adaptively switch between the higher-capacity buffered mode and lower-cost bufferless mode. Jafri et al. [22] propose adaptive flow-control (AFC) to allow routers to switch between backpressure mode (in which they store incoming flits) and back pressureless mode (in which they use deflection), which performs well under both high and low network loads.

Previous proposals are effective in improving throughput and fairness of bufferless NoCs but they are not suited for datacenter environments with mixed workloads where the performance of latency-critical applications might be substantially degraded. In this work, we investigate both congestion control and deflection routing, finding that the latter is more effective in guaranteeing QoS in bufferless NoCs.

2.4. QoS-Aware Bufferless NoCs. Since almost all QoS-support techniques are based on buffer-based VCs, implementing QoS-support in bufferless NoCs remains an open problem.

NoCs are shared by many cores; a QoS-oblivious bufferless NoC may substantially degrade performance for latency-critical applications, even if overall throughput is high. To investigate this, we simulate a 64-core system and measure the impact of NoC contention. We designate h264ref from SPEC CPU2006 [23] to be a high-priority application, and we randomly mix it with other (low priority) applications. Figure 1 illustrates that, as the number of additional applications increases from three to 63, the IPC (instruction per cycle) of h264ref declines by 35%.

Figure 2 illustrates that BLESS-T, a state-of-the-art bufferless routing and congestion-control mechanism, still performs poorly with respect to guaranteeing SLA-level QoS for datacenter environments (here we take mcf from SPEC CPU2006 as the critical application). There are two reasons for this. First, BLESS-T allows data flits from high-priority critical applications to be deflected by low-priority flits. Figure 2 shows that mcf suffers from severe flit deflection at a rate of 51–59% (54% on average) in an 8×8 NoC. Second, since BLESS-T uses IPF as the metric to perform source-throttling, a critical application with low IPF may be

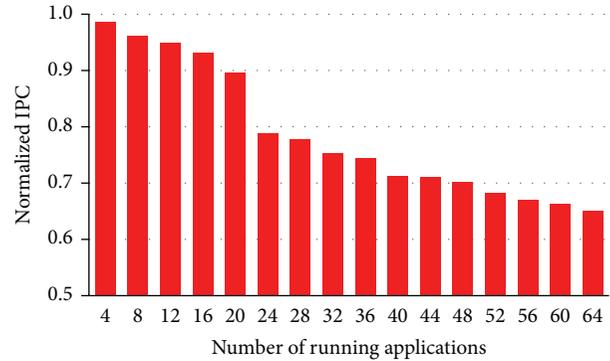


FIGURE 1: Performance decay of h264ref due to NoC contention (experimental setup is in Section 4).

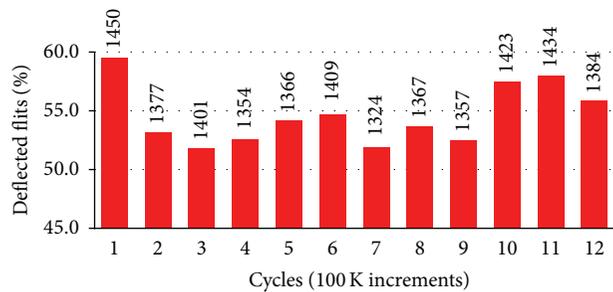


FIGURE 2: Percentage of deflected flits of mcf w/BLESS-T.

chosen as the throttling victim. Thus the application stays in a starvation state in which it is prevented from injecting flits into network. For example, our experimental results show that the throttling rate of mcf is 40%, on average.

On the one hand, bufferless NoCs have the advantages of small area and low power. On the other hand, SLA-level QoS-support is essential for improving datacenter utilization through resource sharing. These factors motivate us to investigate how to design and implement QoS on bufferless NoCs.

3. QBLESS Design

We propose QBLESS, a QoS-aware bufferless NoC design, for datacenter environments. Figure 3 illustrates the organization of our QBLESS scheme. In particular, QBLESS consists of three components: (1) a bufferless routing mechanism is responsible for selecting appropriate output ports for incoming flits in light of priority information (Section 3.1); (2) a congestion-control mechanism implements source throttling, obeying a new set of rules to adjust throttling rates (Section 3.2); and (3) a tagging mechanism conveys application flit priority information to NoC routers (Section 3.3).

To integrate these mechanisms into NoCs, we need to add a set of registers and to modify some components in the routers. Figure 3 shows these modules we add (in red) and the modules we modify (in blue). We present the details of our three mechanisms in the following subsections.

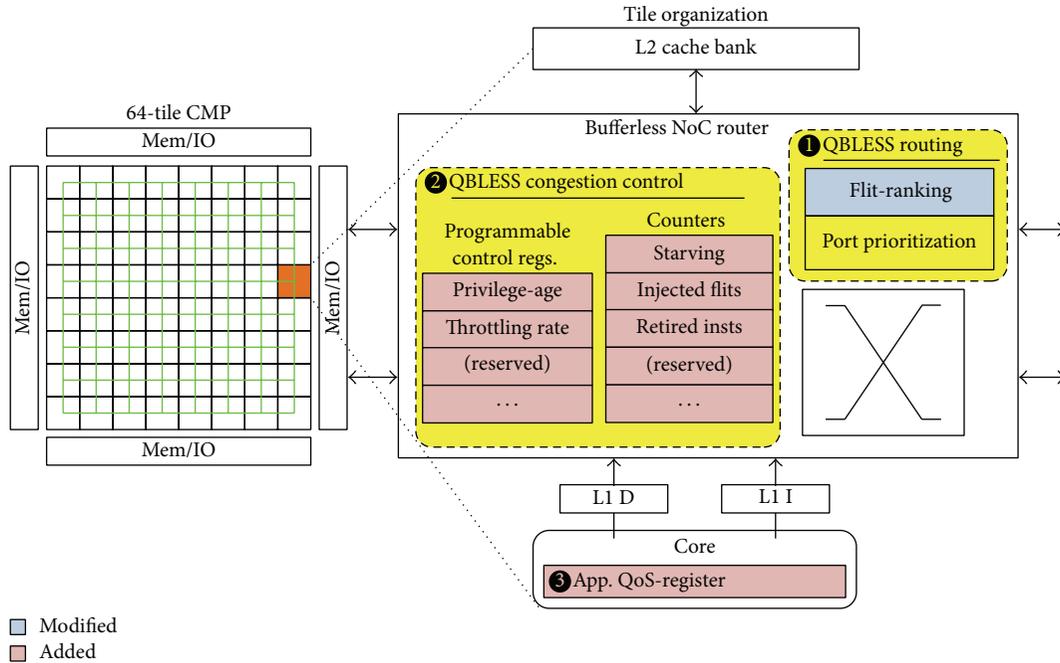


FIGURE 3: QBLESS router architecture.

3.1. QBLESS Routing (QBLESS-R)

3.1.1. Illustrative Example. Figure 4 illustrates the principle of the QBLESS-R mechanism. A high-priority application sends a flit via path (8 → 5 → 2) according to the rules of dimension-order routing, as shown in Figure 4(a). Meanwhile, a low-priority application wants to send a flit through path (3 → 4 → 5 → 2). The flit of the low-priority application is sent one cycle before its competitor so that the two flits arrive at router number 5 at the same time. They contend for the same output port to router number 2. Since there is no buffer, one must be deflected in a wrong direction.

Previous routing algorithms usually adopt age-based arbitration to determine which flit to deflect, regardless of the priority of the data flits. Therefore, in this case, because the age of the low-priority flit is one hop larger than that of the high-priority flit, the high-priority flit is deflected to router number 4, as shown in Figure 4(b). Figure 4(c) shows that QBLESS allows the high-priority flit to go through router number 5 and deflects the low-priority flit to router number 4. Thus, compared to the QoS-unaware routing algorithm, QBLESS removes two hops from the path of the high-priority flit.

To achieve this, a QBLESS router must perform two tasks: ranking flits to select an appropriate flit candidate (flit-ranking) and prioritizing available output ports to select an appropriate one (port-prioritizing).

3.1.2. Flit-Ranking. Previous routing algorithms usually adopt age-based arbitration, for example, BLESS using an oldest-first (OF) ranking policy that performs best in most scenarios in terms of latency, deflection-rate, and

energy-efficiency [5]. However, the OF-only ranking policy is unaware of priority, which means that flits of latency-critical applications will inevitably be deflected.

In QBLESS, a high-priority flit obtains a *privilege-age* when injected into the NoC, which means that the age of the high-priority flit is a certain number of hops ahead of low-priority flits. As shown in Figure 3, the value of this *privilege-age* is stored in a register and is programmable via software.

Determining the value of the *privilege-age* is critical to effectiveness of the QBLESS routing. The value should be large enough to allow high-priority applications to always beat low-priority ones. For low-priority applications, *privilege-age* should be small enough to avoid livelock.

The value of *privilege-age* is determined by the network parameters (size, diameter), the individual application characteristics (IPF and data locality), and the network confliction. In practice, *privilege-age* is an empirical parameter that reflects QBLESS’s ability to guarantee high-priority applications. Since it is programmable, we can dynamically adjust its value according to NoC performance. We evaluate the impact of *privilege-age* in Section 5.2.

3.1.3. Port-Prioritizing. When a flit arrives at the router, the router first tries to assign it to its preferred port. If the preferred port is occupied, the router assigns it to a deflecting port. The routing algorithm must guarantee that low-priority flits are not deflected indefinitely.

3.2. QBLESS Congestion Control. Starvation occurs due to congestion when a router exhausts all its ports and cannot

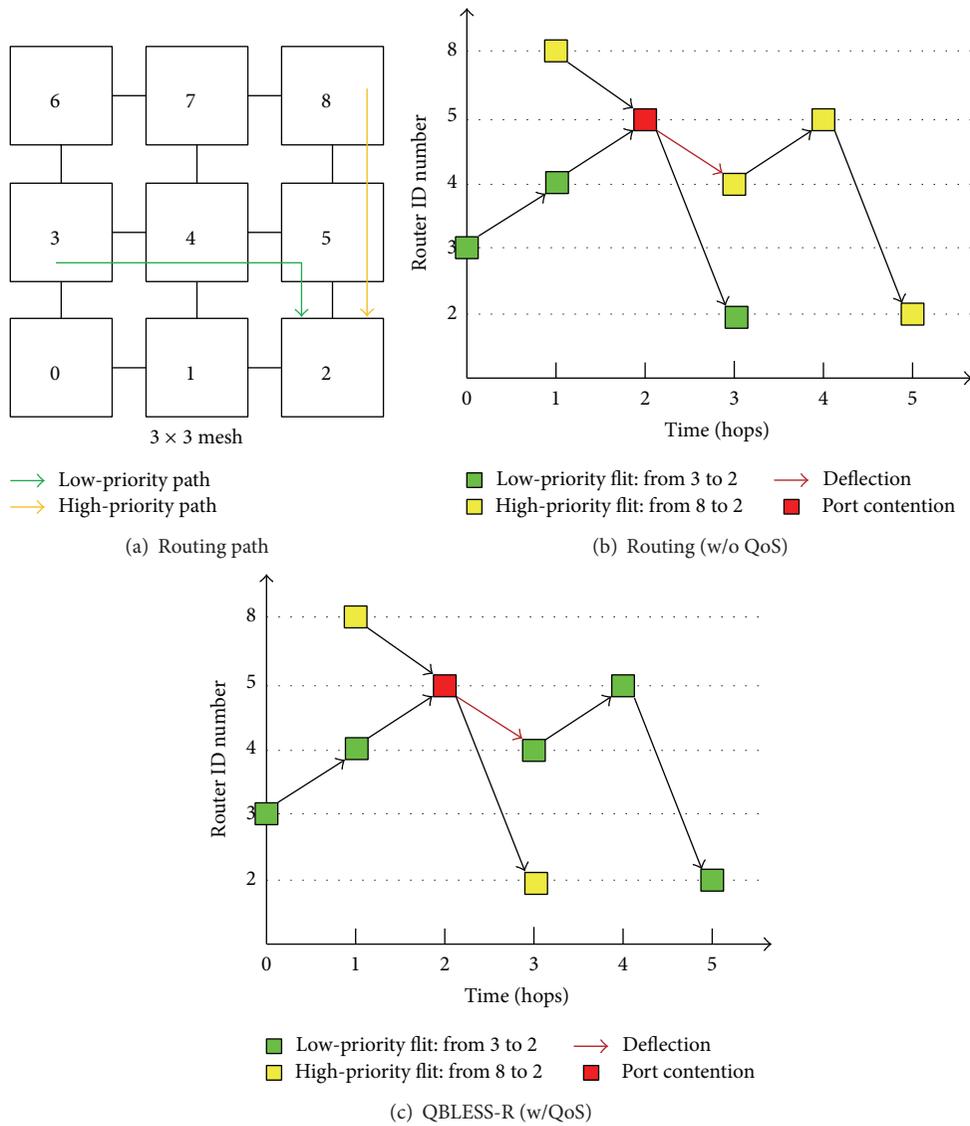


FIGURE 4: Example of QBLESS-R.

inject new flits into the NoC. For bufferless NoCs, congestion increases the deflection-rate, and deflections exacerbate congestion. Thus, congestion control is important for both throughput and latency.

Throttling a specific application is an effective approach for mitigating starvation, but it degrades the performance of the victim application. Previous studies [6, 19–21] try to enforce overall system throughput and fairness according to IPF, MPKI (misses per kilo-instructions), and injection rate. Although such mechanisms (e.g., BLESS-T) are effective for improving fairness, they are unsuitable for datacenter environments.

Figure 5 illustrates the principles of the QBLESS-CC mechanism. Like previous schemes, QBLESS-CC also adopts source-throttling to control congestion. In contrast to those previous schemes, QBLESS-CC can recognize network nodes

injecting high-priority flits and avoid throttling them, as shown in Figure 5(b).

Table 1 illustrates QBLESS-CC rules. Program execution is divided into a series of epochs. During each epoch, each network node performs two tasks: determining throttling rate and monitoring/updating statistics. To achieve these goals, we add a set of registers to each router to record the dynamic throttling rate and to track the number of starvation cycles, injected flits, and retired instructions (see Figure 3). There is a global controller that periodically collects these data to identify congestion spots and to calculate throttling rates. Specifically, QBLESS-CC needs to address the following three issues.

3.2.1. *When to Throttle.* In each epoch, a global controller collects the IPF and starvation rate of each router. If any

TABLE 1: Interval-based QBLESS congestion control.

Each node	Global controller
(1) Dynamically throttle according to global controller information from previous quantum	(1) Collect node measurements from previous quantum
(2) Monitor IPF and starvation throughout this quantum	(2) Identify congestion spots
	(3) Calculate throttling rate for next quantum
	(4) Broadcast throttling rate
	(5) Wait for next quantum and repeat

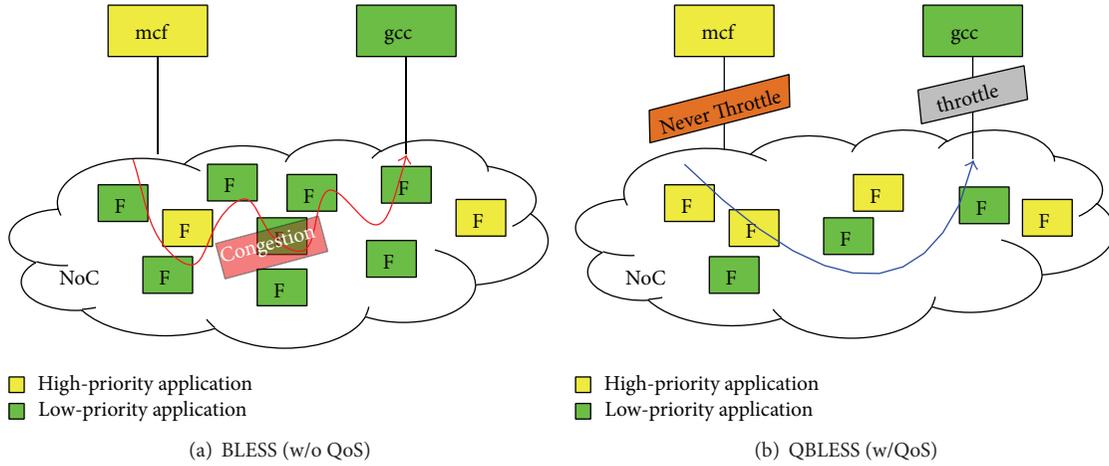


FIGURE 5: Example of QBLESS-CC.

router's starvation rate exceeds a threshold, the network is deemed to be congested. Note that each router has its own threshold:

$$\text{threshold} = \min \left(\alpha + \frac{\beta}{\text{IPF} + \text{priority} \times \lambda}, \gamma \right). \quad (1)$$

Equation (1) defines the relationship between IPF, priority, and threshold. These coefficients are not fixed and can be changed by the operating system.

3.2.2. Whom to Throttle. Generally, high-priority applications are not targeted for throttling. Low-priority applications whose IPFs are lower than the average value are selected as throttling candidates.

3.2.3. How Much to Throttle. Lower IPF indicates less NoC sensitivity, and thus applications with lower IPF can be throttled more than others. In particular, we adopt the algorithm of BLESS-T [6] and add priority to the calculation of the throttling rate as shown in (2). As in (1), all coefficients are programmable:

$$\text{throttle}_{\text{rate}} = \min \left(\rho + \frac{\sigma}{\text{IPF} + \text{Priority} \times \varphi}, \tau \right). \quad (2)$$

3.3. QoS Identification. Each flit has a (potentially multibit) priority tag. For example, one bit can be used to indicate two priority levels. To make the QBLESS scheme easy to understand, we use just one bit to present our design. In practice, priority levels can be extended (Section 5.3).

The *priority tags* are obtained from application QoS registers in the CPU cores. Specifically, we leverage a QoS framework that adds priority information to each process control block (PCB), and we add a corresponding QoS-register to each core (see Figure 3). The priority information is programmed by the operation system (OS). Upon a context switch, the OS stores the value of the QoS-register into the PCB of the old process and then loads the new priority value from the process to be run. On each memory access request, the core reads the value from the QoS-register and sets the priority value for the request. Thus all the NoC packets contain this priority information.

4. Methodology

4.1. Simulator Model. We use MacSim [24], a trace-driven, cycle-level, heterogeneous architecture simulator. MacSim models a detailed pipeline (in-order and out-of-order), a memory system that includes caches, the NoC, and the memory controllers. We model an 8×8 (10×10)-mesh CMP. Table 2 shows the parameters of our system. We run 10 million cycles for each experiment.

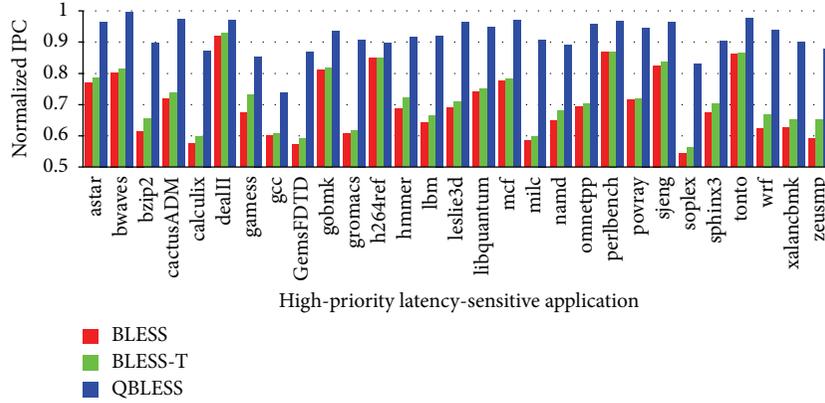


FIGURE 6: Performance of high-priority applications.

TABLE 2: System parameters for evaluation.

Network	Topology	2D mesh, 8 × 8 (10 × 10) size
	Routing algorithm	QBLESS (BLESS)
	Routing latency	2 cycles
Core	Out-of-order, 16 MSHR, 128 instructions window size	
	L1 I-cache and D-cache: 32 KB, 64 B line-size, 2-way, LRU, 2-cycle hit. The L1 caches are private to each core	
L2 cache	Per-block interleaving, shared, distributed, 64 B line-size, perfect	

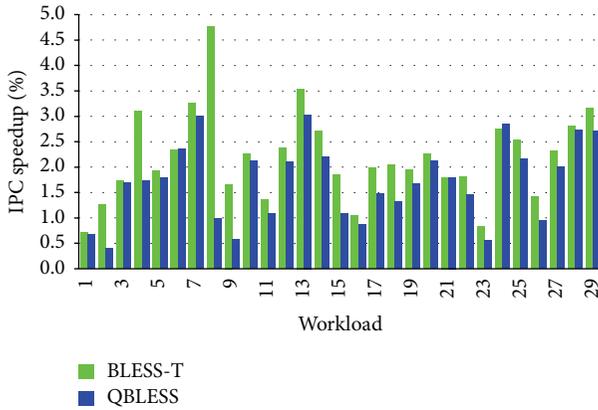


FIGURE 7: Performance of low-priority applications (normalized to BLESS).

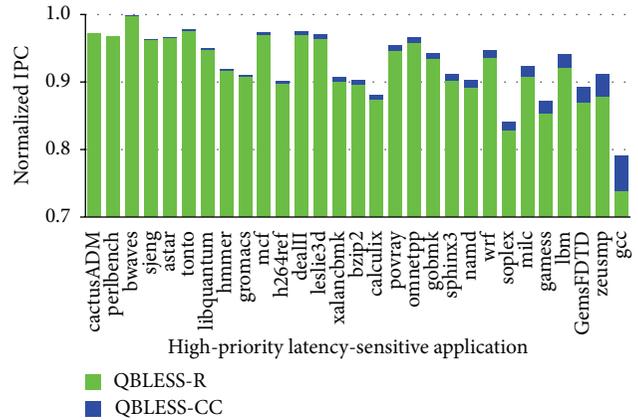


FIGURE 8: Performance breakdown of routing and congestion control (normalized to solo).

4.2. *Workloads.* We evaluate randomly generated multiprogrammed workloads from 29 SPEC CPU2006 on both the 64-core and 100-core systems. Each workload consists of one high-priority application and other low-priority applications. For each application, we capture the instruction trace of a representative execution slice using a Pin tool [25].

4.3. *QBLESS Parameters.* We determine the following algorithm parameters based on empirical evaluations [6]: *privilege-age* is set to 32, and the period of network information collection T is set to 100 K cycles. For the congestion threshold, we set the range limit from $\alpha = 0.01$ to $\gamma = 0.7$. We set the coefficient $\beta = 0.4$ and the priority associated factor

$\lambda = 2$. We set the throttling rate interval to go from $\rho = 0.25$ to $\tau = 0.8$, and the factor $\sigma = 0.9$, $\phi = 2$.

4.4. *Comparison Mechanisms.* To evaluate QBLESS, we implement two previously proposed bufferless routing and congestion-control mechanisms in our simulator: BLESS [5] and BLESS-T [6].

5. Evaluation

In this section, we evaluate the effectiveness and scalability of QBLESS.

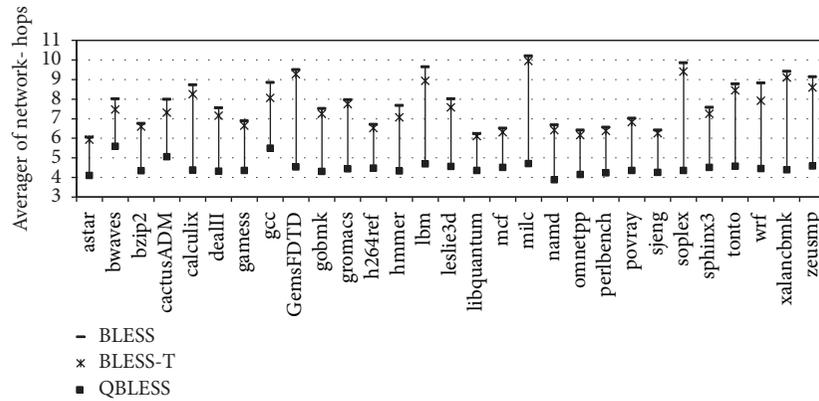


FIGURE 9: Average network-hops of high-priority applications.

5.1. Overall Performance

5.1.1. High-Priority Applications. Figure 6 shows the IPC slowdown of the high-priority applications in each of the 29 workloads. The results are normalized to solo execution (the selected high-priority application runs alone). According to Figure 6, QBLESS reduces IPCs by less than 10% on average. This is much better than BLESS, which is unaware of SLA-level QoS. Although BLESS-T can reduce network congestion to improve system performance by throttling network nodes, it is unable to distinguish high-priority applications from low-priority ones. High-priority applications thus suffer from being heavily throttled. As expected, our results demonstrate that, for high-priority applications, QBLESS performs much better than methods that have no SLA-level QoS guarantee mechanism.

5.1.2. Low-Priority Applications. Although QBLESS can guarantee the QoS requirements of high-priority applications, the overall throughput of other low-priority applications is also improved. Figure 7 illustrates these counterintuitive results; QBLESS improves the throughput of low-priority applications by 0.4%~3.0% and 1.7% on average. Compared to BLESS-T, the overall system throughput of the most low-priority workloads drops negligibly by only 0.4%.

There are two reasons for this: first, QBLESS-CC can reduce network congestion to improve overall system performance; second, QBLESS-R ensures that the flits of low-priority applications arrive at their destinations after a certain number of hops of delay.

Based on Figures 6 and 7, we conclude that QBLESS improves performance for high-priority applications with negligible impact on corunning low-priority applications.

5.2. Analysis

5.2.1. Performance Breakdown of Routing and Throttling. Figure 8 illustrates the performance breakdown of the routing mechanism and the congestion-control mechanism. The bars are sorted in the ascending order of QBLESS-CC. Figure 8 shows that QBLESS-R contributes more than 90% to the

performance improvement of the high-priority applications, indicating the effectiveness of QBLESS-R.

On the other hand, congestion control is also effective for some applications, such as gcc, although the benefit is not that obvious due to relative low network intensity of our workload traces. In fact, as pointed out by Nychis et al. [6], network congestion can cause application throughput reductions for both small and large network loads. So we believe that QBLESS can gain more benefits from QBLESS-CC when network is more heavily congested.

5.2.2. Average Network Hops. As illustrated in Figure 9, the network-hops of QBLESS are 3.9 to 5.6 (4.4 on average), which reduces the average network-hops by 41.7% and 38.9%, respectively, compared to BLESS and BLESS-T. The deflection-rate of high-priority application is largely reduced, since QBLESS-R prioritizes the flits of latency-critical application and assigns the preferred ports.

5.2.3. Privilege-Age. As mentioned in Section 3.1, the value of *privilege-age* is critical to the effectiveness of QBLESS-R but is difficult to be determined. We conduct many experiments and results in Figure 10 show that 32 is a good enough empirical value for *privilege-age*. It is interesting that *privilege-age* has negligible impact on low-priority applications. Therefore, we choose *privilege-age* = 32 for QBLESS evaluation. It is worth noting that *privilege-age* is programmable.

5.3. Scalability

5.3.1. Multiple Priorities. In previous experiments, QBLESS supports only two priorities. We extend QBLESS to support 4 priorities (three-level high priorities and one low priority) by using two priority bits. Figure 11 shows that higher priority yields better performance. For example, the highest priority applications achieve 94.5%~96.7% performance compared to solo while the middle and lower priority applications achieve 85.3%~90.7% and 73.8%~84.2%, respectively. These gradient performance results conclude that QBLESS can be extended to support multipriority easily with very low cost.

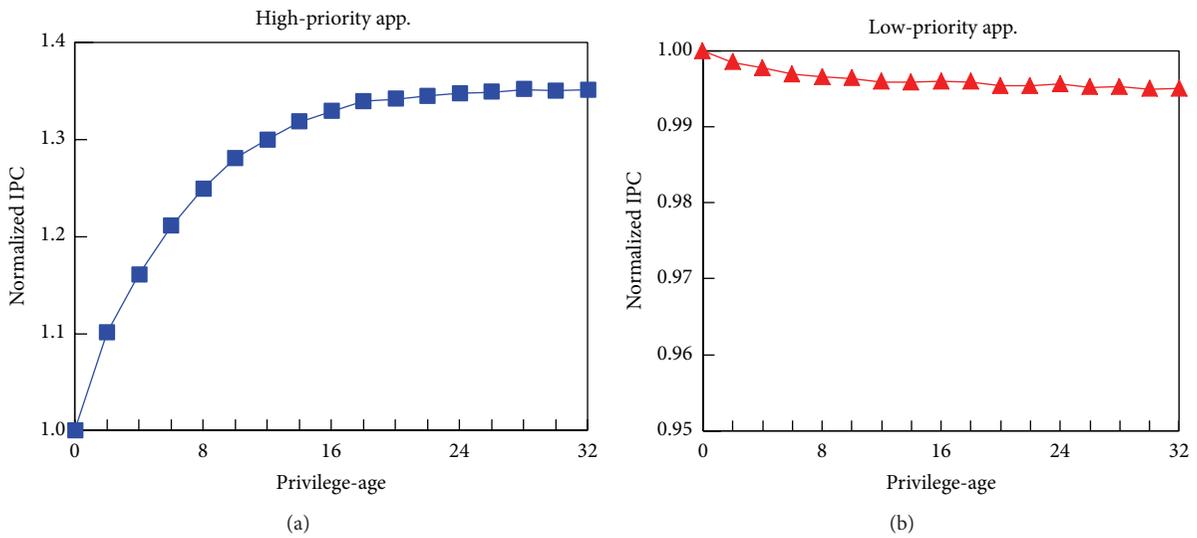


FIGURE 10: The impact of privilege-age.

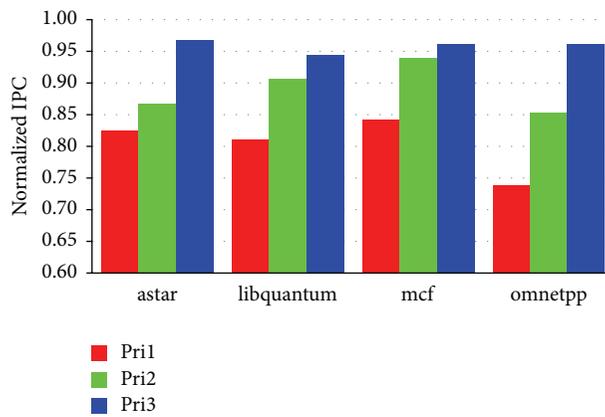


FIGURE 11: The performance impact of different priorities.

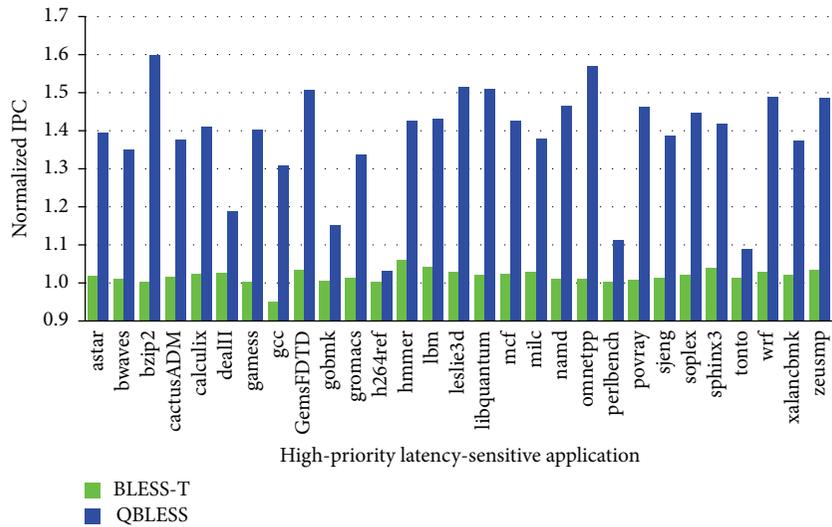


FIGURE 12: Performance of high-priority applications (100 cores, normalized to BLESS).

5.3.2. *100 Cores.* We perform experiments to evaluate QBLESS in a 100-core system. As shown in Figure 12, compared to BLESS and BLESS-T, QBLESS improves the performance of latency-critical application by 3.2%~60.0% (38.2% on average) and 3.0%~59.4% (35.7% on average), respectively, which is more significant than the 64-core system. This means that QBLESS can achieve good performance scalability with SLA-QoS as the number of core increases.

5.4. *Hardware Overhead.* The major source of hardware overhead of QBLESS is the modification of router architecture, which is required to measure the starvation rate at each node and to throttle injection. As shown in Figure 3, in each router, QBLESS requires three 32-bit counters and two 8-bit control registers. Additionally, an 8-bit register is required in each core to store the QoS information derived from the application level. Each tile, containing one process core and one router, requires only 15 bytes ($= 3 \times 4B + 2 \times 1B + 1B$) of storage overhead in total, which is much less than the storage overhead for implementing the buffered router (256 bytes per router).

6. Conclusion

We propose QBLESS, a hardware programmable approach for reducing in-network contention in bufferless NoCs. QBLESS adaptively selects the routed output port and throttling rate of low-priority applications to ensure the QoS of high-priority latency-critical corunners. We examine both application level and network level performance in 8×8 and 10×10 networks and show significant QoS improvements for latency-critical applications on a variety of real workloads.

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 all the anonymous reviewers for their insightful comments and suggestions. This work is supported by the National Natural Science Foundation of China under Grant nos. 61202062, 60903046, and 61202076, CCF-Intel Young Faculty Research Program (YFRP), and the General Program of Science and Technology Development Project of the Beijing Municipal Education Commission (Grant no. KM201210005022).

References

- [1] L. Tang, J. Mars, W. Wang, T. Dey, and M. L. Soffa, "ReQoS: reactive static/dynamic compilation for QoS in warehouse scale computers," in *Proceedings of the 18th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '13)*, pp. 89–100, March 2013.
- [2] D. Wentzlaff, P. Griffin, H. Hoffmann et al., "On-chip interconnection architecture of the tile processor," *IEEE Micro*, vol. 27, no. 5, pp. 15–31, 2007.
- [3] A. Olofsson, R. Trogan, O. Raikhman, and L. Adapteva, "A 1024-core 70 GFLOP/W floating point manycore microprocessor," in *Proceedings of the Workshop on High Performance Embedded Computing (HPEC '11)*, 2011.
- [4] K. Sankaralingam, R. Nagarajan, R. McDonald et al., "Distributed microarchitectural protocols in the TRIPS prototype processor," in *Proceeding of the 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '06)*, pp. 480–491, Orlando, Fla, USA, December 2006.
- [5] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," in *Proceedings of the 36th Annual International Symposium on Computer Architecture (ISCA '09)*, pp. 196–207, June 2009.
- [6] G. P. Nychis, C. Fallin, T. Moscibroda, O. Mutlu, and S. Seshan, "On-chip networks from a networking perspective: Congestion and scalability in many-core interconnects," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 407–418, 2012.
- [7] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *Proceedings of the 38th Design Automation Conference (DAC '01)*, pp. 684–689, June 2001.
- [8] M. Kambadur, T. Moseley, R. Hank, and M. A. Kim, "Measuring interference between live datacenter applications," in *Proceedings of the 24th International Conference for High Performance Computing, Networking, Storage and Analysis (SC '12)*, p. 51, Salt Lake City, Utah, USA, November 2012.
- [9] J. C. Luiz André Barroso and U. Hölzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, Morgan & Claypool, 2nd edition, 2013.
- [10] L. Tang, J. Mars, N. Vachharajani, R. Hundt, and M. L. Soffa, "The impact of memory subsystem resource sharing on datacenter applications," in *Proceedings of the 38th Annual International Symposium on Computer Architecture (ISCA'11)*, pp. 283–294, June 2011.
- [11] L. Tang, J. Mars, X. Zhang, R. Hagmann, R. Hundt, and E. Tune, "Optimizing Google's warehouse scale computers: the NUMA experience," in *Proceedings of the 19th IEEE International Symposium on High Performance Computer Architecture (HPCA '13)*, pp. 188–197, Shenzhen, China, February 2013.
- [12] T. Bjerregaard and J. Sparsø, "A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip," in *Proceedings of the Design, Automation and Test in Europe (DATE '05)*, pp. 1226–1231, March 2005.
- [13] E. Bolotin, Z. Guz, I. Cidon, R. Ginosar, and A. Kolodny, "The power of priority: NoC based distributed cache coherency," in *Proceeding of the First International Symposium on Networks-on-Chip (NOCS '07)*, pp. 117–126, Princeton, NJ, USA, May 2007.
- [14] R. Das, O. Mutlu, T. Moscibroda, and C. R. Das, "Application-aware prioritization mechanisms for on-chip networks," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '09)*, pp. 280–291, December 2009.
- [15] B. Grot, S. W. Keckler, and O. Mutlu, "Preemptive virtual clock: a flexible, efficient, and cost-effective QoS scheme for networks-on-chip," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, Micro-42 (MICRO '09)*, pp. 268–279, December 2009.
- [16] J. Ouyang and Y. Xie, "LOFT: a high performance network-on-chip providing quality-of-service support," in *Proceedings of the 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '10)*, pp. 409–420, Atlanta, Ga, USA, December 2010.

- [17] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu, "Kilo-NOC: a heterogeneous network-on-chip architecture for scalability and service guarantees," in *Proceedings of the 38th Annual International Symposium on Computer Architecture (ISCA '11)*, pp. 401–412, June 2011.
- [18] C. Fallin, C. Craik, and O. Mutlu, "CHIPPER: a low-complexity bufferless deflection router," in *Proceedings of the 17th International Symposium on High-Performance Computer Architecture (HPCA '11)*, pp. 144–155, San Antonio, Tex, USA, February 2011.
- [19] R. Ausavarungnirun, K. K.-W. Chang, C. Fallin, and O. Mutlu, "Adaptive cluster throttling: improving high-load performance in bufferless on-chip networks," SAFARI Technical Report TR 2011-0062011, Computer Architecture Lab (CALCM) Carnegie Mellon University, 2011.
- [20] Y. Kim, H. Kim, and J. Kim, "Clumsy flow control for high-throughput bufferless on-chip networks," *IEEE Computer Architecture Letters*, vol. 12, no. 2, pp. 47–50, 2012.
- [21] K. K. Chang, R. Ausavarungnirun, C. Fallin, and O. Mutlu, "HAT: heterogeneous adaptive throttling for on-chip networks," in *Proceedings of the 24th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD '12)*, pp. 9–18, October 2012.
- [22] S. A. R. Jafri, Y.-J. Hong, M. Thottethodi, and T. N. Vijaykumar, "Adaptive flow control for robust performance and energy," in *Proceedings of the 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '10)*, pp. 433–444, December 2010.
- [23] J. L. Henning, "SPEC CPU2006 benchmark descriptions," *ACM SIGARCH Computer Architecture News*, vol. 34, no. 4, pp. 1–17, 2006.
- [24] H. Kim, J. Lee, N. B. Lakshminarayana, J. Sim, J. Lim, and T. Pho, "MacSim: a CPU-GPU heterogeneous simulation framework," HPArch Research Group, Georgia Institute of Technology, 2012.
- [25] C.-K. Luk, R. Cohn, R. Muth et al., "Pin: building customized program analysis tools with dynamic instrumentation," in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '05)*, pp. 190–200, June 2005.

Research Article

Discrete Bat Algorithm for Optimal Problem of Permutation Flow Shop Scheduling

Qifang Luo, Yongquan Zhou, Jian Xie, Mingzhi Ma, and Liangliang Li

College of Information Science and Engineering, Guangxi University for Nationalities, Nanning, Guangxi 530006, China

Correspondence should be addressed to Qifang Luo; l.qf@163.com

Received 22 June 2014; Accepted 30 July 2014; Published 27 August 2014

Academic Editor: Shifei Ding

Copyright © 2014 Qifang Luo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A discrete bat algorithm (DBA) is proposed for optimal permutation flow shop scheduling problem (PFSP). Firstly, the discrete bat algorithm is constructed based on the idea of basic bat algorithm, which divide whole scheduling problem into many subscheduling problems and then NEH heuristic be introduced to solve subscheduling problem. Secondly, some subsequences are operated with certain probability in the pulse emission and loudness phases. An intensive virtual population neighborhood search is integrated into the discrete bat algorithm to further improve the performance. Finally, the experimental results show the suitability and efficiency of the present discrete bat algorithm for optimal permutation flow shop scheduling problem.

1. Introduction

Scheduling problems are taking the very important effect in both manufacturing systems and industrial process for improving the utilization efficiency of resources [1], such as, aircraft landing scheduling problem, job shop scheduling problem, and flow shop scheduling problem. In the past several decades, scheduling problems are widely researched. Permutation flow shop scheduling problem (PFSP) is one of best known production scheduling problems, which can be viewed as a simplified version of the flow shop problem and has been proved that non-deterministic polynomial (NP) time [2]. Due to its significance in both academic and engineering applications, the permutation flow shop with the criterion of minimizing the makespan, maximum lateness of jobs, or minimizing total flow time, a great diversity of methods have been proposed to solve PFSP and some achievements were obtained.

So far, there are many methods that have been introduced for solving PFSP with the objective of minimizing the makespan. To sum up, these methods can be classified into three categories: exact methods, constructive heuristic methods, and metaheuristic algorithms based on the constructive operation and neighborhood search. Exact methods include

branch and bound method [3], integer linear programming method [4], and so on. Constructive heuristic methods which build some rule to construct a feasible scheduling, such as,

Johnson method, Rajendran NEH can be viewed as the typical cases [5]. Among them, the NEH is one of the most successful constructive methods and can provide comparable results with metaheuristics. The metaheuristics mainly include genetic algorithm (GA) [6], particle swarm optimization algorithm (PSO) [7], differential evolution (DE) [8], and bat algorithm (BA) [9] and so on. Many metaheuristic algorithms are used to solve flow shop scheduling based on the constructive operation and neighborhood search in the past few years. In [6], Wang and Zheng proposed a SGA to solve flow shop scheduling, which used the well-known NEH combined with GA to generate the initial population and applied multicrossover operators to enhance the exploring potential. In [10], Tasgetiren et al. applied the PSO algorithm to solve PFSP for makespan and total flow time minimization by using the smallest position value rule borrowed from the random key representation of GA, and the proposed algorithm was combined with the variable neighborhood-based local search, as called PSO-VNS. Liu et al., in [11], proposed an efficient particle swarm optimization based mimetic algorithm (MA) for PFSP to minimize

the maximum completion time. In [12], two effective heuristics are used during the local search to improve all generated chromosomes in every generation. Yagmahan and Yenisey have proposed a multiobjective ant colony system algorithm to simultaneously minimize objectives of makespan and total flow time [13]. Tasgetiren et al. present a discrete artificial bee colony algorithm hybridized with a variant of iterated greedy algorithms to find the permutation that gives the smallest total flow time [14]. In [15], a novel mechanism is employed in initializing the pheromone trails based on an initial sequence, and the pheromone trail intensities are limited between lower and upper bounds which change dynamically. Moreover, a local search is performed to improve the performance quality of the solution. In [16], Li and Yin applied a differential evolution based memetic algorithm, named ODDE, to solve PFSP by combining with NEH heuristic initialization, opposition-based learning, pairwise local search, and fast local search in ODDE. In [17], Liu et al. a multipopulation PSO based memetic algorithm for permutation flow shop scheduling is proposed. In [18], Mirabi proposed a novel hybrid genetic algorithm to solve the sequence-dependent permutation flow shop scheduling problem. In [19], Victor and Framinan use on insertion tie-breaking rules in heuristics for the permutation flow shop scheduling problem.

In recent years, a bat algorithm (BA) as a new metaheuristic optimization algorithm is proposed [9]. BA is inspired by the intelligent echolocation behavior of microbats when their foraging. After the bat algorithm is proposed by Yang in 2010, bat algorithm is used to solve various optimization problems. For example, Gandomi et al. focus on solving constrained optimization tasks [20]. Yang and Gandomi apply bat algorithm to solve many global engineering optimizations [21]. Mishra et al. present a model for classification using bat algorithm to update the weights of a functional link artificial neural network (FLANN) classifier [22]. Meanwhile, there are improved bat algorithms that are applied to various optimization problems; Xie et al. proposed a DLBA bat algorithm based on differential operator and Lévy flights trajectory to solve function optimization and nonlinear equations [23]. Wang et al. proposed a new bat algorithm with mutation (BAM) to solve the uninhabited combat air vehicle (UCAV) path planning problem [24]. In this paper, we propose a discrete bat algorithm (DBA) to solve PFSP. Here, the DBA is constructed based on the idea of continuous bat algorithm, which divide whole scheduling problem into many subscheduling problems, then NEH heuristic was introduced to solve subscheduling problem. Moreover, some subsequences are operated with certain probability in the pulse emission and loudness phases. An intensive virtual population neighborhood search is integrated into the DBA to further improve the performance. Finally, the experimental results show the effectiveness of the discrete bat algorithm for PFSP.

2. Problem Descriptions and Bat Algorithm

2.1. Permutation Flow Shop Scheduling Problem. The permutation flow shop scheduling problem (PFSP) in the paper

consists of a set of jobs on a set of machines with the objective of minimizing the makespan. In PFSP, n jobs are to be processed on a series of m machines, sequentially. All jobs are processed in the same permutation; meanwhile, every job is processed in one machine only once and each machine can only process one job at a time, and all jobs are processed in an identical processing order on all machines.

The permutation flow shop scheduling problems are often denoted by the symbols $n | m | \text{prmu} | C_{\max}$, where n represents the number of jobs; m is the number of machines; prmu denotes the type of flow shop scheduling problem; and C_{\max} is the makespan. Let $t_{i,j}$ ($1 \leq i \leq n$, $1 \leq j \leq m$) be the times of job i processed on machine j , assuming preparation time for each job is zero or is included in the processing time $t_{i,j}$; $\pi = (j_1, j_2, \dots, j_n)$ is a scheduling permutation of all jobs. Π is set of all scheduling permutation. $C(j_i, k)$ is completion time of job j_i on machine k , and every job will be processed on machine 1 to machine m orderly. The completion time of the permutation flow shop scheduling problem according to the processing sequence $\pi = (j_1, j_2, \dots, j_n)$ is shown as follows:

$$\begin{aligned} C(j_1, 1) &= t_{j_1,1}, \\ C(j_i, 1) &= C(j_{i-1}, 1) + t_{j_i,1}, \quad i = 2, 3, \dots, n, \\ C(j_1, k) &= C(j_1, k-1) + t_{j_1,k}, \quad k = 2, 3, \dots, m, \\ C(j_i, k) &= \max\{C(j_{i-1}, k), C(j_i, k-1)\} + t_{j_i,k}, \\ & \quad i = 2, 3, \dots, n, \quad k = 2, 3, \dots, m, \\ \pi_* &= \arg\{C_{\max}(\pi) = C(j_n, m)\} \longrightarrow \min, \quad \forall \pi \in \Pi, \end{aligned} \quad (1)$$

where π_* is the most suitable arrangement which is the goal of the permutation flow shop problem to find $C_{\max}(\pi_*)$ is the minimal makespan.

2.2. Bat Algorithm (BA). The bat algorithm (BA) is an evolutionary algorithm first introduced by Yang in 2010 [9]. In simulations of BA, under several ideal rules, the updated rules of their positions x_i and velocities v_i in a D-dimensional search space are defined. The new solutions x_i^t and velocities v_i^t at generation t are given by

$$\begin{aligned} f_i &= f_{\min} + (f_{\max} - f_{\min})\beta, \\ v_i^t &= v_i^{t-1} + (x_i^t - x_*)f_i, \\ x_i^t &= x_i^{t-1} + v_i^t, \end{aligned} \quad (2)$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution, f_i denotes frequency of each bat, and the frequency $f_i \in [f_{\min}, f_{\max}]$. Here x_* is the current global best location (solution) which is located after comparing all the solutions among all the n bats.

After the position updating of bat, a random number is generated; if the random number is greater than the pulse

```

Begin
  Initialization. Set the generation counter  $t = 1$ ; Initialize the population of  $NP$  bats
   $P$  randomly and each bat corresponding to a potential solution to the given problems;
  define loudness  $A_i$ , pulse frequency  $Q_i$  and the initial velocities  $v_i$  ( $i = 1, 2, \dots, NP$ );
  set pulse rate  $r_i$ .
  While the termination criterion is not satisfied or  $t < MaxGeneration$  do
    Generate new solutions by adjusting frequency, and updating velocities and location
    Solutions (2),
    if ( $rand > r_i$ ) then
      Select a solution among the best solutions;
      Generate a location solution around the selected best solution
    endif
      Generate a new solution by flying randomly
    if ( $rand < A_i \ \&\& \ f(x_i) < f(x_*)$ )
      Accept the new solution
      Increase  $r_i$  and reduce  $A_i$ 
    endif
      Rank the bats and the find the current best  $x_*$ 
       $t = t + 1$ ;
    endwhile
    Post-processing the results and visualization.
end.

```

ALGORITHM 1: Basic bat algorithm (BA).

```

Compute the total processing time for each job on  $m$  machine;
Generate a sequence  $j = (j_1, j_2, \dots, j_n)$  by sorting the jobs in non-increasing order according to
the total processing time;
The first job is taken.  $\pi_* = \{j_1\}$ ;
for  $i = 1 : n - 1$ 
  /* The implemented operations of NEH and NEH1 is different, the NEH insert a job into all possible
  positions of  $\pi_*$ , but the NEH1 only insert a job into the front and rear of  $\pi_*$ . The other operations are
  consistent both NEH and NEH1. */
  Take job  $j_i$  form  $j$  and insert  $j_i$  into all possible positions of  $\pi_*$ ; // Operation of NEH
  Take job  $j_i$  form  $j$  and insert  $j_i$  into the front and rear of  $\pi_*$ ; // Operation of NEH1
  Evaluate the new sequence  $\pi \leftarrow \pi_* \cup j_i$ ;
  Select the  $\pi_* \leftarrow \pi$  with lowest objective value;
endfor
return  $\pi_*$ ;

```

ALGORITHM 2: The pseudocode of NEH and NEH1.

```

for each individual
  Compute pulse emission rate  $r_i$  by (6);
  if  $rand > r_i$ 
    /* sub-sequence swap */
    Randomly select two sub-sequences defined by frequency  $f$  on  $x_i^n(t)$ ;
    Swap the two sub-sequences to generate a new position;
  else
    /* sub-sequence inserting */
    Randomly select one sub-sequence defined by frequency  $f$  on  $x_i^n(t)$ ;
    Insert this sub-sequence into a random location in remainder sequence;
  endif
endfor

```

ALGORITHM 3: The pseudocode of pulse emission rate local operation.

emission rate r_i , a new position will be generated around the current best solutions, and it can be represented by

$$x = x_* + \varepsilon A_t, \quad (3)$$

where $\varepsilon \in [-1, 1]$ is a random number, while $A_t = \langle A_i^t \rangle$ is the average loudness of all the bats at current generation t .

Furthermore, the loudness A_i and the pulse emission rate r_i will be updated and a solution will be accepted if a random number is less than loudness A_i and $f(x_i) < f(x_*)$. A_i and r_i are updated by

$$A_i^{t+1} = \alpha A_i^t, \quad r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)], \quad (4)$$

where α , γ are constants and $f(\cdot)$ is fitness function. The algorithm repeats until the termination criterion is reached. The basic steps of the bat algorithm (BA) can be described in Algorithm 1.

3. Discrete Bat Algorithm for PFSP

Since standard BA is a continuous optimization algorithm, the standard continuous encoding scheme of BA cannot be used to solve PFSP directly. Meanwhile, many combinational optimization problems are discrete problem, and PFSP is a typical case. In order to apply BA to PFSP, there are two methods: the first method is to solve PFSP using continuous BA, however, this method needs to construct a direct mapping relationship between the job sequence and the vector of individuals in BA; the second method is to construct a discrete BA for PFSP. Therefore, in this paper, a discrete bat algorithm is proposed to solve PFSP with minimal makespan.

In addition, for PFSP, some neighborhood search methods always are used to enhance the quality of the solution, and the performance is remarkable. In this paper, four neighborhood search methods, that is, insert, swap, inverse, and crossover, will be employed. These neighborhood operations are shown in Figure 1. The details of these neighborhoods are as follows.

Swap. Choose two different positions from a job permutation randomly and swap them.

Insert. Choose two different positions from a job permutation randomly and insert the back one before the front.

Inverse. Inverse the subsequence between two different random positions of a job permutation.

Crossover. Choose a subsequence in a random interval from another random job permutation and replace the corresponding part of subsequence.

3.1. Solution Representation in DBA. In original BA, the position of each virtual bat is viewed as a candidate solution of problem; these bat individuals adjust the flight speed by randomly selecting frequency of sonic wave which they emitted and then update the position of bats. Furthermore, the pulse emission rate and loudness are used to control the intensive local search that is process to generate a new

individual around the current global best solution. In DBA, in general, the position $x_i^n(t)$ of individual i denotes a scheduling plan on t th iteration, where n represents the scheduling plan including n jobs. The $x_i^n(t)$ is also viewed as a $\pi = (j_1, j_2, \dots, j_n)$. For example, if $x_1^4(2) = [3 \ 2 \ 1 \ 4]$, which represents the processing order of all jobs on all machines, is $3 \rightarrow 2 \rightarrow 1 \rightarrow 4$, this permutation represents the position of first bat individual in second generation. The velocity $v_i^N(t)$ consists of a part of scheduling plan or whole scheduling plan on t th iteration, where $N \leq n$.

3.2. Population Initialization. In this paper, the DBA is applied to explore the new search space. Initial swarm is often generated randomly, and, in DBA, this initial strategy is adopted. Meanwhile, recent studies have confirmed the superiority of NEH over the most recent constructive heuristic [5]. Many metaheuristic algorithms in order to generate an initial population with certain quality and diversity take advantage of the NEH heuristic to generate some individuals and the rest of the individuals are initialized with random values [16]. In this paper, this kind of initialization strategy is not including in DBA, but NEH is used in position updating of bat. However, a discrete bat algorithm with NEH initialization strategy is experimented. By experiments, we find that the combination of NEH initialization strategy and succeeding operation always deteriorates the population diversity, by tracking offspring, the results showed that all the individuals in the final population were similar.

In [25], NEH heuristic is regarded as the best heuristic for the PFSP. The NEH algorithm is based on the idea that the high processing time on all machines should be scheduled as early in the sequence as possible. The NEH heuristic has two phases.

- (1) The jobs are sorted in nonincreasing sums of their processing time.
- (2) A job sequence is established by evaluating the partial schedules based on the initial order of the first phase. The standard NEH and a variant of standard NEH (NEH1) can be described as shown in Algorithm 2; the only difference of two NEH is that the inserted position of new job in partial schedules is different: NEH1 have only two possibilities of inserting.

3.3. Position Updating of Bat. Scheduling problem with many jobs can be viewed as a combination of many subscheduling problems; as we all know, we can apply dynamic programming to solve this problem. However, in this paper, the idea of partition is adopted, a complete scheduling sequence is divided into many segments, and each subscheduling problem is solved by superior NEH.

In continuous BA, the bat individual randomly selects a certain range of frequency, and its speed is updated according to their selected frequency; at last, a new position is generated using its speed and its own position. In DBA, for each individual, firstly, a frequency f is selected in the range of frequency $[f_{\min}, f_{\max}]$; frequency f denotes the number of

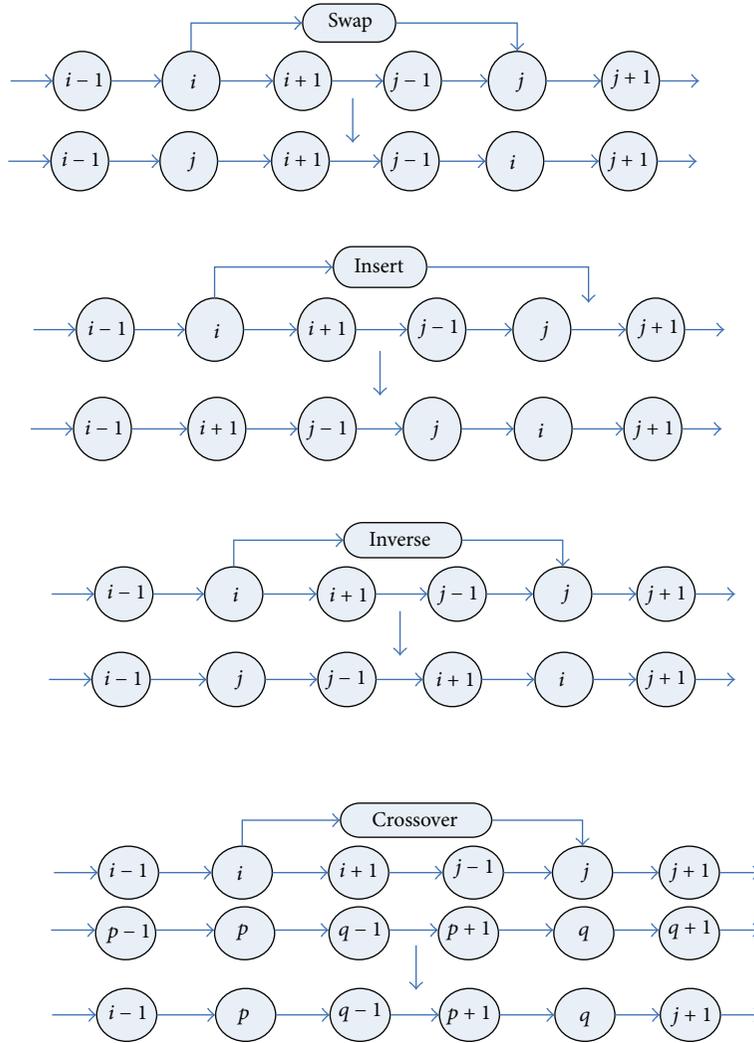


FIGURE 1: Four neighborhood operations (swap, insert, inverse, and crossover).

subsequences, where f_{\min}, f_{\max} are two integers in the range of job amount n ,

$$f = \left\lfloor f_{\max} + (f_{\min} - f_{\max}) \times \left(\frac{t}{t_{\max}} \right) \right\rfloor, \quad (5)$$

where $\lfloor \cdot \rfloor$ denotes rounded down function. Secondly, frequency f decides the starting location and ending location of each subsequence, and the position $x_i^n(t)$ is divided into f subsequences; these subsequences are viewed as the velocity $v_{i,f}^N(t)$ of bat individual, where $N \leq n$. Thirdly, these velocities are updated by NEH; the new velocity is called $v_{\text{tmp},f}^N(t)$. At last, the corresponding part of $x_i^n(t)$ is replaced by $v_{\text{tmp},f}^N(t)$. In order to facilitate understanding, there is a simple instance: $f = 3 \in [2, 4], n = 8, x(t) = [5, 1, 3, 2, 4, 7, 6, 8]$; $v_1 = [5, 1, 3], v_2 = [2, 4, 7], v_3 = [6, 8]$; $v_{\text{tmp},1} = [1, 3, 5], v_{\text{tmp},2} = [4, 2, 7], v_{\text{tmp},3} = [6, 8]$; $v_i^N(t) = [2, 1, 3]$, so $x(t+1) = [1, 3, 5, 4, 2, 7, 6, 8]$.

3.4. Pulse Emission Rate Local Operation. In original BA, the pulse emission rate and loudness are used to control the intensive local search, that is to generate a new individual around the current global best individual $gbest_x$. In DBA, each individual has its own pulse emission rate r_i . The initial pulse emission rate is a positive and smaller number; with the increase of iteration, pulse emission rate r_i will increase to 1. The updating of r_i using

$$r_i(t) = 1 + \exp \left(-\frac{10}{t_{\max}} \times \left(t - \frac{t_{\max}}{2} \right) + r_i(1) \right)^{-1}. \quad (6)$$

Figure 2 presents an example of updating curve of pulse emission rate r_i under maximal iterations is 100, pulse emission rate r_i has a value ranging from 0 to 1. Using this updating formula, the algorithm can not only quickly exploit near the current optimal position in the early iteration, so that speed up the convergence rate, but also can mainly concentrate in diversity in later search and can avoid to fall into local optima.

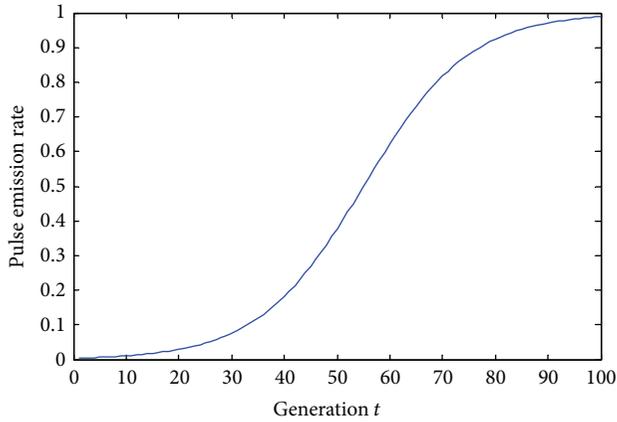


FIGURE 2: Updating curve of pulse emission rate r_i .

The pulse emission rate r_i will control the subsegment local operation. For each individual, randomly generate a random number; if this random number is larger than its r_i , this position of bat individual will be updated by random swap two segments defined by frequency f ; otherwise, the updating operation will be implement by random inserting operation; the pseudo code can be described as shown in Algorithm 3.

3.5. Loudness Local Operation. In DBA, the loudness Ld_i of bat individual i is relative to its own fitness fit_i ; the better fitness, the less loudness. The loudness can be described by

$$Ld_i = \frac{(fit_i - fit_{\min})}{(fit_{\max} - fit_{\min})}, \quad (7)$$

where fit_i is the fitness of individual i and fit_{\min} and fit_{\max} are the minimum and maximum fitness in current population, respectively. In DBA, the loudness reflects the quality of individual. In this subsection, there are two kinds of local search embedded into algorithm, random subsequence inverse and random subsequence inserting. Note that, where inserting operation is different from inserting operation in Section 3.4.

In this part, for each individual, randomly generate a random number; if this random number is larger than its Ld_i , a random length of subsequence is randomly selected in range of $[1, \lfloor D/2 \rfloor]$; this position of bat individual will be updated by inserting operation with random subsequence; otherwise, the updating operation will be implement by random subsequence inverse operation. Note that the subsequence is a portion of the current best position $gbest_x$; however, the corresponding replacement portion is the individual x_i in bat population, and the pseudo code can be described as show in Algorithm 4.

Although this inserting and inverse operation may generate invalid scheduling sequence, those invalid scheduling sequences need to adjust to a feasible solution. The adjustment of the pseudo code can be described as show in Algorithm 5.

In order to facilitate understanding, process of adjustment $x = [3, 6, 3, 2, 1, 5, 5, 3]$, $\{S\} = [1, 2, 3, 5, 6]$, $\{Sid\} = [5, 4, 8, 7, 2]$, $\{R\} = [4, 7, 8]$, $\{Rid\} = [1, 3, 6]$, $IO = [3, 1, 2]$, and $x_{\text{adjust}} = [8, 6, 4, 2, 1, 7, 5, 3]$.

3.6. Intensive Virtual Population Neighborhood Search. In this paper, an intensive virtual population neighborhood search with same population size is easily embedded in DBA for solving PFSP. The purpose of the virtual population neighborhood search is to find a better solution from the neighborhood of the current global best solution. In this part, three neighborhoods, that is, insert, swap, and single-point move backward operate, are employed. These operations are used to improve the diversity of population and enhance the quality of the solution.

In order to enhance the local search ability and get a better solution, a new population is generated based on the current global best solution, and the population size is not less than original bat population; the new population is called virtual population. The new population size $ps_1 = \mu \times ps$, $\mu \geq 1$ is real number.

Firstly, the virtual population is generated by randomly selecting two jobs to perform swap operation. Secondly, the virtual population is generated by randomly selecting a job and insert into another random location. At last, the single-point move backward operation is performed also based on current global best individual $gbest_x$. In the simulation, first of all, a job position i is chosen randomly in $gbest_x$; the selected job i is inserted into the back of job i , orderly, until the population size ps_1 is reached. For example, the population size $ps_1 = 3$, random job position $i = 2$, and $gbest_x = [2, 5, 4, 1, 3]$; the virtual population is generated as follows:

$$\begin{aligned} & [2 \ 4 \ 5 \ 1 \ 3] \\ & [2 \ 4 \ 1 \ 5 \ 3] \\ & [2 \ 4 \ 1 \ 3 \ 5] \end{aligned} \quad (8)$$

3.7. Discrete Bat Algorithm (DBA). In DBA, all individuals once the update either in bat population or in virtual population, these individuals will be evaluated and one solution be accepted as the current global best $gbest_x$ if the objective fitness of it is better than the fitness of the last $gbest_x$. The algorithm terminates until the stopping criterion is reached; the DBA algorithm for PFSP can be described in Algorithm 6.

4. Numerical Simulation Results and Comparisons

To test the performance of the proposed DBA for the permutation flow shop scheduling, computational simulations are carried out with some well-studied problems taken from the OR-Library (<http://people.brunel.ac.uk/~mastjjb/info.html>). In this paper, 29 problems from two classes of PFFSP test problems are selected. The first eight problems are instances Car1, Car2 through to Car8 designed by Carlier [26]. The second 21 problems are instances Rec01, Rec03 through to Rec41 designed by Reeves and Yamada [27]. So

```

for each individual
    Compute loudness  $Ld_i$  by (7);
    if rand >  $Ld_i$ 
        /* random sub-sequence inserting */
        Randomly select a length of sub-sequence;
        Randomly determine the sub-sequence with selected length in  $gbest_x$ ;
        Insert this sub-sequence into a random location in remainder sequence;
    else
        /* random sub-sequence inverse */
        Randomly select a length of sub-sequence;
        Randomly determine the sub-sequence with selected length in  $gbest_x$ ;
        Perform inverse operation on selected sub-sequence;
        Replace original sub-sequence with inverted sub-sequence
    end if
end for
    
```

ALGORITHM 4: The pseudocode of loudness local operation.

```

for each individual
    {S, Sid} ← Find out all jobs and their position in current scheduling sequence;
    {R} ← {1:n} – {S}, where n denotes the number of jobs in current scheduling problem;
    {Rid} ← {1:n} – {Sid};
    Generate an insert order IO randomly;
    Select a job in {R} according to IO and insert into {Rid};
end for
    
```

ALGORITHM 5: The pseudocode of adjustment.

far, these problems have been widely used as benchmarks to certify the performance of algorithms by many researchers.

The DBA is coded in MATLAB 2012a, and in our simulation, numerical experiments are performed on a PC with AMD Athlon(tm) II X4 640 Processor 3.0 GHz and 2.0 GB memory. In the experiment, the termination criterion is set as $(n \times m/2) \times 30$ ms maximum computation time. Setting the time limitation in this way allows the much computation time as the job number or the machine number increases. And, this method is also adopted by many researchers, such as Jarboui et al. [28], Ruiz and Stützle [29]. Each instance is independently run 15 times for every algorithm for comparison.

The comparison method adopts BRE, ARE, and WRE to measure the quality of solution by the percentage difference from C_* ; these expressions as follows:

$$BRE = \frac{C_{\max}^{\text{best}} - C_*}{C_*} \times 100\%,$$

$$ARE = \sum_{i=1}^n \left(\frac{C_{\max}^i - C_*}{C_*} \right) \times \frac{1}{n} \times 100\%, \quad (9)$$

$$WRE = \frac{C_{\max}^{\text{worst}} - C_*}{C_*} \times 100\%,$$

where C_* is the optimal makespan or upper bound value known so far, the makespan of an obtained solution in DBA is C_{\max} , BRE represents the best relative error to C_* , ARE denotes the average relative error to C_* , and WRE represents the worst relative error to C_* . Std denotes the standard deviation of the makespan. These performance measures are employed in our experiments; these results are rounded to the nearest number which contains 2 or 3 digits after the decimal point.

4.1. *Parameter Analysis.* In the subsection, parameters of DBA are determined by experiments, and the impact of each parameter is analyzed. In DBA, parameters ps , μ are tested. ps is population size, A small ps may lead insufficient information provided, and the diversity cannot guarantee. On the other side, a large one indicates diversity is sufficient, but the computing time will increase. μ determines the size of virtual population; the large one can perform large single point neighborhood search, which may achieve a better solution, especially, the current best solution extraordinarily approximated the exact solution; however, an oversize will increase the computing time, and the precision of optimal solution may have lesser improvement. In order to evaluate the sensitivity of parameters, Car5 and Rec11 are chosen to run 15 times and the results are shown in Figures 3 and 4.

```

Begin
  Initialize the population  $ps$ ,  $t = 1$ , other parameters and bat population
  Evaluate fitness for each individual and find out  $gbest\_x$  and  $pbest\_x$ 
while (the termination condition does not satisfy)
  /* Position Updating of Bat */
  for  $i = 1 : ps$ 
    Generate frequency  $f$ ;
    Obtain velocity  $v_{i,f}^N(t)$ ;
    Determine  $v_{tmp,f}^N(t)$  by NEH method;
    Update  $x_i^N(t)$  using  $v_i^N(t)$ ;
    Evaluate fitness of individual and update  $pbest\_x$ ;
    Perform Pulse Emission Rate Local Operation; // Algorithm 2
    Evaluate fitness of individual and update  $pbest\_x$ 
    Perform Loudness Local Operation; // Algorithm 3
     $x = \text{adjustment}(x)$ ; // Algorithm 4
    Evaluate fitness of individual and update  $pbest\_x$ ;
  endfor
  Find out current global best position  $gbest\_x$ ;
  /* Intensive Virtual Population Neighborhood Search */
  for  $i = 1 : ps_1$ 
    Execute swap operation based on  $gbest\_x$ 
  endfor
  Evaluate fitness for each individual and find out  $gbest\_x$ 
  for  $i = 1 : ps_1$ 
    Execute insert operation based on  $gbest\_x$ 
  endfor
  Evaluate fitness for each individual and find out  $gbest\_x$ 
  for  $i = 1 : ps_1$ 
    Execute single-point move backward operation based on  $gbest\_x$ 
  endfor
  Evaluate fitness for each individual and find out  $gbest\_x$ 
   $t = t + 1$ ;
endwhile
  Output result and plot;
end

```

ALGORITHM 6: The DBA for PFSP.

Figures 3 and 4 represent the relative error of test case Car5 and Rec11 after 15 times independent running, which showed the sensitivity of parameters ps and μ . $\mu = 2$ when test parameter ps , and $ps = 10$ when test parameter μ . From the two test cases, for Car5, the performance is better and better while parameter ps gradually increases. But for Rec11, ps equal to 40 or 50 can achieve exact solution, but the performances do not follow the laws of Car5. In DBA, the parameter ps takes a compromise values, $ps = 50$. Similarly, parameter μ equal to 2 is optimal for Car5; however, $\mu = 3$ is optimal for Rec11. In order to balance all test cases, the parameter μ is set as 1 while $ps = 50$.

4.2. Comparisons of DBA, DBA_NEH1, and DBA-IVPNS. In order to evaluate the performance of each strategy, two

variants of DBA are compared, whose abbreviations are as follows.

- (1) DBA: DBA with NEH.
- (2) DBA_NEH1: DBA with NEH1.
- (3) DBA-IVPNS: DBA without intensive virtual population neighborhood search.

At this group experiment, the parameter setting is $ps = 10$, $\mu = 2$, termination criterion is set as $(n \times m/2) \times 10$ ms maximum computation time, and the algorithm is run 15 times independently. The statistical performances of DBA, DBA_NEH1, and DBA-IVPNS are shown in Table 1.

From Table 1, we can find out that the average performance of DBA is better than the other two variants of DBA; for benchmarks Car1 to Car8, the DBA-IVPNS is better; the

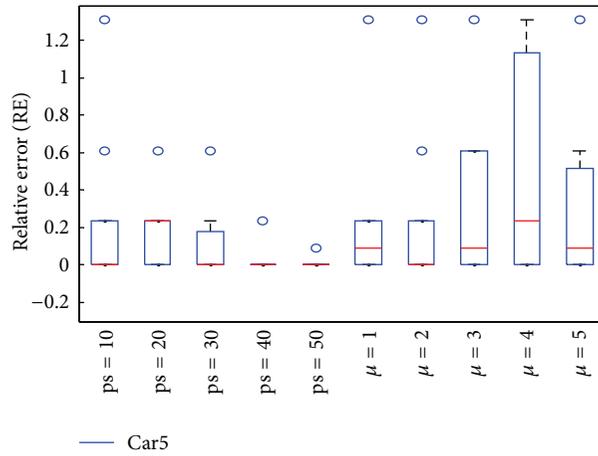


FIGURE 3: Box-and-whisker diagram of Car5.

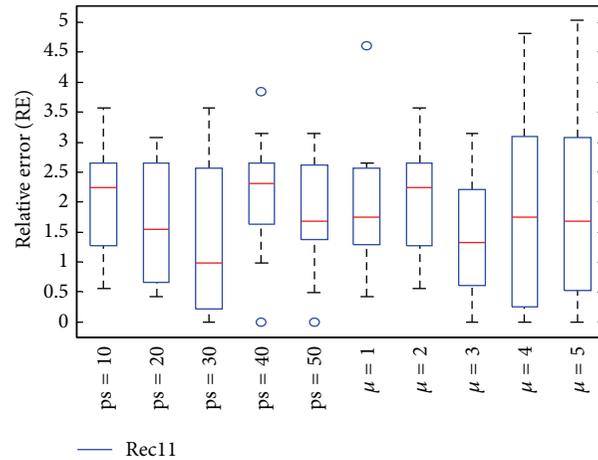


FIGURE 4: Box-and-whisker diagram of Rec11.

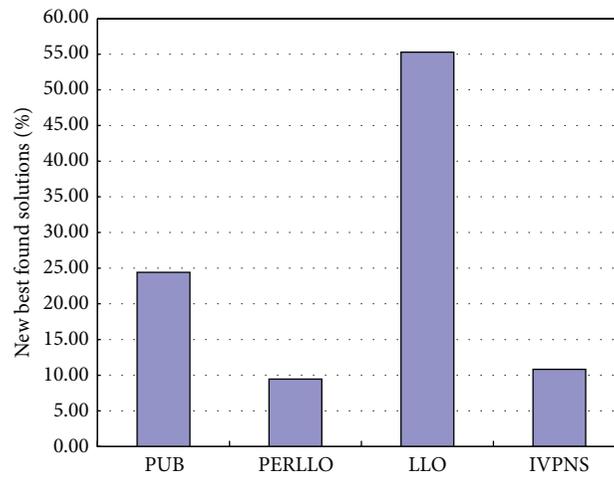


FIGURE 5: The contribution of each strategy move to finding a new best solution.

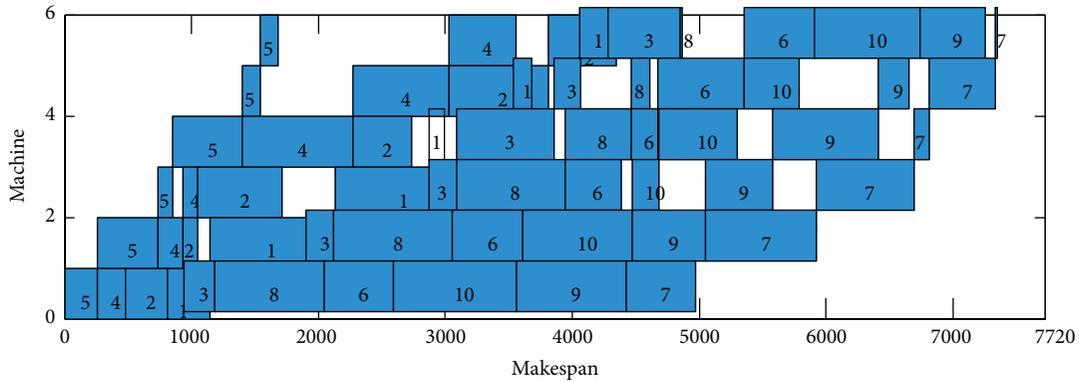


FIGURE 6: Gantt chart of an optimal schedule for Car05, $\pi_* = [5, 4, 2, 1, 3, 8, 6, 10, 9, 7]$.

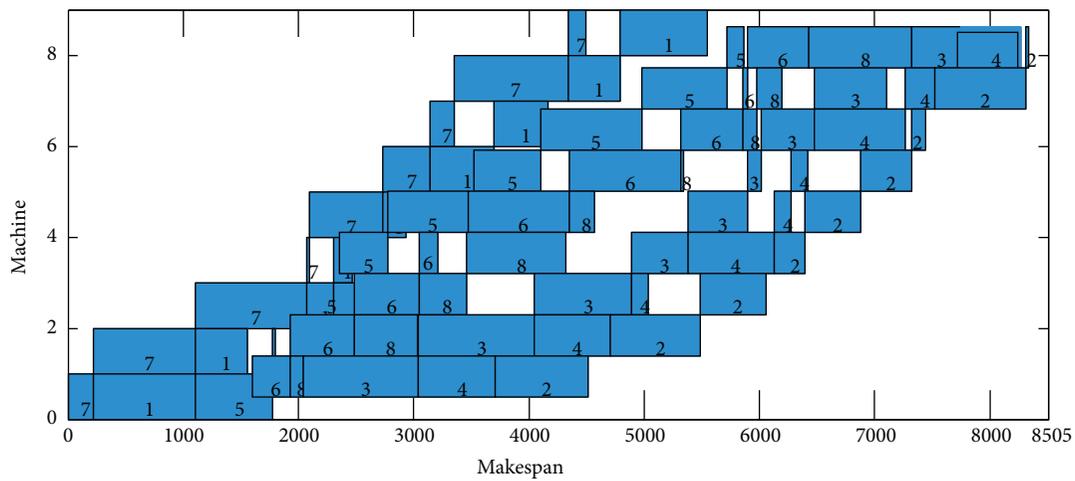


FIGURE 7: Gantt chart of an optimal schedule for Car06, $\pi_* = [7, 1, 5, 6, 8, 3, 4, 2]$.

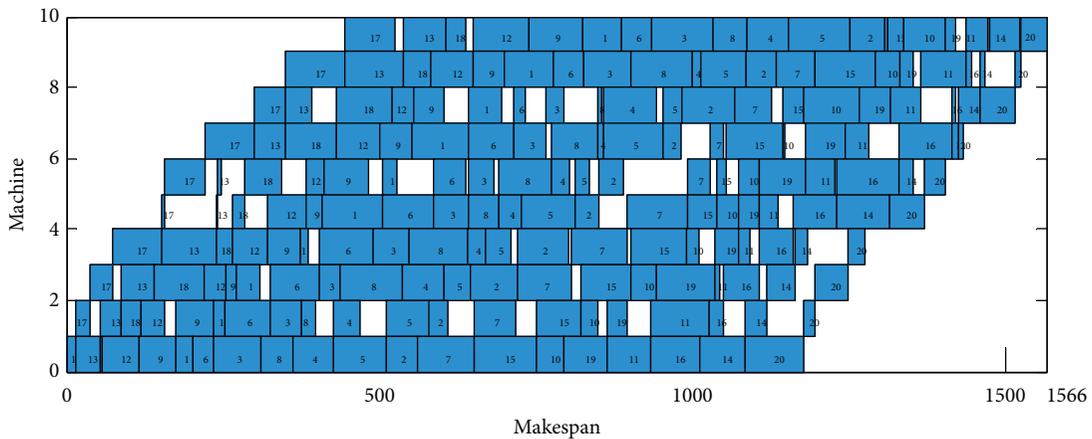


FIGURE 8: Gantt chart of an optimal schedule for Rec7, $\pi_* = [17, 13, 18, 12, 9, 1, 6, 3, 8, 4, 5, 2, 7, 15, 10, 19, 11, 16, 14, 20]$.

reason may be that the IVPNS implementation is single-point operation on the current global best individual $gbest_{x}$; this operation may improve the quality of solution, but this needs much computing time, so the DBA-IVPNS have more time to explore of more new position. However, from Rec1 to Rec41, the DBA is much better than other variants.

For DBA_NEH1, only it has a difference that the position updating of bat by NEH1. The NEH1 has lesser computational complexity than NEH. From experiment results, we can find out that DBA_NEH1 can find better solutions for several benchmarks. In general, the DBA is better than DBA_NEH1 for all benchmarks.

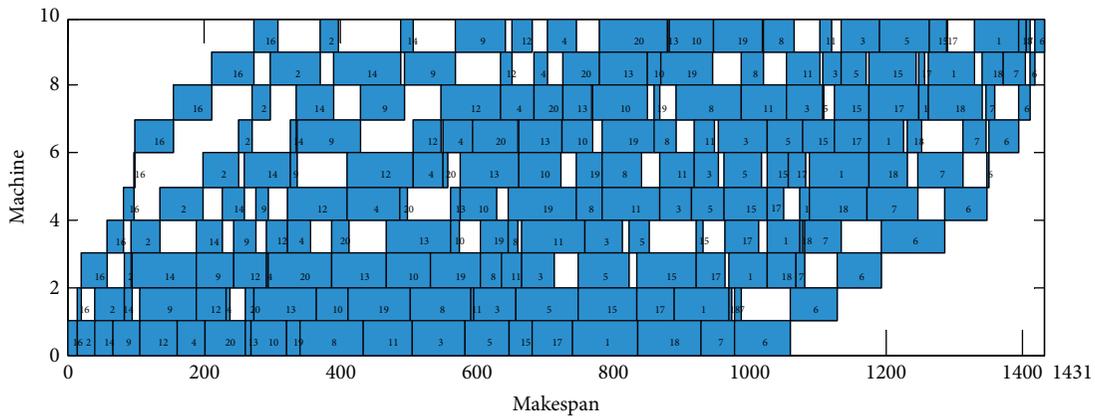


FIGURE 9: Gantt chart of an optimal schedule for Rec11, $\pi_* = [16, 2, 14, 9, 12, 4, 20, 13, 10, 19, 8, 11, 3, 5, 15, 17, 1, 18, 7, 6]$.

TABLE 1: Statistical performances of DBA, DBA_NEHI, and DBA-IVPNS.

Problem	$n m$	C^*	DBA				DBA_NEHI				DBA-IVPNS			
			BRE	ARE	WRE	Std	BRE	ARE	WRE	Std	BRE	ARE	WRE	Std
Car1	11 5	7038	0	0	0	0	0	0	0	0	0	0	0	0
Car2	13 4	7166	0	0.195	2.931	54.22	0	0.391	2.931	73.89	0	0	0	0
Car3	12 5	7312	0	0.476	1.190	44.12	0	0.635	1.190	44.93	0	0.397	1.190	42.45
Car4	14 4	8003	0	0	0	0	0	0	0	0	0	0	0	0
Car5	10 6	7720	0	0.246	1.308	35.70	0	0.664	1.360	45.95	0	0.352	1.308	40.88
Car6	8 9	8505	0	0	0	0	0	0	0	0	0	0	0	0
Car7	7 7	6590	0	0	0	0	0	0	0	0	0	0	0	0
Car8	8 8	8366	0	0	0	0	0	0	0	0	0	0	0	0
Rec1	20 5	1247	0.160	0.209	0.722	1.84	0.160	0.241	1.043	2.85	0.160	0.545	1.925	7.92
Rec3	20 5	1109	0.090	0.481	2.164	7.09	0	0.499	1.803	6.09	0.180	0.385	1.713	4.51
Rec5	20 5	1242	0.242	0.623	2.174	9.49	0.242	0.768	2.496	10.72	0.242	1.100	2.496	10.69
Rec7	20 10	1566	1.149	1.443	3.831	11.41	1.149	2.048	3.831	18.33	1.149	1.537	3.831	13.54
Rec9	20 20	1537	0	2.420	3.709	13.44	0	2.065	3.318	15.98	1.041	2.728	4.815	13.13
Rec11	20 10	1431	0.559	1.975	3.564	13.56	0	2.241	7.617	29.69	0	1.859	4.403	16.64
Rec13	20 15	1930	0.415	2.394	3.938	19.84	0.933	2.525	4.819	19.30	1.762	2.694	4.352	16.29
Rec15	20 15	1950	0.154	2.178	4.615	23.43	0.821	2.410	4.615	24.23	1.231	2.903	4.256	19.21
Rec17	20 15	1902	0.946	2.685	4.206	19.27	0.894	3.582	5.941	25.20	1.577	5.065	6.730	25.65
Rec19	30 10	2093	0.573	2.621	4.252	21.38	1.386	2.599	4.730	19.58	2.484	3.883	5.542	20.28
Rec21	30 10	2017	1.438	2.310	4.412	19.89	1.636	2.568	5.702	24.05	1.785	3.543	5.255	19.50
Rec23	30 10	2011	0.945	3.216	5.868	23.88	1.591	3.090	4.923	19.22	3.282	4.422	6.266	18.48
Rec25	30 15	2513	2.348	3.520	5.213	20.71	1.870	3.489	5.133	23.77	3.780	5.428	6.805	19.57
Rec27	30 15	2373	2.402	3.638	5.057	19.03	1.728	3.217	5.900	23.66	2.023	4.374	5.942	23.93
Rec29	30 15	2287	1.530	4.323	7.084	33.64	2.186	3.615	5.597	24.60	4.766	6.046	7.521	19.20
Rec31	50 10	3045	3.284	4.917	6.502	30.44	3.153	4.926	6.765	38.21	5.353	6.192	7.783	21.95
Rec33	50 10	3114	0.835	1.916	4.143	29.20	1.317	2.338	4.528	26.65	1.927	2.899	4.689	25.99
Rec35	50 10	3277	0.092	0.484	2.014	18.73	0.092	1.082	3.021	36.89	0.244	1.107	2.563	20.99
Rec37	75 20	4951	5.615	7.172	8.140	39.66	5.918	7.387	8.826	37.75	8.503	9.156	10.261	21.68
Rec39	75 20	5087	3.696	5.578	6.408	41.52	4.914	6.083	7.529	35.22	6.979	7.629	8.374	23.50
Rec41	75 20	4960	6.129	7.435	8.952	33.55	6.573	7.589	8.952	29.22	8.105	9.319	10.726	37.80
Average			1.124	2.154	3.531	20.17	1.261	2.278	3.882	22.62	1.951	2.881	4.095	16.68

TABLE 2: Statistical performances of DBA, PSOMA, PSOVNS, and OSA.

Problem	C_{max}	DBA				PSOVNS			PSOMA			OSA		
		BRE	ARE	WRE	Std	BRE	ARE	WRE	BRE	ARE	WRE	BRE	ARE	Std
Car1	7038	0	0	0	0	0	0	0	0	0	0	0	0	0
Car2	7166	0	0	0	0	0	0	0	0	0	0	0	0	0
Car3	7312	0	0.397	1.190	42.45	0	0.420	1.189	0	0	0	0	0.625	47.19
Car4	8003	0	0	0	0	0	0	0	0	0	0	0	0	0
Car5	7720	0	0	0	0	0	0.039	0.389	0	0.018	0.375	0	0.801	50.73
Car6	8505	0	0	0	0	0	0.076	0.764	0	0.114	0.764	0	2.093	274.71
Car7	6590	0	0	0	0	0	0	0	0	0	0	0	1.483	114.21
Car8	8366	0	0	0	0	0	0	0	0	0	0	0	2.297	254.63
Rec1	1247	0	0.080	0.160	0.85	0.160	0.168	0.321	0	0.144	0.160	0.160	0.160	0
Rec3	1109	0	0.081	0.180	0.88	0	0.158	0.180	0	0.189	0.721	0	0.189	1.85
Rec5	1245	0.242	0.242	0.242	0	0.242	0.249	0.420	0.242	0.249	0.402	0.242	0.588	4.62
Rec7	1566	0	0.575	1.149	9.40	0.702	1.095	1.405	0	0.986	1.149	0	0.434	11.59
Rec9	1537	0	0.638	2.407	15.00	0	0.651	1.366	0	0.621	1.691	0	0.690	12.39
Rec11	1431	0	1.167	2.655	11.17	0.071	1.153	2.656	0	0.129	0.978	0	2.215	37.60
Rec13	1938	0.415	1.461	3.782	19.01	1.036	1.790	2.643	0.259	0.893	1.502	0.311	1.793	14.69
Rec15	1953	0.154	1.226	2.103	7.97	0.769	1.487	2.256	0.051	0.628	1.076	0.718	1.569	16.07
Rec17	1909	0.368	1.277	2.154	41.65	0.999	2.453	3.365	0	1.330	2.155	1.840	3.796	36.72
Rec19	2105	0.573	0.929	2.023	33.06	1.529	2.099	2.532	0.430	1.313	2.102	0.287	0.803	9.48
Rec21	2046	1.438	1.671	2.231	4.04	1.487	1.671	2.033	1.437	1.596	1.636	1.438	1.477	1.69
Rec23	2027	0.796	1.173	2.381	39.27	1.343	2.106	2.884	0.596	1.310	2.038	0.497	0.854	10.82
Rec25	2554	1.632	2.921	3.940	18.96	2.388	3.166	3.780	0.835	2.085	3.233	1.194	1.938	15.06
Rec27	2397	1.011	1.419	2.298	21.35	1.728	2.463	3.203	1.348	1.605	2.402	0.843	1.845	21.06
Rec29	2311	1.049	2.580	3.935	22.84	1.968	3.109	4.067	1.442	1.888	2.492	0.612	2.882	38.83
Rec31	3115	2.299	3.392	4.532	23.66	2.594	3.232	4.237	1.510	2.254	2.692	0.296	1.333	30.39
Rec33	3133	0.610	0.728	1.734	39.40	0.835	1.007	1.477	0	0.645	0.834	0.128	0.732	7.32
Rec35	3277	0	0.037	0.092	1.52	0	0.038	0.092	0	0	0	0	0	0
Rec37	5118	3.373	4.872	5.979	40.31	4.383	4.949	5.736	2.101	3.537	4.039	2.000	2.751	25.43
Rec39	5203	2.280	3.851	5.347	45.97	2.850	3.371	5.585	1.553	2.426	2.830	0.767	1.240	12.31
Rec41	5149	3.810	5.095	6.532	42.89	4.173	4.867	5.585	2.641	3.684	4.052	1.734	2.726	39.38

TABLE 3: Optimal job permutations of DBA.

Problem	$n m$	C^*	π_*
Car1	11 5	7038	8, 1, 3, 11, 5, 9, 4, 10, 7, 2, 6
Car2	13 4	7166	7, 3, 4, 11, 9, 1, 8, 12, 5, 2, 13, 10, 6
Car3	12 5	7312	11, 6, 5, 10, 12, 9, 3, 2, 4, 7, 8, 1
Car4	14 4	8003	4, 12, 13, 14, 5, 7, 6, 1, 9, 10, 11, 8, 2, 3
Car5	10 6	7720	5, 4, 2, 1, 3, 8, 6, 10, 9, 7
Car6	8 9	8505	7, 1, 5, 6, 8, 3, 4, 2
Car7	7 7	6590	5, 4, 2, 6, 7, 3, 1
Car8	8 8	8366	7, 3, 8, 5, 2, 1, 6, 4
Rec1	20 5	1247	6, 9, 2, 20, 12, 14, 17, 15, 13, 7, 1, 18, 3, 4, 11, 5, 8, 10, 19, 16
Rec3	20 5	1109	6, 14, 7, 1, 2, 3, 11, 8, 9, 17, 15, 5, 19, 4, 16, 10, 12, 13, 18, 20
Rec7	20 10	1566	17, 13, 18, 12, 9, 1, 6, 3, 8, 4, 5, 2, 7, 15, 10, 19, 11, 16, 14, 20
Rec9	20 20	1537	4, 19, 17, 12, 18, 14, 7, 16, 5, 13, 2, 10, 9, 11, 8, 20, 1, 15, 3, 6
Rec11	20 10	1431	16, 2, 14, 9, 12, 4, 20, 13, 10, 19, 8, 11, 3, 5, 15, 17, 1, 18, 7, 6
Rec35	50 10	3277	13, 14, 40, 39, 50, 36, 46, 35, 37, 26, 2, 18, 19, 8, 41, 10, 25, 20, 38, 29, 33, 15, 27, 9, 21, 17, 42, 22, 32, 3, 1, 23, 4, 12, 5, 49, 11, 45, 43, 16, 34, 6, 44, 30, 7, 48, 47, 28, 24, 31

TABLE 4: The statistical results of score.

Benchmark	DBA				PSOVNS			PSOMA			SGA + NEH		OSA		
	BRE	ARE	WRE	Std	BRE	ARE	WRE	BRE	ARE	WRE	BRE	ARE	BRE	ARE	Std
Car1-Car8	32	31	30	32	32	27	28	32	29	30	30	16	32	19	27
Rec1-Rec41	60	58	62	70	40	37	56	73	66	78	20	4	73	57	77
Car1-Rec29	78	78	78	83	63	54	67	85	75	84	47	19	82	54	81
Car1-Rec41	92	89	92	102	72	64	84	105	95	108	50	20	105	76	104

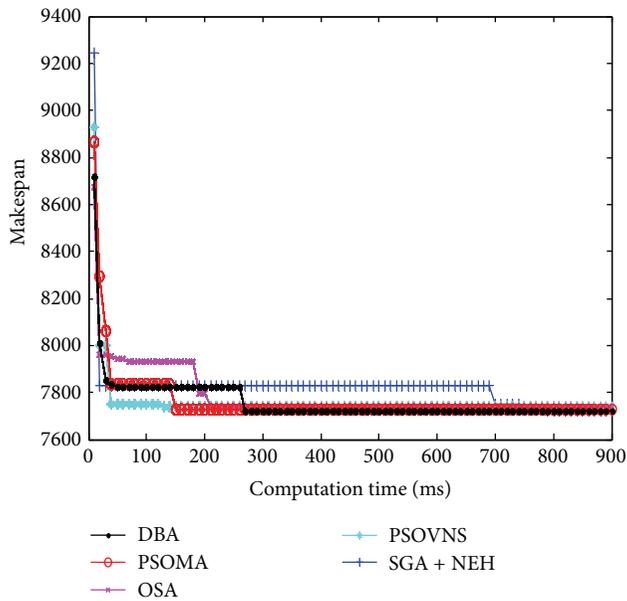


FIGURE 10: The convergence curves of Car5.

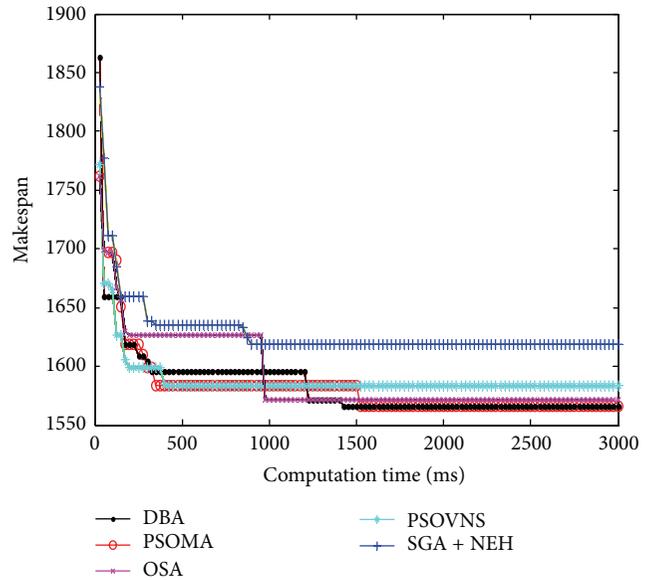


FIGURE 12: The convergence curves of Rec7.

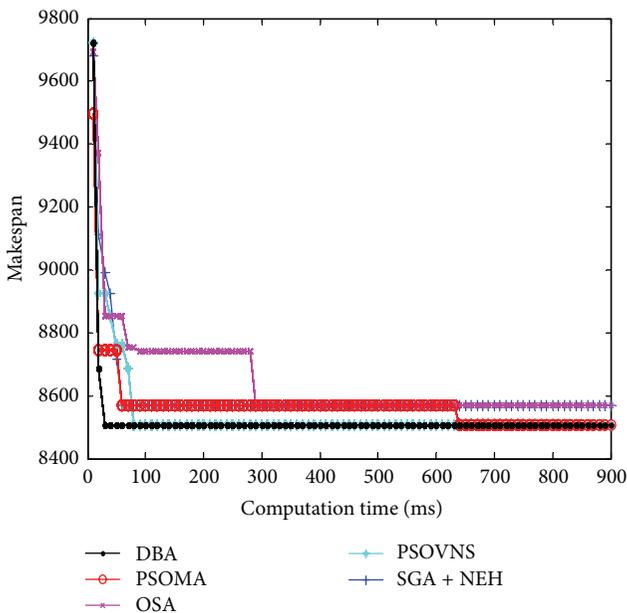


FIGURE 11: The convergence curves of Car6.

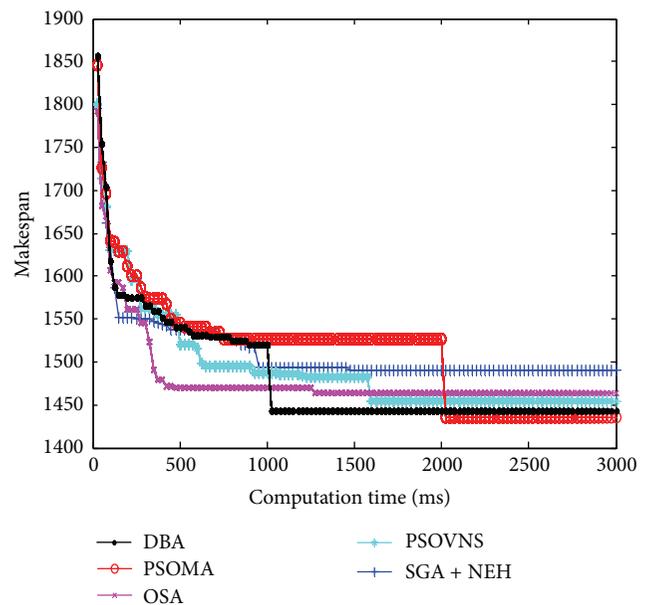


FIGURE 13: The convergence curves of Rec11.

In addition, in order to demonstrate the effect of each strategy in the specific scheduling problem, the frequency of finding a new best solution by applying these moves in DBA is recorded; it can show the contribution of each strategy. The Car1 to Car8 and Rec1 to Rec15 16 benchmarks are chosen to be tested. Each problem was run 10 times; each time a new best solution was found by the algorithm; the move resulting in this improvement was recorded. Figure 5 demonstrates the percentage of contribution.

4.3. Comparisons of DBA, PSOMA, PSOVSNS, OSA. In order to show the effectiveness of DBA, we carry out a simulation to compare our DBA with other state-of-art algorithms, that is, PSOMA proposed by Liu et al. [11], PSOVSNS proposed by Tasgetiren et al., and experimental results reference [5], and SA is a simulated annealing, the experimental results reference [16]. The population size is 50 and the termination criterion is set as $(n \times m/2) \times 30$ ms maximum computation time. The experimental results are listed in Table 2.

From Table 2, for the Car problems, the DBA, PSOVSNS, PSOMA, and OSA all can find the exact solution, and DBA is better than the other algorithm on ARE. For the Rec problems, DBA also can find better solutions. Compared with DBA, PSOVSNS, PSOMA, and OSA, the DBA achieved 14 exact solutions; several optimal job permutations are shown in Table 3. PSOVSNS achieved 11 exact solutions, PSOMA achieved 16 exact solutions, and OSA achieved 13 exact solutions. For all test problems, obtained solutions of DBA are not better than the PSOMA and OSA, but the performance is similar to PSOMA and OSA.

In order to compare each norm (BRE, ARE, WRE, and Std) of corresponding algorithms, for all benchmarks, each norm is scored among corresponding algorithms. The first is score 4, the second is score 3, the third is score 2, the fourth is score 1, and the last is score 0, if several results are same, they have same score. The statistical results are listed in Table 4. From Table 4, for Car problems, the DBA is best on ARE, the DBA and PSOMA are identical on WRE, DBA has better Std compared with OSA. For Rec problems, the OSA and PSOMA have better BRE, the DBA is better than PSOVSNS, the DBA is better than PSOVSNS, OSA, the DBA is also better than PSOVSNS on WRE among DBA, PSOVSNS, and PSOMA, but the Std is not better than OSA. The DBA is best on ARE for Car1 to Rec29 among DBA, PSOVSNS, PSOMA, and OSA, and the Std is better than OSA. On the whole, the achieved solutions of DBA have better quality. For large-scale scheduling problems, the DBA still have the room for improvement; it also is our further work.

The DBA achieved 14 exact solutions, due to the fact that Rec35 have 10 machines and 50 jobs, the margin of paper is restricted, the Gantt chart of an optimal schedule for Rec35 cannot display on this paper, and the Gantt chart of Car5, Car6, Rec7, and Rec11 is selected as instance. These Gantt charts of an optimal schedule are shown in Figures 6, 7, 8, and 9.

Figures 10, 11, 12, and 13 show the convergence curves of Car5, Car6 Rec7, and Rec11. From Figures 10 to 13, the convergence rate of DBA is fast, and the precision of solution

is prominent. The performance of DBA is similar to PSOMA; however, the convergence rate of DBA is faster than PSOMA in the early phase of iteration. The precision of solution is not as good as PSOMA while the scale of scheduling problems is increasing. The DBA is better than SGA + NEH [5], PSOVSNS, and OSA in some aspects.

5. Conclusions

In this paper, we construct a direct relationship between the job sequence and the vector of individuals in DBA; a DBA is proposed to solve PFSP. In order to evaluate the performance of the proposed DBA, we compare DBA with several PFSP algorithms with benchmark problems of PFSP. Experimental results have shown that our algorithm is pretty effective, the performance of each strategy is evaluated, and sensitivity of parameters is analyzed. Moreover, our further work is to study the theoretical aspects as well as the performance of the technique.

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 National Science Foundation of China under Grant no. 61165015, Key Project of Guangxi Science Foundation under Grant no. 2012GXNSFDA053028, and Key Project of Guangxi High School Science Foundation under Grant nos. 20121ZD008, 201203YB072.

References

- [1] H. Stadler, "Supply chain management and advanced planning—basics, overview and challenges," *European Journal of Operational Research*, vol. 163, no. 3, pp. 575–588, 2005.
- [2] K. A. Rinnooy, *Machine Scheduling Problems: Classification, Complexity, and Computations*, Nijhoff, The Hague, The Netherlands, 1976.
- [3] F. Della Croce, M. Ghirardi, and R. Tadei, "An improved branch-and-bound algorithm for the two machine total completion time flow shop problem," *European Journal of Operational Research*, vol. 139, no. 2, pp. 293–301, 2002.
- [4] E. F. Stafford, "On the development of a mixed integer linear programming model for the flowshop sequencing problem," *Journal of the Operational Research Society*, vol. 39, pp. 1163–1174, 1988.
- [5] L. Wang and B. Liu, *Particle Swarm Optimization and Scheduling Algorithms*, Tsinghua University Press, Beijing, China, 2008.
- [6] L. Wang and D. Z. Zheng, "An effective hybrid heuristic for flow shop scheduling," *International Journal of Advanced Manufacturing Technology*, vol. 21, no. 1, pp. 38–44, 2003.
- [7] J. J. Liang, Q. Pan, C. Tiejun, and L. Wang, "Solving the blocking flow shop scheduling problem by a dynamic multi-swarm particle swarm optimizer," *International Journal of Advanced Manufacturing Technology*, vol. 55, no. 5–8, pp. 755–762, 2011.
- [8] M. F. Tasgetiren, P. N. Suganthan, and Q. K. Pan, "An ensemble of discrete differential evolution algorithms for solving the

- generalized traveling salesman problem," *Applied Mathematics and Computation*, vol. 215, no. 9, pp. 3356–3368, 2010.
- [9] X. S. Yang, "A new metaheuristic bat-inspired algorithm. Nature Inspired Cooperative Strategies for Optimization (NICSO)," *Studies in Computational Intelligence*, vol. 284, pp. B65–B74, 2010.
- [10] M. F. Tasgetiren, Y. C. Liang, M. Sevkli, and G. A. Gencyilmaz, "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1930–1947, 2007.
- [11] B. Liu, L. Wang, and Y. Jin, "An effective PSO-based memetic algorithm for flow shop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 37, no. 1, pp. 18–27, 2007.
- [12] L. Tseng and Y. Lin, "A hybrid genetic local search algorithm for the permutation flowshop scheduling problem," *European Journal of Operational Research*, vol. 198, no. 1, pp. 84–92, 2009.
- [13] B. Yagmahan and M. M. Yenisey, "A multi-objective ant colony system algorithm for flow shop scheduling problem," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1361–1368, 2010.
- [14] M. F. Tasgetiren, Q. Pan, P. N. Suganthan, and A. H.-L. Chen, "A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops," *Information Sciences*, vol. 181, no. 16, pp. 3459–3475, 2011.
- [15] F. Ahmadizar, "A new ant colony algorithm for makespan minimization in permutation flow shops," *Computers & Industrial Engineering*, vol. 63, no. 2, pp. 355–361, 2012.
- [16] X. Li and M. Yin, "An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure," *Advances in Engineering Software*, vol. 55, pp. 10–31, 2013.
- [17] R. Liu, C. Ma, W. Ma, and Y. Li, "A multipopulation PSO based memetic algorithm for permutation flow shop scheduling," *The Scientific World Journal*, vol. 2013, Article ID 387194, 11 pages, 2013.
- [18] M. Mirabi, "A novel hybrid genetic algorithm to solve the sequence-dependent permutation flow-shop scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 71, pp. 429–437, 2014.
- [19] F.-V. Victor and J. M. Framinan, "On insertion tie-breaking rules in heuristics for the permutation flowshop scheduling problem," *Computers & Operations Research*, vol. 45, pp. 60–67, 2014.
- [20] A. H. Gandomi, X. S. Yang, A. H. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," *Neural Computing and Applications*, vol. 22, no. 6, pp. 1239–1255, 2013.
- [21] X. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.
- [22] S. Mishra, K. Shaw, and D. Mishra, "A new meta-heuristic bat inspired classification approach for microarray data," *Procedia Technology*, vol. 4, pp. 802–806, 2012.
- [23] J. Xie, Y. Zhou, and H. Chen, "A novel bat algorithm based on differential operator and Lévy flights trajectory," *Computational Intelligence and Neuroscience*, vol. 2013, Article ID 453812, 13 pages, 2013.
- [24] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, "A bat algorithm with mutation for UCAV path planning," *The Scientific World Journal*, vol. 2012, Article ID 418946, 15 pages, 2012.
- [25] M. Nawaz, E. E. Ensore Jr., and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91–95, 1983.
- [26] J. Carlier, "Ordonnancements à contraintes disjonctives," *RAIRO Recherche Opérationnelle*, vol. 12, pp. 333–351, 1978.
- [27] C. R. Reeves and T. Yamada, "Genetic algorithms, path relinking, and the flowshop sequencing problem," *Evolutionary Computation*, vol. 6, no. 1, pp. 45–60, 1998.
- [28] B. Jarbouli, M. Eddaly, and P. Siarry, "An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems," *Computers and Operations Research*, vol. 36, no. 9, pp. 2638–2646, 2009.
- [29] R. Ruiz and T. Stützle, "An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives," *European Journal of Operational Research*, vol. 187, no. 3, pp. 1143–1159, 2008.

Research Article

Feature Selection and Classifier Parameters Estimation for EEG Signals Peak Detection Using Particle Swarm Optimization

Asrul Adam,¹ Mohd Ibrahim Shapiai,² Mohd Zaidi Mohd Tumari,³
Mohd Saberi Mohamad,⁴ and Marizan Mubin¹

¹ Applied Control and Robotics (ACR) Laboratory, Department of Electrical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia

² Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia

³ Faculty of Electrical and Electronic Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia

⁴ Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia

Correspondence should be addressed to Asrul Adam; asrul.adam@siswa.um.edu.my

Received 18 June 2014; Accepted 30 July 2014; Published 19 August 2014

Academic Editor: Shifei Ding

Copyright © 2014 Asrul Adam et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Electroencephalogram (EEG) signal peak detection is widely used in clinical applications. The peak point can be detected using several approaches, including time, frequency, time-frequency, and nonlinear domains depending on various peak features from several models. However, there is no study that provides the importance of every peak feature in contributing to a good and generalized model. In this study, feature selection and classifier parameters estimation based on particle swarm optimization (PSO) are proposed as a framework for peak detection on EEG signals in time domain analysis. Two versions of PSO are used in the study: (1) standard PSO and (2) random asynchronous particle swarm optimization (RA-PSO). The proposed framework tries to find the best combination of all the available features that offers good peak detection and a high classification rate from the results in the conducted experiments. The evaluation results indicate that the accuracy of the peak detection can be improved up to 99.90% and 98.59% for training and testing, respectively, as compared to the framework without feature selection adaptation. Additionally, the proposed framework based on RA-PSO offers a better and reliable classification rate as compared to standard PSO as it produces low variance model.

1. Introduction

The peak detection algorithms have significantly been used on different types of biological signals such as electrooculogram (EOG), electrocardiogram (ECG), and electroencephalogram (EEG). EOG signal is generated by human eye. ECG signal is generated by heart. EEG signal is generated by brain. The peak detection in the EOG signal has been used for detecting the eye blink [1, 2]. In the EOG based signal, a number of electrodes are placed around the eyes. If the eyes move in vertical direction, positive or negative peak points will arise. For the ECG signal, peak detection is typically used to detect the combination of Q, R, and S waves or the so-called QRS complex [3]. The QRS complex is a peak model for ECG signal including Q-valley point, R-peak point, and S-valley

point. Other important peak points in ECG signal are P-peak point and T-peak point. The detection of the QRS complex is critical part in numerous ECG signal processing system. The different pattern of QRS complex will determine the patient heart syndrome. Additionally, the peak detection for the EEG signal has been widely used to detect P300 response [4, 5] and epilepsy response [6]. P300 is a brain response measured by electrodes covering the parietal lobe in the presence of visual and auditory stimuli. A brain with chronic disorder will respond with epilepsy. Therefore, the utilization of peak detection algorithm for the biological signals is compatible in this study.

To date, variety approaches of peak detection algorithms have been proposed. These algorithms can be categorized into four main approaches based on time domain [7–15],

frequency domain [16], time-frequency domain [10, 17], and nonlinear [18]. In time domain approach, the peaks are analyzed in time. In frequency domain approach, the peaks are analyzed in frequency. In time-frequency domain approach, the peaks are analyzed in both time and frequency domain. In nonlinear approach, some statistical parameters of the peaks are analyzed. The general framework of peak detection algorithm usually involves several processes which are signal preprocessing, peak candidate detection, feature extraction, and classification. Various signal preprocessing methods have been employed such as data compression [19], wavelet transform [6], Kalman filter [20], and Hilbert transform [15]. Two methods for peak candidate detection have been used which are three point sliding window method [8] and k-point nonlinear energy operator (k-NEO) method [21]. Various feature extraction techniques have been proposed which are model-based [21], wavelet analysis [22], template matching [23], and power spectra analysis [24]. Several classifiers have been used, which are rule-based [8, 24], artificial neural network (ANN) [10, 11, 25, 26], support vector machine (SVM) [7, 27], and expert system [10]. The highlighted purposes in designing the framework are to achieve the highest performance and to reduce the computational time. Almost all studies in the EEG peak detection literature focus on the problem of detecting peaks in epileptic EEG signals. A review of peak detection algorithms that is employed to the epileptic EEG signal is presented in [28]. The peak detection is just a first step in epileptic event detection. The main goal is to determine the epileptic spikes not the whole peaks. Therefore, for an epileptic event detection system, the epileptic spike detection performance not the peak detection performance is the performance of interest.

In time domain approach, fourteen different peak features are recognized from different peak models [7–10]. The peak model is a set of peak features that represents a peak by its amplitude, width, and slope. Most algorithms [7–13, 21] in time domain approach consider different peak models and the different styles of framework. The peak model is chosen based on the experiences of EEG expert. To date, there is no any peak detection framework that automatically finds the finest existing peak model. The use of the finest peak model will give a chance for the algorithm to achieve a good performance. On the other hand, the chosen peak model is not necessarily suitable for different types of biological signal. Moreover, the finest peak model represents some meaningful information on the signal to be evaluated. Therefore, the adaptation of feature selection technique is important in this study to automatically find the finest peak model. The utilization of feature selection on peak detection algorithm will also reduce the computational time.

In this study, feature selection and classifier parameters estimation method based on standard particle swarm optimization (PSO) and random asynchronous PSO (RA-PSO) algorithm are employed. The process to find the finest peak model and classifier parameter estimation is executed simultaneously. The peak features will be evaluated by a rule-based classifier. The role of the classifier is to distinguish between peak point and non-peak point. Rule-based classifier is employed due to the ability to provide an outstanding

interpretation for the obtained decisions [24]. In addition, the parameter values are tricky to be estimated manually. A PSO algorithm is considered to be appropriate for addressing the problem based on the reason in which the feature selection is a binary search problem and determination of classifier parameter is a continuous search problem [29].

1.1. Peak Model in Time Domain Analysis. Peak model is a set of peak features that represents a peak by its amplitude, width, and slope. In time domain analysis, fourteen different peak features are recognized from different peak models [8–10]. The earliest peak model was introduced by Dumpala et al. in 1982 [8]. The peak model comprises four features, which are (1) the amplitude between the magnitude of peak point and the magnitude of valley point at the first half wave, (2) the width between valley point of first half point and valley point at second half wave, (3) and (4) two slopes between a peak point and valley point in the first half wave and second half wave. A similar definition of the peak amplitude and slopes are also been used in [7, 11, 13].

An additional feature of peak amplitude and two features of peak width have been introduced by Acir et al. [7, 11]. The additional peak amplitude is the amplitude between the magnitude of peak point and the magnitude of valley point of the second half wave. The peak widths are the width between peak point and valley point of first half wave and second half wave. The total features that are introduced by Acir et al. are six features. Acir et al. did not use the width feature that was introduced by Dumpala et al. A similar definition of the peak amplitudes, widths, and slopes has also been used in [21]. In [21], an additional peak feature is added with a set of features that is introduced in [7, 11], which is the area of peak. However, the definition of area integration is not presented in the paper.

In addition, Liu et al. [10] have introduced eleven peak features. The proposed peak model consists of four amplitudes: (1) the amplitude between the magnitude of peak point and the magnitude of valley point at the first half wave; (2) the amplitude between the magnitude of peak point and the magnitude of valley point of the second half wave; (3) the amplitude between the magnitude of peak and the magnitude of turning point at the first half wave, and (4) the amplitude between the magnitude of peak and the magnitude of turning point at the second half wave. The turning point is defined as the point where the slope decreases more than 50% as compared to the slope of the preceding point. The model also consists of three widths: (1) the width between valley point at first half point and valley point at second half wave, (2) the width between turning point at first half wave and turning point at second half wave, and (3) the width between half point at first half wave and half point at second half wave. There are four slopes that are also measured: (1) and (2) two slopes between a peak point and valley point in the first half wave and second half wave, (3) and (4) two slopes between peak point and turning point at first half wave and second half wave.

Another peak model consists of four features, which has been proposed by Dingle et al. [9]. The peak amplitude is

TABLE 1: Summary of different peak models on different style of framework.

Peak model	Type of signal	Description of framework
Dumpala et al. (1982) [8]	Electrical control activity (ECA)	The theory of maxima and minima using three-point sliding window approach has been applied to detect a candidate peak. Two flowcharts of peak detection have been proposed. A predicted peak can be identified if the feature values satisfied the decision threshold values. The strength and weakness of the proposed approach are described as follows: (1) strength: the authors claimed that the proposed peak detection algorithm can be used for other biological signals, (2) weakness: the utilization of peak-to-peak amplitude on the peak model is hard to distinguish between noise and actual peak. In addition, large variation of peak width in the signal may drop the classification performance.
Dingle et al. (1993) [9]	Epileptic EEG	Based on the defined peak model, the features are grouped into two: (1) epileptiform transient parameters and (2) background activity parameters. Two-threshold systems have been employed to detect a candidate peak or candidate epileptiform transient. Expert system which considered both spatial and temporal contextual information has been used to reject the artifacts and classify the transient events. The strength and weakness of the proposed approach are described as follows: (1) strength: moving average amplitude is good in rejecting false peak points. The employed features are claimed to offer good performance in the proposed expert system, (2) weakness: inconsistency of feature slope information as the proposed work claimed that the proposed framework fails to provide slope information.
Liu et al. (2002) [10]	Epileptic EEG	Wavelet transform has been used to decompose the EEG signal. Based on the decomposed signals and the defined peak model, seven features are calculated. These features are used as the input of ANN classifier. Expert system which considered both spatial and temporal contextual information has been used to reject the artifact. Several heuristic rules have been employed to distinguish the type of artifact. After all artifacts are recognized and rejected, the decision will be made to classify the epileptic events. The strength and weakness of the proposed approach are described as follows: (1) strength: the employed features is claimed to offer good performance in the proposed expert system, (2) weakness: it considers that almost all the features may deteriorate the classification performance.
Acir et al. (2005) [11]	Epileptic EEG	A three-stage procedure based on ANN is proposed for the detection of epileptic spikes. The EEG signal is transformed into time-derivative signal. Several rules have been used to detect a peak candidate. The features of peak candidate are calculated based on the defined peak model. These features are fed into two discrete perceptron classifiers to classify into three groups: definite peak, definite non-peak, and possible/possible non-peak. The peak that belongs in the third group is going to be further processed by nonlinear classifier. The strength and weakness of the proposed approach are described as follows: (1) strength: the employed features are claimed to offer good performance in the proposed system, (2) weakness: inconsistency of feature slope information as the proposed work claimed that the proposed framework fails to provide slope information.
Acir (2005) [26]	Epileptic EEG	A two-stage procedure based on a modified radial basis function network (RBFN) is proposed for the detection of epileptic spikes. The EEG signal is transform into time-derivative signal. Several rules have been used to detect a peak candidate. The features of peak candidate are calculated based on the defined peak model. These features are fed into discrete perceptron classifiers to classify into two groups: definite non-peak and peak-like non-peak. The peak that belongs to the second group requires further process by modified RBFN classifier. The strength and weakness of the proposed approach are described as follows: (1) strength: the employed features are claimed to offer good performance in the proposed system, (2) weakness: inconsistency of feature slope information as the proposed work claimed that the proposed framework fails to provide slope information.
Liu et al. (2013) [21]	Epileptic EEG	A two-stage procedure is proposed for the detection of epileptic spike. k-NEO has been used to detect a candidate peak. The peak features are calculated based on the defined peak model. These features are then used as the input of the AdaBoost classifier. The strength and weakness of the proposed approach are described as follows: (1) strength: the peak model considers feature based on peak area, (2) weakness: the definition of area integration is not presented in the paper.

the difference between the peak point and the floating mean. The floating mean is the average EEG which is centered at the peak point that is also called moving average curve (MAC) [12]. The width is calculated based on the difference between the valley point at the first half wave and the

valley point at the second half wave. The two slopes are the slopes between a peak point and valley point in the first half wave and second half wave. Summary of different peak models on different style of framework is briefly described in Table 1. The strength and weakness are also highlighted

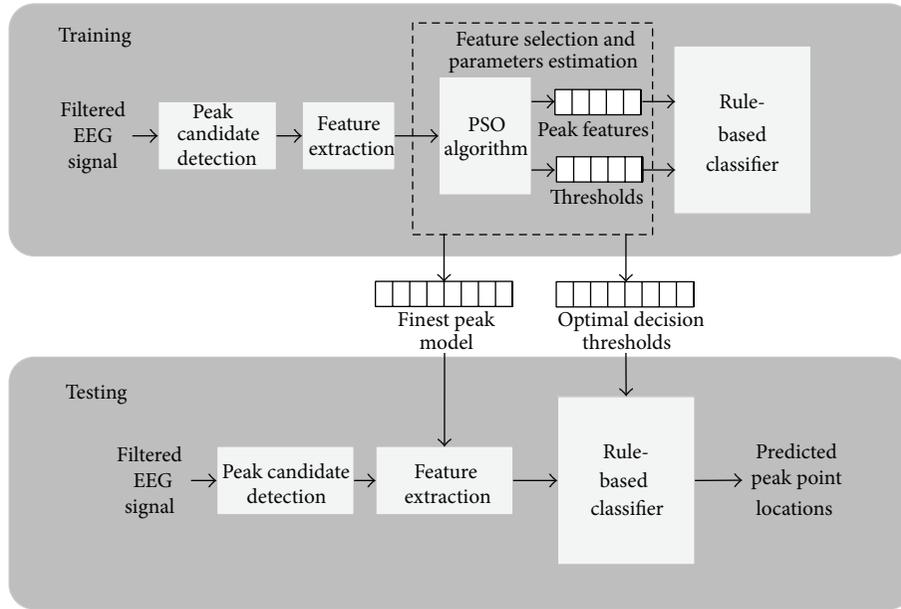


FIGURE 1: Feature selection and parameters estimation framework for peak detection algorithm.

in Table 1. Generally, the authors claimed that the selected peak feature offers good classification performance on the proposed framework. However, the previous works did not provide the justification on the selected features.

2. Methodology

Figure 1 shows the framework of the proposed techniques for EEG signal peak detection. There are two phases of the process which are training and testing phases. The training phase is firstly run to find the finest peak model and the optimal decision threshold values. Next, the testing phase is utilized for unseen EEG signal.

The framework can be divided into four stages: peak candidate detection, features extraction of peak candidate, feature selection and parameters estimation, and classification. In the first stage, the detection of peak candidates is performed to differentiate between a peak candidate and a non-peak candidate. The second stage is the extraction of peak candidate features. In the third stage, PSO algorithm is adapted during the training phase for feature selection and classifier parameters' estimation. Finally, the peak candidates are classified between predicted peak and predicted non-peak at particular locations by rule-based classifier.

2.1. Peak Candidate Detection. The first step to detect peaks is to find candidate peaks. Consider a discrete-time signal, $x(I)$, of L points. The i th candidate peak point, PP_i , as shown in Figure 2, is identified using three-points sliding window method [8]. Those three-points are denoted as $x(I - 1)$, $x(I)$, and $x(I + 1)$ for $I = 1, 2, \dots, L$. A candidate peak point is identified when $x(PP_i - 1) < x(PP_i) > x(PP_i + 1)$ and two associated valley points, $VP1_i$ and $VP2_i$, are in between as shown in Figure 2. Both valley points exist when $x(VP1_i -$

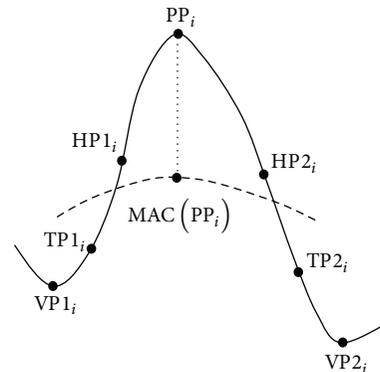


FIGURE 2: Model-based parameters.

$$1) > x(VP1_i) < x(VP1_i + 1) \text{ and } x(VP2_i - 1) > x(VP2_i) < x(VP2_i + 1).$$

2.2. Feature Extraction. Based on the existing peak models, the total peak features are fourteen. The peak features of a peak candidate are calculated based on the eight model-based parameters as shown in Figure 2. The parameters consist of the i th candidate peak point, PP_i , the two associated valley points, $VP1_i$ and $VP2_i$, the half point at first half wave ($HP1_i$), the half point at second half wave ($HP2_i$), the turning point at first half wave ($TP1_i$), the turning point at second half wave ($TP2_i$), and the moving average curve ($MAC(PP_i)$). The peak features can be categorized into three groups; amplitude, width, and slope. There are five different amplitudes, five different widths, and four different slopes that can be calculated based on the model-based parameters. All equations and description of peak features are tabulated in Table 2. Referring to Table 3, the peak model, which is

TABLE 2: Equations and descriptions of peak features.

Peak feature	Equation	Description
Amplitudes	$f_1 = x(PP_i) - x(VP1_i) $	Amplitude between the magnitude of peak and the magnitude of valley at the first half wave
	$f_2 = x(PP_i) - x(VP2_i) $	Amplitude between the magnitude of peak and the magnitude of valley of the second half wave
	$f_3 = x(PP_i) - x(TP1_i) $	Amplitude between the magnitude of peak and the magnitude of turning point at the first half wave
	$f_4 = x(PP_i) - x(TP2_i) $	Amplitude between the magnitude of peak and the magnitude of turning point at the second half wave
	$f_5 = x(PP_i) - MAC(PP_i) $	Amplitude between the magnitude of peak and the magnitude of moving average
Widths	$f_6 = VP1_i - VP2_i $	Width between valley point of first half point and valley point at second half wave
	$f_7 = PP_i - VP1_i $	Width between peak point and valley point at first half wave
	$f_8 = PP_i - VP2_i $	Width between peak point and valley point of second half wave
	$f_9 = TP1_i - TP2_i $	Width between turning point at first half wave and turning point at the second half wave
	$f_{10} = HP1_i - HP2_i $	Width between half point of first half wave and half point of second half wave
Slopes	$f_{11} = \frac{x(PP_i) - x(VP1_i)}{PP_i - VP1_i}$	Slope between a peak point and valley point at the first half wave
	$f_{12} = \frac{x(PP_i) - x(VP2_i)}{PP_i - VP2_i}$	Slope between a peak point and valley point at the second half wave
	$f_{13} = \frac{PP_i - TP1_i}{x(PP_i) - x(TP1_i)}$	The slope between peak point and turning point at the first half wave
	$f_{14} = \frac{PP_i - TP2_i}{x(PP_i) - x(TP2_i)}$	The slope between peak point and turning point at the second half wave

TABLE 3: List of different peak models and sets of features.

Peak model	Set of features	Number of features
Dumpala et al. (1982) [8]	f_1, f_6, f_{11}, f_{12}	4
Acir et al. (2005) [7, 11, 26]	$f_1, f_2, f_7, f_8, f_{13}, f_{14}$	6
Liu et al. (2002) [10]	$f_1, f_2, f_3, f_4, f_6, f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}$	11
Dingle et al. (1993) [9]	f_5, f_6, f_{11}, f_{12}	4

introduced by Dumpala et al. [8] and Dingle et al. [9], consists of four features. The peak model, which is specified by Acir et al. [7, 11], consists of six features. The peak model, which is specified by Liu et al. [10], consists of eleven features.

2.3. Feature Selection and Parameters Estimation Using Particle Swarm Optimization. In this stage, the peak features and classifier parameters are simultaneously found using two different PSO algorithms which are standard PSO and RA-PSO algorithms. At the end of this stage, the finest peak model and the optimal classifier parameters are obtained. The optimal classifier parameters represent the optimal decision threshold values.

The PSO algorithm was firstly introduced by Kennedy and Eberhart in 1995 [30]. The PSO algorithm has been numerously enhanced fundamentally [31, 32] and applied in many fields [33–35]. Fundamentally, the PSO algorithm follows several steps as described in Algorithm 1: (1) initialization, (2) calculation of the fitness function, (3) updating the personal best (*pbest*) for each particle and global best (*gbest*), (4) updating the particle's velocity and the particle's

```

(1) Initialization
(2) while not stopping criteria do
(3)   for each ith particle in a population do
(4)     calculate fitness function
(5)     update pbest and gbest
(6)   end for
(7)   for each particle in a population do
(8)     update the ith particle's velocity and
(9)     update the ith particle's position
(10)  end for
(11) end while

```

ALGORITHM 1: Standard PSO Algorithm.

position, and (5) performing termination based on a stopping criterion.

In PSO, particles search for the best solution and update the position information from iteration to iteration. Each particle in the population consists of a vector position and vector velocity in *d* dimension. The position of particle *i* at

```

(1) Initialization
(2) while not stopping criteria do
(3)   while not meet N times do
(4)     Randomly choose ith particle in a population
(5)     for ith particle in a population do
(6)       calculate fitness function
(7)       update pbest and gbest
(8)       update the ith particle's velocity and
(9)       update the ith particle's position
(10)    end for
(11)  end while
(12) end while

```

ALGORITHM 2: Random Asynchronous PSO (RA-PSO).

TABLE 4: Representation of particle position.

Particle	Peak features (binary type)				Thresholds (continuous type)			
	1	2	...	nf	$nf + 1$	$nf + 2$...	$nf \times 2$
s_i^k	$x_{i,1}^k$	$x_{i,2}^k$...	$x_{i,d}^k$	$x_{i,1}^k$	$x_{i,2}^k$...	$x_{i,D}^k$

iteration k is denoted as $s_i^k = \{x_{i,1}^k, x_{i,2}^k, x_{i,3}^k, \dots, x_{i,d}^k\}$, while the velocity of particle i at iteration k is denoted as $v_i^k = \{v_{i,1}^k, v_{i,2}^k, v_{i,3}^k, \dots, v_{i,d}^k\}$. The *pbest* of particle i is represented as $p_i^k = \{p_{i,1}^k, p_{i,2}^k, p_{i,3}^k, \dots, p_{i,d}^k\}$ and the *gbest* is denoted as $p_g^k = \{p_{g,1}^k, p_{g,2}^k, p_{g,3}^k, \dots, p_{g,d}^k\}$. To obtain the updated position of a particle, s_i^{k+1} , each particle changes its velocity as the follows:

$$v_i^{k+1} = \omega v_i^k + c_1 r_1 (p_i^k - x_i^k) + c_2 r_2 (p_g^k - x_i^k), \quad (1)$$

where c_1 is a cognitive coefficient, c_2 is a social coefficient, r_1 and r_2 are random values $[0, 1]$, and ω is a decrease inertial weight [36, 37] calculated as follows:

$$\omega = \omega_{\max} - \left(\frac{\omega_{\max} - \omega_{\min}}{k_{\max}} \right) \times k, \quad (2)$$

where ω_{\max} and ω_{\min} denote the maximum and minimum values of inertia weight, respectively, and k_{\max} is the maximum iteration. Then, the particle's position is updated based on (3). Note that this equation is only valid for continuous version of PSO algorithm:

$$s_i^{k+1} = s_i^k + v_i^{k+1}. \quad (3)$$

For a binary version of PSO [38], the particle position is updated based on the following equation:

$$T(v_i^{k+1}) = \left| \tanh(v_i^{k+1}) \right|, \quad (4)$$

$$s_i^{k+1} = \begin{cases} (s_i^k)^{-1} & \text{if rand} < T(v_i^{k+1}) \\ s_i^k & \text{rand} \geq T(v_i^{k+1}). \end{cases} \quad (5)$$

Equation (4) is a transfer function which is the main part of the binary version. Several studies have proven that this transfer function significantly improves the performance of

the standard binary PSO. Equation (5) is used to update the particle position according to the given rules, where s_i^k and v_i^k represent the vector position and velocity of i th particle at iteration k and $(s_i^k)^{-1}$ is the complement of s_i^k . The particle position maintains the current position when the velocity is lower than random value and its complement the position when the velocity is greater than random value. This method has been introduced by Mirjalili and Lewis (2013) that is also named as v-shaped transfer function [39].

Synchronous update in standard PSO algorithm indicates that all particles move to their new position after all particles are evaluated, as described in Algorithm 1. However, in RA-PSO [40], a particle immediately updates its position after it is evaluated without the need to wait until the evaluation of all particles is completed. Moreover, an i th particle in a population is randomly chosen with a total N times before i th particle is evaluated. N is the total number of particles. Some particles might be chosen more than once while some particles might not be chosen at all. The RA-PSO algorithm is described in Algorithm 2.

To perform the feature selection and parameters estimation simultaneously, both versions of PSO algorithm are employed to the standard PSO and RA-PSO algorithms. Table 4 illustrates the representation of particle position. The i th particle at iteration k , s_i^k , in PSO represents two types of dimensions which are binary and continuous type of dimension [29], $s_i^k = \{x_{i,1}^k, x_{i,2}^k, \dots, x_{i,d}^k, x_{i,1}^k, x_{i,2}^k, \dots, x_{i,D}^k\}$. The $d = 1, 2, 3, \dots, nf$ is a d th dimension of binary type, and the $D = nf + 1, nf + 2, nf + 3, \dots, nf \times 2$ is a D th dimension of continuous type. nf is the total number of peak features. The particle dimension is a two times number of features. The number of thresholds is equal of the number of features.

In the initialization stage of PSO algorithm, some of the parameters are initialized: (1) the initial PSO parameters and (2) the initial particle position. The initial PSO parameters

consist of the maximum inertia weight, ω_{\max} , the minimum inertia weight, ω_{\min} , the velocity clamping, v_{\max} the velocity vector for each particle, the *pbest* score for each particle, *gbest* score, the cognitive component, c_1 , and the social component, c_2 . The random values, r_1 and r_2 , are randomly distributed values from 0 to 1. All particles are randomly placed within the search space.

For the calculation of fitness function, geometric mean (*Gmean*) is employed. The *Gmean* is calculated as follows:

$$\begin{aligned} \text{TPR} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{TNR} &= \frac{\text{TN}}{\text{TN} + \text{FP}}, \\ \text{Gmean} &= \sqrt{\text{TPR} \times \text{TNR}}, \end{aligned} \quad (6)$$

where true peak (TP) is a correctly detected peak point, true non-peak (TN) is a correctly detected non-peak point, false peak (FP) is a wrongly detected the non-peak point, false non-peak (FN) is a wrongly detected peak point, TPR is a true peak rate, and TNR is a true non-peak rate.

2.4. Rule-Based Classifier. A rule-based classifier is employed to distinguish whether the candidate peak is a true peak or true non-peak from the extracted features. Each feature has a corresponding threshold value in the classification process. Given a set of features, a true peak only can be identified if all the feature values are greater than or equal to the decision threshold values. Otherwise, the candidate peak belongs to true non-peak. The form of the rule is

$$\begin{aligned} \text{IF } f_1 \geq \text{th}_1 \text{ AND } f_2 \geq \text{th}_2 \text{ AND } \dots \text{ AND} \\ f_M \geq \text{th}_M \text{ THEN Candidate Peak is a True Peak,} \end{aligned} \quad (7)$$

where f_i is denoted as a one of sixteen peak features, th_i is denoted as one of the decision threshold values of this peak feature, and true peak is predicted peak at a particular peak point location.

3. Experimental Setup

In this section, two experiments are conducted for peak detection of EEG signal. For first experiment, the framework is executed without feature selection. For second experiment, the experiment is executed with feature selection. The experimental protocols are discussed in the next subsection. The training and testing EEG signal are prepared to evaluate the performance of the proposed framework. Then, the results are discussed and analyzed.

Each experiment is conducted in 10 independent runs. For each run, 30 particles are used to perform feature selection and parameters estimation. For each particle, the total number of dimensions is depending on the number of features in a feature set. The maximum iteration was set to 1000. For the initial value of PSO parameters, the maximum inertia weight, ω_{\max} , is 0.9 and the minimum inertia weight, ω_{\min} , is 0.4. The cognitive component, c_1 , and the social

TABLE 5: Parameters setting of standard PSO and RA-PSO algorithms.

Initial PSO parameters	
Parameters	Value
Decrease inertia weight, ω	0.9~0.4
Cognitive component, c_1	2
Social component, c_2	2
Random value, r_1 and r_2	Random number [0, 1]
Velocity vector for each particle	0
Initial <i>pbest</i> score for each particle	0
Initial <i>gbest</i> score	0
Range of search space for $nf + 1$ to $nf + 5$	[0 30]
Range of search space for $nf + 6$ to $nf + 12$	[0 781.25]
Range of search space for $nf + 13$ to $nf \times 2$	[0 24.16]

component, c_2 , are set to 2. These values are proposed by Shi and Eberhart in 1999 [41]. The random values, r_1 and r_2 , are randomly distributed values from 0 to 1. The velocity clamping, v_{\max} , for binary version is set to 6 [39]. The velocity vector for each particle, the *pbest* score for each particle, and *gbest* score is set to 0. The parameters setting of standard PSO and RA-PSO algorithms are tabulated in Table 5.

3.1. Experimental Protocols. This study uses the eye movement EEG signal as a case study to evaluate the proposed framework. The observation of the eye movement EEG signal indicates that the most observable signal pattern is the peak point which signifies the brain response on eye movements. The known peak point locations through the response of the brain can be translated into an output, for example, wheelchair movement.

The experimental protocol to acquire this EEG signal was reviewed and approved by the Medical Ethic Committee (MEC) in the University of Malaya Medical Centre (UMMC). The subject gave a written consent prior to the data collection session. This EEG signal was acquired in the Applied Control and Robotic (ACR) Laboratory, Department of Electrical Engineering, Faculty of Engineering, University of Malaya, Malaysia. A healthy subject was involved voluntarily in this data collection session who is a postgraduate student in the Faculty of Engineering.

The EEG signal recording was conducted using the g.MOBILab portable signal acquisition system. The EEG signal was recorded from C4 channel. The EEG signal of channel CZ was used as a reference. The ground electrode was located on the forehead. The electrode was placed using the 10–20 international electrode placement system. The sampling frequency was set to 256 Hz.

Before the session begins, the subject was advised to get good rest. Thus, he can give full focus during the session. The subject was also advised to wash his hair. During the data collection session, the subject was required to be ready within 0 to 4 seconds for waiting for an external cue. The cue is a command for a subject to move their eyes to the right position. Within the standby time, the subject is required not to move their eyes into a frontal position.

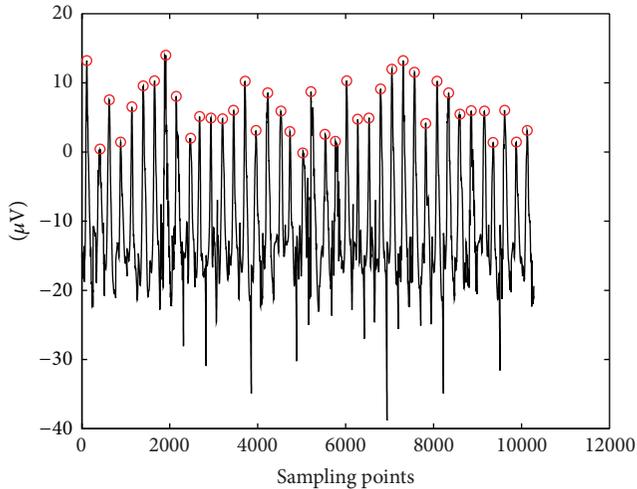


FIGURE 3: Filtered EEG signal.

TABLE 6: Signal specifications.

Specification	Channel C4
Total sampling point	10240
Total length signal (second)	40
Number of peak points in the signal	40
Sampling frequency (Hz)	256

When the time is exactly 5 seconds, the external cue appears on the screen monitor. The instruction allows the subject to move back their eyes in a frontal position. The external cue appears for 40 times. The total length of EEG recording is 40 seconds. As a cleanliness procedure, the electrodes and head-cap that are used in the session were washed. The filtered EEG signal is shown in Figure 3. Forty locations of definite peak points are highlighted in the red circle. The next process is to prepare the training and testing data.

From the data collection, 40 definite peak point locations have been identified by EEG expert. In 40-second signal there are 10240 sampling points, $x(I)$. There are only 40 peak points and the remaining of 10200 sampling points are the non-peak points. For preparing the training and testing signal, the training signal is selected from 1 to 5120 sampling points while the remaining EEG signal is used for testing signal. The signal specification is summarized in Table 6.

4. Results and Discussions

To evaluate the proposed framework for training and testing phase, four different measures are used including the average G_{mean} , the maximum G_{mean} , the minimum G_{mean} , and the standard deviation (STDEV).

4.1. Results of Peak Detection Algorithm without Feature Selection. Four peak models are employed for evaluating the peak models performance in the proposed framework. The training and testing performance based on those four

different measures for each model is shown in Table 7. The standard PSO algorithm is used to find the optimal threshold values for each peak model. The obtained results for each peak model are compared with the results of peak detection algorithm and the feature selection framework based on standard PSO. Notably, in this section, only standard PSO is considered in the peak detection algorithm without feature selection framework.

Referring to Table 7, the training performance for average, maximum, minimum, and STDEV is 84.01%, 89.15%, 80.58%, and 4.43% for Dumpala et al.'s peak model; 74.4%, 80.59%, 67.08%, and 3.71% for Acir et al.'s peak model; and 90.98%, 94.76%, 83.66%, and 5.51% for Dingle et al.'s peak model, respectively. The testing performance for average, maximum, minimum, and STDEV is 81.22%, 91.83%, 74.15%, and 9.13% for Dumpala et al.'s peak model; 68.59%, 77.43%, 54.77%, and 6.97% for Acir et al.'s peak model; and 88.78%, 94.75%, 77.44%, and 7.98% for Dingle et al.'s peak model, respectively.

Overall, the average performance of the training phase for Dumpala et al.'s peak model, Acir et al.'s peak model, and Dingle et al.'s peak model is greater than the average performance of their testing phase. However, for the peak model, Liu et al.'s peak model, will give zero percent performance for training and testing phase. This result indicates the limitation of rule-based classifier when dealing with both feature sets. During the training process on the feature sets, the particles in the PSO algorithm do not meet the optimum decision threshold values and the particles might also be trapped at local optima. Based on the preceding rule, a true peak only can be identified if all the feature values are greater than or equal to the decision threshold values. So, if one of the feature values does not satisfy the decision threshold value, the classifier will decide the peak candidate as a non-peak point. When this happens to all peak candidates, the TP is equal to zero. G_{mean} will give zero percent performance even if TN is equal to some values. The end results indicate the employment of the presented rule is only valid for Dumpala et al.'s peak model, Acir et al.'s peak model, and Dingle et al.'s peak model.

Compared to the test average performance of the peak models, the highest test performance is obtained by Dingle et al.'s peak model, which is 88.78%, then follows by Dumpala et al.'s peak model, which is 81.22%. The worst test performance is obtained by Acir et al.'s peak model, which is 68.59%. It can be concluded: from the findings of experimental results, the finest peak model for the filtered EEG signal is Dingle et al.'s peak model, and the worst peak model for the filtered EEG signal is Acir et al.'s peak model. True peak rate and true non-peak rate of test performance are shown in Table 8. It can be concluded that, from the finding experimental results, the chosen peak models limit the designed framework to obtain the best accuracy. Therefore, the feature selection technique using standard PSO is employed into the designed framework.

4.2. Results of Peak Detection Algorithm with Feature Selection. The results of peak detection algorithm with feature selection are categorized into two subsections which are the results of

TABLE 7: Training and testing performance of peak detection for each peak model (without feature selection).

Peak model	Training (%)				Testing (%)			
	Average	Max	Min	STDEV	Average	Max	Min	STDEV
Dumpala et al. (1982) [8]	84.01	89.15	80.58	4.43	81.22	91.83	74.15	9.13
Acir et al. (2005) [7, 11, 26]	74.4	80.59	67.08	3.71	68.59 ^{worst}	77.43	54.77	6.97
Liu et al. (2002) [10]	0	0	0	0	0	0	0	0
Dingle et al. (1993) [9]	90.98	94.76	83.66	5.1	88.78^{best}	94.75	77.44	7.98

TABLE 8: TPR and TNR test results for EEG signal (without feature selection).

Peak model	TPR (%)	TNR (%)
Dumpala et al. (1982) [8]	65.0	99.7
Acir et al. (2005) [7, 11, 26]	50.0	99.9
Liu et al. (2002) [10]	0.0	0.0
Dingle et al. (1993) [9]	80.0	99.3

feature selection using standard PSO and the results of feature selection using RA-PSO. Also, the results from the two PSO algorithms in the proposed framework are discussed.

4.2.1. Feature Selection Using Standard PSO. The feature sets of 10 runs using the standard PSO algorithm are shown in Table 9. The result shows the variety of the optimal combination of features that give the higher classification performance, mostly higher than 99.69%. The maximum training accuracy is 99.98%. The most significant peak feature is the feature f_5 because all the 10 runs appear as a selected feature by PSO. Feature f_5 is the amplitude that is calculated from the difference between peak points (PP) and moving average curve (MAC). Another most significant feature is feature f_2 , which is the calculated amplitude between a peak point and valley point at the second half wave. The feature f_6 is chosen 4 times. The feature f_6 is chosen 4 times. The features f_4 and f_9 are chosen 2 times. The feature f_{10} is only selected at 9th run.

Based on the results in Table 9, the combination of peak features (f_2 , f_5 , and f_6) appears 4 times, the combination of peak features (f_2 , f_5 , and f_9) appears 2 times, and the combination of peak features (f_2 and f_5) appears 2 times. Therefore, there are 3 optimal combinations of features that can be chosen.

Table 10 has the optimal threshold values for the optimal combination of the features. The threshold values are selected based on the selected peak features that are highlighted in the table.

The average of training and testing results of 10 runs using standard PSO algorithm is tabulated in Table 11. The results of standard PSO show the average training accuracy is 99.91%. The maximum training accuracy is 99.98%. The minimum training accuracy is 99.69%, and the standard deviation is 8.07%. On the other hand, the testing accuracy is 93.73%. The maximum testing accuracy is 99.92%. The minimum testing accuracy is 77.41%.

In terms of peak and the non-peak rate (TP and TN) for training results, the classifier accurately predicted all 20 peak points and 5113 non-peak points. The results also show that the classifier misclassified 27 non-peak points. The maximum of the true peak point is 20 and true non-peak point is 5118. The minimum of true peak point is 20, and true non-peak point is 5109.

For testing results, the classifier accurately predicted 18 peak points and 5110 non-peak points. The maximum of the true peak point is 20 and true non-peak point is 5114. The minimum of true peak point is 12 and true non-peak point is 5106. In general, the average testing result that corresponds to the selected peak features using the proposed feature selection framework is greater than the average testing result of Dingle's peak model which is 93.73% and 88.78%. The feature set of the Dingle's peak model is f_5 , f_6 , f_{11} , and f_{12} while the feature set that gives a higher training performance in this experiment is f_2 and f_5 .

However, the proposed framework based on standard PSO produces slightly high variance model as it measures from the STDEV index. The STDEV is evaluated for measuring the algorithm consistency where lowest STDEV value indicates a good generalization algorithm. Based on the results of the STDEV in Table 13, the STDEV values of the standard PSO are 8.07% and 7.18% for training and testing, respectively. This results show that the high standard deviation of the accuracy is recorded between maximum and minimum of classification rate. The experimental results are reasonable due to the limitation of the standard PSO algorithm.

4.2.2. Feature Selection Using RA-PSO. Table 12 shows the feature selection results of 10 runs based on the RA-PSO algorithm. The feature set was highlighted of each run. The threshold values for all selected features are also given in Table 13. The highest G_{mean} value of training phase is 99.91%. The significant peak features are f_5 and f_8 . The corresponding threshold values are 9.20 and 4. Note that feature f_5 is the amplitude that is calculated from the difference between peak points (PP) and moving average curve (MAC). Another most significant feature is feature f_8 , which is the width between peak point and valley point of second half wave. The features f_2 , f_4 , and f_8 are chosen 3 times. The feature f_{12} is only selected at second run.

Three similar results were obtained out of ten runs. Other significant feature sets that are obtained in this result are the combination of peak features (f_2 and f_5) and (f_4 and f_5). These feature sets also appear 3 times.

TABLE 9: Training results: the feature sets of 10 runs using standard PSO.

Run	Amplitudes				Peak features						Slopes	Gmean (%)		
	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}			f_{11}	f_{12}
#1	0	1	0	0	1	0	0	0	1	0	0	0	0	99.89
#2	0	0	0	1	1	0	0	0	0	0	0	0	0	99.91
#3	0	1	0	0	1	0	0	0	1	0	0	0	0	99.69
#4	0	1	0	0	1	1	0	0	0	0	0	0	0	99.92
#5	0	1	0	0	1	1	0	0	0	0	0	0	0	99.91
#6	0	1	0	0	1	0	0	0	0	0	0	0	0	99.95
#7	0	1	0	0	1	1	0	0	0	0	0	0	0	99.94
#8	0	1	0	0	1	1	0	0	0	0	0	0	0	99.91
#9	0	0	0	1	1	0	0	0	0	1	0	0	0	99.96
#10	0	1	0	0	1	0	0	0	0	0	0	0	0	99.98

TABLE 10: Training results: the optimal decision threshold values of 10 runs using standard PSO.

Run	Optimal threshold values													
	th ₁	th ₂	th ₃	th ₄	th ₅	th ₆	th ₇	th ₈	th ₉	th ₁₀	th ₁₁	th ₁₂	th ₁₃ -th ₁₄	
#1	—	0.40	—	—	9.07	—	—	—	9	—	—	—	—	
#2	—	—	—	0.27	9.20	—	—	—	—	—	—	—	—	
#3	—	1.24	—	—	9.27	—	—	—	17	—	—	—	—	
#4	—	0.37	—	—	8.93	12	—	—	—	—	—	—	—	
#5	—	0.43	—	—	9.18	12	—	—	—	—	—	—	—	
#6	—	1.25	—	—	11.34	—	—	—	—	—	—	—	—	
#7	—	0.93	—	—	9.07	11	—	—	—	—	—	—	—	
#8	—	0.38	—	—	9.10	8	—	—	—	—	—	—	—	
#9	—	—	—	0.43	9.13	—	—	—	—	8	—	—	—	
#10	—	0.90	—	—	10.07	—	—	—	—	—	—	—	—	

TABLE 11: Average training and testing results of 10 runs with feature selection using standard PSO.

Algorithm	Results	Training					Testing				
		Gmean (%)	TN	FP	TP	FN	Gmean (%)	TN	FP	TP	FN
Standard PSO	AVG	99.91	5113	27	20	0	93.73	5110	30	18	2
	MAX	99.98	5118	22	20	0	99.92	5114	26	20	0
	MIN	99.69	5109	31	20	0	77.41	5106	34	12	8
	STDEV	8.07					7.18				

TABLE 12: Training results: the feature sets of 10 runs using RA-PSO.

RA-PSO	Amplitudes				Peak features						Slopes	Gmean (%)		
	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}			f_{11}	f_{12}
#1	0	0	0	0	1	0	0	1	0	0	0	0	0	99.91
#2	0	0	0	0	1	0	0	0	0	0	0	1	0	99.87
#3	0	0	0	1	1	0	0	0	0	0	0	0	0	99.90
#4	0	0	0	1	1	0	0	0	0	0	0	0	0	99.90
#5	0	0	0	0	1	0	0	1	0	0	0	0	0	99.91
#6	0	1	0	0	1	0	0	0	0	0	0	0	0	99.90
#7	0	1	0	0	1	0	0	0	0	0	0	0	0	99.90
#8	0	0	0	1	1	0	0	0	0	0	0	0	0	99.90
#9	0	1	0	0	1	0	0	0	0	0	0	0	0	99.90
#10	0	0	0	0	1	0	0	1	0	0	0	0	0	99.91
AVERAGE Gmean														99.90

TABLE 13: Training results: the optimal decision threshold values of 10 runs using RA-PSO.

Run	Optimal threshold values using RA-PSO												
	th ₁	th ₂	th ₃	th ₄	th ₅	th ₆	th ₇	th ₈	th ₉	th ₁₀	th ₁₁	th ₁₂	th ₁₃ -th ₁₄
#1	—	—	—	—	9.20	—	—	4	—	—	—	—	—
#2	—	—	—	—	9.21	—	—	—	—	—	—	0.6	—
#3	—	—	—	0.38	9.21	—	—	—	—	—	—	—	—
#4	—	—	—	0.22	9.20	—	—	—	—	—	—	—	—
#5	—	—	—	—	9.20	—	—	4	—	—	—	—	—
#6	—	0.36	—	—	9.04	—	—	—	—	—	—	—	—
#7	—	0.39	—	—	9.22	—	—	—	—	—	—	—	—
#8	—	—	—	0.25	9.20	—	—	—	—	—	—	—	—
#9	—	0.27	—	—	9.19	—	—	—	—	—	—	—	—
#10	—	—	—	—	9.20	—	—	4	—	—	—	—	—

TABLE 14: Average training and testing results of 10 runs with feature selection using RA-PSO.

Algorithm	Results	Training					Testing				
		<i>Gmean</i> (%)	TN	FP	TP	FN	<i>Gmean</i> (%)	TN	FP	TP	FN
RA-PSO	AVG	99.90	5110	30	20	0	98.59	5106	34	19	1
	MAX	99.91	5111	29	20	0	99.86	5107	33	20	0
	MIN	99.87	5107	33	20	0	97.33	5103	37	19	1
	STDEV	1.15					1.33				

Table 14 shows the average training and testing results of 10 runs with feature selection using RA-PSO algorithm. The average *Gmean* value of the RA-PSO algorithm is 99.90% and 98.59% for training and testing, respectively. The maximum *Gmean* value of the RA-PSO algorithm is 99.91% and 99.86% for training and testing, respectively. The minimum *Gmean* value of the RA-PSO algorithm is 99.87% and 97.33% for training and testing, respectively.

In terms of peak and the non-peak rate (TP and TN) for training results, the classifier accurately predicted all 20 peak points and 5110 non-peak points. The results also show that the classifier misclassified 30 non-peak points. The maximum of the true peak point is 20 and true non-peak point is 5111. The minimum of true peak point is 20 and true non-peak point is 5107.

For testing results, the classifier accurately predicted 19 peak points and 5106 non-peak points. The maximum of the true peak point is 20 and true non-peak point is 5107. The minimum of true peak point is 19 and true non-peak point is 5103.

As compared to the framework, using standard PSO, RA-PSO is found to offer lower variance model. The recorded STDEV values of the RA-PSO are 1.15% and 1.33% for training and testing, respectively. Therefore, the RA-PSO may offer a reliable and reasonable model as compared to standard PSO with consistent classification rate.

5. Conclusions

In this study, the framework of feature selection and parameters estimation is proposed for EEG signal peak detection algorithm. The proposed framework involves peak candidate

detection, feature extraction, feature selection, and classification. The framework is developed based on PSO algorithm and a rule-based classifier. In general, the binary PSO based algorithm was utilized for selecting the peak features while the continuous PSO based algorithm was utilized for optimizing the classifier parameters. Two PSO based algorithms are employed in the proposed framework: (1) standard PSO and (2) RA-PSO. Fourteen peak features were employed in this study. All these peak features were taken from the existing peak models in the time domain approach. The available peak features are then automatically selected in combinatorial form using the proposed framework. Based on the experiment results of peak detection algorithm without feature selection, the best peak model is Dingle et al.'s [9] peak model where the highest performance obtained is 88.78%. Meanwhile, the experimental results with feature selection show the proposed framework with standard PSO can further improve the Dingle et al.'s model. However, the recorded results are inconsistent due to high variances of the classification accuracy. The unreliability of the standard PSO can be further improved based on the proposed framework using RA-PSO. In general, the proposed feature selection technique offers a better performance as compared to any peak models without feature selection. For future work, the proposed framework will be employed in more case studies and will invent more classification methods.

Conflict of Interests

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

Acknowledgments

This project is funded by the Ministry of Education Malaysia for High Impact Research Grant (UM-D000016-16001), University of Malaya, Research Acculturation Grant Scheme (RDU121403), Universiti Malaysia Pahang, Fundamental Research Grant Scheme (VOT 4F331), Universiti Teknologi Malaysia, and MyPhD scholarship from Ministry of Education Malaysia. The authors would like to thank the Faculty of Engineering, the University of Malaya, for supporting this research. The authors also would like to acknowledge the Editor and anonymous reviewers for their valuable comments and suggestions.

References

- [1] A. Bulling, J. A. Ward, H. Gellersen, and G. Tröster, "Eye movement analysis for activity recognition using electrooculography," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 741–753, 2011.
- [2] H. Zeng and A. G. Song, "Removal of EOG artifacts from EEG recordings using stationary subspace analysis," *The Scientific World Journal*, vol. 2014, Article ID 259121, 9 pages, 2014.
- [3] R. Tafreshi, A. Jaleel, J. Lim, and L. Tafreshi, "Automated analysis of ECG waveforms with atypical QRS complex morphologies," *Biomedical Signal Processing and Control*, vol. 10, pp. 41–49, 2014.
- [4] N. Xu, X. Gao, B. Hong, X. Miao, S. Gao, and F. Yang, "BCI competition 2003—data set IIb: enhancing P300 wave detection using ICA-based subspace projections for BCI applications," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1067–1072, 2004.
- [5] Q. G. Ma and Q. Shang, "The influence of negative emotion on the Simon effect as reflected by P300," *The Scientific World Journal*, vol. 2013, Article ID 516906, 6 pages, 2013.
- [6] K. P. Indiradevi, E. Elias, P. S. Sathidevi, S. Dinesh Nayak, and K. Radhakrishnan, "A multi-level wavelet approach for automatic detection of epileptic spikes in the electroencephalogram," *Computers in Biology and Medicine*, vol. 38, no. 7, pp. 805–816, 2008.
- [7] N. Acir and C. Güzeliş, "Automatic spike detection in EEG by a two-stage procedure based on support vector machines," *Computers in Biology and Medicine*, vol. 34, no. 7, pp. 561–575, 2004.
- [8] S. R. Dumpala, S. Narasimha Reddy, and S. K. Sarna, "An algorithm for the detection of peaks in biological signals," *Computer Programs in Biomedicine*, vol. 14, no. 3, pp. 249–256, 1982.
- [9] A. A. Dingle, R. D. Jones, G. J. Carroll, and W. R. Fright, "A multistage system to detect epileptiform activity in the EEG," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 12, pp. 1260–1268, 1993.
- [10] H. S. Liu, T. Zhang, and F. S. Yang, "A multistage, multimethod approach for automatic detection and classification of epileptiform EEG," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 12 I, pp. 1557–1566, 2002.
- [11] N. Acir, I. Öztura, M. Kuntalp, B. Baklan, and C. Güzeliş, "Automatic detection of epileptiform events in EEG by a three-stage procedure based on artificial neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 1, pp. 30–40, 2005.
- [12] W. Lu, M. M. Nystrom, P. J. Parikh et al., "A semi-automatic method for peak and valley detection in free-breathing respiratory waveforms," *Medical Physics*, vol. 33, no. 10, pp. 3634–3636, 2006.
- [13] L. Xu, M. Q.-H. Meng, R. Liu, and K. Wang, "Robust peak detection of pulse waveform using height ratio," in *Proceedings of the 30th IEEE Annual International Conference of the Engineering in Medicine and Biology Society*, pp. 2856–3859, British Columbia, Canada, 2008.
- [14] R. Barea, L. Boquete, S. Ortega, E. López, and J. M. Rodríguez-Ascariz, "EOG-based eye movements codification for human computer interaction," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2677–2683, 2012.
- [15] M. S. Manikandan and K. P. Soman, "A novel method for detecting R-peaks in electrocardiogram (ECG) signal," *Biomedical Signal Processing and Control*, vol. 7, no. 2, pp. 118–128, 2012.
- [16] A. Juozapavičius, G. Bacevičius, D. Bugelskis, and R. Samaitienė, "EEG analysis—automatic spike detection," *Journal of Nonlinear Analysis: Modelling and Control*, vol. 16, no. 4, pp. 375–386, 2011.
- [17] L. Senhadji and F. Wendling, "Epileptic transient detection: wavelets and time-frequency approaches," *Neurophysiologie Clinique*, vol. 32, no. 3, pp. 175–192, 2002.
- [18] M. Putignano, A. Intermite, and P. Welsch, "A non-linear algorithm for current signal filtering and peak detection in SiPM," *Journal of Instrumentation*, vol. 7, pp. 1–19, 2012.
- [19] R. E. Bonner, L. Crevasse, M. IrenéFerrer, and J. C. Greenfield Jr., "A new computer program for analysis of scalar electrocardiograms," *Computers and Biomedical Research*, vol. 5, no. 6, pp. 629–653, 1972.
- [20] V. P. Oikonomou, A. T. Tzallas, and D. I. Fotiadis, "A Kalman filter based methodology for EEG spike enhancement," *Computer Methods and Programs in Biomedicine*, vol. 85, no. 2, pp. 101–108, 2007.
- [21] Y.-C. Liu, C.-C. K. Lin, J.-J. Tsai, and Y.-N. Sun, "Model-based spike detection of epileptic EEG data," *Sensors*, vol. 13, pp. 12536–12547, 2013.
- [22] N. Sinno and K. Tout, "Analysis of epileptic events using wavelet packets," *The International Arab Journal of Information Technology*, vol. 5, no. 4, pp. 165–169, 2008.
- [23] Z. Ji, X. Wang, T. Sugi, S. Goto, and M. Nakamura, "Automatic spike detection based on real-time multi-channel template," in *Proceedings of the 4th International Conference on Biomedical Engineering and Informatics (BMEI '11)*, pp. 648–652, IEEE, October 2011.
- [24] T. P. Exarchos, A. T. Tzallas, D. I. Fotiadis, S. Konitsiotis, and S. Giannopoulos, "EEG transient event detection and classification using association rules," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 3, pp. 451–457, 2006.
- [25] C. J. James, R. D. Jones, P. J. Bones, and G. J. Carroll, "Detection of epileptiform discharges in the EEG by a hybrid system comprising mimetic, self-organized artificial neural network, and fuzzy logic stages," *Clinical Neurophysiology*, vol. 110, no. 12, pp. 2049–2063, 1999.
- [26] N. Acir, "Automated system for detection of epileptiform patterns in EEG by using a modified RBFN classifier," *Expert Systems with Applications*, vol. 29, no. 2, pp. 455–462, 2005.
- [27] J. F. Gao, Y. Yang, P. Lin, P. Wang, and C. X. Zheng, "Automatic removal of eye-movement and blink artifacts from EEG signals," *Brain Topography*, vol. 23, no. 1, pp. 105–114, 2010.

- [28] S. B. Wilson and R. Emerson, "Spike detection: a review and comparison of algorithms," *Clinical Neurophysiology*, vol. 113, no. 12, pp. 1873–1881, 2002.
- [29] C.-L. Huang and J.-F. Dun, "A distributed PSO-SVM hybrid system with feature selection and parameter optimization," *Applied Soft Computing*, vol. 8, no. 4, pp. 1381–1391, 2008.
- [30] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICW '95)*, pp. 1942–1948, Perth, Australia, November–December 1995.
- [31] K. S. Lim, Z. Ibrahim, S. Buyamin et al., "Improving vector evaluated particle swarm optimisation by incorporating non-dominated solutions," *The Scientific World Journal*, vol. 2013, Article ID 510763, 19 pages, 2013.
- [32] M. S. Mohamad, S. Omatu, S. Deris, M. Yoshioka, A. Abdullah, and Z. Ibrahim, "An enhancement of binary particle swarm optimization for gene selection in classifying cancer classes," *Algorithms for Molecular Biology*, vol. 8, article 15, 2013.
- [33] Z. Ibrahim, N. K. Khalid, J. A. A. Mukred et al., "A DNA sequence design for DNA computation based on binary vector evaluated particle swarm optimization," *International Journal of Unconventional Computing*, vol. 8, no. 2, pp. 119–137, 2012.
- [34] A. Adam, A. F. Zainal Abidin, Z. Ibrahim, A. R. Husain, Z. Md Yusof, and I. Ibrahim, "A particle swarm optimization approach to Robotic Drill route optimization," in *Proceedings of the 4th International Conference on Mathematical Modelling and Computer Simulation (AMS '10)*, pp. 60–64, May 2010.
- [35] M. N. Ayob, Z. M. Yusof, A. Adam et al., "A particle swarm optimization approach for routing in VLSI," in *Proceedings of the 2nd International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN '10)*, pp. 49–53, July 2010.
- [36] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 69–73, Anchorage, Alaska, USA, May 1998.
- [37] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proceedings of the 7th Annual Conference on Evolutionary Programming*, pp. 591–601, San Diego, Calif, USA, 1998.
- [38] J. Kennedy and R. C. Eberhart, "Discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International Conference on Computational Cybernetics and Simulation*, pp. 4104–4108, IEEE, Orlando, Fla, USA, October 1997.
- [39] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1–14, 2013.
- [40] J. Rada-Vilela, M. Zhang, and W. Seah, "A performance study on synchronicity and neighborhood size in particle swarm optimization," *Soft Computing*, vol. 17, no. 6, pp. 1019–1030, 2013.
- [41] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the IEEE Conference on Evolutionary Computation*, pp. 1945–1950, Washington, DC, USA, 1999.

Research Article

A Community Detection Algorithm Based on Topology Potential and Spectral Clustering

Zhixiao Wang, Zhaotong Chen, Ya Zhao, and Shaoda Chen

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China

Correspondence should be addressed to Zhixiao Wang; softstone416@163.com

Received 12 May 2014; Accepted 12 July 2014; Published 22 July 2014

Academic Editor: Shifei Ding

Copyright © 2014 Zhixiao Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Community detection is of great value for complex networks in understanding their inherent law and predicting their behavior. Spectral clustering algorithms have been successfully applied in community detection. This kind of methods has two inadequacies: one is that the input matrixes they used cannot provide sufficient structural information for community detection and the other is that they cannot necessarily derive the proper community number from the ladder distribution of eigenvector elements. In order to solve these problems, this paper puts forward a novel community detection algorithm based on topology potential and spectral clustering. The new algorithm constructs the normalized Laplacian matrix with nodes' topology potential, which contains rich structural information of the network. In addition, the new algorithm can automatically get the optimal community number from the local maximum potential nodes. Experiments results showed that the new algorithm gave excellent performance on artificial networks and real world networks and outperforms other community detection methods.

1. Introduction

Most networks show community structure [1]. The results of community detection are meaningful for forecasting the behavior and evolution trend of complex networks [2]. For example, in World Wide Web, community detection can be used to improve the performance of search engine, in social networks, community detection can be used to forecast the information propagation among users [3], in electronic commerce area, community detection can be used to select potential user for advertising; and in bioengineering area, community detection can be used to recognize functions of protein [4].

In recent years, many methods inspired by different paradigms are put forward for community detection [5]. Among these efforts, spectral clustering has shown to be successful [6], for it is very simple to implement and can be solved by standard linear algebra methods.

The traditional spectral clustering methods are based on kinds of input matrixes, such as the adjacency matrix, the standard Laplacian matrix, and the normalized Laplacian matrix. The standard Laplacian matrix is defined as

$L = D - A$, and the normalized Laplacian matrix is defined as $L = D^{-1}A$, where A is adjacency matrix and D is a diagonal matrix with elements D_{ii} being the degree of the i th node.

Almost all above matrixes are constructed with the adjacency matrix and diagonal matrix of networks. These matrixes can only reflect the local relationship between a node and its direct neighbors, as [6] pointed out, "the eigenvalues and eigenvectors of traditional input matrixes cannot provide sufficient structural information for community detection." As a result, the accuracy of community detection may decrease.

What is more, the community number k must be set in advance for the spectral clustering method based on standard Laplacian matrix. The normalized Laplacian matrix can solve this problem to some extent, which has k nontrivial eigenvalues close to the biggest eigenvalue 1. The eigenvector elements corresponding to these eigenvalues present ladder distribution. The proper community number of communities can be estimated by the ladders. However, when the community structure of network is not clear, the eigenvector elements cannot show obvious ladder distribution but an approximately continuous curve [7]. In this case, we cannot get

the proper community number from the ladder distribution of eigenvector elements.

In order to solve these problems, this paper puts forward a novel community detection algorithm based on topology potential and spectral clustering. The algorithm constructs the normalized Laplacian matrix with topology potential of network nodes. The topology potential of a node is the sum of potential components produced by neighbors at the position of this node. The topology potential describes the complicated interaction among nodes and contains rich structural information of the network. This structural information is meaningful for community detection. In addition, the new algorithm can automatically get the optimal community number from the local maximum potential nodes, whether the community structure of network is obvious or not. Experiments results showed that the new algorithm can improve the accuracy of community detection and has significant adaptability.

This paper is organized as follows. Section 2 describes related works. Section 3 introduces the concept of topology potential. Section 4 shows the new community detection algorithm based on topology potential and spectral clustering. Section 5 is simulation experiment and results. Section 6 comes to the conclusion of this paper.

2. Related Works

Spectral clustering algorithms have been successfully applied to community detection. From the perspective of input matrix, spectral clustering methods can be divided into the adjacency matrix [8], the standard Laplacian matrix [9], the normalized Laplacian matrix [10], the modularity matrix [11], and the correlation matrix [12]. Reference [13] found that the normalized Laplacian matrix significantly outperforms the other matrixes in identifying the community structure of networks.

In order to improve the performance of spectral clustering, many nontraditional spectral clustering algorithms have been proposed [6], such as complement based spectral clustering [14], complex eigenvector based spectral clustering [15], semisupervised spectral clustering [16], and eigenspace-based spectral clustering [17]. Zarei and Samani [14] gave out a spectral method based on the network complement and anticommunity concept, declaring “the spectrum of matrixes corresponding to a network complement reveals the communities more accurately than that of a matrix corresponding to the network itself.” Zarei et al. [15] also put forward a spectrum method based on complex eigenvectors and found that the complex eigenvectors of network matrixes showed better performance in community detection. Mavroeidis [16] proposed a semisupervised spectral clustering, and its results showed that the partial supervision cannot only improve the quality of spectral clustering but also accelerate the spectral clustering. Ma et al. [17] presented an eigenspace-based spectral method for community detection, which can identify both the overlapping and hierarchical community without increasing the time complexity. All these methods try to integrate some additional topology structure information into input matrixes.

Except for methods mentioned above, there are some other newly developed spectral methods for community detection. Gong et al. [18] proposed a spectral algorithm utilizing multiple eigenvectors to identify the communities in networks, which performed better for more spectral information is used. Newman [19] found that, within the spectral approximations, community detection by modularity maximization, community detection by statistical inference, and normalized-cut graph partitioning are identical. With the large-deviation theory, Bo et al. [20] established a relationship between the hierarchical community structure of a network and the local mixing properties.

Recently, a novel theory-topology potential theory was introduced to complex network for community detection [21]. Because of its inherent advantage in time complexity and performance, this theory has attracted plenty of attention. Gan et al. [21] put forward a topology-potential-based community detection algorithm. With the algorithm, the community structure can be uncovered by “detecting all local high potential areas margined by low potential nodes.” Han et al. [22] proposed an overlapping community detection algorithm, which divides networks into separate communities by “spreading outward from each local maximum potential node.” Zhang et al. [23] proposed a variable scale network overlapping community identification method based on topology potential. In order to identify overlapping nodes, this method defined an identity uncertainty measure related to topology potential. These above topology-potential-based methods show better performance in community detection; however, there is a weakness for almost all these methods; that is, they definitely need additional strategies or parameters to determine the community attachment of nodes, such as the benefit function in [21] and the parameter ξ in [23].

Different from above works, this paper puts forward a novel community detection algorithm, which combines spectral clustering and topology potential, making best use of their advantages and bypassing their disadvantages. The new algorithm constructs the normalized Laplacian matrix with topology potential of network nodes. The topology potential contains rich structural information of the network, which is meaningful for community detection. What's more, the new algorithm can automatically judge the optimal community number from the local maximum potential nodes, whether the community structure of complex network is obvious or not.

3. Topology Potential

The topology potential field theory is an important branch of the field theory. People abstracted the classical field as a mathematical model to describe noncontact interaction between objects [24]. Any complex network has its relatively stable topology structure; nodes in the network are not isolated, and there exist relationships between nodes linked by edges. Therefore, the topology potential field theory was introduced into complex network to describe the interaction and association among network nodes [22].

Given a network $G = (V, E)$, where $V = \{v_i \mid i = 1, \dots, n\}$ is a set of nodes, n is the total number of nodes,

$E = \{(v_i, v_j) \mid v_i, v_j \in V\}$ is a set of edges. According to the topology potential field theory, the topology potential of any node is defined as follows:

$$\varphi(v_i) = \sum_{l=1}^k m(v_l) \times e^{-(d_{il}/\sigma)^2}, \quad (1)$$

where $\varphi(v_i)$ is the topology potential of node v_i , $1 \leq i \leq n$; the node v_l is a node within the influence scope of node v_i , and k is the total number of nodes within the influence scope, $1 \leq k \leq n - 1$, $1 \leq l \leq k$; d_{il} is the hops between v_i and v_l ; $m(v_l)$ is the mass of node v_l ; generally speaking, it is set to 1, and the mass difference between nodes is ignored; σ is an impact factor used to control the influence scope of node, the maximum scope is $\lfloor 3\sigma/\sqrt{2} \rfloor$ hops.

The impact factor σ will affect topology potential field and the influence scope of node. If σ is small, the interaction and association among nodes is weak. And when $\sigma \rightarrow 0$, there is even no interaction and association. Conversely, if σ is big, the interaction and association become strong, and in extreme conditions, all nodes associate with each. Therefore, we need to select suitable value, so as to make the distribution of topology potential value reflect the structure characteristics of network. Potential entropy has been introduced to evaluate the rationality of topology potential value distribution [21].

Suppose the topology potential of nodes v_1, v_2, \dots, v_n are $\varphi(v_1), \varphi(v_2), \dots, \varphi(v_n)$, respectively; the potential entropy H is defined as

$$H = -\sum_{i=1}^n \frac{\varphi(v_i)}{Z} \cdot \log\left(\frac{\varphi(v_i)}{Z}\right), \quad (2)$$

where n is the total number of nodes; $Z = \sum_{i=1}^n \varphi(v_i)$ is a normalization factor. When topology potential field achieves the smallest potential entropy, the impact factor value is optimal [25].

As can be seen from the formula (1), the topology potential of a node totally depends on the topology structure of its surroundings, which reflects the influence ability of another node over it. Obviously, the topology potential contains rich structural information of the network, which offers a desirable solution to the insufficient structural information in the traditional Laplacian matrix. If we construct the Laplacian matrix by using topology potential of network nodes, the additional structural information can be provided for community detection. So, this paper puts forward a novel algorithm based on topology potential and spectral clustering to improve the performance of community detection, and Section 4 will describe the new algorithm in detail.

4. Community Detection Algorithm

In this section, we will give out a novel community detection algorithm based on topology potential and spectral clustering. The new algorithm is described as follows.

Input: complex network $G = (V, E)$, the corresponding node set $V = \{v_i \mid i = 1, \dots, n\}$, edge set $E = \{(v_i, v_j) \mid v_i, v_j \in V\}$.

Output: a community partition of G .

Algorithm Description:

- (1) calculate the topology potential of node with formula (1);
- (2) search all local maximum potential nodes of G . Suppose we find k local maximum potential nodes;
- (3) construct the potential component matrix P and topology potential matrix T of G ;
- (4) compute the normalized Laplacian matrix $L = T^{-1}P$ of G ;
- (5) compute the first k eigenvectors u_1, \dots, u_k of L , k is the total number of local maximum potential nodes;
- (6) map all nodes in V to \mathbb{R}^k corresponding to eigenvectors u_1, \dots, u_k ;
- (7) cluster the nodes in \mathbb{R}^k with the k -means algorithm into communities C_1, \dots, C_k .

Compared with the traditional spectral clustering method, the new algorithm constructs the normalized Laplacian matrix with the topology potential of nodes and can automatically get the optimal community number from the local maximum potential node.

The following part of the section will focus on the normalized Laplacian matrix construction and local maximum potential node search of the new community detection algorithm.

4.1. Normalized Laplacian Matrix Construction. In order to add additional structural information of networks, the normalized Laplacian matrix L is redefined as follows:

$$L = T^{-1}P, \quad (3)$$

where the adjacency matrix A used in the conventional normalized Laplacian matrix is replaced by the potential component matrix P and the degree matrix D by the topology potential matrix T .

The topology potential matrix T is an n -dimensional diagonal matrix, and the diagonal element $t_{i,i} = \varphi(v_i)$, that is, the topology potential of node v_i .

The potential component matrix P is $n \times n$ matrix, and the matrix elements $p_{i,j}$ are the potential component produced by node v_j at the position node v_i , which is defined as follows:

$$p_{i,j} = m(v_j) \times e^{-(d_{ij}/\sigma)^2}, \quad 1 \leq i, j \leq n, \quad (4)$$

where $m(v_j)$ is the mass of node; d_{ij} is the hops between node v_j and node v_i ; σ is an impact factor used to control the influence scope of node. If $i = j$, then $p_{i,j} = 0$; if node v_i is out of the influence scope ($\lfloor 3\sigma/\sqrt{2} \rfloor$) of node v_j , then $p_{i,j} = 0$.

Figure 1 shows a simple network model, which contains only six nodes. Here, we take the figure as an example to show the construction of the potential component matrix P and topology potential matrix T . For this network, the selected optimal impact factor $\sigma = 1.39$; thus, the influence scope of node $\lfloor 3\sigma/\sqrt{2} \rfloor = 2$. We can use formula (1) to get the topology

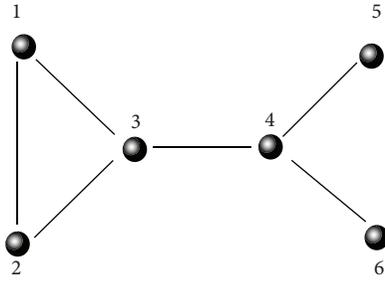


FIGURE 1: A simple network model.

potential of all the six nodes. The topology potential of node 1 is 1.3181, and the topology potentials of the other five nodes are 1.3181, 2.0402, 2.0402, 0.8413, and 0.8413, respectively. Thus, we can get the topology potential matrix T of Figure 1:

$$T = \begin{pmatrix} 1.3181 & & & & & \\ & 1.3181 & & & & \\ & & 2.0402 & & & \\ & & & 2.0402 & & \\ & & & & 0.8413 & \\ & & & & & 0.8413 \end{pmatrix}. \tag{5}$$

The topology potential of node 1 is the summation of potential component produced by node 2 (0.5960), node 3 (0.5950), and node 4 (0.1261). Similarly, the topology potential of node 3 is the summation of potential component produced by node 1 (0.5960), node 2 (0.5960), node 4 (0.5950), node 5 (0.1261), and node 6 (0.1261). Thus, we can get the potential component matrix P :

$$P = \begin{pmatrix} 0 & 0.5960 & 0.5960 & 0.1261 & 0 & 0 \\ 0.5960 & 0 & 0.5960 & 0.1261 & 0 & 0 \\ 0.5960 & 0.5960 & 0 & 0.5960 & 0.1261 & 0.1261 \\ 0.1261 & 0.1261 & 0.5960 & 0 & 0.5960 & 0.5960 \\ 0 & 0 & 0.1261 & 0.5960 & 0 & 0.1261 \\ 0 & 0 & 0.1261 & 0.5960 & 0.1261 & 0 \end{pmatrix}. \tag{6}$$

Based on formula (2), the normalized Laplacian matrix L is

$$L = T^{-1}P = \begin{pmatrix} 0 & 0.4521 & 0.4521 & 0.0957 & 0 & 0 \\ 0.4521 & 0 & 0.4521 & 0.0957 & 0 & 0 \\ 0.2921 & 0.2921 & 0 & 0.2921 & 0.0618 & 0.0618 \\ 0.0618 & 0.0618 & 0.2921 & 0 & 0 & 0.2921 \\ 0 & 0 & 0.1487 & 0.7026 & 0 & 0.1487 \\ 0 & 0 & 0.1487 & 0.7026 & 0.1487 & 0 \end{pmatrix}. \tag{7}$$

4.2. *Local Maximum Potential Node Search.* The hill-climbing method is a traditional algorithm for local maximum point search, which may leave out some local maximum points, and search performance is greatly influenced by initial point selection. We give out a new local maximum potential node search algorithm with review to local maximum potential nodes' characteristics.

The key steps of the new search algorithm are shown as follows.

- (1) All network nodes are initialized to "unvisited."
- (2) Randomly choose an "unvisited" node v_i and compare the topology potential of v_i with its neighbors'. If the topology potential of v_i is higher than all neighbors', then jump to step (3); otherwise, jump to step (4).
- (3) Add v_i to the local maximum potential node set K and mark v_i as well as its all neighbors "visited."
- (4) Mark v_i "visited," and mark neighbors with lower topology potential than v_i 's "visited."
- (5) Repeat steps (2), (3), and (4), until all nodes in network are marked "visited."
- (6) If there are two local maximum potential nodes whose distance, that is, hops, is smaller than $\lfloor 3\sigma/\sqrt{2} \rfloor$, then we delete the smaller one from K .
- (7) Output the final local maximum potential node set K .

More details about local maximum potential node search can be referred to [24].

5. Simulation Experiments

In this section, a series of experiments will be carried out to empirically evaluate the performance of the new algorithm. Simulation program was implemented with MATLAB. The experiment data include two kinds of complex networks: artificial networks and real world networks. The artificial networks were generated by ad hoc model [26] and LFR Benchmark generator [27]. LFR Benchmark is a network generator, which produces networks with power-law degree distribution and with implanted communities within the network [27]. The real world networks come from <http://www-personal.umich.edu/~mejn/netdata/>. The normalized mutual information (NMI) [28], a widely used measure, is calculated for the community partition by each algorithm.

5.1. *Ad Hoc Network.* The generated ad hoc network, with 128 nodes, is split into 4 communities containing 32 nodes each. The parameter z_{out} is the average edge that links one node with other nodes of different communities. As z_{out} increases, the community structure of the ad hoc network becomes ambiguous gradually. In the experiment, we changed z_{out} from 0 to 8 and observed the corresponding NMI produced by six methods: our algorithm, traditional spectral method, the k -means based on diffusion distance (DD k -means) [26], the k -means based on dissimilarity index (DI k -means) [26], Fast Newman algorithm [29], and Extremal Optimization method [30].

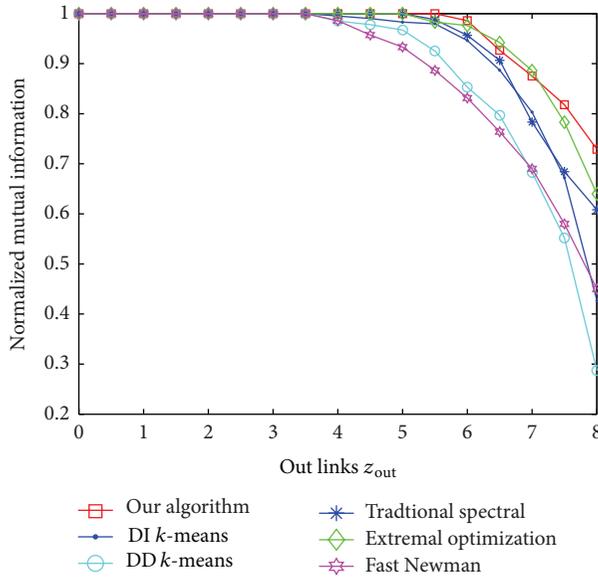


FIGURE 2: The NMI of the six methods with the change of z_{out} .

The experiment results are shown in Figure 2, where y -axis represents the value of NMI, and each point represents an average 30 simulation experiments. Compared with the other five methods, our algorithm is only slightly worse than the Extremal Optimization method for $6.4 < z_{out} < 7.1$. Our algorithm has a good performance for the ad hoc network, and the accurate rate is more than 98% for $z_{out} < 5.5$.

5.2. LFR Network. In generated LFR networks, the node degree and community size distribute according to power law. A mixing parameter μ is defined as the ratio between the external degree of a node with respect to its community and the total degree of the node [26], $0 \leq \mu \leq 1$. As μ increases, the community structure of the LFR network becomes ambiguous gradually. There are many other parameters used to control the generated LFR networks: the number of nodes N , the average node degree k , the maximum node degree max_k , the minimum community size min_c , and the maximum community size max_c [26].

In our experiments, we changed μ from 0 to 0.8 and observed the corresponding NMI produced by seven methods: our algorithm, traditional spectral method, Danon algorithm, Louvain algorithm, Infomap algorithm, Clique Percolation algorithm [28], and Fast Newman algorithm [29]. We used the default parameter configuration where $N = 1000$, $k = 15$, $max_k = 50$, $min_c = 20$, and $max_c = 50$.

The experiment results are shown in Figure 3, where y -axis represents the value of NMI. Compared with other six algorithms, our algorithm performs quite well, and its accuracy is only slightly worse than that of the Clique Percolation, Louvan, and Infomap in the case of $0.25 < \mu < 0.45$. Because of the complexity of topology potential distribution in the topology potential field, local maximum potential nodes may not necessarily be the real central nodes of

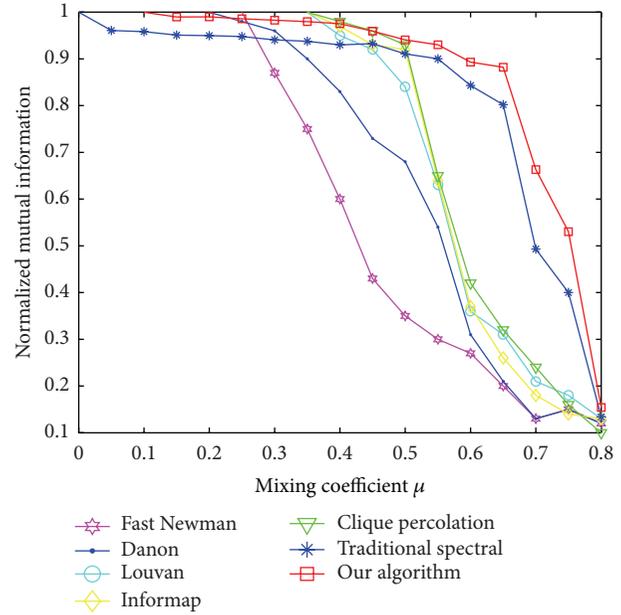


FIGURE 3: The NMI of the seven methods with the change of μ .

communities in some cases, resulting in the split or merger of some actual communities and the fluctuation of NMI value.

5.3. American College Football Network. The American College Football network [18] contains 115 teams, among which 616 games were carried out. In the network, nodes represent teams and edges games. All teams are organized into 12 conferences, and each of which contains about 8–12 teams. These 12 conferences are Atlantic Coast, Big East, Big Ten, Big Twelve, Conference USA, Independents, Mid American, Mountain West, Pacific Ten, Southeastern, Sun Belt, and Western Athletic.

We compared our algorithm with other three algorithms, including the traditional spectral algorithm, the spectral algorithm based on modularity Q [18], and CMITP (community members identification based on topology potential) [22].

Firstly, we compared our algorithm with the traditional spectral algorithm. The latter cannot obtain the football network community number from the ladder distribution of eigenvector elements; therefore, we set its community number the same as our method. Figures 4 and 5 show the community detection results by our algorithm and traditional spectral algorithm, respectively. Each node represents a competing team, using its name as label. The teams in same community are marked the same color. For this network, the traditional spectral algorithm gets six correct communities: Mountain West, Atlantic Coast, Southeastern, Pacific Ten, Big Ten, and Conference USA. Compared with the traditional spectral algorithm, our algorithm gets three new correct communities: Big Twelve, Big East, and Mid American. For the conference Western Atlantic, our algorithm gets 9 correct teams, with only 1 team missing. Both algorithms split the conference Sun Belt and Independents.

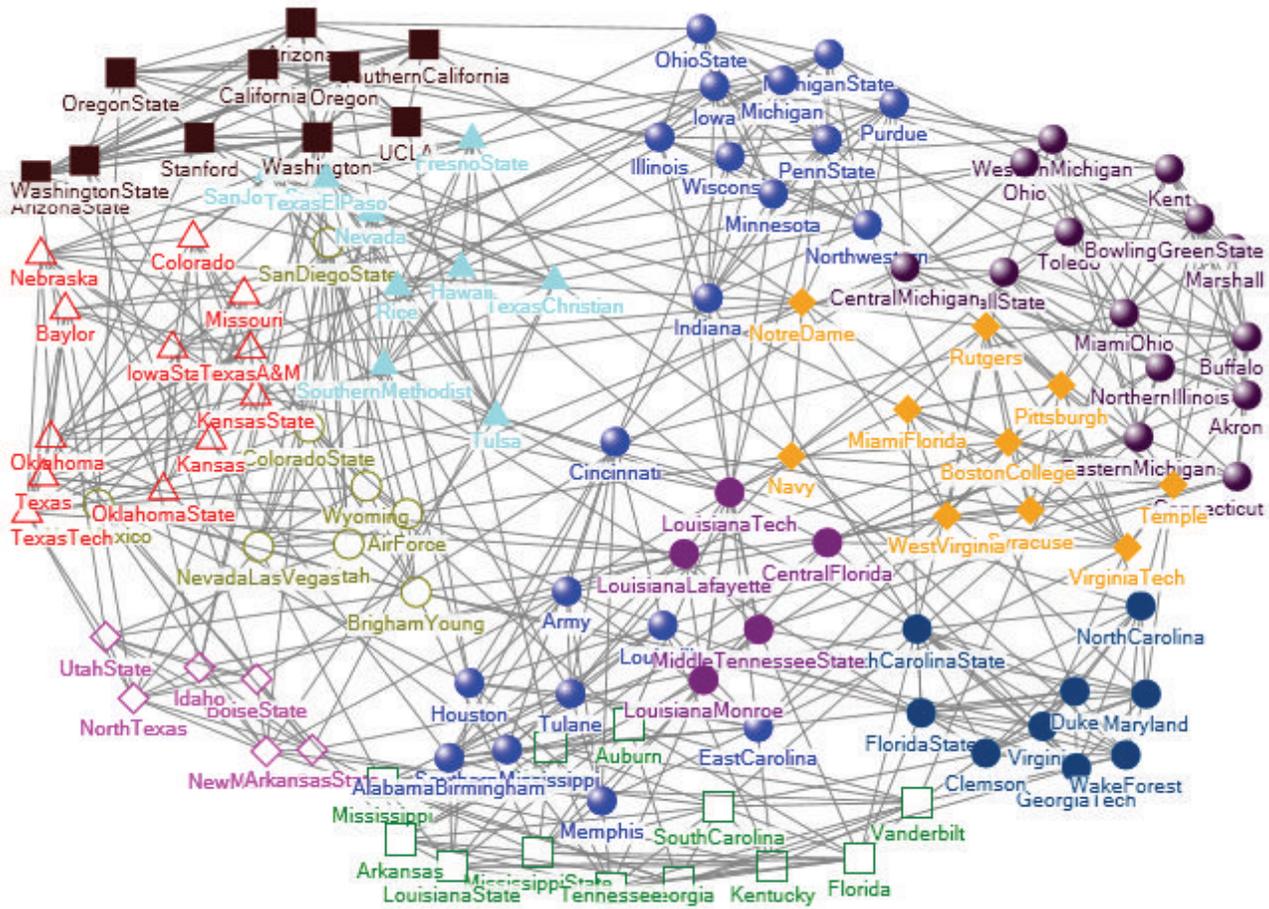


FIGURE 4: The community detection results by our algorithm.

Secondly, we compared our algorithm with the spectral algorithm based on modularity. Tables 1 and 2 show the results of our algorithm and the spectral algorithm based on modularity, respectively. The conference names are listed in the leftmost column, and columns $a \sim k$ represent the communities found by the two algorithms. Each found community consists of teams from one or more conferences as indicated by the numbers in the corresponding column [18]. The spectral algorithm based on modularity divided this network into 10 communities, and six communities are correctly detected: Atlantic Coast, Big East, Big Ten, Big Twelve, Mid American, and Pacific Ten. Compared with the spectral algorithm based on modularity, our algorithm found 11 communities and got a new correct communities Mountain West.

The CMITP method divided this network into 17 communities, and there are many overlapping nodes between communities, such as nodes “Hawaii” and “Nevada.” Table 3 shows the community number, Q and NMI of four different algorithms. Compared with other three methods, our algorithm got the highest NMI 0.9292. In addition, our

TABLE 1: The community detection results of our algorithm.

	a	b	c	d	e	f	g	h	i	j	k
Atlantic Coast					9						9
Big East						8					8
Big Ten		11									11
Big Twelve									12		12
Conference USA	9						1				10
Independents				1	2	1			1		5
Mid American				13							13
Mountain West										8	8
Pac Ten								10			10
Southeastern			12								12
Sun Belt							3			4	7
Western Atlantic	9									1	10
	9	20	12	14	9	10	5	10	12	6	115

algorithm found 11 communities, which is the closest to the real community number 12.

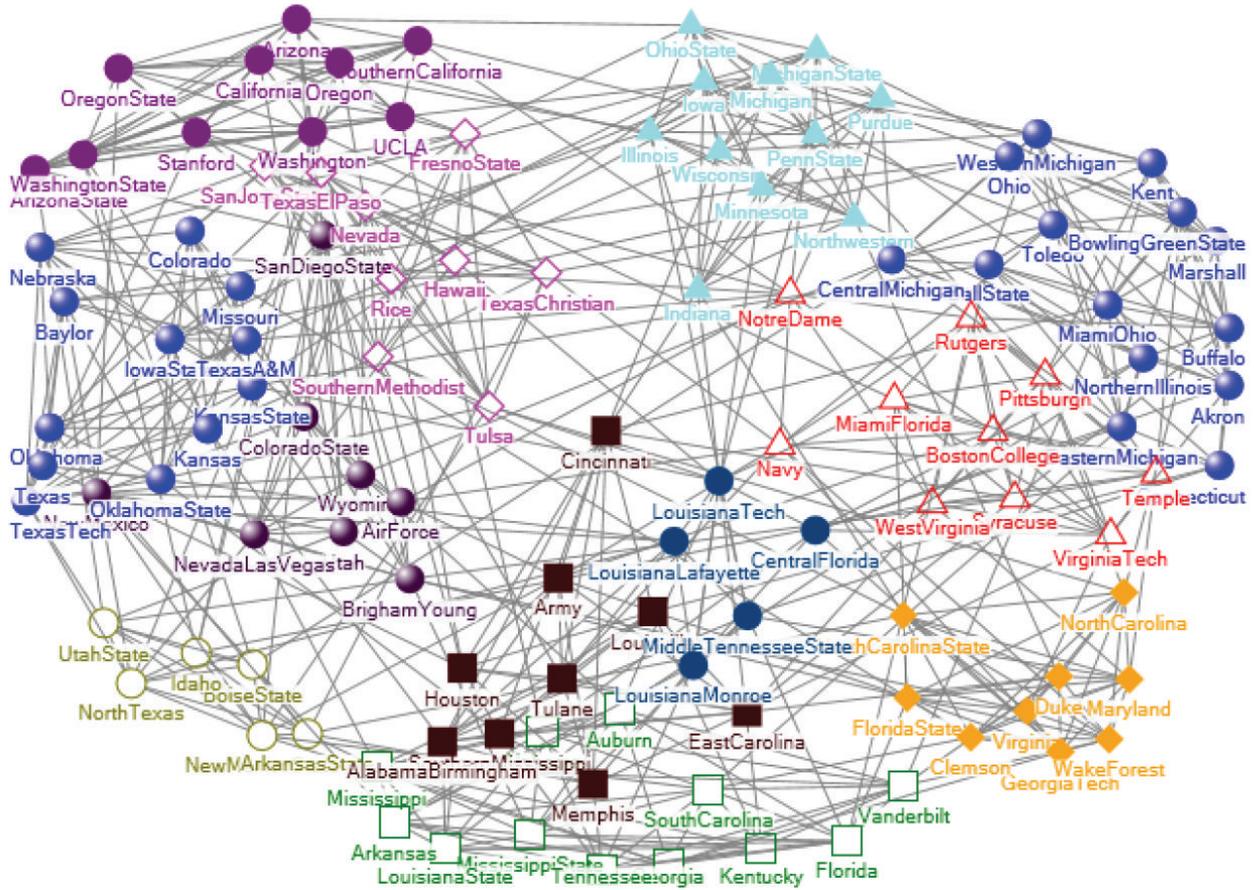


FIGURE 5: The community detection results by the traditional spectral algorithm.

TABLE 2: The community detection results of the spectral algorithm based on modularity.

	a	b	c	d	e	f	g	h	i	j
Atlantic Coast							9			9
Big East				8						8
Big Ten		11								11
Big Twelve			12							12
Conference USA						1			9	10
Independents				2				2	1	5
Mid American								13		13
Mountain West									8	8
Pacific Ten	10									10
Southeastern					12					12
Sunbelt				3					4	7
Western Atlantic				1	8				1	10
	10	11	12	10	16	9	9	15	9	14
										115

5.4. The Influence of Impact Factor σ on Algorithm Performance. The impact factor σ will affect topology potential field and the influence scope of node. With different impact factor σ , the distribution of topology potential value will be different. These changes may bring out different community

TABLE 3: The community number, Q and NMI of four algorithms.

	Community number	Q	NMI
The real community	12	0.5540	1.0000
Our algorithm	11	0.5879	0.9292
Traditional spectral	11	0.5792	0.8879
Spectral based on modularity	10	0.5870	0.8800
CMITP	17	0.5538	—

detecting results. We take a real world network, the Zachary karate club network, to analyze the influence of impact factor σ on algorithm performance. Figure 6 shows the NMI of our algorithm with different impact factor σ .

Figure 6 shows that if $\sigma \leq 0.4716$, the NMI is 0; if $0.4716 < \sigma \leq 1.66$, the NMI is 1; if $1.66 < \sigma \leq 1.90$, the NMI is 0.8372; if $1.90 < \sigma \leq 1.934$, the NMI is 0.6459; if $\sigma > 1.934$, the NMI is 0.1701. The analysis is as follows. The maximum influence scope of node is $\lfloor 3\sigma/\sqrt{2} \rfloor$ hops. When $\sigma \leq 0.4716$, the influence scope of node $\lfloor 3\sigma/\sqrt{2} \rfloor = 0$; it means that all nodes are isolated and have same topology potential value. For Zachary network, the optimal σ is 1.02 according to formula (2). When $0.4716 < \sigma \leq 1.66$, we can detect accurate

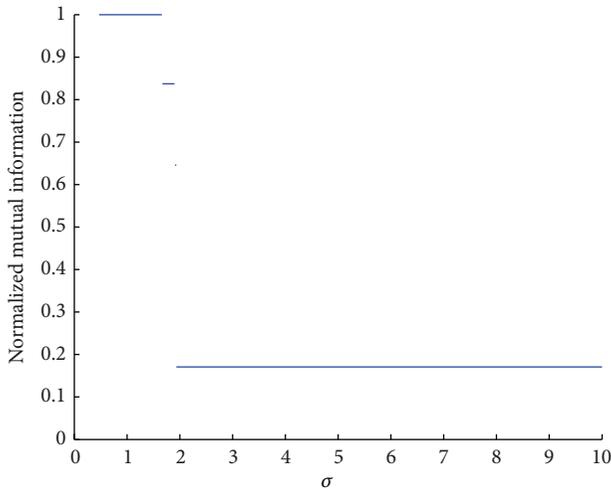


FIGURE 6: The influence of σ on algorithm performance.

community structure, and the NMI is 1. But as σ further increases, one node can associate with almost all the other nodes. In this case, the distribution of topology potential value cannot truly reflect the structure characteristics of network; therefore, the community detecting results are bad. In a word, as long as the impact factor σ is set near the optimal value, our algorithm can get good outcomes.

6. Conclusion

Identifying community structure is crucial for understanding complex networks. Recently, spectral clustering algorithms have been successfully applied in community detection. The traditional spectral clustering methods cannot provide sufficient structural information for community detection and cannot always get the community number from the ladder distribution of eigenvector elements. Aiming at these inadequacies, this paper puts forward a novel community detection algorithm based on topology potential and spectral clustering. The new algorithm constructs the normalized Laplacian matrix with network nodes' topology potential, which contains rich structural information of the network. In addition, the new algorithm can automatically judge the optimal community number from the local maximum potential nodes. Experiments on ad hoc network, LFR network, and the American college football network showed that the new algorithm can improve the accuracy of community detection and has significant adaptability.

Conflict of Interests

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

Acknowledgment

This work was supported by the Fundamental Funds for the Central Universities (2014QNB23).

References

- [1] Z. Wang, Y. Zhao, Z. Chen, and Q. Niu, "An improved topology-potential-based community detection algorithm for complex network," *The Scientific World Journal*, vol. 2014, Article ID 121609, 7 pages, 2014.
- [2] R. W. Myster, "A refined methodology for defining plant communities using postagricultural data from the neotropics," *The Scientific World Journal*, vol. 2012, Article ID 365409, 9 pages, 2012.
- [3] G. Palla, A. Barabási, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.
- [4] W.-D. Zhou and L. Nakhleh, "Convergent evolution of modularity in metabolic networks through different community structures," *BMC Evolutionary Biology*, vol. 12, no. 1, article 181, 2012.
- [5] G. K. Orman, V. Labatut, and H. Cherifi, "Comparative evaluation of community detection algorithms: a topological approach," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2012, no. 8, Article ID P08001, 2012.
- [6] X. Ma and L. Gao, "Non-traditional spectral clustering algorithms for the detection of community structure in complex networks: a comparative analysis," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2011, no. 5, Article ID P05012, 2011.
- [7] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [8] S. Chauhan, M. Girvan, and E. Ott, "Spectral properties of networks with community structure," *Physical Review E*, vol. 85, no. 2, Article ID 029906, 10 pages, 2012.
- [9] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, "Synchronization reveals topological scales in complex networks," *Physical Review Letters*, vol. 96, Article ID 114102, 4 pages, 2006.
- [10] X.-Q. Cheng and H.-W. Shen, "Uncovering the community structure associated with the diffusion dynamics on networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2010, no. 4, Article ID P04024, 2010.
- [11] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [12] H.-W. Shen, X.-Q. Cheng, and B.-X. Fang, "Covariance, correlation matrix, and the multiscale community structure of networks," *Physical Review E*, vol. 82, Article ID 016114, 2010.
- [13] S. Hua-Wei and C. Xue-Qi, "Spectral methods for the detection of network community structure: a comparative analysis," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2010, no. 10, Article ID P10020, 2010.
- [14] M. Zarei and K. A. Samani, "Eigenvectors of network complement reveal community structure more accurately," *Physica A: Statistical Mechanics and Its Applications*, vol. 388, no. 8, pp. 1721–1730, 2009.
- [15] M. Zarei, K. A. Samani, and G. R. Omid, "Complex eigenvectors of network matrices give better insight into the community structure," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, Article ID P10018, 2009.
- [16] D. Mavroeidis, "Accelerating spectral clustering with partial supervision," *Data Mining and Knowledge Discovery*, vol. 21, no. 2, pp. 241–258, 2010.
- [17] X. Ma, L. Gao, and X. Yong, "Eigenspaces of networks reveal the overlapping and hierarchical community structure more precisely," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2010, no. 8, Article ID P08012, 2010.

- [18] X. Gong, K. Li, M.-H. Li, and C.-H. Lai, "A spectral algorithm of community identification," *EPL (Europhysics Letters)*, vol. 101, no. 4, Article ID 48001, 2013.
- [19] M. E. J. Newman, "Spectral methods for community detection and graph partitioning," *Physical Review E*, vol. 88, Article ID 042822, 2013.
- [20] Y. Bo, J. Liu, and J. Feng, "On the spectral characterization and scalable mining of network communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 2, pp. 326–337, 2012.
- [21] W. Y. Gan, N. He, D. Y. Li, and J. M. Wang, "Community discovery method in networks based on topological potential," *Journal of Software*, vol. 20, no. 8, pp. 2241–2254, 2009.
- [22] Y. Han, D. Li, and T. Wang, "Identifying different community members in complex networks based on topology potential," *Frontiers of Computer Science in China*, vol. 5, no. 1, pp. 87–99, 2011.
- [23] J. Zhang, H. Li, J. Yang, J. Bai, L. Zhang, and Y. Chu, "Variable scale network overlapping community identification based on identity uncertainty," *Acta Electronica Sinica*, vol. 40, no. 12, pp. 2512–2518, 2012.
- [24] Z.-X. Wang, Z.-T. Chen, Y. Zhao, and Q. Niu, "A novel local maximum potential point search algorithm for topology potential field," *International Journal of Hybrid Information Technology*, vol. 7, no. 2, pp. 1–8, 2014.
- [25] Z.-X. Wang, S.-X. Xia, and Q. Niu, "A novel ontology analysis tool," *Applied Mathematics & Information Sciences*, vol. 8, no. 1, pp. 255–261, 2014.
- [26] J. Liu, "Comparative analysis for k-means algorithms in network community detection," in *Advances in Computation and Intelligence*, vol. 6382 of *Lecture Notes in Computer Science*, pp. 158–169, Springer, 2010.
- [27] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Physical Review E*, vol. 80, no. 1, Article ID 016118, 2009.
- [28] H. J. Li, J. Zhang, Z. P. Liu, L. Chen, and X. S. Zhang, "Identifying overlapping communities in social networks using multi-scale local information expansion," *The European Physical Journal B*, vol. 85, no. 6, article 190, pp. 190–198, 2012.
- [29] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, Article ID 066133, pp. 1–66133, 2004.
- [30] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical Review E*, vol. 72, no. 2, Article ID 027104, 2005.