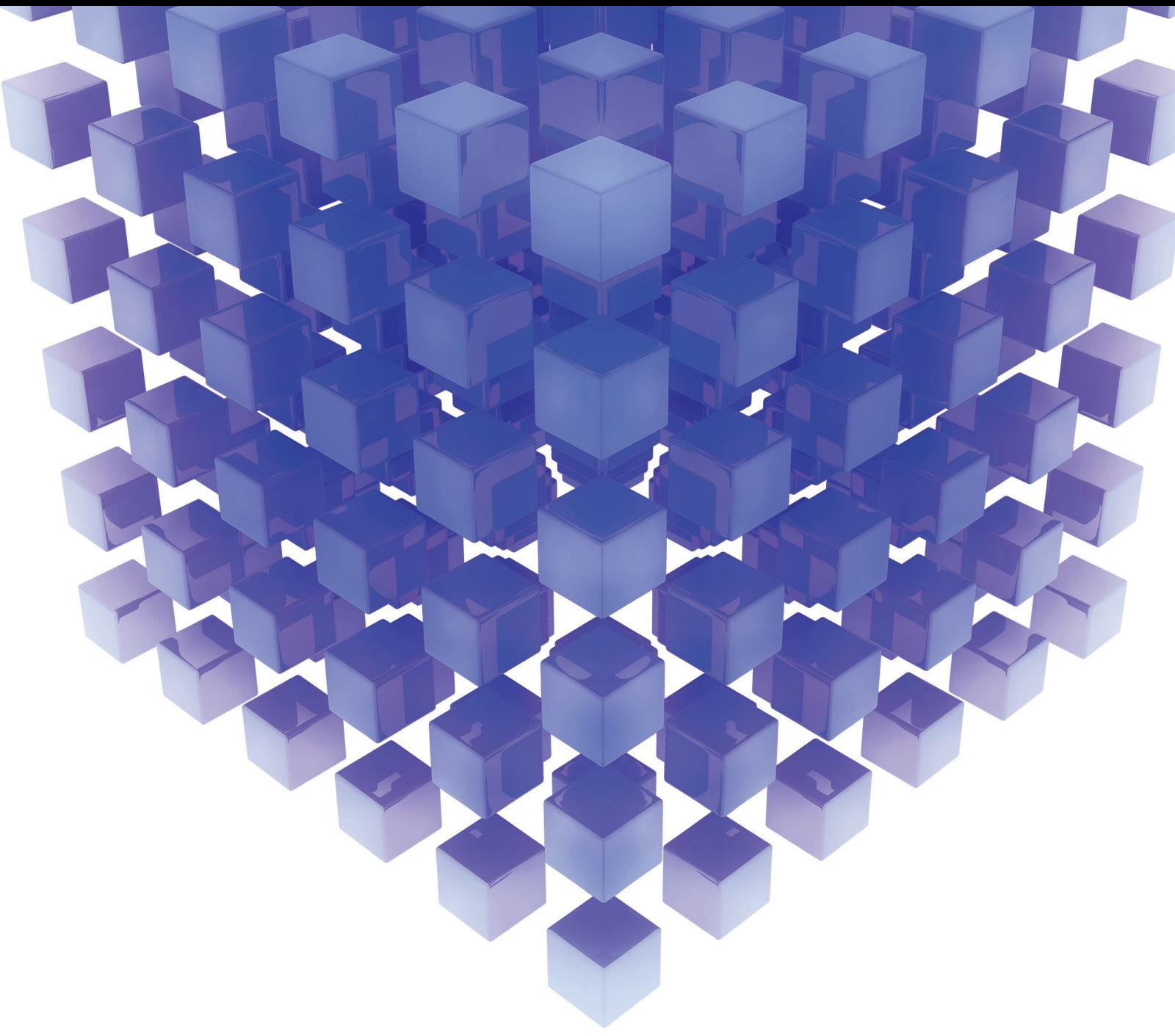
Mathematical Problems in Engineering

Mathematical Modeling and Analysis of Soft Computing

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



Mathematical Modeling and Analysis of Soft Computing

Mathematical Problems in Engineering

Mathematical Modeling and Analysis of Soft Computing

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

Editorial Board

Contents

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 578321, 2 pages
http://dx.doi.org/10.1155/2015/578321

Editorial

Mathematical Modeling and Analysis of Soft Computing

Shifei Ding,[1,2] Zhongzhi Shi,[2] Ke Chen,[3] and Ahmad Taher Azar[4]

[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 Science, Beijing 100190, China*
[3] *Centre for Mathematical Imaging Techniques, Department of Mathematical Sciences, University of Liverpool, Liverpool L69 7ZL, UK*
[4] *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

Welcome to this special issue. Soft computing is one of the hot research fields in advanced artificial intelligence, while mathematical modeling and analysis (MMA) plays key role in soft computing. This special issue aims to promote the research, development, and applications of MMA for soft computing 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 from among the highest rated papers in submitted manuscripts. The selection of papers featured here cover the topics of the main soft computing theories and experimental studies of some application systems.

Truss layout optimization is a typical difficult constrained mixed-integer nonlinear program. A. Xiao et al. present fitness estimation based particle swarm optimization algorithm with an adaptive penalty function approach (FEPSO-AP). The evaluation of particles is partly substituted by the estimation of similar particles and the iteration information is merged into the penalty function to find a good balance between the exploration and exploitation of the constrained design domain. The present method solves some problems as how we can simplify truss layout optimization and the computational cost of truss analysis. Flower pollination algorithm (FPA) is a new nature-inspired intelligent algorithm which uses the whole update and evaluation strategy on solutions. This algorithm may deteriorate the convergence speed and the quality of solution due to interference phenomena among dimensions for solving multidimension function optimization problems. To overcome this shortage, R. Wang and Y. Zhou point out a dimension by dimension improvement based flower pollination algorithm.

In view of the small sample classification problem, a neural network ensemble optimized classification algorithm based on partial least squares (PLS) and ordinary least squares (OLS) is proposed by W. Jia et al. This algorithm can reduce the feature dimension of small sample data and improve the recognition precision of classification system. X. Xiang and J. Wu build a framework of moment neuronal networks with intra- and inter-interactions that shows how the spontaneous activity is propagated across the homogeneous and heterogeneous network. The application of chemical flooding in petroleum reservoirs turns into hot topic of the recent researches. In order to improve the estimation efficiency of chemical flooding in oil reservoirs, M. A. Ahmadi put forward a predictive model based on swarm intelligence and artificial neural network (ANN).

Multilevel association rules mining is an important domain to discover interesting relations between data elements with multiple levels abstractions. Y. Xu et al. describe novel genetic based multilevel association rules mining to avoid the excessive computation in the big data. The authors use the category tree to describe the multilevel application data sets as the domain knowledge, then put forward a special tree encoding schema based on the category tree to build the heuristic multilevel association mining algorithm, and lastly propose the genetic algorithm based on the tree encoding schema. N. H. Moin et al. propose a hybrid genetic algorithm with multiparents crossover for job shop scheduling problem

that is one of the well-known hard combinatorial scheduling problems. This algorithm embeds a schedule generation procedure to reduce the search space and ensures the sustainability. Y.-H. Kim and Y. Yoon propose a new distance metric, based on the linkage of genes, in the search space of genetic algorithms. This second-order distance measure is derived from the gene interaction graph and first-order distance, which is a natural distance in chromosomal spaces.

J. Chai et al. design empirical mode decomposition least square support vector machine (EMD LSSVM) model to analyze the CSI 300 index. The EMD-LSSVM model with grid search (GS) algorithm has good performance compared with other optimization methods including simplex, GS, particle swarm optimization (PSO), and genetic algorithm (GA) in predicting stock market movement direction. Focusing on the bifurcation analysis of an SIS epidemic model with bilinear incidence rate and saturated treatment, Y. Xiao et al. point out the global dynamics of an SIS model with bilinear incidence rate and saturated treatment function which shows the effect of delayed treatment when the rate of treatment is lower and the number of infected individuals is getting larger. Z. Shi and Z. Ji point out a new two-stage least squares iterative algorithm to identify the parameters of the identification of Hammerstein finite impulse response moving average (H-FIR-MA) systems. Compared with other methods, the present method can obtain highly accurate parameter estimates and fast convergence rate. Accurate identification of the aggregation residues could help fully decipher the molecular mechanisms. In order to predict aggregation residues from the sequence information, B. Liu et al. introduce two computational approaches: Aggre_Easy and Aggre_Balance. The prediction of aggregation prone sites plays a crucial role in the research of drug-targets.

V. Dobrić et al. present formalization of prototype theory of categorization that is accomplished by using Boolean consistent fuzzy logic—interpolative Boolean algebra. Proposed formalism secures the principles of categorization and shows that Boolean laws are fundamental in the categorization process. H. Wang et al. propose a novel hybrid approach based on mutual information and improved gravitational search algorithm to improve the efficiency and accuracy of CTQs identification. The results of experiment show that the present method has a strong search capability. Accelerated life test is commonly used for the estimation of high-reliability product; the article by G. Wang et al. titled "Accelerated Lifetime Data Analysis with a Nonconstant Shape Parameter" explores a simple and efficient approach to estimate the coefficients of acceleration models by using weight least square method.

The traditional social cognitive optimization (SCO) is not guaranteed to converge to the global optimization solution with probability one. J. Sun et al. design an improved SCO algorithm which is guaranteed to converge to the global optimization solution and has better stable optimization outcomes than the traditional SCO for nonlinear programming problems. The convergence proof for the improved SCO based on Solis and Wets' research results is given. F.-A. Deng et al. propose an improved harmony search algorithm based on NGHS algorithm (INGHS). The novel harmony search algorithm has a good performance for optimization problems into which nine construction examples of algebra system are converted. Global best strategy and dynamic parameters adjustment are employed in INGHS.

Many scholars have made significant contributions to research, education, and development of soft computing. This special issue is dedicated to those pioneers and scientific and technical workers in research in this area.

Acknowledgments

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. Finally, we would like to thank all those who helped to make this special issue possible, especially the authors and the reviewers of the articles.

Shifei Ding
Zhongzhi Shi
Ke Chen
Ahmad Taher Azar

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 680624, 6 pages
http://dx.doi.org/10.1155/2015/680624

Research Article

Linkage-Based Distance Metric in the Search Space of Genetic Algorithms

Yong-Hyuk Kim[1] and Yourim Yoon[2]

[1]*Department of Computer Science & Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 139-701, Republic of Korea*
[2]*Department of Computer Engineering, College of Information Technology, Gachon University, 1342 Seongnam-daero, Sujeong-gu, Seongnam-si, Gyeonggi-do 461-701, Republic of Korea*

Correspondence should be addressed to Yourim Yoon; yryoon@gachon.ac.kr

Received 31 July 2014; Accepted 7 September 2014

Academic Editor: Shifei Ding

We propose a new distance metric, based on the linkage of genes, in the search space of genetic algorithms. This second-order distance measure is derived from the gene interaction graph and first-order distance, which is a natural distance in chromosomal spaces. We show that the proposed measure forms a metric space and can be computed efficiently. As an example application, we demonstrate how this measure can be used to estimate the extent to which gene rearrangement improves the performance of genetic algorithms.

1. Introduction

Distance metrics are fundamental tools for organizing search spaces, because the introduction of a metric is the simplest way to induce a topology [1]. Different metrics produce different topologies and thus change the shape of the search space. When a space is to be searched by a genetic algorithm (GA), a good distance metric facilitates navigation of the space [2–5] and can also improve the effectiveness of search [6–12]. Hamming distance is a popular metric in a discrete space that is to be searched by a GA. Hamming distance has also been widely used in analyses of solution spaces [13–15].

Fitness distance correlation (FDC), proposed by Jones and Forrest [14], is a measure of the effectiveness of a distance metric in a space to be searched by a GA. An FDC is obtained by measuring the correlation between fitness and the distance to the nearest global optimum for a number of sample solutions. FDC coefficients range from −1 to 1, where higher values suggest increased difficulty in maximizing fitness and decreased difficulty in minimizing fitness. When a GA is hybridized with a local optimization, the population consists entirely of local optima, and it is then more useful to determine FDCs of local-optimum spaces.

In this paper, we propose a new distance measure which takes account of gene interaction and show that it forms a metric space. We use this metric to compute FDCs of search space and show that FDCs obtained in this way have improved correlation with the improvement in GA performance that can be obtained by gene rearrangement. The remainder of this paper is organized as follows. In Section 2, we review gene rearrangement in GAs. In Section 3, we propose a new distance measure for GAs, show that it forms a metric space, and demonstrate an application. Finally, we draw conclusions in Section 4.

2. Gene Rearrangement

Holland's schema theorem [16] shows that schemata (i.e., groups of genes) with high fitness, short defining length, and low order have high probabilities of survival in a standard GA.

These durable schemata are called *building blocks*. They make a major contribution to fitness and have a high degree of mutual interaction. The performance of a GA is strongly dependent on the survival and reproduction of these building blocks.

The survival probability of a gene group through a crossover is strongly affected by the positions of genes in the chromosome. Schemata consisting of genes in scattered positions tend to be too long to survive. Thus, the strategy used for placing genes significantly affects the performance of a GA. Inversion is an operator which changes the location of genes while a GA is running [17], and the process of rearranging genes dynamically to improve performance is called *linkage learning* [18]. Messy GA [19] is an example of a technique that implicitly uses dynamic gene rearrangement.

It has been observed that the performance of GAs on problems with a locus-based encoding can be improved by rearranging the indices of the genes before running the GA. Static gene rearrangement was first suggested by Bui and Moon [20, 21], who rearrange genes within a chromosomal representation to improve the quality of schemata and to help the GA to preserve the better schemata. Many studies on the static rearrangement of gene positions [20–24] have showed performance improvements. However, the improvement in performance achieved in this way has been shown to vary greatly between problem instances. This motivated us to develop a distance metric to improve our ability to estimate how much improvement in the performance of a GA on a particular problem instance can be expected through gene rearrangement.

3. A Linkage-Based Distance Measure

3.1. Second-Order Distance Measure. The most usual first-order distance measure in discrete space is the Hamming distance which is also a natural distance in chromosomal space, although there are other first-order distance measures, such as the quotient metric in redundant encoding [11]. We now define a second-order distance measure derived from first-order distance. Given a problem instance p, consider the unweighted undirected graph G_p representing first-order gene interaction [23], which is the pairwise interaction of genes. For convenience, we will assume that each gene has an interaction with itself, so that $\{g, g\} \in E(G_p)$ for each gene $g \in V(G_p)$. Let A_p be the adjacency matrix of G_p and consider A_p as a binary matrix over $\mathbb{Z}_2$ [25–27].

Definition 1. Suppose that the inverse of A_p exists as a binary matrix over $\mathbb{Z}_2$; that is, $A_p \in GL_n(\mathbb{Z}_2)$. One defines the second-order distance measure $d_p^{(2)}$ as follows:

$$d_p^{(2)}(x, y) := \left\| A_p^{-1}(x \oplus y) \right\|, \tag{1}$$

where $\oplus$ is a vector summation operator, which performs a Boolean XOR (i.e., $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, and $1 + 1 = 0$) in each coordinate, and $\|\cdot\|$ is a norm derived from the first-order distance metric $d^{(1)}$ (i.e., $\|\cdot\| = d^{(1)}(\cdot, 0)$).

Theorem 2. *$d_p^{(2)}$ is a metric.*

Proof. It is enough to show the following four conditions [1].

(i) Nonnegativity: since $d_p^{(2)}(x, y) = d^{(1)}(A_p^{-1}(x \oplus y), 0)$ and $d^{(1)}$ is a metric, $0 \le d_p^{(2)}(x, y) < \infty$ for all x and y in X.

(ii) Identity of indiscernibles: consider

$$\begin{aligned} d_p^{(2)}(x, y) = 0 &\iff \left\| A_p^{-1}(x \oplus y) \right\| = 0 \\ &\iff A_p^{-1}(x \oplus y) = 0 \\ &\iff x \oplus y = 0 \\ &\iff x = y. \end{aligned} \tag{2}$$

(iii) Symmetry: consider

$$\begin{aligned} d_p^{(2)}(x, y) &= \left\| A_p^{-1}(x \oplus y) \right\| \\ &= \left\| A_p^{-1}(y \oplus x) \right\| \\ &= d_p^{(2)}(y, x). \end{aligned} \tag{3}$$

(iv) Triangle inequality: consider

$$\begin{aligned} &d_p^{(2)}(x, y) + d_p^{(2)}(y, z) \\ &\quad = \left\| A_p^{-1}(x \oplus y) \right\| + \left\| A_p^{-1}(y \oplus z) \right\| \\ &\quad \ge \left\| A_p^{-1}(x \oplus y) \oplus A_p^{-1}(y \oplus z) \right\| \\ &\quad = \left\| A_p^{-1}((x \oplus y) \oplus (y \oplus z)) \right\| \\ &\quad = \left\| A_p^{-1}((x \oplus z) \oplus (y \oplus y)) \right\| \\ &\quad = \left\| A_p^{-1}(x \oplus z) \right\| \\ &\quad = d_p^{(2)}(x, z). \end{aligned} \tag{4}$$

□

If the inverse of A_p does not exist, we can extend the scope of the distance metric using the following well-defined formulation:

$$d_p^{(2)}(x, y) := \min \left\| \left(\arg\min_z \left\| (x \oplus y) \oplus A_p z \right\| \right) \right\|. \tag{5}$$

We note that if the inverse of A_p exists, then $z := A_p^{-1}(x \oplus y)$, which implies $(x \oplus y) \oplus A_p z = 0$, and hence $\arg\min_z \|(x \oplus y) \oplus A_p z\| = A_p^{-1}(x \oplus y)$. Our second-order distance and its extension can be computed in $O(n^3)$ by a variant of Gauss-Jordan elimination [28], where n is the number of genes.

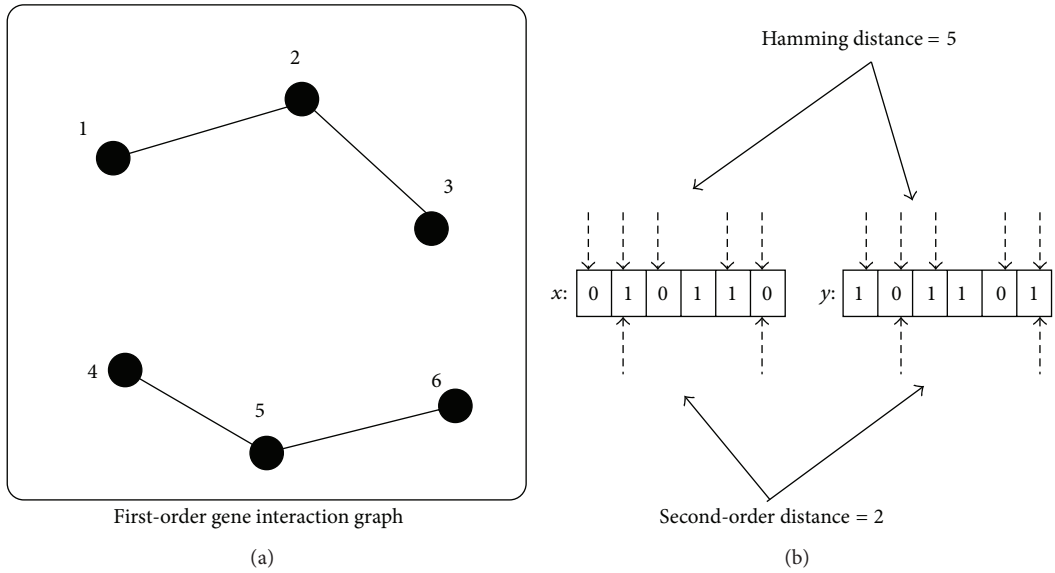


Figure 1: (a) An example of a first-order gene interaction graph G_p and (b) distances between two example chromosomes x and y.

3.2. An Application. Intuitively, our measure of the distance between two chromosomes can be understood as the minimum number of bits that must be changed to transform one chromosome into the other in the genetic process using optimal gene rearrangement.

Given an undirected graph $G = (V, E)$ with edge weights $(w_{ij})_{(i,j)\in E}$, the max-cut problem is that of finding a subset $S \subset V$ which maximizes the sum of the edge weights which traverse the cut $(S, V \setminus S)$ [29–31]. Consider the 6-node max-cut problem instance p, which is to maximize the following expression:

$$x_1 \oplus x_2 + x_2 \oplus x_3 - x_4 \oplus x_5 - x_5 \oplus x_6, \tag{6}$$

where a vertex v_i belongs to the position $x_i \in \{0, 1\}$ and $\oplus$ is the Boolean XOR operator. In this problem instance, edges $\{v_1, v_2\}$ and $\{v_2, v_3\}$ increase the fitness and edges $\{v_4, v_5\}$ and $\{v_5, v_6\}$ reduce the fitness. In the max-cut problem, we can consider that the given graph removing edge weights shows the first-order gene interaction (see, e.g., Figure 1(a)). Figure 1(b) shows an example in which the Hamming and second-order distances between two chromosomes x and y are obtained by optimal gene arrangement of the gene interaction graph G_p. In this example,

$$A_p = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

$$A_p^{-1} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \tag{7}$$

$x \oplus y = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}^T$, $A_p^{-1}(x \oplus y) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}^T$, and hence $\|A_p^{-1}(x \oplus y)\| = 2$. If we use the normalized Hamming distance (developed for the 2-grouping problem) [32, 33] as the first-order distance measure, the FDC of this problem is −0.50. But when our second-order distance is used, the FDC becomes −0.95.

Given a graph $G = (V, E)$ and its adjacency matrix $A = (a_{ij})$, the graph bipartitioning problem is that of minimizing the following expression:

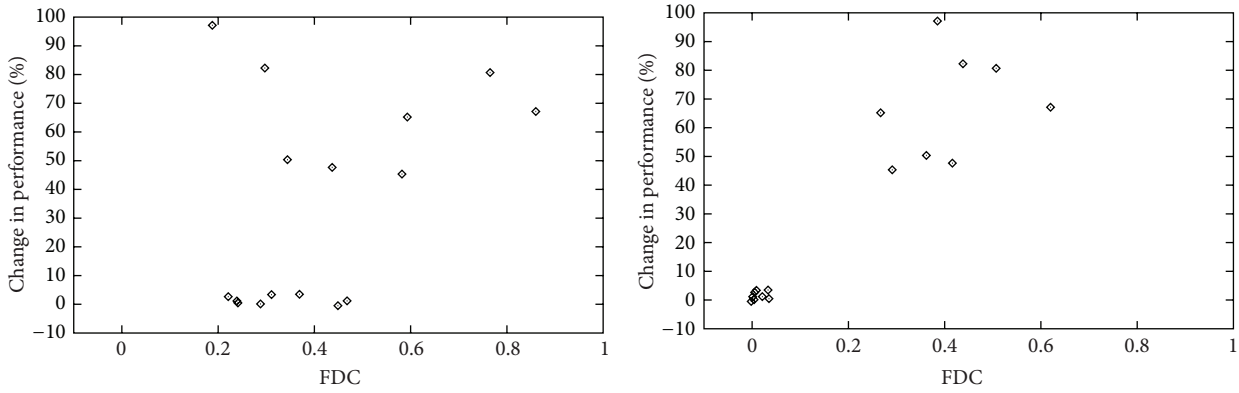
$$\frac{1}{2}\sum_{i=1}^{|V|}\sum_{j=1}^{|V|} a_{ij}\left(x_i \oplus x_j\right) + \gamma\left(\sum_{i=1}^{|V|} x_i - \frac{|V|}{2}\right)^2, \tag{8}$$

where $a_{ij} \in \{0, 1\}$, a vertex v_i belongs to the position $x_i \in \{0, 1\}$, and γ is a positive constant introduced to penalize unbalanced partitions. If we ignore the second balancing term altogether, we can regard the given graph as the first-order gene interaction graph of the given problem instance. Bui and Moon [21] tried gene rearrangement in a GA for graph bipartitioning and obtained dramatic improvements in performance for some graphs. We hypothesized that FDCs calculated using our second-order distance would help identify graphs that could benefit most from gene rearrangement, in terms of GA performance. Figure 2 shows the relationship

Table 1: Effect of gene rearrangement on FDCs computed using first- and second-order distance.

Graph	FDC with $d^{(1)}$	FDC with $d_p^{(2)}$	Improvement (%)† from [21]
G500.2.5	0.369	0.033	3.495
G500.05	0.449	−0.002	−0.487
G500.10	0.221	0.005	2.674
G500.20	0.288	0.004	0.117
G1000.2.5	0.241	0.035	0.469
G1000.05	0.239	0.001	1.167
G1000.10	0.311	0.009	3.362
G1000.20	0.468	0.021	1.201
U500.05	0.297	0.438	82.258
U500.10	0.437	0.416	47.649
U500.20	0.593	0.267	65.198
U500.40	0.860	0.620	67.143
U1000.05	0.188	0.385	97.144
U1000.10	0.344	0.362	50.340
U1000.20	0.582	0.291	45.341
U1000.40	0.765	0.507	80.668

†Change in GA performance obtained by gene rearrangement.



(a) First-order distance $d^{(1)}$ (b) Second-order distance $d_p^{(2)}$

Figure 2: Correlation of gene rearrangement with FDC values computed using first- and second-order distance.

between FDC and the performance improvement of a GA on 16 benchmark graphs (8 random graphs and 8 random geometric graphs) that were used in [34–40].

Here, the performance improvement means the difference in percentage between the average performances of a GA with and without gene rearrangement (data from [21]). The FDC values were approximated from 10,000 randomly generated local optima. When the first-order (normalized Hamming) distance was used, there was little correlation with the change in performance, but our second-order distance provided a clear correlation (see Figure 2(b) and Table 1).

4. Concluding Remarks

In most previous work, distances among chromosomes in GAs have usually been first-order distances, and in particular Hamming distance. We have proposed a second-order distance measure for GAs, which we consider to be more meaningful. We have showed that this distance measure forms a metric space and that it can be computed efficiently.

Using second-order distance allows us to see problem spaces from a different viewpoint. We have demonstrated its value in predicting the effectiveness of gene rearrangement, and we envisage it providing further understanding of the working mechanism of GAs.

Disclosure

A preliminary version of this paper appeared in the *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1393–1399, 2005.

Conflict of Interests

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

Acknowledgment

This research was supported by the Gachon University research fund of 2014 (GCU-2014-0121).

References

[1] D. W. Kahn, *Topology: An Introduction to the Point-set and Algebraic Areas*, Dover Publications, New York, NY, USA, 1995.

[2] Y.-H. Kim and B.-R. Moon, "New usage of Sammon's mapping for genetic visualization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1136–1147, 2003.

[3] A. Moraglio and R. Poli, "Topological interpretation of crossover," in *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 1, pp. 1377–1388, 2004.

[4] M. Wineberg and F. Oppacher, "Distance between populations," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1481–1492, 2003.

[5] Y. Yoon and Y.-H. Kim, "Geometricity of genetic operators for real-coded representation," *Applied Mathematics and Computation*, vol. 219, no. 23, pp. 10915–10927, 2013.

[6] S.-S. Choi and B.-R. Moon, "Normalization in genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 862–873, 2003.

[7] Y.-H. Kim, A. Moraglio, A. Kattan, and Y. Yoon, "Geometric generalisation of surrogate model-based optimisation to combinatorial and program spaces," *Mathematical Problems in Engineering*, vol. 2014, Article ID 184540, 10 pages, 2014.

[8] Y.-H. Kim, A. Moraglio, Y. Yoon, and B.-R. Moon, "Geometric crossover for multiway graph partitioning," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1217–1224, July 2006.

[9] A. Moraglio, Y.-H. Kim, Y. Yoon, and B.-R. Moon, "Geometric crossovers for multiway graph partitioning," *Evolutionary Computation*, vol. 15, no. 4, pp. 445–474, 2007.

[10] Y. Yoon and Y.-H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1473–1483, 2013.

[11] Y. Yoon, Y.-H. Kim, A. Moraglio, and B.-R. Moon, "Quotient geometric crossovers and redundant encodings," *Theoretical Computer Science*, vol. 425, pp. 4–16, 2012.

[12] Y. Yoon, Y.-H. Kim, A. Moraglio, and B.-R. Moon, "A theoretical and empirical study on unbiased boundary-extended crossover for real-valued representation," *Information Sciences*, vol. 183, pp. 48–65, 2012.

[13] K. D. Boese, A. B. Kahng, and S. Muddu, "A new adaptive multi-start technique for combinatorial global optimizations," *Operations Research Letters*, vol. 16, no. 2, pp. 101–113, 1994.

[14] T. Jones and S. Forrest, "Fitness distance correlation as a measure of problem difficulty for genetic algorithms," in *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 184–192, Pittsburgh, Pa, USA, July 1995.

[15] P. Merz and B. Freisleben, "Fitness landscape analysis and memetic algorithms for the quadratic assignment problem," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 337–352, 2000.

[16] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[17] J. Bagley, *The behavior of adaptive systems which employ genetic and correlation algorithms [Ph.D. thesis]*, University of Michigan, Ann Arbor, Mich, USA, 1967.

[18] G. R. Harik and D. E. Goldberg, "Learning linkage," in *Foundations of Genetic Algorithms*, vol. 4, pp. 247–262, Morgan Kaufmann, San Francisco, Calif, USA, 1996.

[19] D. E. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: motivation, analysis, and first results," *Complex Systems*, vol. 3, no. 5, pp. 493–530, 1989.

[20] T. N. Bui and B.-R. Moon, "Hyperplane synthesis for genetic algorithms," in *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 102–109, July 1993.

[21] T. N. Bui and B. R. Moon, "Genetic algorithm and graph partitioning," *IEEE Transactions on Computers*, vol. 45, no. 7, pp. 841–855, 1996.

[22] T. N. Bui and B. R. Moon, "New genetic approach for the Traveling salesman problem," in *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, pp. 7–12, June 1994.

[23] Y.-H. Kim, Y.-K. Kwon, and B.-R. Moon, "Problem-independent schema synthesis for genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1112–1122, Chicago, Ill, USA, July 2003.

[24] B.-R. Moon and C. K. Kim, "A two-dimensional embedding of graphs for genetic algorithms," in *Proceedings of the International Conference on Genetic Algorithms*, pp. 204–211, 1997.

[25] Y.-H. Kim and K. Seo, "Two congruence classes for symmetric binary matrices over $\mathbb{F}_2$," *WSEAS Transactions on Mathematics*, vol. 7, no. 6, pp. 339–343, 2008.

[26] Y.-H. Kim and Y. Yoon, "Effect of changing the basis in genetic algorithms using binary encoding," *KSII Transactions on Internet and Information Systems*, vol. 2, no. 4, pp. 184–193, 2008.

[27] Y. Yoon and Y.-H. Kim, "A mathematical design of genetic operators on $GL_n(\mathbb{Z}_2)$," *Mathematical Problems in Engineering*, vol. 2014, Article ID 540936, 8 pages, 2014.

[28] M. Anderson and T. Feil, "Turning lights out with linear algebra," *Mathematics Magazine*, vol. 71, no. 4, pp. 300–303, 1998.

[29] S.-H. Kim, Y.-H. Kim, and B.-R. Moon, "A hybrid genetic algorithm for the max cut problem," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 416–423, 2001.

[30] K. Seo, S. Hyun, and Y.-H. Kim, "A spanning tree-based encoding of the MAX CUT problem for evolutionary search," in *Proceedings of the International Conference on Parallel Problem Solving from Nature*, vol. 7491 of *Lecture Notes in Computer Science*, pp. 510–518, 2012.

[31] K. Seo, S. Hyun, and Y.-H. Kim, "An edge-set representation based on spanning tree for searching cut space," *IEEE Transactions on Evolutionary Computation*, 2014.

[32] Y.-H. Kim and B.-R. Moon, "Investigation of the fitness landscapes and multi-parent crossover for graph bipartitioning," in *Genetic and Evolutionary Computation—GECCO 2003*, vol. 2723 of *Lecture Notes in Computer Science*, pp. 1123–1135, Springer, Berlin, Germany, 2003.

[33] Y.-H. Kim and B.-R. Moon, "Investigation of the fitness landscapes in graph bipartitioning: an empirical study," *Journal of Heuristics*, vol. 10, no. 2, pp. 111–133, 2004.

[34] I. Hwang, Y.-H. Kim, and B.-R. Moon, "Multi-attractor gene reordering for graph bisection," in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference*, pp. 1209–1215, July 2006.

[35] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, "Optimization by simulated annealing: an experimental evaluation, part I. Graph partitioning," *Operations Research*, vol. 37, no. 6, pp. 865–892, 1989.

[36] Y.-H. Kim, "An enzyme-inspired approach to surmount barriers in graph bisection," in *Proceedings of the International Conference on Computational Science and Its Applications*, vol. 5072 of *Lecture Notes in Computer Science*, pp. 841–851, 2008.

[37] Y.-H. Kim and B.-R. Moon, "A hybrid genetic search for graph partitioning based on lock gain," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 167–174, 2000.

[38] Y.-H. Kim and B.-R. Moon, "Lock-gain based graph partitioning," *Journal of Heuristics*, vol. 10, no. 1, pp. 37–57, 2004.

[39] Y. Yoon and Y.-H. Kim, "New bucket managements in iterative improvement partitioning algorithms," *Applied Mathematics and Information Sciences*, vol. 7, no. 2, pp. 529–532, 2013.

[40] Y. Yoon and Y.-H. Kim, "Vertex ordering, clustering, and their application to graph partitioning," *Applied Mathematics and Information Sciences*, vol. 8, no. 1, pp. 135–138, 2014.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 516374, 8 pages
http://dx.doi.org/10.1155/2015/516374

Research Article

Least Squares Based and Two-Stage Least Squares Based Iterative Estimation Algorithms for H-FIR-MA Systems

Zhenwei Shi[1,2] and Zhicheng Ji[1]

[1] *School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China*
[2] *Wuxi Electrical and Higher Vocational School, Wuxi 214028, China*

Correspondence should be addressed to Zhicheng Ji; zcji@jiangnan.edu.cn

Received 25 May 2014; Accepted 17 September 2014

Academic Editor: Shifei Ding

This paper studies the identification of Hammerstein finite impulse response moving average (H-FIR-MA for short) systems. A new two-stage least squares iterative algorithm is developed to identify the parameters of the H-FIR-MA systems. The simulation cases indicate the efficiency of the proposed algorithms.

1. Introduction

System modeling [1–5] and parameter estimation [6–10] are basic for controller design [11, 12]. Nonlinear Hammerstein model identification has received much attention due to its ability to describe a wide class of nonlinear systems and has extensive applications in many engineering problems [13, 14]. The Hammerstein models are special class of nonlinear systems; the nonlinear block is usually static nonlinearity and is followed by a linear system [15]. For example, Wang et al. discussed the identification problem for a Hammerstein nonlinear system with a dynamic subspace state space [16]; Greblicki investigated a class of continuous time Hammerstein system identification [17].

There are a lot of research topics about linear or nonlinear system identification [18, 19] and control [20, 21]. For example, Ding et al. derived the gradient search based and the Newton based identification methods for Hammerstein systems [22]; Wang and Ding proposed a hierarchical least squares identification method for Hammerstein-Wiener systems by using the hierarchical identification principle and the auxiliary model identification idea [23]; Based on the data filtering technique and the key-term separation principle, Wang et al. investigated a filtering based recursive least squares identification algorithm for Hammerstein output error moving average systems [24]. The proposed algorithm can identify not only the system model parameters but also the noise model parameters and the internal variables.

The iterative algorithm is one of the basic methods for system analysis and synthesis, and nonlinear optimization [25–28]. In [29], Wang and Ding presented a gradient based and least squares based iterative identification algorithms for Wiener systems through the use of the hierarchical identification principle. In [30], Ding et al. discussed the Newton iterative identification algorithm of a class of Wiener nonlinear systems with moving average noises from input-output measurement data. Li et al. derived iterative parameter identification methods for nonlinear functions [31]. Pan et al. proposed a digital image correlation using iterative least squares and pointwise least squares for displacement field and strain field measurements [32]. In the field of control, Zhang et al. applied the iterative algorithm to the predictive control field [33, 34].

Recently, the multistage identification strategy is widely applied to the system identification field [35, 36]. For example, Ding and Duan studied a new-type two-stage least squares based iterative algorithm for identifying the system model parameters and the noise model parameters [37].

The main concern of this paper is to investigate the parameter identification problem of Hammerstein finite

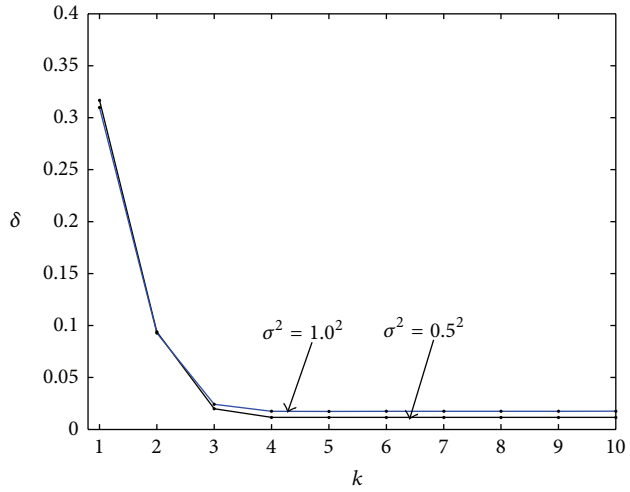


Figure 1: The LSI estimation errors δ versus k ($L = 3000$, $\sigma^2 = 0.5^2$, and $\sigma^2 = 1.0^2$).

impulse response moving average (H-FIR-MA) systems. The memoryless polynomial input nonlinearity is followed by a linear dynamical system, as is explained in Figure 1. Both the least squares iterative and the two-stage least squares iterative algorithms are proposed to estimate the parameters of the H-FIR-MA systems.

The layout of this paper is organized as follows. Section 2 describes the identification model of H-FIR-MA systems. Section 3 provides the least squares iterative algorithm for the H-FIR-MA systems. Section 4 introduces the two-stage least squares iterative algorithm for the H-FIR-MA systems. In Section 5, we apply the proposed algorithms to an example to illustrate their implementation. Finally, concluding remarks are offered in Section 6.

2. System Description and Identification Model

Some notation is given. $\|\mathbf{X}\|^2 = \mathrm{tr}[\mathbf{X}\mathbf{X}^T]$ stands for the norm of a matrix $\mathbf{X}$; "$a := x$" or "$x := a$" expresses that "a is defined as x"; $\mathbf{I}$ represents the identity matrix of appropriate sizes and $\mathbf{1}_n$ is defined as an n-dimensional identity column vector.

Consider an H-FIR-MA system, which is described by

$$\begin{aligned} y(t) &= x(t) + w(t) \\ &= \sum_{i=1}^{n_b} b_i \overline{u}(t-i) + b_0 \overline{u}(t) + \sum_{i=1}^{n_d} d_i v(t-i) + v(t), \end{aligned} \tag{1}$$

$$\begin{aligned} x(t) &= \sum_{i=1}^{n_b} b_i \overline{u}(t-i) + b_0 \overline{u}(t), \\ w(t) &= \sum_{i=1}^{n_d} d_i v(t-i) + v(t), \end{aligned} \tag{2}$$

$$\overline{u}(t) = \mathbf{f}(u(t))\,\boldsymbol{\gamma}, \tag{3}$$

where $\{u(t)\}$ and $\{y(t)\}$ are the input and output sequences of the systems, $\{v(t)\}$ is an uncorrelated stochastic noise sequence with zero mean and variance σ^2, and $x(t)$, $w(t)$, and $\overline{u}(t)$ are unmeasurable. The output $\overline{u}(t)$ of the nonlinear block is a linear combination, with unknown coefficients $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_m]^T \in R^{m\times 1}$, of a known basis $\mathbf{f}(u(t)) := [f_1(u(t)), f_2(u(t)), \ldots, f_m(u(t))] \in R^{1\times m}$ in the system input $u(t)$, and can be expressed as

$$\begin{aligned} \overline{u}(t) &= \mathbf{f}(u(t))\,\boldsymbol{\gamma} = \gamma_1 f_1(u(t)) + \gamma_2 f_2(u(t)) \\ &\quad + \cdots + \gamma_m f_m(u(t)) \\ &= \sum_{j=1}^{m} \gamma_j f_j(u(t)). \end{aligned} \tag{4}$$

Assume that the orders n_b, n_d, and m are known in (1) and (4) and $u(t) = 0$, $y(t) = 0$, $\overline{u}(t) = 0$, and $v(t) = 0$ for $t \le 0$. In order to get unique parameter estimates, here we let $b_0 = 1$ [24]. The item $\overline{u}(t)$ in (1) is chosen as the key term; substituting (4) into (1) gets

$$y(t) = \sum_{i=1}^{n_b} b_i \overline{u}(t-i) + \mathbf{f}(u(t))\,\boldsymbol{\gamma} + \sum_{i=1}^{n_d} d_i v(t-i) + v(t). \tag{5}$$

Define the parameter vector $\boldsymbol{\Theta}$ and the information vector $\boldsymbol{\varphi}(t)$ as follows:

$$\begin{aligned} \boldsymbol{\Theta} &:= \left[\boldsymbol{\theta}^T,\ \boldsymbol{\gamma}^T\right]^T \in R^{n_b+m+n_d}, \\ \boldsymbol{\theta} &:= \left[b_1, b_2, \ldots, b_{n_b}, d_1, d_2, \ldots, d_{n_d}\right]^T \in R^{n_b+n_d}, \\ \boldsymbol{\varphi}(t) &:= \left[\overline{u}(t-1), u(t-2), \ldots, u(t-n_b),\right. \\ &\quad v(t-1), v(t-2), \ldots, v(t-n_d), \\ &\quad \left. f_1(u(t)), f_2(u(t)), \ldots, f_m(u(t))\right]^T \in R^{n_b+n_d+m}. \end{aligned} \tag{6}$$

From (5), we obtain the following identification model:

$$\begin{aligned} y(t) &= \boldsymbol{\varphi}^T(t)\left[\boldsymbol{\theta}^T,\ \boldsymbol{\gamma}^T\right]^T + v(t) \\ &= \boldsymbol{\varphi}^T(t)\,\boldsymbol{\Theta} + v(t). \end{aligned} \tag{7}$$

Define the cost function:

$$J(\boldsymbol{\Theta}) := \sum_{t=1}^{L} v^2(t) = \sum_{t=1}^{L}\left[y(t) - \boldsymbol{\varphi}^T(t)\,\boldsymbol{\Theta}\right]^2. \tag{8}$$

In what follows, we derive the algorithms for identifying the H-FIR-MA system using the least squares and two-stage least squares iterative estimation algorithms.

3. The Least Squares Iterative Estimation Algorithm

In this section, referring to the method in [27], we give simply the least squares iterative (LSI) estimation algorithm for the H-FIR-MA system for comparison.

Consider the data from $t = 1$ to $t = L$ ($L \geq n$) and define the stacked information matrices $\boldsymbol{\Phi}(L)$, the stacked output vector $\mathbf{Y}(L)$, and the stacked white noise vector $\mathbf{V}(L)$ as

$$\begin{aligned}\boldsymbol{\Phi}(L) &:= \left[\boldsymbol{\varphi}(1), \boldsymbol{\varphi}(2), \ldots, \boldsymbol{\varphi}(L)\right]^T \in R^{L\times n},\\ n &= n_b + m + n_d,\end{aligned} \tag{9}$$

$$\mathbf{Y}(L) := \left[y(1), y(2), \ldots, y(L)\right]^T \in R^L, \tag{10}$$

$$\mathbf{V}(L) := \left[v(1), v(2), \ldots, v(L)\right]^T \in R^L. \tag{11}$$

Hence, (7) can be rewritten as

$$\mathbf{Y}(L) = \boldsymbol{\Phi}(L)\,\boldsymbol{\Theta} + \mathbf{V}(L). \tag{12}$$

According to the estimation model in (12), the cost function in (8) can be written as

$$J_1(\boldsymbol{\Theta}) = \|\mathbf{V}(L)\|^2 = \|\mathbf{Y}(L) - \boldsymbol{\Phi}(L)\,\boldsymbol{\Theta}\|^2. \tag{13}$$

To minimize $J_1(\boldsymbol{\Theta})$, letting its partial derivative of $J_1(\boldsymbol{\Theta})$ with respect to $\boldsymbol{\Theta}$ be zero, we have

$$\widehat{\boldsymbol{\Theta}} = \left[\boldsymbol{\Phi}^T(L)\,\boldsymbol{\Phi}(L)\right]^{-1}\boldsymbol{\Phi}^T(L)\,\mathbf{Y}(L). \tag{14}$$

It is impossible to obtain the estimate $\widehat{\boldsymbol{\Theta}}$, because the information matrix $\boldsymbol{\Phi}(L)$ (i.e., $\boldsymbol{\varphi}(t)$ in (6)) contains the unmeasurable inner variables $\overline{u}(t-i)$ and the noise terms $v(t-i)$. Here we adopt the auxiliary model idea and the hierarchical identification principle: let $k = 1, 2, 3\ldots$ be iteration variable, let $\widehat{\boldsymbol{\Theta}}_k$ be the iterative estimate of $\boldsymbol{\Theta}$ at iterative k and $\widehat{\overline{u}}_k(t-i)$, and let $\widehat{v}_k(t-i)$ be the k iterative estimates of $\overline{u}(t-i)$ and $v(t-i)$. We replace $\overline{u}(t-i)$ and $v(t-i)$ in (6) with their estimates and obtain the estimates $\widehat{\boldsymbol{\varphi}}_k(t)$ and $\widehat{\boldsymbol{\Phi}}_k(L)$ as follows:

$$\begin{aligned}&\widehat{\boldsymbol{\varphi}}_k(t)\\ &:= \left[\widehat{\overline{u}}_{k-1}(t-1), \widehat{\overline{u}}_{k-1}(t-2), \ldots, \widehat{\overline{u}}_{k-1}(t-n_b),\right.\\ &\quad f_1(u(t)), f_2(u(t)), \ldots, f_m(u(t)),\\ &\quad \left.\widehat{v}_{k-1}(t-1), \widehat{v}_{k-1}(t-2), \ldots, \widehat{v}_{k-1}(t-n_d)\right]^T \in R^n,\\ &\widehat{\boldsymbol{\Phi}}_k(L) := \left[\widehat{\boldsymbol{\varphi}}_k(1), \widehat{\boldsymbol{\varphi}}_k(2), \ldots, \widehat{\boldsymbol{\varphi}}_k(L)\right]^T \in R^{L\times n},\\ &\qquad n = n_b + n_d + m,\\ &\widehat{v}_k(t-i) = y(t-i) - \widehat{\boldsymbol{\varphi}}_k^T(t-i)\,\widehat{\boldsymbol{\Theta}}_k(t), \quad i = 1, 2, 3, \ldots n_b,\\ &\widehat{\overline{u}}_k(t-i) = \widehat{\gamma}_{1(k)} f_1(u(t-i)) + \widehat{\gamma}_{2(k)} f_2(u(t-i))\\ &\qquad + \cdots + \widehat{\gamma}_{m(k)} f_m(u(t-i))\\ &\qquad = \mathbf{f}(u(t-i))\,\widehat{\boldsymbol{\gamma}}_k, \quad i = 1, 2, 3, \ldots m.\end{aligned} \tag{15}$$

Replacing $\boldsymbol{\Phi}(L)$ in (14) with $\widehat{\boldsymbol{\Phi}}_k(L)$ and combining (10) and (15), we can obtain the LSI estimation algorithm of identifying $\boldsymbol{\Theta}$ for the H-FIR-MA system as follows [27]:

$$\widehat{\boldsymbol{\Theta}}_k = \left[\widehat{\boldsymbol{\Phi}}_k^T(L)\,\widehat{\boldsymbol{\Phi}}_k(L)\right]^{-1}\widehat{\boldsymbol{\Phi}}_k^T(L)\,\mathbf{Y}(L), \tag{16}$$

$$\mathbf{Y}(L) = \left[y(1), y(2), \ldots, y(L)\right]^T, \tag{17}$$

$$\widehat{\boldsymbol{\Phi}}_k(L) = \left[\widehat{\boldsymbol{\varphi}}_k(1), \widehat{\boldsymbol{\varphi}}_k(2), \ldots, \widehat{\boldsymbol{\varphi}}_k(L)\right]^T, \tag{18}$$

$$\begin{aligned}\widehat{\boldsymbol{\varphi}}_k(t) &= \left[\widehat{\overline{u}}_{k-1}(t-1), \widehat{\overline{u}}_{k-1}(t-2), \ldots,\right.\\ &\quad \left.\widehat{\overline{u}}_{k-1}(t-n_b), \widehat{v}_{k-1}(t-1),\right.\\ &\quad \left.\widehat{v}_{k-1}(t-2), \ldots, \widehat{v}_{k-1}(t-n_d), \mathbf{f}_k(u(t))\right]^T\end{aligned} \tag{19}$$

$$\mathbf{f}_k(u(t)) = \left[f_1(u(t)), f_2(u(t)), \ldots, f_m(u(t))\right], \tag{20}$$

$$\widehat{v}_k(t-i) = y(t-i) - \widehat{\boldsymbol{\varphi}}_k^T(t-i)\,\widehat{\boldsymbol{\Theta}}_k(t), \tag{21}$$

$$\begin{aligned}\widehat{\overline{u}}_k(t-i) &= \widehat{\gamma}_{1(k)} f_1(u(t-i)) + \widehat{\gamma}_{2(k)} f_2(u(t-i))\\ &\quad + \cdots + \widehat{\gamma}_{m(k)} f_m(u(t-i))\\ &= \mathbf{f}(u(t-i))\,\widehat{\boldsymbol{\gamma}}_k.\end{aligned} \tag{22}$$

The computation procedures of the LSI algorithm in (16)–(22) are summarized as follows.

Step 1. Let $k = 1$ and set the initial values $\widehat{\boldsymbol{\Theta}}_0 = \mathbf{1}_n/p_0$, $\widehat{v}_0(t) = 1/p_0$, $\widehat{\overline{u}}_0(t) = 1/p_0$; p_0 is a large number (i.e., $p_0 = 10^6$).

Step 2. Collect the input and output data $u(t)$ and $y(t)$ ($t = 1, 2, \ldots, L$), compute $\mathbf{f}_k(u(t))$ by (20), structure $\mathbf{Y}(L)$ and $\widehat{\boldsymbol{\varphi}}_k(t)$ by (17) and (19), respectively, and form $\widehat{\boldsymbol{\Phi}}_k(L)$ by (18).

Step 3. Update the parameter identification $\widehat{\boldsymbol{\Theta}}_k$ by (16).

Step 4. Compute $\widehat{v}_k(t)$ and $\widehat{\overline{u}}_k(t)$ by (21) and (22), respectively.

Step 5. Increase k by 1 and jump to Step 2.

4. The Two-Stage Least Squares Iterative Estimation Algorithm

Here, we derive a two-stage least squares iterative (TS-LSI) estimation algorithm for the H-FIR-MA system. From (5) and (6), we can obtain the following identification model:

$$\begin{aligned}y(t) &= \sum_{i=1}^{n_b} b_i \overline{u}(t-i) + \mathbf{f}(u(t))\,\boldsymbol{\gamma} + \sum_{i=1}^{n_d} d_i v(t-i) + v(t),\\ &= \Psi^T(t)\,\boldsymbol{\theta} + \mathbf{f}(u(t))\,\boldsymbol{\gamma} + v(t).\end{aligned} \tag{23}$$

Define the information vector $\Psi(t)$ and $\mathbf{f}(u(t))$ as

$$\begin{aligned}\Psi(t) &= \left[\overline{u}(t-1), \overline{u}(t-1), \ldots, \overline{u}(t-n_b),\right.\\ &\qquad \left. v(t-1),\ v(t-2), \ldots, v(t-n_d)\right]^T \in R^{n_b+n_d},\\ \mathbf{f}(u(t)) &= \left[f_1(u(t)), f_2(u(t)), \ldots, f_m(u(t))\right] \in R^{1\times m}.\end{aligned} \tag{24}$$

Define two intermediate variables $y_1(t) := y(t) - \mathbf{f}(u(t))\boldsymbol{\gamma}$ and $y_2(t) := y(t) - \Psi^T(t)\boldsymbol{\theta}$; then the system in (23) can be decomposed into two "suppositional" subsystems:

$$\begin{aligned}y_1(t) &= \Psi^T(t)\boldsymbol{\theta} + v(t),\\ y_2(t) &= \mathbf{f}(u(t))\boldsymbol{\gamma} + v(t).\end{aligned} \tag{25}$$

The estimates of two "suppositional" subsystems in (25) can be obtained by minimizing the cost function:

$$J_2(\boldsymbol{\theta}) = \sum_{t=1}^{L} v^2(t) = \sum_{t=1}^{L}\left[y_1(t) - \Psi^T(t)\boldsymbol{\theta}\right]^2, \tag{26}$$

$$J_3(\boldsymbol{\gamma}) = \sum_{t=1}^{L} v^2(t) = \sum_{t=1}^{L}\left[y_2(t) - \mathbf{f}(u(t))\boldsymbol{\gamma}\right]^2. \tag{27}$$

Consider the data from $t = 1$ to $t = L$ $(L \geq n)$ and in (25) define the stacked information matrices $\boldsymbol{\Psi}(L)$ and $\mathbf{F}(u(L))$ and the stacked vector $\mathbf{Y}_1(L)$ and $\mathbf{Y}_2(L)$ as

$$\begin{aligned}\boldsymbol{\Psi}(L) &:= \left[\Psi(1), \Psi(2), \ldots, \Psi(L)\right]^T \in R^{L\times(n_b+n_d)},\\ \mathbf{Y}_1(L) &:= \left[y_1(1), y_1(2), \ldots, y_1(L)\right]^T \in R^L,\\ \mathbf{F}(u(L)) &:= \left[\mathbf{f}^T(u(1)), \mathbf{f}^T(u(2)), \ldots, \mathbf{f}^T(u(L))\right]^T \in R^{L\times m},\\ \mathbf{Y}_2(L) &:= \left[y_2(1), y_2(2), \ldots, y_2(L)\right]^T \in R^L.\end{aligned} \tag{28}$$

Two intermediate variables can be rewritten as

$$\mathbf{Y}_1(L) := \mathbf{Y}(L) - \mathbf{F}(u(L))\boldsymbol{\gamma}, \tag{29}$$

$$\mathbf{Y}_2(L) := \mathbf{Y}(L) - \boldsymbol{\Psi}(L)\boldsymbol{\theta}. \tag{30}$$

From (25), we have

$$\mathbf{Y}_1(L) = \boldsymbol{\Psi}(L)\boldsymbol{\theta} + \mathbf{V}(L), \tag{31}$$

$$\mathbf{Y}_2(L) = \mathbf{F}(u(L))\boldsymbol{\gamma} + \mathbf{V}(L). \tag{32}$$

According to the estimation model in (31), the cost function in (26) can be written as

$$J_2(\boldsymbol{\theta}) = \|\mathbf{V}(L)\|^2 = \left\|\mathbf{Y}_1(L) - \boldsymbol{\Psi}(L)\boldsymbol{\theta}\right\|^2. \tag{33}$$

To minimize $J_2(\boldsymbol{\theta})$, let its partial derivative of $J_2(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ be zero:

$$\frac{\partial J_2(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\boldsymbol{\Psi}^T(L)\left[\mathbf{Y}_1(L) - \boldsymbol{\Psi}(L)\boldsymbol{\theta}\right] = 0. \tag{34}$$

From (34), the least squares estimate of the parameter vector $\boldsymbol{\theta}$ can be expressed as

$$\widehat{\boldsymbol{\theta}} = \left[\boldsymbol{\Psi}^T(L)\boldsymbol{\Psi}(L)\right]^{-1}\boldsymbol{\Psi}^T(L)\mathbf{Y}_1(L). \tag{35}$$

Here, put (29) into (35) and (35) gives

$$\widehat{\boldsymbol{\theta}} = \left[\boldsymbol{\Psi}^T(L)\boldsymbol{\Psi}(L)\right]^{-1}\boldsymbol{\Psi}^T(L)\left[\mathbf{Y}(L) - \mathbf{F}(u(L))\boldsymbol{\gamma}\right]. \tag{36}$$

In accordance with the same derivation process of $\widehat{\boldsymbol{\theta}}$, we can easily get the estimation formula of

$$\widehat{\boldsymbol{\gamma}} = \left[\mathbf{F}^T(u(L))\mathbf{F}(u(L))\right]^{-1}\mathbf{F}^T(u(L))\left[\mathbf{Y}(L) - \boldsymbol{\Psi}(L)\boldsymbol{\theta}\right]. \tag{37}$$

However, (36) and (37) contain the unknown parameter $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$, respectively, it is impossible to $\widehat{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{\gamma}}$. According to the method in Section 3, we can summarize the two-stage least squares iterative estimation algorithm for estimating $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ of the H-FIR-MA systems as follows:

$$\widehat{\boldsymbol{\theta}}_k = \left[\widehat{\boldsymbol{\Psi}}_k^T(L)\widehat{\boldsymbol{\Psi}}_k(L)\right]^{-1}\widehat{\boldsymbol{\Psi}}_k^T(L)\left[\mathbf{Y}(L) - \mathbf{F}_k(u(L))\widehat{\boldsymbol{\gamma}}_{k-1}\right], \tag{38}$$

$$\widehat{\boldsymbol{\gamma}}_k = \left[\mathbf{F}_k^T(u(L))\mathbf{F}_k(u(L))\right]^{-1}\mathbf{F}_k^T(u(L))\left[\mathbf{Y}(L) - \widehat{\boldsymbol{\Psi}}_k(L)\widehat{\boldsymbol{\theta}}_k\right], \tag{39}$$

$$\mathbf{Y}(L) = \left[y(1), y(2), \ldots, y(L)\right]^T, \tag{40}$$

$$\widehat{\boldsymbol{\Psi}}_k(L) = \left[\widehat{\Psi}_k(1), \widehat{\Psi}_k(2), \ldots, \widehat{\Psi}_k(L)\right]^T, \tag{41}$$

$$\begin{aligned}\widehat{\Psi}_k(t) = \left[\right.&\widehat{\overline{u}}_{k-1}(t-1), \widehat{\overline{u}}_{k-1}(t-2), \ldots, \widehat{\overline{u}}_{k-1}(t-n_b),\\ &\left.\widehat{v}_{k-1}(t-1), \widehat{v}_{k-1}(t-2), \ldots, \widehat{v}_{k-1}(t-n_d)\right]^T,\end{aligned} \tag{42}$$

$$\mathbf{F}_k(u(L)) = \left[\mathbf{f}_k^T(u(1)), \mathbf{f}_k^T(u(2)), \ldots, \mathbf{f}_k^T(u(L))\right]^T, \tag{43}$$

$$\mathbf{f}_k(u(t)) = \left[f_1(u(t)), f_2(u(t)), \ldots, f_m(u(t))\right], \tag{44}$$

$$\widehat{v}_k(t-i) = y(t-i) - \widehat{\Psi}_k^T(t-i)\widehat{\boldsymbol{\theta}}_k - \mathbf{f}(u(t-i))\widehat{\boldsymbol{\gamma}}_k, \tag{45}$$

$$\begin{aligned}\widehat{\overline{u}}_k(t-i) &= \widehat{\gamma}_{1(k)} f_1(u(t-i)) + \widehat{\gamma}_{2(k)} f_2(u(t-i))\\ &\quad + \cdots + \widehat{\gamma}_{m(k)} f_m(u(t-i))\\ &= \mathbf{f}(u(t-i))\widehat{\boldsymbol{\gamma}}_k.\end{aligned} \tag{46}$$

The computation procedures of the TS-LSI algorithm in (38)–(46) are summarized as follows.

Step 1. Let $k = 1$, and set the initial values $\widehat{v}_0(t) = 1/p_0$, $\widehat{\boldsymbol{\theta}}_0 = \mathbf{1}_{n_b+n_d}/p_0$, $\widehat{\boldsymbol{\gamma}}_0 = \mathbf{1}_m/p_0$, $\widehat{\overline{u}}_0(t) = 1/p_0$, $p_0 = 10^6$ (i.e., $p_0 = 10^6$).

Step 2. Collect the input and output data $u(t)$ and $y(t)$ $(t = 1, 2, \ldots, L)$, compute $\mathbf{f}_k(u(t))$ by (44), and form $\mathbf{Y}(L)$ by (40) and $\widehat{\Psi}_k(t)$ by (42).

Table 1: The LSI estimation and errors with $\sigma^2 = 0.5^2$ and $\sigma^2 = 1.0^2$ ($L = 3000$).

σ^2	k	b_1	b_2	d_1	γ_1	γ_2	γ_3	$\delta \times 100\%$
0.5	1	−1.4978	0.5020	−0.0174	0.8217	0.5027	0.3252	31.5090
	5	−1.4950	0.5016	−0.6198	0.8202	0.4979	0.3228	1.5288
	10	−1.4953	0.5014	−0.6199	0.8202	0.4972	0.3228	1.5276
1.0	1	−1.4956	0.5040	−0.0348	0.8435	0.5055	0.3195	30.694
	5	−1.4890	0.5027	−0.6198	0.8405	0.4954	0.3147	2.5092
	10	−1.4896	0.5022	−0.6199	0.8406	0.4935	0.3146	2.5156
True values		−1.5000	0.5000	−0.6400	0.8000	0.5000	0.3310	

Table 2: The LSI estimation and errors with $\sigma^2 = 0.5^2$ and $\sigma^2 = 1.0^2$ ($L = 7000$).

σ^2	k	b_1	b_2	d_1	γ_1	γ_2	γ_3	$\delta \times 100\%$
0.5	1	−1.4976	0.4977	−0.0028	0.8038	0.5034	0.3325	32.2250
	5	−1.4951	0.4973	−0.6296	0.7956	0.5050	0.3345	0.7077
	10	−1.4946	0.4969	−0.6296	0.7957	0.5052	0.3345	0.7215
1.0	1	−1.4953	0.4955	−0.0058	0.8077	0.5069	0.3340	32.084
	5	−1.4891	0.4943	−0.6296	0.7914	0.5094	0.3380	1.1290
	10	−1.4883	0.4935	−0.6291	0.7912	0.5091	0.3380	1.0966
True values		−1.5000	0.5000	−0.6400	0.8000	0.5000	0.3310	

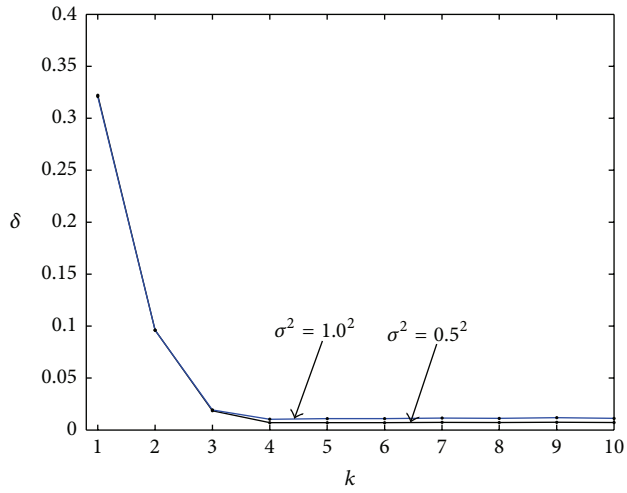


Figure 2: The LSI estimation errors δ versus k ($L = 7000$, $\sigma^2 = 0.5^2$, and $\sigma^2 = 1.0^2$).

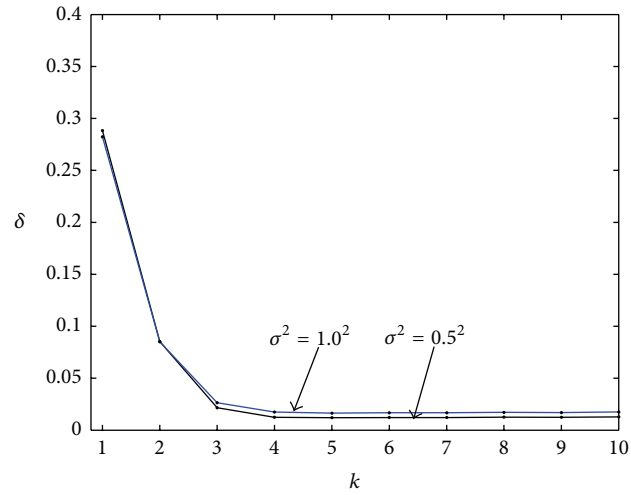


Figure 3: The TS-LSI estimation errors δ versus k ($L = 3000$, $\sigma^2 = 0.5^2$, and $\sigma^2 = 1.0^2$).

Step 3. Structure $\widehat{\mathbf{\Psi}}_k(L)$ and $\mathbf{F}_k(u(L))$, respectively, by (41) and (43).

Step 4. Update the parameter identification $\widehat{\boldsymbol{\theta}}_k$, $\widehat{\boldsymbol{\gamma}}_k$ by (38) and (39), respectively.

Step 5. Compute $\widehat{v}_k(t)$ and $\widehat{\overline{u}}_k(t)$ by (45) and (46), respectively.

Step 6. Increase k by 1 and jump to Step 2.

5. Simulations and Case Study

In this section, we consider an H-FIR-MA system, where the static nonlinearity $\mathbf{f}(u(t)) := [f_1(u(t)), f_2(u(t)), \ldots, f_m(u(t))] \in R^{1\times m}$ is chosen as polynomials. More precisely

$$\begin{aligned} y(t) &= \sum_{i=1}^{2} b_i \overline{u}(t-i) + \overline{u}(t) + \sum_{i=1}^{1} d_i v(t-i) + v(t) \\ &= b_1 \overline{u}(t-1) + b_2 \overline{u}(t-2) + \overline{u}(t) + d_1 v(t-1) + v(t) \\ &= -1.5\overline{u}(t-1) + 0.5\overline{u}(t-2) + \overline{u}(t) \\ &\quad - 0.64v(t-1) + v(t), \end{aligned}$$

Table 3: The TS-LSI estimation and errors with $\sigma^2 = 0.5^2$ and $\sigma^2 = 1.0^2$ ($L = 3000$).

σ^2	k	b_1	b_2	d_1	γ_1	γ_2	γ_3	$\delta \times 100\%$
0.5	1	−1.5067	0.5064	−0.0874	0.8803	0.4947	0.2939	28.307
	5	−1.4961	0.5008	−0.6194	0.8185	0.4955	0.3235	1.4828
	10	−1.4955	0.5033	−0.6192	0.8182	0.5015	0.3237	1.4736
1.0	1	−1.5048	0.5083	−0.1033	0.9010	0.4971	0.2882	27.708
	5	−1.4904	0.5016	−0.6196	0.8370	0.4927	0.3161	2.3477
	10	−1.4901	0.5048	−0.6197	0.8366	0.5000	0.3163	2.3106
True values		−1.5000	0.5000	−0.6400	0.8000	0.5000	0.3310	

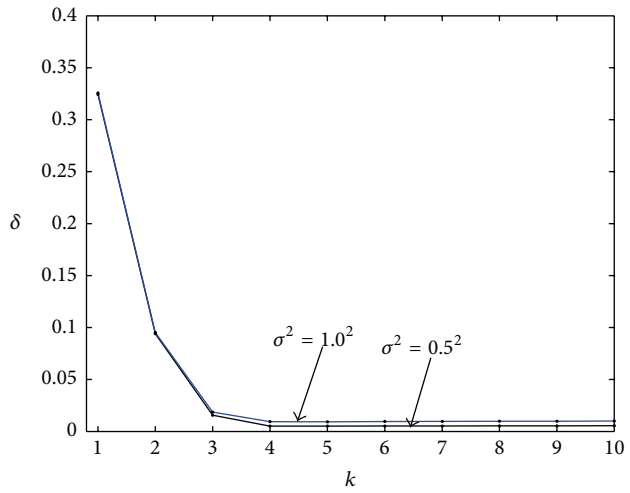


Figure 4: The TS-LSI estimation errors δ versus k ($L = 7000$, $\sigma^2 = 0.5^2$, and $\sigma^2 = 1.0^2$).

$$\begin{aligned}
\overline{u}(t) &= \mathbf{f}(u(t))\,\boldsymbol{\gamma} = \left[u(t), u(t)^2, u(t)^3\right]\boldsymbol{\gamma} \\
&= \gamma_1 u(t) + \gamma_2 u^2(t) + \gamma_3 u^3(t) \\
&= 0.8u(t) + 0.5u^2(t) + 0.331u^3(t), \\
\boldsymbol{\Theta} &= \left[\boldsymbol{\theta}^T,\ \boldsymbol{\gamma}^T\right]^T \\
&= \left[b_1, b_2, d_1, \gamma_1, \gamma_2, \gamma_3\right]^T \\
&= [-1.5, 0.5, -0.64, 0.8, 0.5, 0.331]^T.
\end{aligned} \tag{47}$$

In simulation, the input $\{u(t)\}$ is taken as an uncorrelated measured stochastic signal sequence with zero mean, the noise $\{v(t)\}$ is a white noise sequence with zero mean and variances $\sigma^2 = 0.5^2$ and $\sigma^2 = 1.0^2$, respectively, and the corresponding noise-to-signal ratios are $\delta_{\text{ns}} = 31.73\%$ and $\delta_{\text{ns}} = 63.46\%$. The noise-to-signal ratios can be calculated by the following formula:

$$\delta_{\text{ns}} := \sqrt{\frac{D[w(t)]}{D[x(t)]}} \times 100\% = \frac{\sigma_w}{\sigma_x} \times 100\%, \tag{48}$$

where $D[w(t)] = \sigma_w^2$ and $D[x(t)] = \sigma_x^2$ are, respectively, expressed as the variances of $w(t)$ and $x(t)$ in (1).

Take two different data lengths $L = 3000$ and $L = 7000$. The parameter values $\boldsymbol{\Theta} = \left[\boldsymbol{\theta}^T,\ \boldsymbol{\gamma}^T\right]^T = [b_1, b_2, d_1, \gamma_1, \gamma_2, \gamma_3]^T$ are estimated using the two different methods described in the paper, namely, the LSI and the TS-LSI methods in Sections 3 and 4. We apply the LSI method to estimate the parameters of this case; the parameter estimation with different data length and noise variances σ^2 are shown in Tables 1 and 2, and the estimation errors δ versus iteration k are shown in Figures 1 and 2, where $\delta := \|\widehat{\boldsymbol{\Theta}}_k - \boldsymbol{\Theta}\|/\|\boldsymbol{\Theta}\|$.

Similarly, the parameter estimation and estimation errors δ of the TS-LSI method with different data length and noise variances σ^2 are shown in Tables 3 and 4 and Figures 3 and 4.

From the simulation results in Tables 1–4 and Figures 1–4, we can draw the following conclusions.

(1) The parameter estimation errors given by the LSI and TS-LSI algorithms become small as iterations increase.

(2) The parameter estimation errors given by the LSI and TS-LSI algorithms become closer to their true values with the data length L increasing.

(3) It is easy to see that a high noise level results in a low consistence rate of the parameter estimates to the true parameters for both of the proposed algorithms.

(4) When the data length goes to infinity, the estimation errors converge to zero. The simulations of results in Tables 1–4 and Figures 1–4 indicate that the proposed algorithms based iterative algorithm should stop for about $k = 3 \sim 4$. The fluctuation of the estimation errors is caused for large k due to the stationary of noise.

All in all, this shows that the proposed algorithms are effective.

6. Conclusions

The LSI and the TS-LSI identification algorithms are developed for H-FIR-MA systems. The simulation results indicate that the proposed algorithms can obtain highly accurate parameter estimates and fast convergence rate and illustrate the proposed algorithms' performance. Compared with other methods, the LSI and TS-LSI methods must compute the

Table 4: The TS-LSI estimation and errors with $\sigma^2 = 0.5^2$ and $\sigma^2 = 1.0^2$ ($L = 7000$).

σ^2	k	b_1	b_2	d_1	γ_1	γ_2	γ_3	$\delta \times 100\%$
0.5	1	−1.4949	0.4998	0.0034	0.7899	0.4984	0.3378	32.552
	5	−1.4940	0.4977	−0.6350	0.7964	0.5023	0.3344	0.5547
	10	−1.4934	0.4977	−0.6339	0.7962	0.5015	0.3341	0.4968
1.0	1	−1.4913	0.4982	0.0077	0.7919	0.5004	0.3408	32.772
	5	−1.4867	0.4951	−0.6350	0.7929	0.5045	0.3379	1.0305
	10	−1.4858	0.4950	−0.6349	0.7927	0.5038	0.3360	0.9382
True values		−1.5000	0.5000	−0.6400	0.8000	0.5000	0.3310	

matrix inversion. The proposed methods are simple in principle and the basic idea can be applied to other fields [38–41].

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 National High Technology Research and Development Program of China (China 863 Program) (no. 2013AA040405), the Guangdong-Hong Kong Breakthrough Bidding Project in Key Areas (no. 2012A080107015), the fundamental research funds for the central universities (JUSRP51310A), and the Chinese State Grain Administration commonweal research project (no. 201313012).

References

[1] F. Ding, *System Identification—New Theory and Methods*, Science Press, Beijing, China, 2013.

[2] F. Ding, *System Identification: Performances Analysis for Identification Methods*, Science Press, Beijing, China, 2014.

[3] Y. B. Hu, "Iterative and recursive least squares estimation algorithms for moving average systems," *Simulation Modelling Practice and Theory*, vol. 34, pp. 12–19, 2013.

[4] Y. Hu, B. Liu, Q. Zhou, and C. Yang, "Recursive extended least squares parameter estimation for Wiener nonlinear systems with moving average noises," *Circuits, Systems, and Signal Processing*, vol. 33, no. 2, pp. 655–664, 2014.

[5] F. Ding, "Hierarchical estimation algorithms for multivariable systems using measurement information," *Information Sciences*, vol. 277, pp. 396–405, 2014.

[6] F. Ding, "State filtering and parameter estimation for state space systems with scarce measurements," *Signal Processing*, vol. 104, pp. 369–380, 2014.

[7] C. Wang and T. Tang, "Recursive least squares estimation algorithm applied to a class of linear-in-parameters output error moving average systems," *Applied Mathematics Letters*, vol. 29, pp. 36–41, 2014.

[8] C. Wang and T. Tang, "Several gradient-based iterative estimation algorithms for a class of nonlinear systems using the filtering technique," *Nonlinear Dynamics*, vol. 77, no. 3, pp. 769–780, 2014.

[9] Y. Shi and H. Fang, "Kalman filter-based identification for systems with randomly missing measurements in a network environment," *International Journal of Control*, vol. 83, no. 3, pp. 538–551, 2010.

[10] Y. Liu, F. Ding, and Y. Shi, "An efficient hierarchical identification method for general dual-rate sampled-data systems," *Automatica*, vol. 50, no. 3, pp. 962–970, 2014.

[11] Y. Shi and B. Yu, "Output feedback stabilization of networked control systems with random delays modeled by Markov chains," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1668–1674, 2009.

[12] Y. Shi and B. Yu, "Robust mixed H_2/H_∞ control of networked control systems with random time delays in both forward and backward communication links," *Automatica*, vol. 47, no. 4, pp. 754–760, 2011.

[13] J. Vörös, "Recursive identification of systems with noninvertible output nonlinearities," *Informatica*, vol. 21, no. 1, pp. 139–148, 2010.

[14] R. Zhang, A. Xue, and S. Wang, "Dynamic modeling and nonlinear predictive control based on partitioned model and nonlinear optimization," *Industrial and Engineering Chemistry Research*, vol. 50, no. 13, pp. 8110–8121, 2011.

[15] B. Yu, H. Fang, Y. Lin, and Y. Shi, "Identification of Hammerstein output-error systems with two-segment nonlinearities: algorithm and applications," *Control and Intelligent Systems*, vol. 38, no. 4, pp. 194–201, 2010.

[16] D. Wang, F. Ding, and L. Ximei, "Least squares algorithm for an input nonlinear system with a dynamic subspace state space model," *Nonlinear Dynamics*, vol. 75, no. 1-2, pp. 49–61, 2014.

[17] W. Greblicki, "Continuous-time Hammerstein system identification," *IEEE Transactions on Automatic Control*, vol. 45, no. 6, pp. 1232–1236, 2000.

[18] F. Ding, X. Liu, and J. Chu, "Gradient-based and least-squares-based iterative algorithms for Hammerstein systems using the hierarchical identification principle," *IET Control Theory & Applications*, vol. 7, no. 2, pp. 176–184, 2013.

[19] F. Ding, "Hierarchical multi-innovation stochastic gradient algorithm for Hammerstein nonlinear system modeling," *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 1694–1704, 2013.

[20] X. Luan, P. Shi, and F. Liu, "Stabilization of networked control systems with random delays," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 4323–4330, 2011.

[21] X. Luan, S. Zhao, and F. Liu, "H_∞-infinity control for discrete-time markov jump systems with uncertain transition probabilities," *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1566–1572, 2013.

[22] F. Ding, X. P. Liu, and G. Liu, "Identification methods for Hammerstein nonlinear systems," *Digital Signal Processing*, vol. 21, no. 2, pp. 215–238, 2011.

[23] D. Q. Wang and F. Ding, "Hierarchical least squares estimation algorithm for hammerstein-wiener systems," *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 825–828, 2012.

[24] D. Wang, F. Ding, and Y. Chu, "Data filtering based recursive least squares algorithm for Hammerstein systems using the key-term separation principle," *Information Sciences*, vol. 222, pp. 203–212, 2013.

[25] J. Ding, C. Fan, and J. Lin, "Auxiliary model based parameter estimation for dual-rate output error systems with colored noise," *Applied Mathematical Modelling*, vol. 37, no. 6, pp. 4051–4058, 2013.

[26] J. Ding and J. Lin, "Modified subspace identification for periodically non-uniformly sampled systems by using the lifting technique," *Circuits, Systems, and Signal Processing*, vol. 33, no. 5, pp. 1439–1449, 2014.

[27] K. P. Deng and F. Ding, "Newton iterative identification method for an input nonlinear finite impulse response system with moving average noise using the key variables separation technique," *Nonlinear Dynamics*, vol. 76, no. 2, pp. 1195–1202, 2014.

[28] L. Xie and H. Yang, "Gradient-based iterative identification for nonuniform sampling output error systems," *Journal of Vibration and Control*, vol. 17, no. 3, pp. 471–478, 2011.

[29] D. Wang and F. Ding, "Least squares based and gradient based iterative identification for Wiener nonlinear systems," *Signal Processing*, vol. 91, no. 5, pp. 1182–1189, 2011.

[30] F. Ding, J. Ma, and Y. Xiao, "Newton iterative identification for a class of output nonlinear systems with moving average noises," *Nonlinear Dynamics*, vol. 74, no. 1-2, pp. 21–30, 2013.

[31] J. Li, R. Ding, and Y. Yang, "Iterative parameter identification methods for nonlinear functions," *Applied Mathematical Modelling*, vol. 36, no. 6, pp. 2739–2750, 2012.

[32] B. Pan, A. Asundi, H. Xie, and J. Gao, "Digital image correlation using iterative least squares and pointwise least squares for displacement field and strain field measurements," *Optics and Lasers in Engineering*, vol. 47, no. 7-8, pp. 865–874, 2009.

[33] R. Zhang, P. Li, A. Xue, A. Jiang, and S. Wang, "A simplified linear iterative predictive functional control approach for chamber pressure of industrial coke furnace," *Journal of Process Control*, vol. 20, no. 4, pp. 464–471, 2010.

[34] R. Zhang, A. Xue, J. Wang, S. Wang, and Z. Ren, "Neural network based iterative learning predictive control design for mechatronic systems with isolated nonlinearity," *Journal of Process Control*, vol. 19, no. 1, pp. 68–74, 2009.

[35] F. Ding, "Two-stage least squares based iterative estimation algorithm for CARARMA system modeling," *Applied Mathematical Modelling*, vol. 37, no. 7, pp. 4798–4808, 2013.

[36] Y. Gu, F. Ding, and J. H. Li, "State filtering and parameter estimation for linear systems with d-step state-delay," *IET Signal Processing*, vol. 8, no. 6, pp. 639–646, 2014.

[37] F. Ding and H. Duan, "Two-stage parameter estimation algorithms for Box-Jenkins systems," *IET Signal Processing*, vol. 7, no. 8, pp. 646–654, 2013.

[38] D. Q. Zhu and M. Kong, "Adaptive fault-tolerant control of nonlinear system: An improved CMAC based fault learning approach," *International Journal of Control*, vol. 80, no. 10, pp. 1576–1594, 2007.

[39] D. Zhu, J. Liu, and S. X. Yang, "Particle swarm optimization approach to thruster fault-tolerant control of unmanned underwater vehicles," *International Journal of Robotics and Automation*, vol. 26, no. 3, pp. 426–432, 2011.

[40] D. Q. Zhu, Q. Liu, and Z. Hu, "Fault-tolerant control algorithm of the manned submarine with multi-thruster based on quantum-behaved particle swarm optimisation," *International Journal of Control*, vol. 84, no. 11, pp. 1817–1829, 2011.

[41] D. Zhu, H. Huang, and S. X. Yang, "Dynamic task assignment and path planning of multi-AUV system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 504–514, 2013.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 857325, 9 pages
http://dx.doi.org/10.1155/2015/857325

Research Article

Prediction of "Aggregation-Prone" Peptides with Hybrid Classification Approach

Bo Liu,[1] Wenyi Zhang,[2] Longjia Jia,[2] Jianan Wang,[2] Xiaowei Zhao,[2] and Minghao Yin[2]

[1] *School of Physical Education, Northeast Normal University, Changchun 130117, China*
[2] *School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China*

Correspondence should be addressed to Xiaowei Zhao; zhaoxw303@nenu.edu.cn and Minghao Yin; mhyinnenu@163.com

Received 23 July 2014; Revised 29 September 2014; Accepted 29 September 2014

Academic Editor: Shifei Ding

Protein aggregation is a biological phenomenon caused by misfolding proteins aggregation and is associated with a wide variety of diseases, such as Alzheimer's, Parkinson's, and prion diseases. Many studies indicate that protein aggregation is mediated by short "aggregation-prone" peptide segments. Thus, the prediction of aggregation-prone sites plays a crucial role in the research of drug targets. Compared with the labor-intensive and time-consuming experiment approaches, the computational prediction of aggregation-prone sites is much desirable due to their convenience and high efficiency. In this study, we introduce two computational approaches Aggre_Easy and Aggre_Balance for predicting aggregation residues from the sequence information; here, the protein samples are represented by the composition of *k-spaced amino acid pairs* (CKSAAP). And we use the hybrid classification approach to predict aggregation-prone residues, which integrates the naïve Bayes classification to reduce the number of features, and two undersampling approaches EasyEnsemble and BalanceCascade to deal with samples imbalance problem. The Aggre_Easy achieves a promising performance with a sensitivity of 79.47%, a specificity of 80.70% and a MCC of 0.42; the sensitivity, specificity, and MCC of Aggre_Balance reach 70.32%, 80.70% and 0.42. Experimental results show that the performance of Aggre_Easy and Aggre_Balance predictor is better than several other state-of-the-art predictors. A user-friendly web server is built for prediction of aggregation-prone which is freely accessible to public at the website.

1. Introduction

Protein aggregation is a phenomenon caused by misfolding proteins aggregation. Many studies indicate that protein aggregations can cause amyloid fibrils which are associated with a wide variety of diseases, such as Alzheimer's, Parkinson's, and prion diseases [1]. Although the amyloidogenic proteins do not share the homology in sequences or common native fold patterns, they are remarkably similar in β structure [1]. Experiments demonstrate that protein aggregation is mediated by short "aggregation-prone" peptide segments. So the identification of aggregation prone in the protein sequences is the key to finding protein aggregation phenomenon. As we know, traditional experimental identification and characterization of aggregation prone are labor-intensive and expensive. Therefore, the aggregation residues prediction by computational technology attracted more and more attention in the past few years.

Over the past ten years, a large number of computational approaches have been developed to analyze and predict the aggregation prone. Broadly, from the perspective of feature extraction, these approaches can be divided into three categories: experiment-based methods, structure-based methods, and physical-chemical attribute-based methods. For example, Aggrescan [2] proposed by Conchillo-Solé et al. and the saturation mutagenesis analysis [3] performed by López de la Paz and Serrano were both validated by experiments. Among structure-based methods, Galzitskaya et al. [4] used a new parameter, "*mean packing density,*" to detect both amyloidogenic and disordered regions in a protein sequence; SALSA [5], Hexapeptide Conf. Energy [1], and SecStr [6] were proposed by β-sheet structure analysis. NetCSSP [7] developed by Kim et al. used CSSP algorithm and 3D structure to predict the amyloid fibril formation. On the other hand, physical-chemical attribute-based methods such as PaFigure [8] proposed by Tian et al. and Tango [9]

Table 1: The number of aggregation sites and nonaggregation sites in training and testing dataset.

Dataset	Number of the proteins	Number of the aggregation sites	Number of the nonaggregation sites
Training datasets	25	923	5074
Testing datasets	8	335	1499
All datasets	33	1258	6573

developed by Fernandez-Escamilla et al. took into account the physical-chemical principles to predict the aggregation prone. Recently, Tsolis et al. developed two methods named AMYLPRED [10] and AMYLPRED2 [11] which integrated 5 predictors and 11 predictors, respectively.

However, the above methods did not consider that the dataset of aggregation-prone prediction was imbalanced, and some methods were based on the structure information which had high computational complexity. For these reasons, we develop two approaches Aggre_Easy and Aggre_Balance based on the sequence information to predict the aggregation residues. In this study, the protein samples are represented by the composition of *k-spaced amino acid pairs* (CKSAAP) [12–14]. Then, we use a hybrid classification approach to solve sample imbalance problem. The hybrid classification approach integrates the naïve Bayes classification to reduce the number of features and undersampling strategy to deal with the class-imbalance problem. Two undersampling algorithms, EasyEnsemble and BalanceCascade, are both utilized in this paper. The Aggre_Easy achieves a promising performance with a sensitivity of 79.47%, a specificity of 80.70%, and a MCC of 0.42; the sensitivity, specificity, and MCC of Aggre_Balance reach 70.32%, 80.70%, and 0.42. Experimental results show that the performance of Aggre_Easy and Aggre_Balance predictor is better than several other state-of-the-art predictors. A user-friendly web server is built for prediction of aggregation prone which is freely accessible to public at the following website: http://202.198.129.220:8080/AggrePrediction/.

2. Materials and Method

2.1. Datasets. In this paper, we select 33 amyloidogenic proteins to predict "aggregation-prone" peptides. And all the proteins are extracted from Uniprot/Swiss-Prot (Mar, 20, 2013). Moreover, in order to facilitate comparison with the AMYLPRED2, we select the same dataset. For aggregation-prone peptides prediction, 25 proteins are used for training and the remaining 8 proteins for testing. Similar to [11], all experimentally verified aggregation sites in this paper are regarded as positive samples, and the other nonaggregation sites in the same proteins are taken as the negative samples (as can be seen in Supporting Information Text S1 in Supplementary Material available online at http://dx.doi.org/10.1155/2015/857325). The number of proteins samples in each dataset is shown in the Table 1.

We define a possible aggregation-prone peptide $(2\times w+1)$ as the aggregation bond is flanked by "w" residues upstream and "w" residues downstream from the aggregation site. In this paper, we select four different values of w (2, 3, 4, and 5), and the window sizes are the 5, 7, 9, and 11. If the aggregation site is located at the N- or C-terminus of the protein and the length of the peptide is smaller than $(2\times w+1)$, one or multiple "O" characters are added to complement the peptide $(2 \times w + 1)$.

2.2. Protein Encoding Schema. To develop a powerful predictor, an effective mathematical expression to formulate the protein sequences plays an important role, which can truly reflect their intrinsic correlation with the attribute to be predicted [15, 16]. In this research, we use the encoding scheme based the *composition of k-spaced amino acid pairs* (CKSAAP) [12–14], which is successfully used for predicting more types of posttranslational modifications (PTMs) sites (e.g., prediction of palmitoylation sites [13], ubiquitination sites [12], and Phosphorylation Sites [17]). We describe the detailed procedures as follows.

Generally, we define an aggregation prone with a sequence fragment of $(2 \times w + 1)$ amino acids. There are 441 possible amino acid pair types (i.e., $AA, AC, AD, \ldots, OO$). Note that the pairs are extended to the *k-spaced amino acid pairs* (i.e., pairs that are separated by k other amino acids). We can use the vector $(m^k_{AA} m^k_{AC} m^k_{AD} \cdots m^k_{OO})_{441}$ to describe a feature vector. For instance, m^k_{AC} denotes that AC pairs occur m times, and amino acid A is separated by k other amino acid from the amino acid C in the sequences fragment. In this study, based on previous experience, the amino acid pairs for $k = 3, 4, 5$ are jointly considered. So the total dimension of the proposed feature vector is $441 \times (k - 1)$.

2.3. Hybrid Classification Approach. From Section 2.1, we can see that the negative samples are about five times of positive samples, so the traditional learning algorithm, such as SVM, cannot get good performance in this kind of imbalance dataset. For the huge number of features transformed from the CKSAAP encoding, many feature selections methods are carried out to overcome this problem by reducing the dimension of the features. In this paper, we design a hybrid classification approach integrating naïve Bayes classification and two undersampling methods EasyEnsemble and BalanceCascade to predict aggregation sites, which has also been successfully used for text document classification [18, 19]. It takes advantage of both the simplicity of the Bayes technique and the efficient strategy of the undersampling to deal with class-imbalance problem. In Figure 1, the black frame illustrates the process of hybrid classification approach. Firstly, all training proteins or peptides are represented by CKSAAP encoding schema. Secondly, the features of *composition of k-spaced amino acid pairs* are used as the input data for Bayes classification, where the dimensions of process of Bayes classification are based on the number of available classes in the classification task [20]. Finally, we use the undersampling approaches for the predictors. For the query protein, we can use the training model to predict whether or not it is

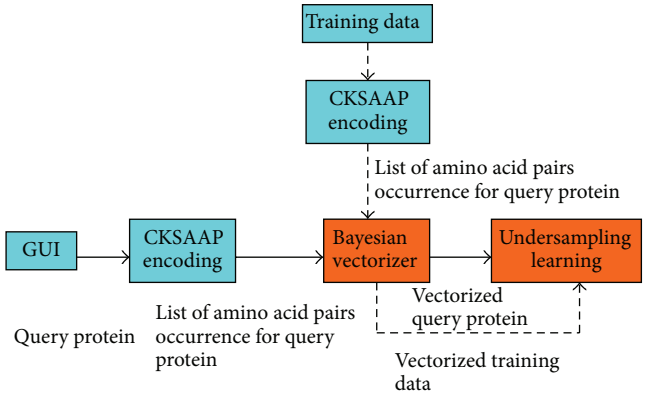


Figure 1: Proposed hybrid classification approach block diagram.

an aggregation protein. The details would be shown in the following sections.

2.3.1. The Bayes Classification Approach. The naïve Bayes classifier [21] starts with the initial step of encoding the sample by extracting the *composition of k-spaced amino acid pairs* (CKSAAP). The list of AAPs (amino acid pairs) is constructed with the assumption that input data contains $\text{aap}_1, \text{aap}_2, \ldots, \text{aap}_{n-1}, \text{aap}_n$, where n is the CKSAAP encoding schema length. And it can be used to create a table; containing the probabilities of the amino acid pair (AAP) in each class. And Table 2 shows the details.

Based on the list of AAP numbers, the trained probabilistic classifier calculates the posterior probability of the particular AAP of the sample being annotated to a particular class by using the formula (1), since each AAP in the input sample contributes to the sample's class probability:

$$\Pr(\text{Class}, \text{AAP}) = \frac{\Pr(\text{AAP}, \text{Class}) \cdot \Pr(\text{Class})}{\Pr(\text{AAP})}. \quad (1)$$

The prior probability, $\Pr(\text{Class}_i)$ can be computed from

$$\Pr(\text{Class}_i) = \frac{\text{Total\_of\_AAPs\_in\_Class}_i}{\text{Total\_of\_AAPs\_in\_Training\_Dataset}}. \quad (2)$$

Meanwhile, we calculate the $\Pr(\text{aap}_j)$, which is represented by the probability of each aap_j in all classes; it is expressed as

$$\Pr(\text{aap}_j) = \frac{\sum \text{numbers\_of\_aap}_i\text{\_in\_all\_Classes}}{\sum \text{numbers\_of\_all\_AAPs\_in\_all\_Classes}}. \quad (3)$$

The total occurrence of a particular AAP in every class can be calculated by searching the training database, which is composed from the lists of AAP occurrences for every class. As previously mentioned, the list of AAP numbers for a class is generated from the analysis of all training samples in the particular class during the initial training stage. The same method can be used to retrieve the sum of numbers of all samples in every class in the training database.

To calculate the likelihood of a particular Class_i with respect to a particular aap_j, the lists of AAP number from the training database is searched to retrieve the numbers of aap_j in Class_i and the sum of all AAPs in Class_i. This information will contribute to the value of $\Pr(\text{aap}_j, \text{Class}_i)$ given in

$$\Pr(\text{aap}_j, \text{Class}_i) = \frac{\text{numbers\_of\_aap}_j\text{\_in\_Class}_i}{\sum \text{numbers\_of\_all\_aaps\_in\_Class}_i}. \quad (4)$$

Based on the derived Bayes' formula for classification and the value of the prior probability Pr(Class), the likelihood Pr(AAP, Class) and evidence Pr(AAP), along with the posterior probability, Pr(Class, AAP) of each AAP in the input data being annotated to each class can be measured.

The probability for an input sample to be annotated to a particular Class_i is calculated by dividing the sum of each of the "Probability" column with the length of the query, n, which is shown in

$$\Pr(\text{Class}_i, \text{Sample}) = \frac{\Pr(\text{Class}_i, \text{aap}_1, \text{aap}_2, \text{aap}_3, \ldots, \text{aap}_{n-1}, \text{aap}_n)}{n}, \quad (5)$$

where $\text{aap}_1, \text{aap}_2, \text{aap}_3, \ldots, \text{aap}_{n-1}, \text{aap}_n$ are the AAPs that are extracted from the input sample.

The Pr(Class, Sample) is the probability value for a sample to be annotated to a class. And if we have class list as Class_1, Class_2, $\text{Class}_3, \ldots, \text{Class}_N$, each sample would have N associated probability values, where sample will have $\Pr(\text{Class}_1, \text{Sample})$, $\Pr(\text{Class}_2, \text{Sample})$, $\Pr(\text{Class}_3, \text{Sample}), \ldots$, and $\Pr(\text{Class}_N, \text{Sample})$. All the probability values of a sample are combined to construct a multidimensional array, which represents the probability distribution of the sample in the vector space. In this way, all the training samples are vectorized into their probability distribution in vector space, in the format of numerical multidimensional arrays, with the number of dimensions depending on the number of classes.

2.3.2. EasyEnsemble and BalanceCascade. Liu et al. proposed EasyEnsemble algorithm and BalanceCascade algorithm [22], which were the undersampling algorithms and were widely used for classification task. EasyEnsemble algorithm extracted several subsets from majority class examples by themselves; for each subset, a classifier was built, and all generated classifiers created an ensemble learning system and then combined them for the final decision by using Adaboost [23]. BalanceCascade algorithm depending on supervised learning methods extracted examples from majority class examples and then created ensemble classifiers with training datasets [24]. The pseudocodes for EasyEnsemble and BalanceCascade were shown in Algorithms 1 and 2.

2.4. Evaluation. In this study, we adopt the 10-fold cross-validation. The dataset is randomly divided into ten equal sets, out of which nine sets are used for training and the remaining one for testing. This procedure is repeated ten times and the final prediction result is the average accuracy of the ten testing sets [25–32].

Four parameters, sensitivity (Sn), specificity (Sp), Q, and Mathew correlation coefficient (MCC), are used to measure

Table 2: Table of AAPs numbers and probabilities.

AAP	Probability Class_1	Probability Class_2	Probability Class_3	$\cdots$	Probability Class_{k-1}	Probability Class_k
aap_1	$\Pr(\text{Class}_1, \text{aap}_1)$	$\Pr(\text{Class}_2, \text{aap}_1)$	$\Pr(\text{Class}_3, \text{aap}_1)$	$\cdots$	$\Pr(\text{Class}_{k-1}, \text{aap}_1)$	$\Pr(\text{Class}_k, \text{aap}_1)$
aap_2	$\Pr(\text{Class}_1, \text{aap}_2)$	$\Pr(\text{Class}_2, \text{aap}_2)$	$\Pr(\text{Class}_3, \text{aap}_2)$	$\cdots$	$\Pr(\text{Class}_{k-1}, \text{aap}_2)$	$\Pr(\text{Class}_k, \text{aap}_2)$
aap_3	$\Pr(\text{Class}_1, \text{aap}_3)$	$\Pr(\text{Class}_2, \text{aap}_3)$	$\Pr(\text{Class}_3, \text{aap}_3)$	$\cdots$	$\Pr(\text{Class}_{k-1}, \text{aap}_3)$	$\Pr(\text{Class}_k, \text{aap}_3)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
aap_{n-1}	$\Pr(\text{Class}_1, \text{aap}_{n-1})$	$\Pr(\text{Class}_2, \text{aap}_{n-1})$	$\Pr(\text{Class}_3, \text{aap}_{n-1})$	$\cdots$	$\Pr(\text{Class}_{k-1}, \text{aap}_{n-1})$	$\Pr(\text{Class}_k, \text{aap}_{n-1})$
aap_n	$\Pr(\text{Class}_1, \text{aap}_n)$	$\Pr(\text{Class}_2, \text{aap}_n)$	$\Pr(\text{Class}_3, \text{aap}_n)$	$\cdots$	$\Pr(\text{Class}_{k-1}, \text{aap}_n)$	$\Pr(\text{Class}_k, \text{aap}_n)$
Total	$\sum \Pr(\text{Class}_1, \text{AAP})$	$\sum \Pr(\text{Class}_2, \text{AAP})$	$\sum \Pr(\text{Class}_3, \text{AAP})$	$\cdots$	$\sum \Pr(\text{Class}_{k-1}, \text{AAP})$	$\sum \Pr(\text{Class}_k, \text{AAP})$
Probability of input sample	$\sum \Pr(\text{Class}_1, \text{AAP})/n$	$\sum \Pr(\text{Class}_2, \text{AAP})/n$	$\sum \Pr(\text{Class}_3, \text{AAP})/n$	$\cdots$	$\sum \Pr(\text{Class}_{k-1}, \text{AAP})/n$	$\sum \Pr(\text{Class}_k, \text{AAP})/n$

Input: Training dataset $S_r = \{(x, y)\}$, the number of individuals T, the number of iterations S_i
(1) Begin
(2) For $i = 1 : T$
(3) Creating a subset S_i^- from negative dataset of S_r^- by using Bootstrap sampling technique, and the number S_i^- is equal to the S_r^+
(4) Use the Adaboost with the weak classifiers $h_{i,j}$ and corresponding weights $\alpha_{i,j}$ to train the individual model N_i, the ensemble's threshold is θ_i, i.e.

$$N_i(x) = \operatorname{sgn}\left(\sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(x) - \theta_i\right).$$

(5) End For
(6) Output: An ensemble like:

$$N(x) = \operatorname{sgn}\left(\sum_{i=1}^{T}\sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(x) - \sum_{i=1}^{T} \theta_i\right)$$

(7) End

Algorithm 1: The EasyEnsemble algorithm.

Input: Training dataset $S_r = \{(x, y)\}$, the number of individuals T, the number of iterations S_i
(1) Begin
(2) $f = \sqrt[T-1]{|P|/|N|}$ f is the false positive rate (the error rate of misclassifying a majority class example to the minority class) that N_i should achieve
(3) For $I = 1 : T$
(4) Creating a subset S_i^- from negative dataset of S_r^- by using Bootstrap sampling technique, and the number S_i^- is equal to the S_r^+
(5) Use the Adaboost with the weak classifiers $h_{i,j}$ and corresponding weights $\alpha_{i,j}$ to train the individual model N_i, the ensemble's threshold is θ_i, i.e.

$$N_i(x) = \operatorname{sgn}\left(\sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(x) - \theta_i\right)$$

(6) Adjust θ_i such that N_i's false positive rate is f.
(7) Remove from S_r^- all examples that are correctly classified by N_i
(8) End for
(9) Output: A single ensemble like:

$$N(x) = \operatorname{sgn}\left(\sum_{i=1}^{T}\sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(x) - \sum_{i=1}^{T} \theta_i\right)$$

(10) End

Algorithm 2: The BalanceCascade algorithm.

the performance of our model. They are defined by the following formulas:

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}},$$

$$Q = \frac{\mathrm{Sn} + \mathrm{Sp}}{2},$$

$$\mathrm{MCC} = \frac{(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FN} \times \mathrm{FP})}{\sqrt{(\mathrm{TP} + \mathrm{FN}) \times (\mathrm{TN} + \mathrm{FP}) \times (\mathrm{TP} + \mathrm{FP}) \times (\mathrm{TN} + \mathrm{FN})}}, \quad (6)$$

where TP, TN, FP, and FN denote the number of true positive, true negative, false positive, and false negative, respectively. For a given dataset, all these values can be obtained from the decision function with fixed cutoff [33–37].

3. Result and Discussion

3.1. The Performance in the Testing Dataset. In this research, we select 33 amyloidogenic proteins for the prediction of "aggregation-prone" peptides. And 25 amyloidogenic proteins are selected for training; there are 923 positive samples and 5074 negative samples; and the rest of 8 amyloidogenic proteins are selected for testing; thus, there are 335 positive samples and 1499 negative samples. The details are shown in Table 1. We define a possible aggregation-prone peptide $(2 \times w + 1)$ as the aggregation bond; "w" is 3, 4, and 5, and the window size is 7, 9, and 11. Next, we use the encoding scheme based on the *composition of k-spaced amino acid pairs* (CKSAAP) to formulate the aggregation-prone peptide, and the "k" is 3, 4, and 5. In Tables 3 and 4, we compare the values of MCC to determine the best values of w and k.

Table 3: The performance for EasyEnsemble learning algorithm in testing dataset.

Window size	The value of the k	Q	MCC	TP	FN	TN	FP
5	3	0.5318	0.0514	136	199	986	513
7	3	0.5438	0.0725	131	204	1044	455
9	3	0.5426	0.0706	130	205	1045	454
11	3	0.5457	0.0779	122	213	1092	407
5	4	0.5443	0.0720	140	195	1006	493
7	**4**	**0.5497**	**0.0827**	**133**	**202**	**1052**	**447**
9	4	0.5347	0.0595	115	220	1090	409
11	4	0.5347	0.0598	112	223	1101	398
5	5	0.5417	0.0677	139	196	1002	497
7	5	0.5188	0.0313	117	218	1034	465
9	5	0.5303	0.0512	116	219	1069	430
11	5	0.5308	0.0542	104	231	1128	371

Table 4: The performance for BalanceCascade learning algorithm in testing dataset.

Window size	The value of the k	Q	MCC	TP	FN	TN	FP
5	3	0.5292	0.0506	107	228	1107	392
7	3	0.5382	0.0694	100	235	1166	333
9	3	0.5197	0.0660	104	231	1142	356
11	3	0.5357	0.0641	101	234	1152	347
5	4	0.5366	0.0649	106	229	1135	367
7	**4**	**0.5405**	**0.0738**	**101**	**234**	**1168**	**331**
9	4	0.5319	0.0580	97	238	1163	336
11	4	0.5253	0.0489	81	254	1213	286
5	5	0.5379	0.0667	108	227	1130	369
7	5	0.5197	0.0353	94	241	1139	360
9	5	0.5293	0.0543	92	243	1175	324
11	5	0.5253	0.0470	81	254	1208	291

We use the hybrid classification approach (naïve Bayes vectorizer and two undersampling algorithms called EasyEnsemble and BalanceCascade) to improve the classification accuracy and performance in the imbalance dataset. For the EasyEnsemble approach, the CART is used to train weak classifiers; the number subset T is 4; the number of iterations S_i is 10 in the each Adaboost ensemble method; the same parameters are used for the BalanceCascade approach. Meanwhile, we perform a 10-fold stratified cross validation. Within each fold, the classification method is repeated 10 times considering that the sampling of subsets introduces randomness. The whole cross validation process is repeated 10 times, and the averages of these 10 cross validations are the final performance of the method.

The average performance of the different parameter is summarized in Tables 3 and 4. When the window size is 7 and the k value was 4, the value of MCC is the highest, 0.0827 for EasyEnsemble learning algorithm and 0.0738 for BalanceCascade learning algorithm in the testing dataset. Thus, we select 7 (window size) and 4 (the k value) as the final parameters of classifier, which is used to comprise with other predictors by 10-fold cross validation in all datasets.

The average Sn of the EasyEnsemble learning algorithm and BalanceCascade learning algorithm is shown in Figures 2 and 4. When the window size is smaller, the value of Sn is higher; for example, window size is 5 and 7, and the Sn is about 0.39~0.41 for EasyEnsemble and 0.27~0.32 for BalanceCascade; on the contrary, when the window size is 9 and 11, the Sn is about 0.34~0.38 for EasyEnsemble and 0.24~0.31 for BalanceCascade. Also in Figures 3 and 5, the average Sp of the EasyEnsemble learning algorithm and BalanceCascade learning algorithm is summarized. It is about 0.66~0.70 for EasyEnsemble and 0.73~0.77 for BalanceCascade, when the window size is 5 and 7; however, it is about 0.69~0.75 for EasyEnsemble and 0.76~0.80 for BalanceCascade, when the window size is 9 and 11. It indicates that smaller window size would be beneficial to predict positive sample; also, the larger the window size is, the more redundant the information is. What's more, the value of Sn is higher for EasyEnsemble than for BalanceCascade, about 10% higher; however, the value of Sp is lower for EasyEnsemble than for BalanceCascade, about 7% lower; it illustrates that the EasyEnsemble would improve the prediction performance of sensitivity, and BalanceCascade would improve the prediction performance of specificity.

3.2. Comparison with Other Predictors. As the result in Table 5, the prediction sensitivity and MCC of Aggre_Easy

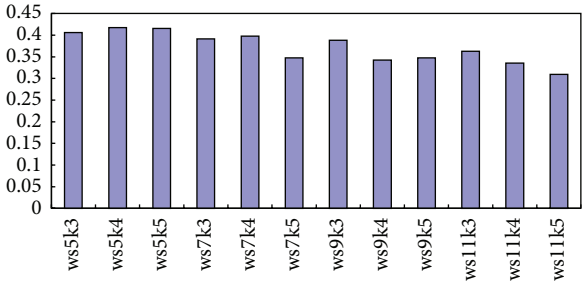


Figure 2: The value of Sn for EasyEnsemble learning algorithm in testing dataset.

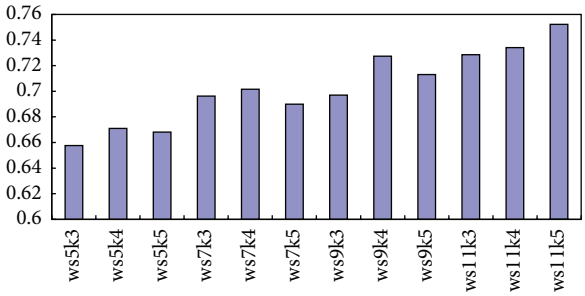


Figure 3: The value of Sp for EasyEnsemble learning algorithm in testing dataset.

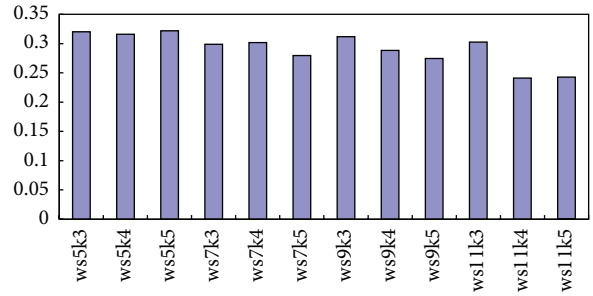


Figure 4: The value of Sn for BalanceCascade learning algorithm in testing dataset.

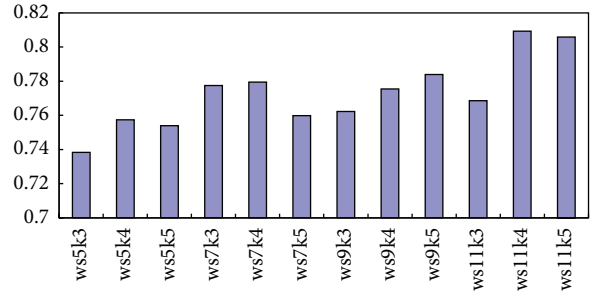


Figure 5: The value of Sp for BalanceCascade learning algorithm in testing dataset.

and Aggre_Balance are the highest compared to others, the Sp is 79.46%, MCC is 0.42 for the Aggre_Easy, the Sp is 70.32%, MCC is 0.42 for the Aggre_Balance. It indicates that our predictor has good performance to predict the positive samples in the imbalance dataset. However, the value of specificity is lower than others. For Aggre_Easy, the value of specificity (Sp = 74.43%) is lower than Amyloidogenic Pattern, Average Packing Density, Beta-strand contiguity, SecStr, Tango, AMYLPRED, and AMYLPRED2, slightly lower than Aggrescan, AmyloidMutants, and Hexapeptide Conf. Energy, and higher than NetCSSP, PaFigure, and Waltz. For Aggre_Balance, the value of specificity (Sp = 80.70%) is lower than Amyloidogenic Pattern, Average Packing Density, Beta-strand contiguity, SecStr, Tango, AMYLPRED, and AMYLPRED2 and higher than other methods. More importantly, the reasonably good performance of Aggre_Easy and Aggre_Balance reflects that the method effectively captures the information of aggregation sites, and we propose that the hybrid classification approach can take advantage of the simplicity of the Bayes technique and the sensitivity of the undersampling ensemble learning algorithm.

In Table 5, the false positives (FP) is large; the main reason was because of the fact that only a relative small portion of them have been studied and confirmed experimentally to be amyloidogenic [11]. On the other hand, we would propose the window redirection operator to improve the prediction performance in the future.

3.3. Web Server for Aggregation-Prone Prediction. An effective prediction servers, Aggre_Easy and Aggre_Balance, are available at http://202.198.129.220:8080/AggrePrediction/. And it is hosted on Apache 2.2 web server by using Windows 2003 server environment. In the web server, the models based on the datasets with the optimal parameters are used to predict sites in submitted sequences. As is displayed in Figures 6 and 7, users could submit the uncharacteristic sequences with FASTA format, and the system would return the prediction results. A region in the polypeptide sequence was considered an aggregation prone if there are 5 or more sequentially continuous residues to be prediction aggregation prone.

4. Conclusion

Accurate identification of the aggregation residues could help fully decipher the molecular mechanisms. Though some researchers have focused on this problem, the overall prediction performance is still not satisfied. In this paper, we develop approaches Aggre_Easy and Aggre_Balance to predict the aggregation prone from the primary sequence information. The Aggre_Easy achieves a promising performance with a sensitivity of 79.47%, a specificity of 80.70%, and a MCC of 0.42; the sensitivity, specificity, and MCC of Aggre_Balance reach 70.32%, 80.70%, and 0.42. Experimental results show that the performances of Aggre_Easy and Aggre_Balance predictor are better than several other state-of-the-art predictors and our methods are helpful for the prediction of aggregation prone.

Table 5: The performance comparison of Aggre_Easy and Aggre_Balance with existing 12 predictors.

Method	Sn (%)	Sp (%)	Q (%)	MCC	TP	TN	FP	FN
Aggrescan	35.37	79.26	57.32	0.13	445	5210	1363	813
AmyloidMutants	41.65	74.91	58.28	0.14	524	4924	1649	734
Amyloidogenic Pattern	13.99	94.95	54.22	0.12	176	6208	365	1082
Average Packing Density	28.70	84.12	56.41	0.12	361	5529	1044	897
Beta-strand contiguity	33.15	85.62	59.39	0.18	417	5628	945	841
Hexapeptide Conf. Energy	39.27	78.69	58.98	0.15	494	5172	1401	764
NetCSSP	51.27	65.22	58.25	0.12	645	4287	2286	613
PaFigure	51.75	71.43	61.59	0.18	651	4695	1878	607
SecStr	11.37	94.40	52.88	0.09	143	6205	368	1115
Tango	13.67	95.57	54.62	0.14	172	6282	291	1086
Waltz	56.44	65.42	60.93	0.16	710	4300	2273	548
AMYLPRED	32.99	86.23	59.61	0.19	415	5668	905	843
AMYLPRED2	39.27	84.48	61.88	0.22	494	5553	1020	764
Aggre_Easy	79.46	74.43	76.95	0.42	1000	4892	1681	258
Aggre_Balance	70.32	80.70	75.51	0.42	885	5304	1269	373

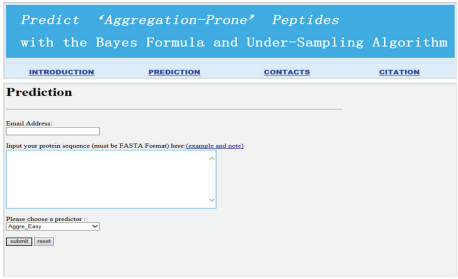


Figure 6: The interface of user input.

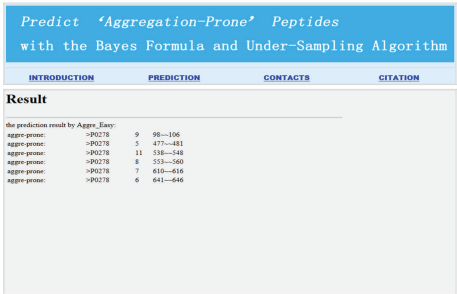


Figure 7: The example of prediction result by Aggre_Easy.

Supporting Information

Text S1: All datasets are consisting of the 33 proteins and their sites information.

Text S2: The prediction result of aggregation prone for Aggre_Easy, Aggre_Balance, AMYLPRED and AMYLPRED2, by comparison. Simply, we remove the single prediction positive site, and, in the future, we will propose the window redirection operator to improve the prediction performance.

Conflict of Interests

The authors declare no conflict of interests.

Acknowledgments

This research is partially supported by National Natural Science Foundation of China (61403077 and 61403076), the Fundamental Research Funds for the Central Universities (14QNJJ029), and the Postdoctoral Science Foundation of China (2014M550166).

References

[1] Z. Zhang, H. Chen, and L. Lai, "Identification of amyloid fibril-forming segments based on structure and residue-based statistical potential," *Bioinformatics*, vol. 23, no. 17, pp. 2218–2225, 2007.

[2] O. Conchillo-Solé, N. S. de Groot, F. X. Avilés, J. Vendrell, X. Daura, and S. Ventura, "AGGRESCAN: a server for the prediction and evaluation of "hot spots" of aggregation in polypeptides," *BMC Bioinformatics*, vol. 8, article 65, 2007.

[3] M. López de la Paz and L. Serrano, "Sequence determinants of amyloid fibril formation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 1, pp. 87–92, 2004.

[4] O. V. Galzitskaya, S. O. Garbuzynskiy, and M. Y. Lobanov, "Prediction of amyloidogenic and disordered regions in protein chains," *PLoS Computational Biology*, vol. 2, article e177, 2006.

[5] S. Zibaee, O. S. Makin, M. Goedert, and L. C. Serpell, "A simple algorithm locates β-strands in the amyloid fibril core of α-synuclein, Aβ, and tau using the amino acid sequence alone," *Protein Science*, vol. 16, no. 5, pp. 906–918, 2007.

[6] S. J. Hamodrakas, C. Liappa, and V. A. Iconomidou, "Consensus prediction of amyloidogenic determinants in amyloid fibril-forming proteins," *International Journal of Biological Macromolecules*, vol. 41, no. 3, pp. 295–300, 2007.

[7] C. Kim, J. Choi, S. J. Lee, W. J. Welsh, and S. Yoon, "NetCSSP: web application for predicting chameleon sequences and amyloid fibril formation," *Nucleic Acids Research*, vol. 37, no. 2, pp. W469–W473, 2009.

[8] J. Tian, N. Wu, J. Guo, and Y. Fan, "Prediction of amyloid fibril-forming segments based on a support vector machine," *BMC Bioinformatics*, vol. 10, supplement 1, article S45, 2009.

[9] A.-M. Fernandez-Escamilla, F. Rousseau, J. Schymkowitz, and L. Serrano, "Prediction of sequence-dependent and mutational effects on the aggregation of peptides and proteins," *Nature Biotechnology*, vol. 22, no. 10, pp. 1302–1306, 2004.

[10] K. K. Frousios, V. A. Iconomidou, C.-M. Karletidi, and S. J. Hamodrakas, "Amyloidogenic determinants are usually not buried," *BMC Structural Biology*, vol. 9, article 44, 2009.

[11] A. C. Tsolis, N. C. Papandreou, V. A. Iconomidou, and S. J. Hamodrakas, "A consensus method for the prediction of 'Aggregation-Prone' peptides in globular proteins," *PLoS ONE*, vol. 8, no. 1, Article ID e54175, 2013.

[12] Z. Chen, Y.-Z. Chen, X.-F. Wang, C. Wang, R.-X. Yan, and Z. Zhang, "Prediction of ubiquitination sites by using the composition of K-spaced amino acid pairs," *PLoS ONE*, vol. 6, no. 7, Article ID e22930, 2011.

[13] X.-B. Wang, L.-Y. Wu, Y.-C. Wang, and N.-Y. Deng, "Prediction of palmitoylation sites using the composition of k-spaced amino acid pairs," *Protein Engineering, Design and Selection*, vol. 22, no. 11, pp. 707–712, 2009.

[14] Y. Z. Chen, Y. R. Tang, Z. Y. Sheng, and Z. Zhang, "Prediction of mucin-type O-glycosylation sites in mammalian proteins using the composition of k-spaced amino acid pairs," *BMC bioinformatics*, vol. 9, article 101, 2008.

[15] W. Zhang, X. Xu, M. Yin, N. Luo, J. Zhang, and J. Wang, "Prediction of methylation sites using the composition of K-spaced amino acid pairs," *Protein and Peptide Letters*, vol. 20, no. 8, pp. 911–917, 2013.

[16] Y. Xu, J. Ding, L.-Y. Wu, and K.-C. Chou, "iSNO-PseAAC: predict cysteine S-nitrosylation sites in proteins by incorporating position specific amino acid propensity into pseudo amino acid composition," *PLoS ONE*, vol. 8, no. 2, Article ID e55844, 2013.

[17] X. Zhao, W. Zhang, X. Xu, Z. Ma, and M. Yin, "Prediction of protein phosphorylation sites by using the composition of k-spaced amino acid Pairs," *PLoS ONE*, vol. 7, no. 10, Article ID e46302, 2012.

[18] L. H. Lee, R. Rajkumar, and D. Isa, "Automatic folder allocation system using Bayesian-support vector machines hybrid classification approach," *Applied Intelligence*, vol. 36, no. 2, pp. 295–307, 2012.

[19] D. Isa, V. P. Kallimani, and L. H. Lee, "Using the self organizing map for clustering of text documents," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9584–9591, 2009.

[20] D. Isa, L. H. Lee, V. P. Kallimani, and R. Rajkumar, "Text document preprocessing with the bayes formula for classification using the support vector machine," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 9, pp. 1264–1272, 2008.

[21] D. Isa, L. H. Lee, and V. P. Kallimani, "A polychotomizer for case-based reasoning beyond the traditional Bayesian classification approach," *Journal of Computer and Information Science*, vol. 1, no. 1, pp. 57–68, 2008.

[22] X. Liu, J. Wu, and Z. Zhou, "Exploratory under-sampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 539–550, 2008.

[23] T. Liu, "Feature selection based on mutual information for gear faulty diagnosis on imbalanced dataset," *Journal of Computational Information Systems*, vol. 8, no. 18, pp. 7831–7838, 2012.

[24] G. Sang, L. Gao, and Z. Liu, "A Bias-ensemble learning algorithm for imbalanced data processing," *Journal of Computational Information Systems*, vol. 9, no. 5, pp. 2025–2032, 2013.

[25] P. Huang and M. H. Yin, "An upper (lower) bound for Max (Min) CSP," *Science China Information Sciences*, vol. 7, pp. 1–9, 2014.

[26] X. Li and M. Yin, "Multiobjective binary biogeography based optimization for feature selection using gene expression data," *IEEE Transactions on Nanobioscience*, vol. 12, no. 4, pp. 343–353, 2013.

[27] X. Li and M. H. Yin, "Modified differential evolution with self-adaptive parameters method," *Journal of Combinatorial Optimization*, 2014.

[28] X. Li, J. Wang, J. Zhou, and M. Yin, "A perturb biogeography based optimization with mutation for global numerical optimization," *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 598–609, 2011.

[29] X. Li and M. Yin, "A hybrid cuckoo search via Lévy flights for the permutation flow shop scheduling problem," *International Journal of Production Research*, vol. 51, no. 16, pp. 4732–4754, 2013.

[30] J. Zhou, P. Huang, M. Yin, and C. Zhou, "Phase transitions of EXPSPACE-complete problems," *International Journal of Foundations of Computer Science*, vol. 21, no. 6, pp. 1073–1088, 2010.

[31] J. P. Zhou, M. H. Yin, and C. G. Zhou, "New worst upper bound for #SAT," in *Proceedings of AAAI*, 2010.

[32] J. Zhou, M. Yin, X. Li, and J. Wang, "Phase transitions of EXPSPACE-complete problems: a further step," *International Journal of Foundations of Computer Science*, vol. 23, no. 1, pp. 173–184, 2012.

[33] L. Hu, P. Soderhjelm, and U. Ryde, "On the convergence of QM/MM energies," *Journal of Chemical Theory and Computation*, vol. 7, no. 3, pp. 761–777, 2011.

[34] L. Hu, P. Söderhjelm, and U. Ryde, "Accurate reaction energies in proteins obtained by combining QM/MM and large QM calculations," *Journal of Chemical Theory and Computation*, vol. 9, no. 1, pp. 640–649, 2013.

[35] X. Zheng, L. H. Hu, X. J. Wang, and G. H. Chen, "A generalized exchange-correlation functional: the Neural-Networks approach," *Chemical Physics Letters*, vol. 390, no. 1–3, pp. 186–192, 2004.

[36] H. Z. Li, L. Li, Z. Y. Zhong, Y. Han, L. Hu, and Y. H. Lu, "An accurate and efficient method to predict Y-NO bond homolysis bond dissociation energies," *Mathematical Problems in Engineering*, vol. 2013, Article ID 860357, 10 pages, 2013.

[37] H. Z. Li, L. H. Hu, W. Tao et al., "A promising tool to achieve chemical accuracy for density functional theory calculations on Y-NO homolysis bond dissociation energies," *International Journal of Molecular Sciences*, vol. 13, no. 7, pp. 8051–8070, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 381271, 13 pages
http://dx.doi.org/10.1155/2015/381271

Research Article

Dynamics of Moment Neuronal Networks with Intra- and Inter-Interactions

Xuyan Xiang[1] and Jianguo Wu[2]

[1]*College of Mathematics and Computational Science, Hunan University of Arts and Science, Changde 415000, China*
[2]*Department of Mathematics, Hunan College of Finance and Economics, Changsha 410205, China*

Correspondence should be addressed to Jianguo Wu; wjghn@163.com

Received 6 July 2014; Accepted 7 September 2014

Academic Editor: Shifei Ding

A framework of moment neuronal networks with intra- and inter-interactions is presented. It is to show how the spontaneous activity is propagated across the homogeneous and heterogeneous network. The input-output firing relationship and the stability are first explored for a homogeneous network. For heterogeneous network without the constraint of the correlation coefficients between neurons, a more sophisticated dynamics is then explored. With random interactions, the network gets easily synchronized. However, desynchronization is produced by a lateral interaction such as Mexico hat function. It is the external intralayer input unit that offers a more sophisticated and unexpected dynamics over the predecessors. Hence, the work further opens up the possibility of carrying out a stochastic computation in neuronal networks.

1. Introduction

The theory of moment neuronal networks (MNN) was developed recently [1], which takes into account both the first and the second order statistics of spike trains, generalizing the case of Poisson synaptic inputs to more biologically plausible renewal processes. Therefore, the MNN framework can be considered as an attempt toward a general framework of computation with stochastic systems. Further a more biologically reasonable MNN with intra- and interlayer interactions was developed by introducing intralayer inputs [2], which is analogous to the "networks with context units" [3] in ANNs. It shows that even a single unit with such system, as analogous to [4–6], is able to perform various complex nonlinear tasks like the XOR problem and that it can reach the trade-off between the output bias and variance by determining the optimal penalty factor due to a specific learning task.

In this letter we will explore how the spontaneous activity is propagated across the feedforward network with both homogeneous and heterogeneous connections, derived from the intralayers inputs. The input-output firing relationship and the stability are first explored for a homogeneous network. Synchronization or desynchronization is then presented for heterogeneous network by random or lateral interactions. It is the external intralayer input unit that offers a more sophisticated and unexpectedly dynamics over MNN.

This paper is organized as follows. The framework of MNN with intra- and interlayer interactions is presented in Section 2. The dynamics of homogeneous and heterogeneous network are explored in Section 3.

2. Moment Neuronal Networks with Intra- and Inter-Interactions

For the decay rate L, when the membrane potential $V_i^{k+1}(t)$ of the ith neuron in the $(k+1)$th layer is between its resting state V_r and its threshold V_t, it satisfies the following dynamics:

$$dV_i^{(k+1)}(t) = -L\left[V_i^{(k+1)}(t) - V_r\right]dt + dI_{i,\mathrm{syn}}^{(k+1)}(t), \quad (1)$$

where the synaptic input $I_{i,\mathrm{syn}}^{(k+1)}(t)$ is given by

$$dI_{i,\mathrm{syn}}^{(k+1)}(t) = \sum_{j=1}^{p^{(k),ir}} \omega_{ij}^{E,(k),ir} dN_j^{E,(k),ir}(t) - \sum_{j=1}^{q^{(k),ir}} \omega_{ij}^{I,(k),ir} dN_j^{I,(k),ir}(t)$$

$$+\sum_{j=1}^{p^{(k+1),ia}}\omega_{ij}^{E,(k+1),ia}dN_j^{E,(k+1),ia}(t)$$
$$-\sum_{j=1}^{q^{(k+1),ia}}\omega_{ij}^{I,(k+1),ia}dN_j^{I,(k+1),ia}(t). \quad (2)$$

Here $\omega_{ij}^{E,(k),ir}$ and $\omega_{ij}^{I,(k),ir}$ are the magnitudes of excitatory postsynaptic potentials (EPSPs) and inhibitory postsynaptic potentials (IPSPs), $\{N_i^{E,(k),ir}(t)\}$ and $\{N_i^{I,(k),ir}(t)\}$ are renewal processes arriving from the ith and jth synapse, and $p^{(k)}$ and $q^{(k)}$ are the total number of active excitatory and inhibitory synapses in the kth layer. All notations with the superscript *ia* imply that they receive the external stimulus from intralayer interactions rather than from the network itself, similar to the "networks with context unit" [3]. It would present a more biologically reasonable network architecture than the moment neuronal networks. When $V_i^{(k+1)}(t)$ crosses the membrane threshold V_t from below, a spike is generated and the membrane resets to its resting potential V_r.

The renewal process $\{N_i^{E,(k),ix}(t)\}$ can be approximated as [1, 2]

$$dN_i^{E,(k),ix}(t) \sim \mu_i^{E,(k),ix}dt + \sigma_i^{E,(k),ix}dB_i^{E,(k),ix}(t), \quad (3)$$

with

$$\mu_{ij}^{E,(k),ix} = \frac{1}{\left\langle T_{ij}^{E,(k),ix}\right\rangle + T_{\text{ref}}},$$
$$\left(\sigma_{ij}^{E,(k),ix}\right)^2 = \frac{\text{Var}\left(T_i^{E,(k),ix}\right)}{\left\langle T_{ij}^{E,(k),ix}\right\rangle^3}, \quad (4)$$

where $\langle T_i^{E,(k),ix}\rangle$ and $\text{Var}(T_i^{E,(k),ix})$ are the mean and variance of the interspike intervals (ISIs) of the renewal process $\{N_i^{E,(k),ix}(t)\}$, respectively, the superscript *ix* could be either *ir* or *ia*, and T_{ref} is the refractory period.

For the conciseness of notation, we take the intrainput from the $(k+1)$th layer as external context input paralleled to the kth layer, which is denoted by the superscript *ia*. Furthermore, we suppose that $q^{(k),ix} = p^{(k),ix}$, $\mu^{(k),ix} \equiv \mu^{I,(k),ix} = \mu^{E,(k),ix}$, $\sigma^{(k),ix} \equiv \sigma^{I,(k),ix} = \sigma^{E,(k),ix}$, and $\omega_{ij}^{I,(k),ir} = r^{ir}\omega_{ij}^{E,(k),ir} \equiv r\omega_{ij}^{(k),ir} \equiv r\omega_{ij}^{(k)}$, $\omega_{ij}^{I,(k),ia} = r^{ia}\omega_{ij}^{E,(k),ia} \equiv r^{ia}\omega_{ij}^{(k),ia}$, $\rho_{mn}^{(k)} \equiv \rho_{mn}^{(k),ir}$; then (2) can be approximated as (see [2])

$$dI_{i,\text{syn}}^{(k+1)}(t) \equiv \overline{\mu}_i^{(k)}dt + \overline{\sigma}_i^{(k)}dB(t), \quad (5)$$

where

$$\overline{\mu}_i^{(k)} = \sum_{j=1}^{p^{(k)}}\omega_{ij}^{(k)}\mu_j^{(k)}(1-r) + \sum_{j=1}^{p^{(k),ia}}\omega_{ij}^{(k),ia}\mu_j^{(k),ia}\left(1-r^{ia}\right), \quad (6)$$

$$\left(\overline{\sigma}_i^{(k)}\right)^2 = \sum_{m,n=1}^{p^{(k)}}\omega_{im}^{(k)}\sigma_m^{(k)}\omega_{in}^{(k)}\sigma_n^{(k)}\rho_{mn}^{(k)}\left(1+r^2\right)$$
$$+\sum_{m,n=1}^{p^{(k),ia}}\omega_{im}^{(k),ia}\sigma_m^{(k),ia}\omega_{in}^{(k),ia}\sigma_n^{(k),ia}\rho_{mn}^{(k),ia}\left[1+\left(r^{ia}\right)^2\right]. \quad (7)$$

Here r^{ia} is the ratio between inhibitory inputs and excitatory inputs from the intralayer; $\rho_{js}^{(k),ia}$ is correlation coefficient between the jth and sth input from intralayer.

In terms of Siegert's expression [7], we have the expression of the mean and variance of the output ISIs:

$$\left\langle T_i^{(k+1)}\right\rangle = \frac{2}{L}\int_{A_i^{(k)}(V_r)}^{A_i^{(k)}(V_t)}a(x)\,dx,$$
$$\text{Var}\left(T_i^{(k+1)}\right) = \frac{4}{L^2}\int_{A_i^{(k)}(V_r)}^{A_i^{(k)}(V_t)}h(x)\,dx, \quad (8)$$

where

$$A_i^{(k)}(y) = \frac{\left(yL-\overline{\mu}_i^{(k)}\right)}{\left(\overline{\sigma}_i^{(k)}\sqrt{L}\right)},$$
$$a(x) = \exp\left(x^2\right)\int_{-\infty}^{x}\exp\left(-u^2\right)du, \quad (9)$$
$$h(x) = \exp\left(x^2\right)\int_{-\infty}^{x}\exp\left(-u^2\right)a^2(u)\,du.$$

As discussed in [2], for the correlation between the inputs and outputs, we can assume that the following *heuristic* relationship holds:

$$\rho_{ij}^{(k+1)} = \frac{\sum_{m,n}\omega_{im}^{(k)}\sigma_m^{(k)}\omega_{jn}^{(k)}\sigma_n^{(k)}\rho_{mn}^{(k)}}{\sqrt{\sum_{m,n}\omega_{im}^{(k)}\sigma_m^{(k)}\omega_{in}^{(k)}\sigma_n^{(k)}\rho_{mn}^{(k)}}\sqrt{\sum_{m,n}\omega_{jm}^{(k)}\sigma_m^{(k)}\omega_{jn}^{(k)}\sigma_n^{(k)}\rho_{mn}^{(k)}}}. \quad (10)$$

Note that the right-hand side of (10) is the correlation of the inputs to the ith and the jth neurons in the $(k+1)$th layer, which includes the intra- and interinput correlations.

The above relationships between inputs and outputs lay the foundation of the moment neural networks with intra- and interinteractions (MNNIII).

3. Dynamics of Networks

The question we intend to address here is how a spontaneous activity can be maintained in a feedforward network. For all simulations, the values of the decay rate, threshold, and resting potential were set equal to $L = 1/20\,\text{ms}^{-1}$, $V_t = 20\,\text{mV}$, and $V_r = 0\,\text{mV}$, respectively. The same parameters have been employed elsewhere [8] and are thought to be

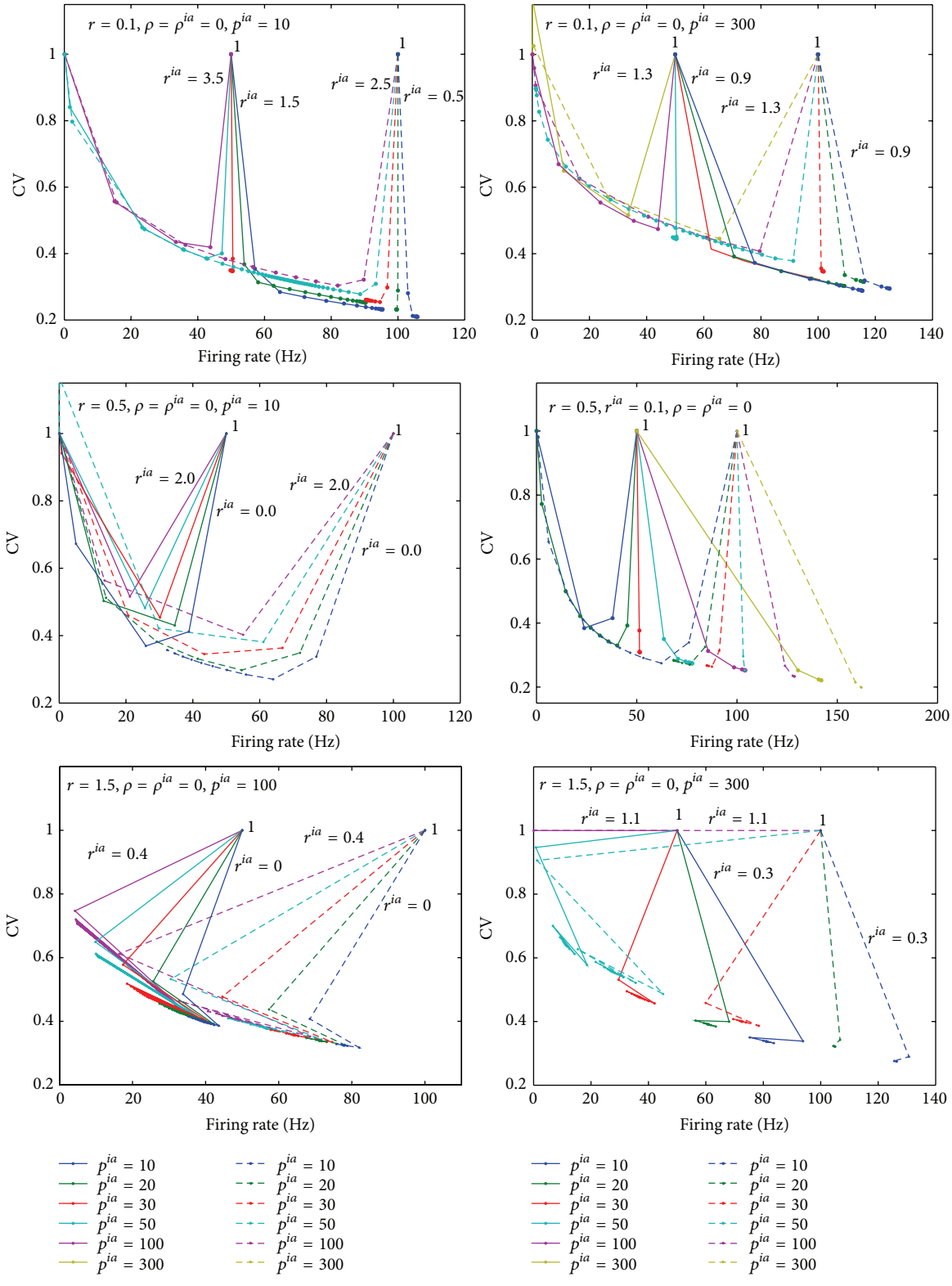


FIGURE 1: Propagation of activity in a homogeneous feedforward MNNIII. For each layer k, the output firing rate ($\mu_1^{(k)}$) and the coefficient of variation ($C_1^{(k)}$) are reported on the abscissa and the ordinate, respectively. Points labeled 1 correspond to the first layer; points corresponding to successive layers are connected by lines. Results were obtained for $p = 100$, $\rho_{ij} = \rho_{ij}^{ia} = 0$, $\mu_1^{(1)} = 50$ Hz (solid lines), 100 Hz (dashed lines), and $r = 0.1$ (top), $r = 0.5$ (middle), and $r = 1.5$ (bottom). In each panel, the ratios from intralayer (r^{ia}) are evenly spaced.

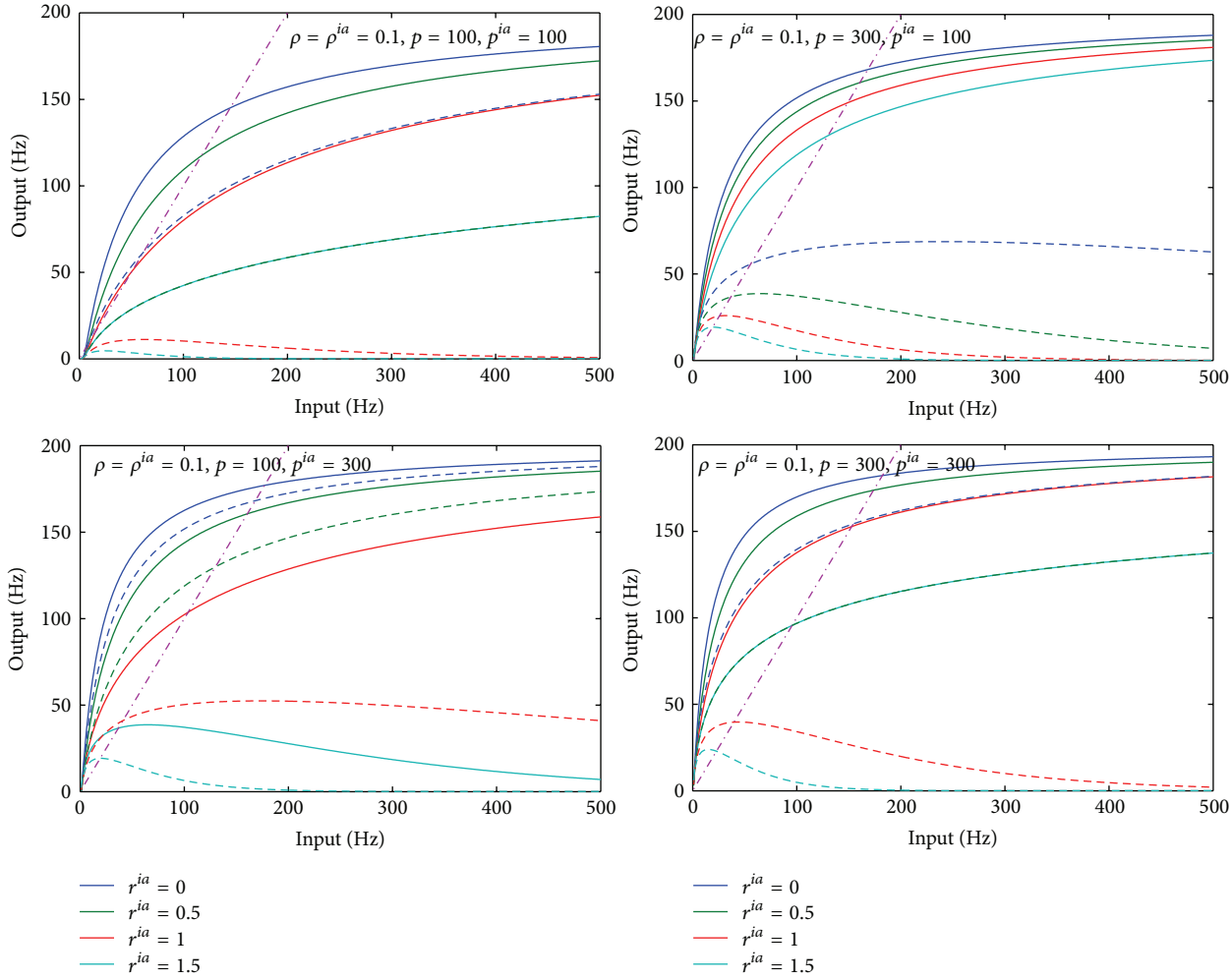


FIGURE 2: The input-output firing rate relationship for a homogeneous MNNIII varied from numbers of intralayer neurons p^{ia} and the ratio of intrainputs r^{ia}. Results were obtained for $\rho_{ij} = \rho_{ij}^{ia} = 0.1$ and $r = 0.5$ (solid lines), $r = 1.5$ (dashed lines).

in the physiological range for visual cortex cells and in agreement with most published results [9, 10]. The coefficient of variation, $\mathrm{CV} = \sqrt{\mathrm{var}(T)}/\langle T\rangle$, is used to quantify the irregularity of a spike train. If $\mathrm{CV} = 0$, the spike train is regular; otherwise the spike train is random. As analogous to [1], we stop our simulation when the firing rate is slower than 0.001 Hz. All simulations were carried out with MATLAB.

3.1. Dynamics of Homogeneous Network. In this section, we focus on a homogeneous network, where all weights, afferent means, and variances were set to be identical. Now the quantities in (6) reduce to $\mu_1^{(k)} = \mu_j^{(k)}$, $\sigma_1^{(k)} = \sigma_j^{(k)}$, $\mu_1^{(k),ia} = \mu_j^{(k),ia}$, $\sigma_1^{(k),ia} = \sigma_j^{(k),ia}$, and $\omega_{12}^{(k)} = \omega_{ij}^{(k)} = \omega^{(k)} = \omega$, $\omega_{12}^{(k),ia} = \omega_{ij}^{(k),ia} = \omega^{(k),ia} = \omega^{ia}$, $i \neq j$. As we discussed before, the propagation of correlation becomes trivial in such case since all cells become fully correlated after the first layer. To avoid this, we clamped the correlation coefficient; that is, we set $\rho_{12}^{(k)} = \rho_{ij}^{(k)} = \rho^{(k)} = \rho$, $\rho_{12}^{(k),ia} = \rho_{ij}^{(k),ia} = \rho^{(k),ia} = \rho^{ia}$, $i \neq j$. In simulations we set $\rho^{ia} = \rho = 0$ or 0.1, in agreement with experimental data reported in the literature [1, 11, 12], and we assume that each intralayer input, including numbers of neurons, afferent means, and variances, is identical; that is, $p^{(1),ia} = p^{(k),ia} \equiv p^{ia}$, $\mu_1^{(1),ia} = \mu_1^{(k),ia}$, and $\sigma_1^{(1),ia} = \sigma_1^{(k),ia}$.

First we choose $r = 0.1, 0.5, 1.5$ to illustrate how the activity is propagated across the networks, varied from r^{ia} and p^{ia}, on assumption that $C_1^{(k)} = C_1^{(k),ia} = 1$. Unless otherwise specified, the initial CV in all simulations is always 1, and the weight is 0.5. In Figure 1, we show the results obtained for various values of $\mu^{(1)}$ and $\sigma^{(1)}$ (we reported the coefficient of variation $C^{(k)} = \sigma^{(k)}/\mu^{(k)}$). As stated in [1], each data point $(\mu^{(k)}, C^{(k)})$ is connected with $(\mu^{(k+1)}, C^{(k+1)})$ to illustrate how the activity is propagated across the networks. After the first few layers, for $r = 0.1$ (top), neurons are found to be either silent or firing at relatively high frequency for all r^{ia} and p^{ia}. For $r = 0.5$ (middle), however, the situation has changed. On the one hand, the left panel (middle) shows that even if there is no inhibitory input ($r^{ia} = 0$) for a small number of intraneurons ($p^{ia} = 10$), to be silent is certainty on fewer intrainput neurons. On the other hand, the right panel (middle) shows that not all networks are certainty to

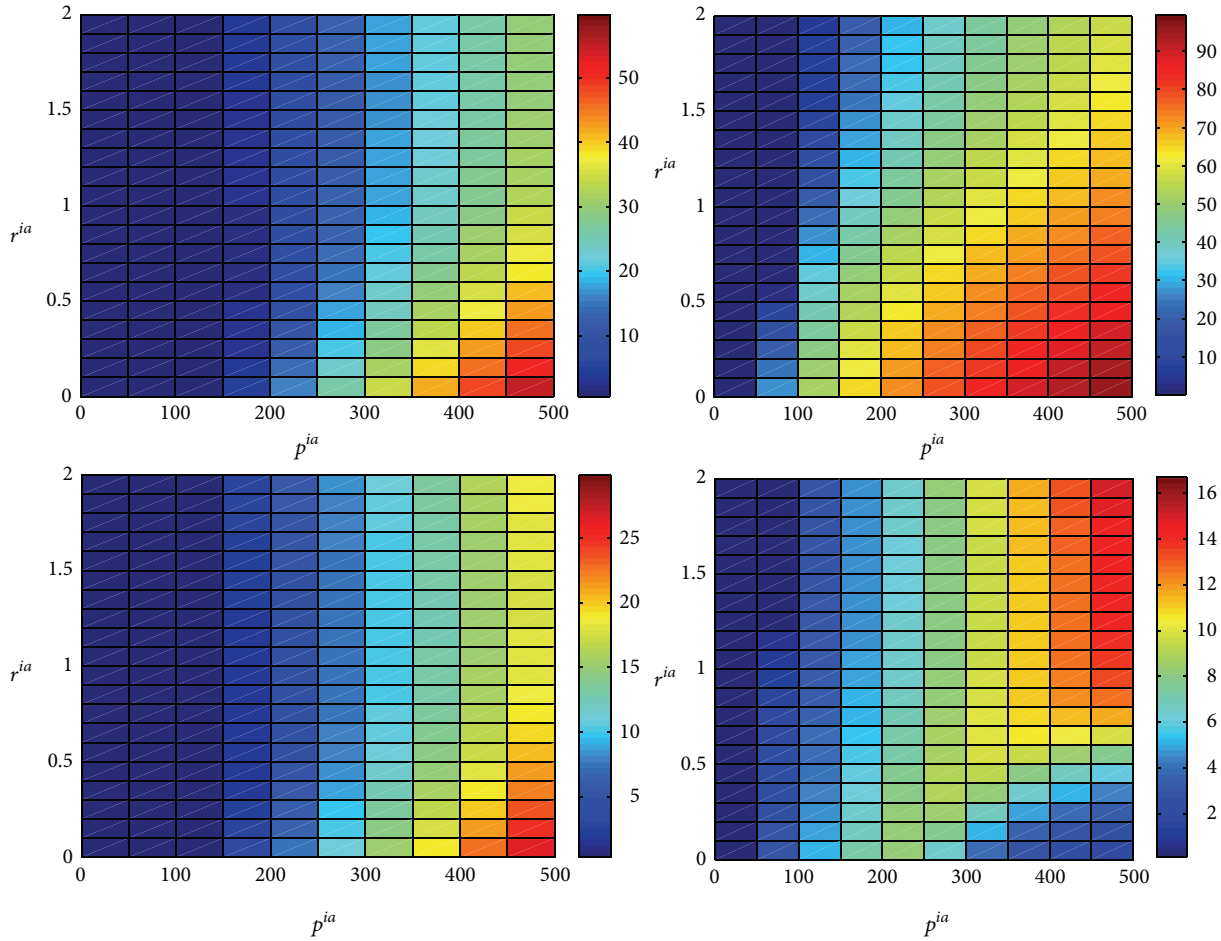


FIGURE 3: Stable outputs for a homogeneous MNNIII varied from numbers of intralayer neurons p^{ia} and the ratio of intralayer r^{ia}. Results were obtained for $r = 0.8$ (top) and 1.5 (bottom) and $p = 100$ (left) and 300 (right), with $\rho_{ij} = \rho_{ij}^{ia} = 0.1$.

be silent. As stated earlier, for $r = 0.5$ with fewer intrainput neurons ($p^{ia} = 10$), the network is certainty to be silent (left). However, with increasing size of intrainput units, the network might be fixing or firing at relatively high frequency (right). Instead, they fire at low or relatively high frequency; for example, $p^{ia} \geq 30$ for $r_1^{ia} = 0.1$. It actually shows that the networks can be stable at low frequency; see discussions later. In addition, for $r = 1.5$ (bottom), there is another kind of properties that neurons fire at relatively low frequency after a few times oscillations. Unlike a homogeneous MNN [1], neurons are certainty to be silent with a larger ratio (e.g., $r \geq 0.4$) after the first few layers. It is obvious that the intrainputs cause a more sophisticated dynamics of network than MNN.

Figure 1 implies that the stable solution existed in some networks. Thus we explore the stable solution of such network. Figure 2 shows how the input-output firing rate relationship is varied from p^{ia} and r^{ia}, for $r = 0.5$ and $r = 1.5$. As analogous to [1], roughly speaking, the stable solution appears at low firing rate for networks of increasing size (either p or p^{ia} size) with strong inhibition (either r or r^{ia}). However, the preferred appearance of stable solution at low frequency is not the increasing size of both inter- and intralayer; for $r = 1.5$ (top right, dashed lines), for example, any ratio r^{ia} from intralayer (from 0 to 1.5) appears as a fixed point at low firing rate when $p = 300$, $p^{ia} = 100$ (top right), instead of $p = 300$, $p^{ia} = 300$ (bottom right).

We further explore the stability of such network by showing that the stable outputs varied from (p^{ia}, r^{ia}). Figure 3 shows the stable outputs versus r^{ia} and p^{ia}, with a fixed ratio from interlayer inputs $r = 0.8$ (top) and $r = 1.5$ (bottom) and with the number of neurons from interlayer inputs $p = 100$ (left) and $p = 300$ (right). The initial inputs fix 5 Hz in the current and latter simulations. It is found that the intralayer input unit appears to have a sophisticated role in the stability of such network, depending on the ratio r^{ia} and the size p^{ia} of context unit, together with the ratio r and the size p of network itself, that is, interlayer. For example, for $r = 0.8$ (top), the stable output appears to increase with the increasing size p^{ia} and the decreasing ratio r^{ia} of context unit, which plays a general role; for $r = 1.5$ and $p = 300$ (bottom right panel), however, the situation has changed to play a negative role; see $p^{ia} \geqslant 300$.

We also explore the stability of such network by showing that the stable outputs varied from numbers of intralayer neurons p^{ia} and of interlayer p. Figure 4 shows that the network with a small size of intra- and interlayer will be

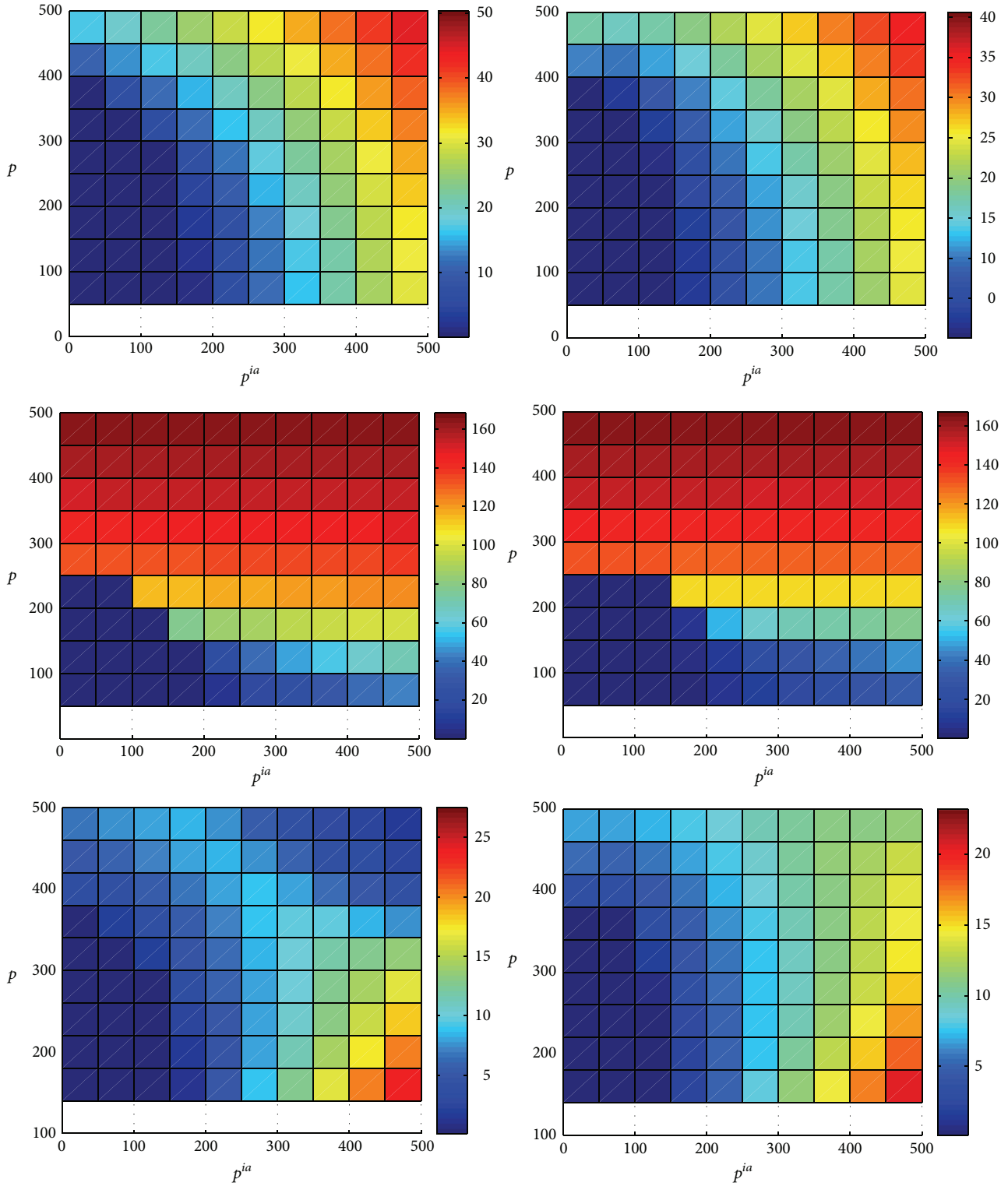


FIGURE 4: Stable outputs for a homogeneous MNNIII varied from numbers of intralayer neurons p^{ia} and of interlayer p. Results were obtained for $r = 1$ (top), $r = 0.5$ (middle) and 1.5 (bottom) and $r^{ia} = 0.5$ (left) and $r^{ia} = 1.5$ (right), with $\rho_{ij} = \rho_{ij}^{ia} = 0.1$.

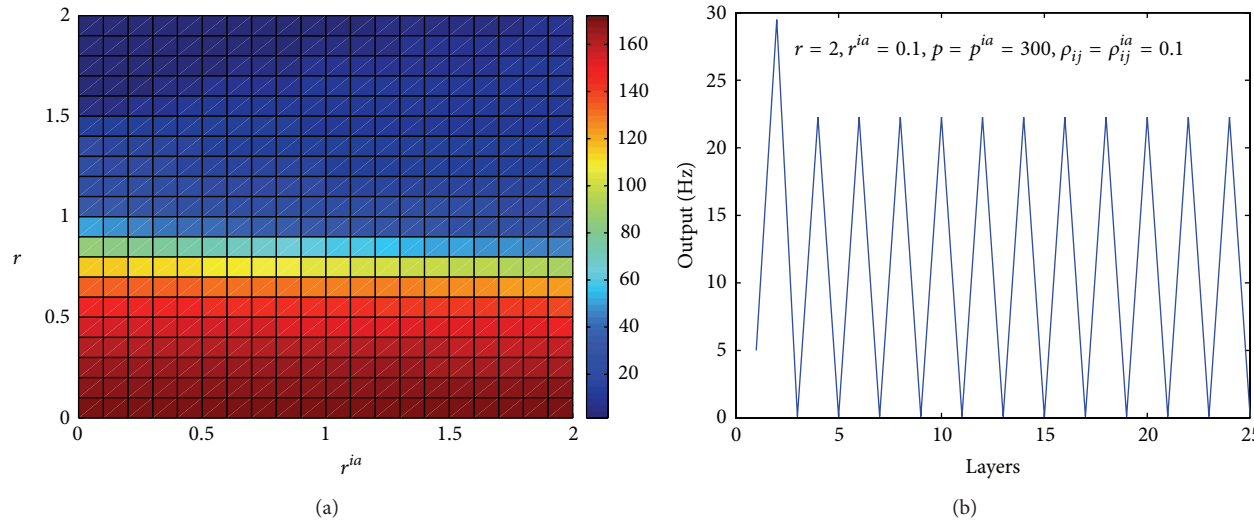


FIGURE 5: (a) Stable outputs for a homogeneous MNNIII varied from the ratio of both inter- and intralayer (r and r^{ia}). Results were obtained for $p = p^{ia} = 300$, with $\rho_{ij} = \rho_{ij}^{ia} = 0.1$. (b) The appearance of oscillation for a homogeneous MNNIII.

either silent or firing at low frequency. Top panel shows that the stable output of such network seems to be symmetry between numbers of intralayers and interlayers when $r = 1$. However, there are obvious asymmetry for $r < 1$ or $r > 1$; see middle and bottom panels in Figure 4. Middle panel shows that the network with a relatively large size of interlayer, for $r < 1$, will be firing at relatively high rate and that the size of intralayer has negligible effect on the stable outputs such as $p > 250$. Bottom panel presents unexpected results of $r > 1$ that relatively high frequency is firing at the large size of intralayer and small size interlayer, rather than the large size of intralayer and interlayer. In particular, for $p > 300$ and $p^{ia} > 300$, the network is unexpectedly firing at low frequency when $r = 1.5$ and $r^{ia} = 0.5$; see the left of bottom panel. All those imply that the current network can lead to more sophisticated dynamics than MNN and the intrainput unit plays a very sophisticated role in dynamics of such network.

Finally, we explore the stability of such network by showing that the stable outputs varied from the ratio of both inter- and intralayers, that is, r and r^{ia}. Figure 5(a) shows the stable outputs versus r and r^{ia} with $p = p^{ia} = 300$. It can be seen that the network will be either silent or firing at low frequency when $r > 1$ and will be firing at relatively high rate when $r < 1$, which implies that the ratio (between inhibitory and excitatory input) from the network itself plays a more important role than that from intrainput unit with such parameters. However, the network will converge to silent or to low firing frequency with strong inhibition $r > 1$ or to high frequency with weak inhibition $r < 1$. Unlike the stable output of MNN in [1] (a high level of inhibition is necessary to maintain a spontaneous activity, that is, a low output firing rate), a weaker inhibition ($r < 1$ approaching 1), even with weaker intrainhibition ($r^{ia} < 1$ approaching 1), can converge to low firing frequency; see, for example, $r \in (0.8, 1)$ and $r^{ia} \in (0.5, 1)$. It is the intrainput unit to maintain the low frequency activity. Thus there is an important gain in computational power over MNN. Note the appearance of oscillation for a homogeneous MNNIII as singularity, for example, with $r = 2$, $r^{ia} = 0.1$, $p = p^{ia} = 300$, and $\rho_{ij} = \rho_{ij}^{ia} = 0.1$ (Figure 5(b)).

3.2. Dynamics of Heterogeneous Network. We now turn to explore a more sophisticated dynamics of a heterogeneous network. We shed light on, for instance, how synchronization or desynchronization is produced by random or lateral connections when we remove the constraint on the correlations between neurons.

In accordance with the results reported in the previous publications, we fixed the total number of neurons in each interlayer to $p^{(k)} = p = 324$. To show the influences of intralayer inputs on networks, we also fixed the total number of neurons in each intralayer to $p^{(k),ia} = p^{ia} = 324$. We further assumed that each intralayer receives the same inputs as the first intralayer; that is, $w_{ij}^{(k),ia} = w_{ij}^{(1),ia}$, $\mu_i^{(k),ia} = \mu_i^{(1),ia}$, $C_i^{(k),ia} = C_i^{(1),ia}$, and $\rho_{ij}^{(k),ia} = \rho_{ij}^{(1),ia}$ for $i, j = 1, \ldots, 324$.

First we generated random connections $w_{ij}^{(k)}, w_{ij}^{(1),ia} \in [0, 0.5]$ and inputs $\mu_i^{(1)}, \mu_i^{(1),ia} \in [0, 100]$ Hz. We also assumed that $C_1^{(1)} = C_1^{(1),ia} = 1$ and that $\rho_{ij}^{(1)}, \rho_{ij}^{(1),ia} \in [0, 0.1]$ for $i, j = 1, \ldots, 324$.

Figure 6 shows the output CV versus the mean firing rate in the first 10 layers of the network itself (left) and the correlation coefficients for one cell in the first, second, and fourth (or fifth) layer (right) for $r = 1.2$, $r^{ia} = 1.5$; $r = 0.5$, $r^{ia} = 0.5$; and $r = 1.5$, $r^{ia} = 0.9$. Simulations show that all neurons synchronize after the 5th layer, that is, $\rho_{ij}^{(k)} = 1$ for $k \geq 5$, and that the network is stable. Top panels show that the network will be stable at relatively low frequency (<100 Hz) after the 4th layer; middle panels show that the network will be stable at relatively high rate (>100 Hz) after the 5th layer; however, the last panels show that the network can be stable at a very low frequency (<1 Hz). For example,

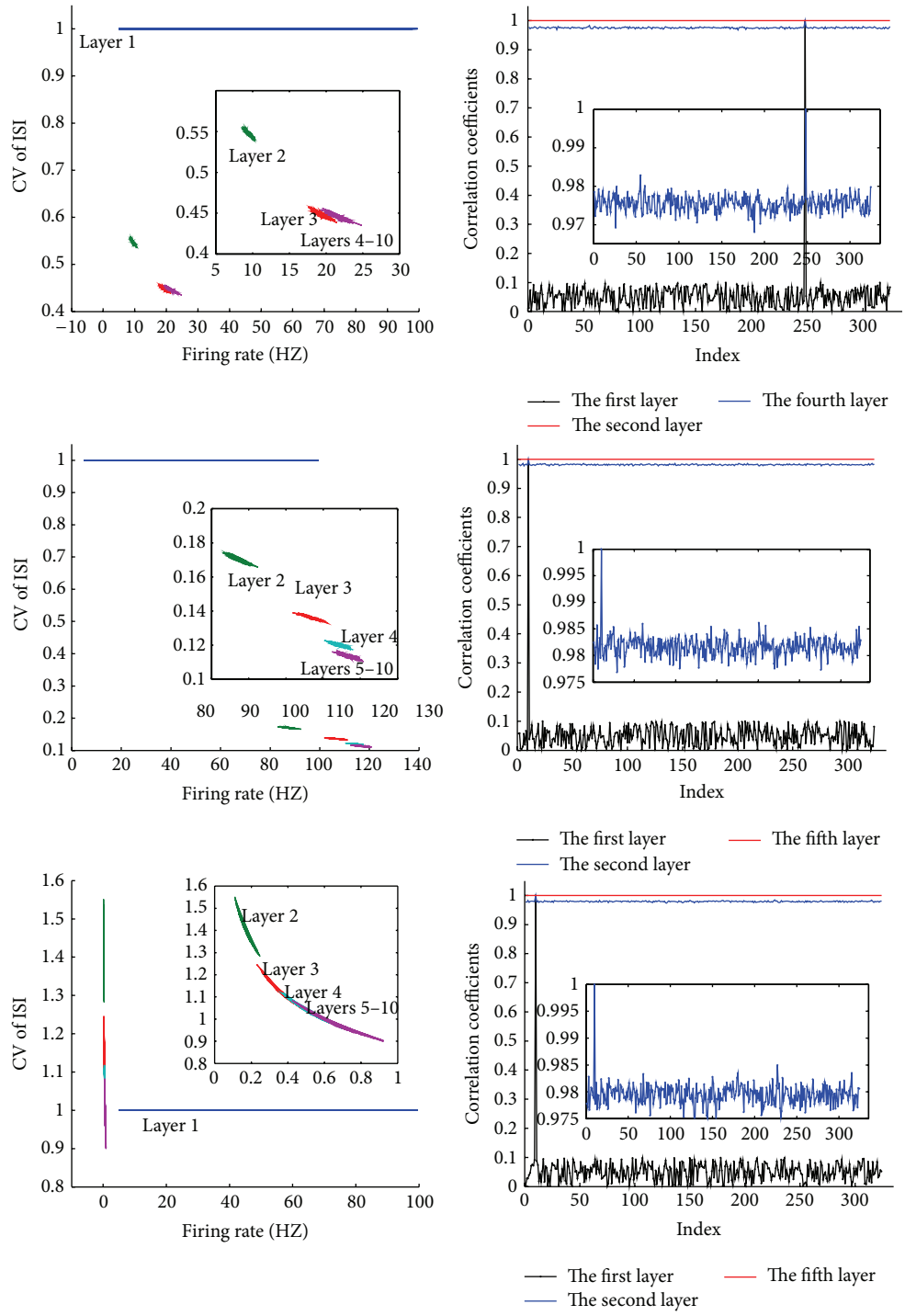


Figure 6: Propagation of activity in a heterogeneous MNNIII with random connections. The output CV versus the mean firing rate in the first 10 layers (left). The correlation coefficients for one cell (right). Results were obtained for $r = 1.2$, $r^{ia} = 1.5$; $r = 0.5$, $r^{ia} = 0.5$, and $r = 1.5$, $r^{ia} = 0.9$.

Figure 7 shows the mean firing rate for different layers, corresponding to Figure 6 (top). Note the different scale for each layer. It can be seen from Figures 6 and 10 (top) that the ratios r and r^{ia} change the stable outputs but do not change the synchronization. This result, in general, is in agreement with numerical experiments of feedforward spiking neuronal networks, showing that neurons get synchronized quite easily [13]. In fact, desynchronization rather than synchronization seems to be the major problem for a spiking neuronal network.

In order to avoid synchronization, we used the same Mexican hat weight distribution as in [1]. To this end, we

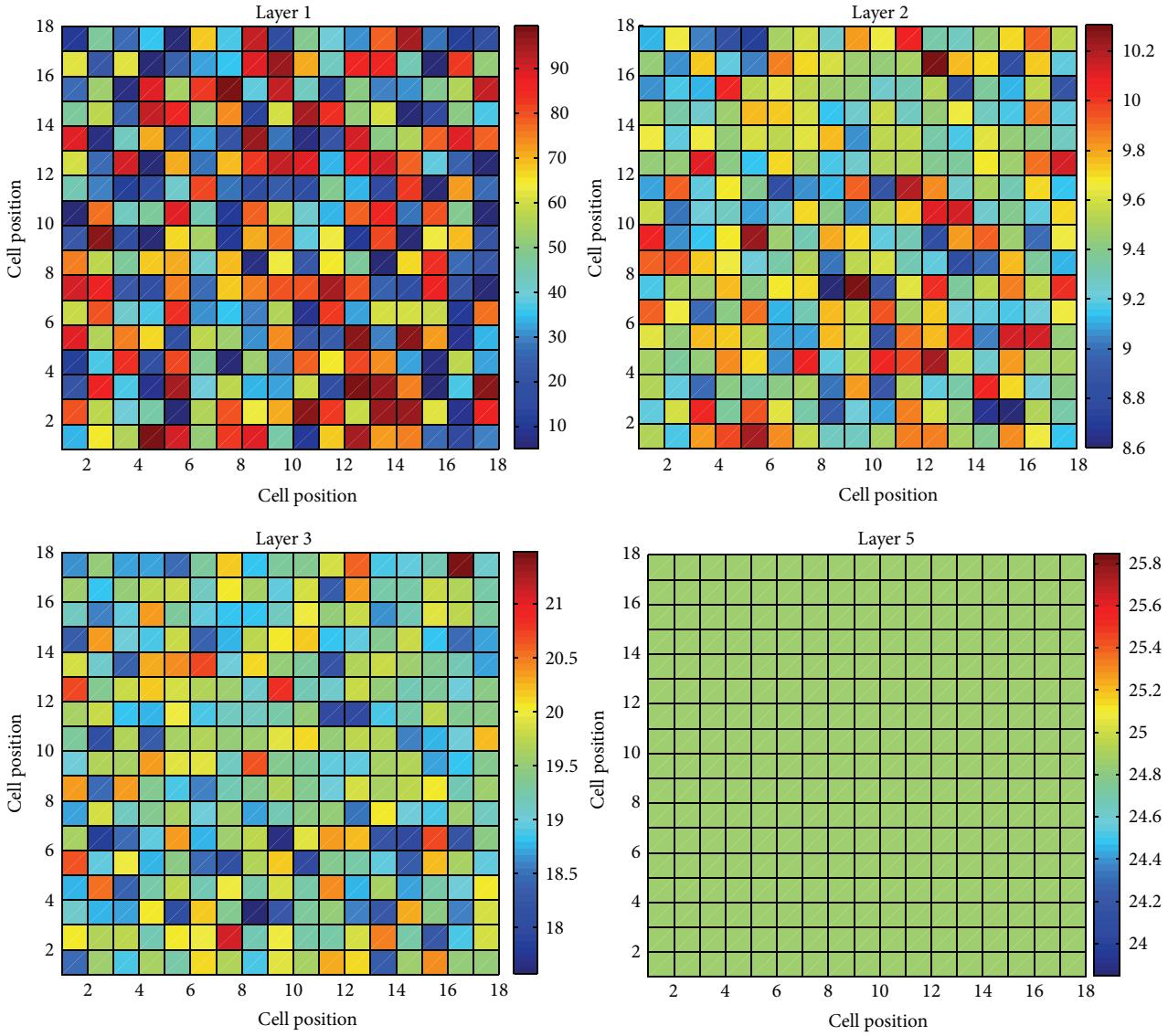


FIGURE 7: The mean firing rate for different layers, corresponding to top panel in Figure 6.

first rearranged all neurons $\{1, 2, \ldots, p\}$ from the interlayer on a two-dimensional square lattice by assigning to neuron $i \in \{1, 2, \ldots, p\}$ coordinates as

$$x(i) = \left(\frac{i-1}{n}\right),$$
$$y(i) = (i-1) - nx(i) \equiv \mathrm{mod}(i-1, n), \tag{11}$$

with $p^{(k)} = p = n^2$. Then for $i, j = 1, 2, \ldots, p$, we set

$$\omega_{ij}^{(k)} = \omega_{ij} = M\left(0.1\left[x(i) - x(j)\right], 0.1\left[y(i) - y(j)\right]\right), \tag{12}$$

where $M(x, y)$ is the Mexico hat function, defined for $x, y \in R$ as

$$M(x, y) = \frac{m}{\sqrt{2\pi}\gamma}\left[\exp\left(-\frac{x^2+y^2}{2\gamma^2}\right) - \frac{1}{2}\exp\left(-\frac{x^2+y^2}{8\gamma^2}\right)\right], \tag{13}$$

with $m > 0$, $\gamma > 0$ modulation parameters. For simulations we used the same modulation $m = 3$, $\gamma = 0.5$ as [1] and the same other parameters as in the previous subsection.

Figure 8 shows the propagation of activity in a heterogeneous MNNIII with Mexico hat type connections for $r = 1.2$, $r^{ia} = 1.5$. The results indicate that the activity in the network becomes stable after the 6th layer (top left). Also, as indicated by the values of the correlation coefficients in the top right panel, the neurons do not synchronize. Middle

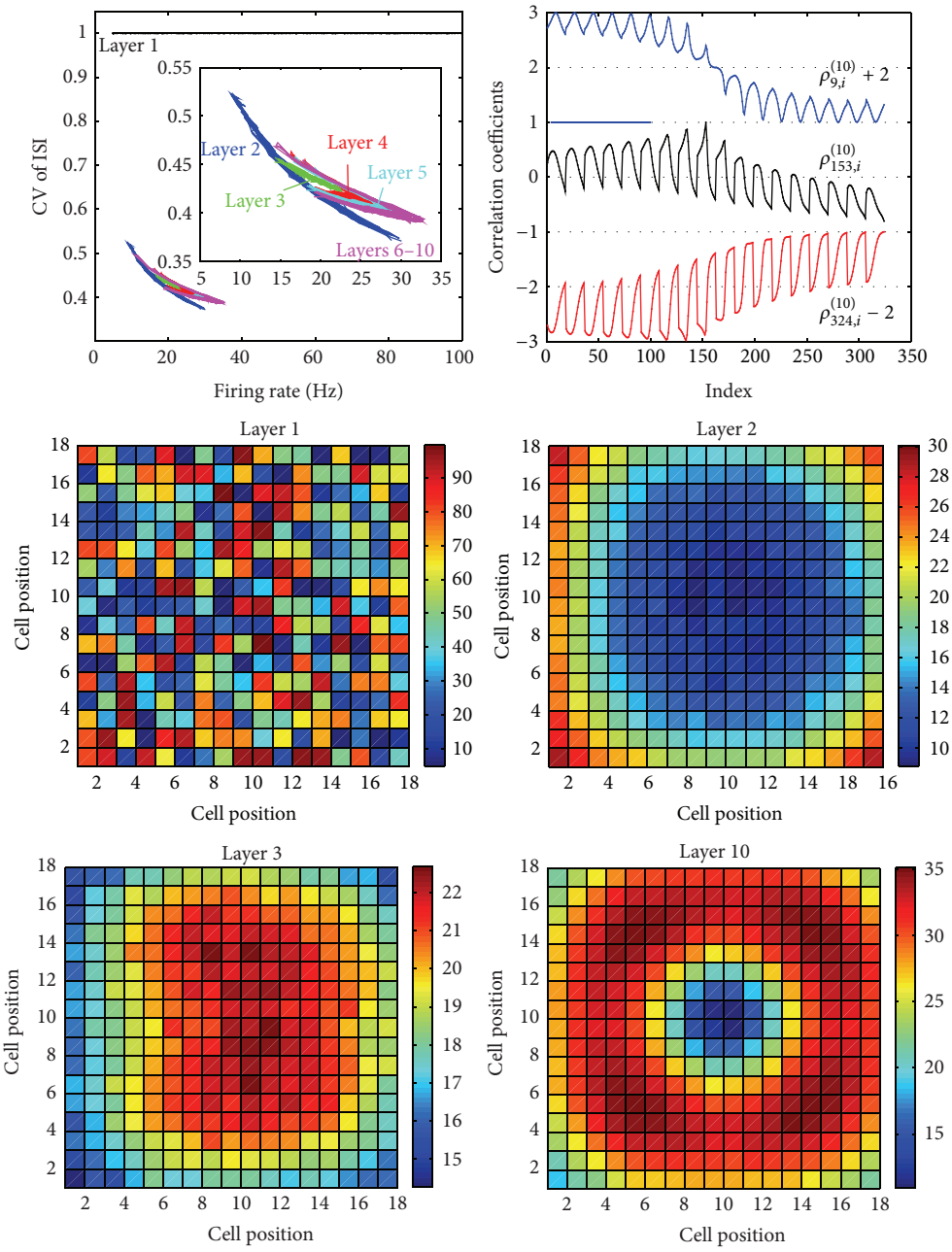


FIGURE 8: Propagation of activity in a heterogeneous MNNIII with Mexico hat type connections. The output CV versus the mean firing rate in the first 10 layers (top left). The correlation coefficients for three cells in layer 10 (top right). The mean firing rate for different layers (middle and bottom). Results were obtained for $r = 1.2$, $r^{ia} = 1.5$.

and bottom panels show the mean firing rate for different layers. Figure 9 presents the correlation coefficients between the central cell and the other cells in layer 3 and in layer 10, corresponding to Figure 8; on average the correlation coefficient between neurons was around zero.

It is interesting to note that, from the first to the second layer, there is a general reduction in firing rate. Then, after a few transition layers, the neuronal activity becomes stable. In summary, the activity in a heterogeneous MNNIII becomes stationary after a few layers. With random interactions, the network gets easily synchronized. However, with lateral interactions by Mexico hat function, desynchronization is presented; that is, firing rates of individual neurons tend to spread out.

To compare the results with others, Figure 10 shows the propagation of activity in a heterogeneous MNNIII with random connections (top) and with Mexico hat type connections (bottom) for $r = 1.5$, $r^{ia} = 1.5$. It only shows the output CV versus the mean firing rate in the first 10 layers (left) and the stable mean firing rate in layer 10 (right),

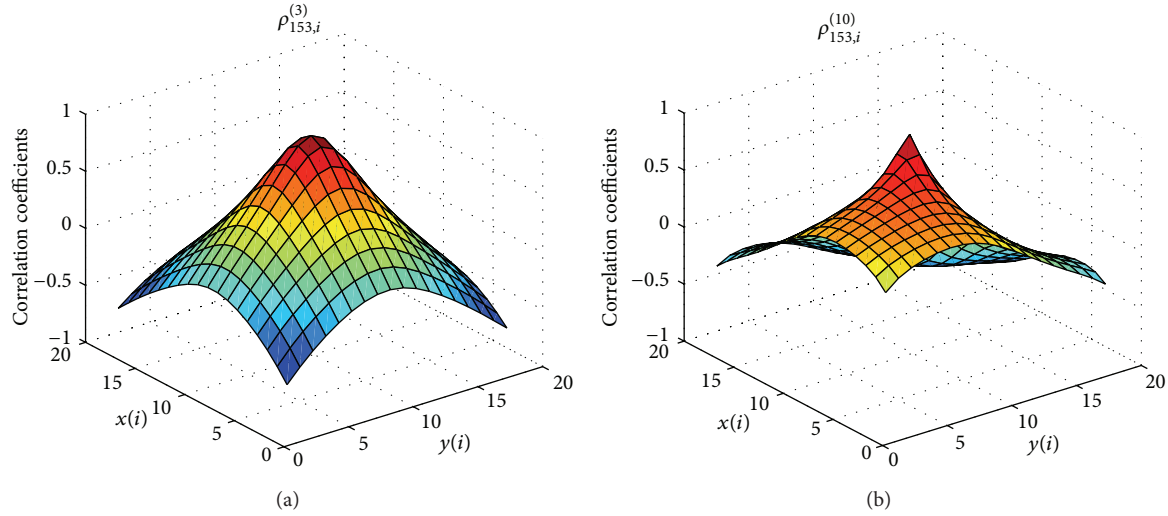


Figure 9: The correlation coefficients between the central cell and the other cells in layer 3 (left) and in layer 10 (right), corresponding to Figure 8.

respectively. Figure 10 implies that the MNNIII with random connections gets synchronized after the 6th layer, and with Mexico hat type connections it becomes stationary and fires with the bell-like firing rates after the 7th layer.

First, the weight connections are key to propagate the activity of such network. A homogeneous network with clamped weight connections, as stated in Section 3.1, converges to silent or to low/high firing frequency. However, a heterogeneous network gets synchronized or desynchronized due to random or lateral weight connections. It is the lateral weight connections introduced by the Mexican hat that push cells to fire with more widely spread firing rates than in a network with random interactions; see the former discussions in this subsection. In other words, weight connections determine the type of dynamics of network (say, convergence, synchronization, and desynchronization) and subsequently change the size of stable outputs. For $r = 1.5$, $r^{ia} = 1.5$, for example, a homogeneous network with initial input 50 Hz converges to 16.09 Hz after the 5th layer (approximate to $r = 1.5$, $r^{ia} = 1.5$ in the left panel of Figure 5); Figure 10 shows that a network with random connections gets synchronized at 9.36 Hz after the 6th layer (top) and with Mexico hat type connections it gets desynchronized at the stable bell-like outputs with a mean value of 19.56 Hz after the 7th layer (bottom). It is the weight connections that change the stable outputs of network. Additionally, the network simulations reported in this subsection show that a highly irregular output firing is also the result of correlations between neuronal activities.

Second, all simulations show that the ratio of inhibition from inter- or intralayer (r or r^{ia}) does not determine the type of dynamics but influences the stable outputs of network. For homogeneous network, it was verified in Section 3.1. For a network with random connections, Figures 6 and 10 (top) show that the ratio r or r^{ia} changes the stable outputs but does not change the synchronization. Figures 8 and 10 (bottom) show that the lateral weight connections introduced by the Mexican hat push cells to fire with more widely spread firing rates than in a network with random interactions and that the ratio r only changes the stable outputs; for example, the mean of stable bell-like outputs in Figure 8 (18.05 Hz) is bigger than that in bottom panel of Figure 10 (11.45 Hz). It is the inhibition from network itself or from external intrainput that can enhance or weaken the (stable) firing outputs of network.

Finally, all parameters influence the speed of the stability of network. For homogeneous network, it was included in discussions in Section 3.1. For heterogeneous network, the weight connections can influence it, when other parameters are identical; see, for example, Figure 8, which shows that the speed of synchronization by random connections is slightly faster than that of desynchronization by lateral connections. Further the ratio of inhibition influences it; see, for example, Figures 6 and 10 (top), which show its validity for random connections, and Figures 8 and 10 (bottom) do so for the Mexican hat type connections. In addition, the heterogeneous MNNIII simulations reported in this subsection show that synchronization/desynchronization seems to be slightly slower or faster than MNN [1] with other identical parameters, since the external intralayer input unit can enhance or weaken the firing outputs. Thus, the network reported here offers a more sophisticated and unexpected dynamics over the MNN.

4. Discussion

In the current paper, we have focused on the dynamics of feedforward MNNIII, akin to the dynamics of MNN. Synchronization or desynchronization is presented by random or lateral interactions. Due to more biologically reasonable intralayer inputs, such network offers a more sophisticated and unexpected dynamics over the MNN. As stated earlier, the intrainput unit plays a very sophisticated role in dynamics of such network. It can enhance or weaken the firing outputs

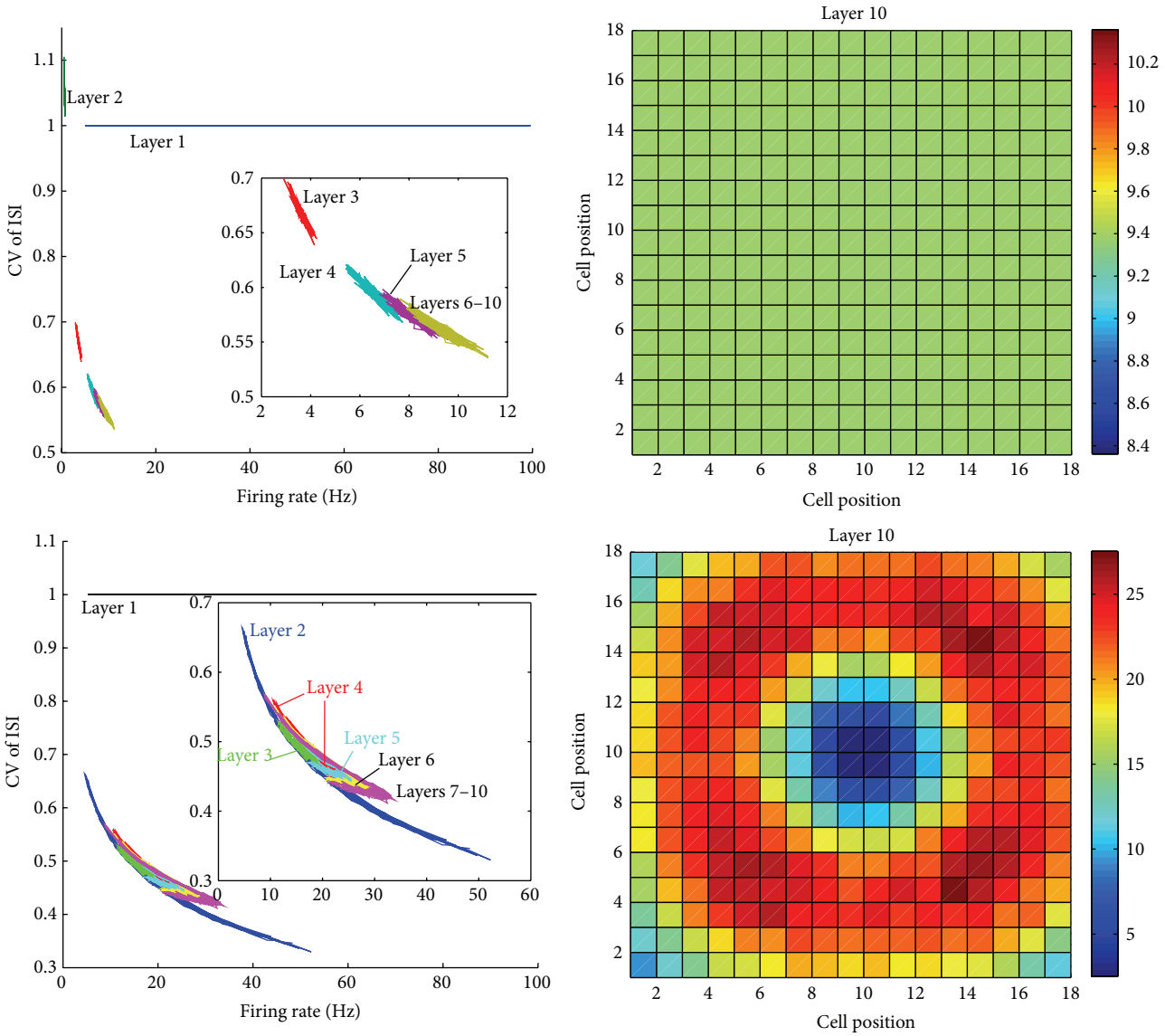


FIGURE 10: Propagation of activity in a heterogeneous MNNIII with random connections (top) and Mexico hat type connections (bottom). The output CV versus the mean firing rate in the first 10 layers (left). The stable mean firing rate in layer 10 (right). Results were obtained for $r = 1.5$, $r^{ia} = 1.5$.

by varying on the ratio r^{ia} and/or the size p^{ia} of intralayer, incorporating with those of interlayer, that is, r and/or p. In the future we might shed light on the application in engineering, the dynamics, and/or learning rule of MNNIII with other network architectures such as recurrent and RBF network [14].

Conflict of Interests

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

Acknowledgments

The authors would like to thank the anonymous reviewers for their useful comments and suggestions. This work was supported by the National Natural Science Foundation of China (11171101 and 61403136), Provincial Natural Science Foundation of Hunan (13JJ3114), Scientific Research Fund of Hunan University of Arts and Science (13ZD02), and the Construction Program of the Key Discipline in Hunan University of Arts and Science—Applied Mathematics.

References

[1] J. Feng, Y. Deng, and E. Rossoni, "Dynamics of moment neuronal networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 73, no. 4, Article ID 041906, 17 pages, 2006.

[2] X. Y. Xiang and Y. C. Deng, " Training moment nenronal networks with intra- and interlayer interactions," *Information: An International Interdisciplinary Journal*, vol. 15, pp. 363–374, 2012.

[3] T. P. Trappenberg, *Fundamentals of Computational Neuroscience*, Oxford University Press, Oxford, UK, 2002.

[4] X. Y. Xiang, Y. C. Deng, and X. Q. Yang, "Second order spiking perceptrons," *Soft Computing*, vol. 13, pp. 1219–1230, 2009.

[5] X. Y. Xiang and Y. C. Deng, "The learning of moment neuronal networks," *Nerucomputing*, vol. 73, pp. 2597–2613, 2010.

[6] X. Y. Xiang, Y. Chen, and L. F. Liu, "An updated learning rule for moment neuronal networks," *Journal of Information and Computational Science*, vol. 8, no. 13, pp. 2509–2516, 2011.

[7] J. F. Feng, *Computational Neuroscience-A Comprehensive Approach*, Chapman & Hall/CRC Press, London, UK, 2003.

[8] W. Gerstner and W. Kistler, *Spiking Neuron Models*, Cambridge University Press, London, UK, 2003.

[9] V. Livak, H. Sompolinsky, I. Segev, and M. Abeles, "On the transmission of rate code in long feedforward networks with excitatory-inhibitory balance," *The Journal of Neuroscience*, vol. 23, pp. 3006–3015, 2003.

[10] M. N. Shadlen and W. T. Newsome, "Noise, neural codes and cortical organization," *Current Opinion in Neurobiology*, vol. 4, pp. 569–579, 1994.

[11] J. F. Feng and D. Brown, "Impact of correlated inputs on the output of the integrate-andfire models," *Neural Computation*, vol. 12, pp. 671–692, 2000.

[12] E. Zohary, M. N. Shadlen, and W. T. Newsome, "Correlated neuronal discharge rate and its implications for psychophysical performance," *Nature*, vol. 370, pp. 140–143, 1994.

[13] H. C. Tuckwell, *Introduction to Theoretical Neurobiology*, Cambridge University Press, London, UK, 1988.

[14] I. Sadeghkhani, A. Ketabi, and R. Feuillet, "Radial basis function neural network application to measurement and control of shunt reactor overvoltages based on analytical rules," *Mathematical Problems in Engineering*, vol. 2012, Article ID 647305, 14 pages, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 210680, 12 pages
http://dx.doi.org/10.1155/2015/210680

Research Article

Hybrid Genetic Algorithm with Multiparents Crossover for Job Shop Scheduling Problems

Noor Hasnah Moin, Ong Chung Sin, and Mohd Omar

Institute of Mathematical Sciences, Faculty of Science, University of Malaya, 50603 Kuala Lumpur, Malaysia

Correspondence should be addressed to Noor Hasnah Moin; noor_hasnah@um.edu.my

Received 13 June 2014; Revised 8 September 2014; Accepted 8 September 2014

Academic Editor: Shifei Ding

The job shop scheduling problem (JSSP) is one of the well-known hard combinatorial scheduling problems. This paper proposes a hybrid genetic algorithm with multiparents crossover for JSSP. The multiparents crossover operator known as extended precedence preservative crossover (EPPX) is able to recombine more than two parents to generate a single new offspring distinguished from common crossover operators that recombine only two parents. This algorithm also embeds a schedule generation procedure to generate full-active schedule that satisfies precedence constraints in order to reduce the search space. Once a schedule is obtained, a neighborhood search is applied to exploit the search space for better solutions and to enhance the GA. This hybrid genetic algorithm is simulated on a set of benchmarks from the literatures and the results are compared with other approaches to ensure the sustainability of this algorithm in solving JSSP. The results suggest that the implementation of multiparents crossover produces competitive results.

1. Introduction

The job shop scheduling problem (JSSP) is one of the well-known hard combinatorial scheduling problems and it has been studied extensively due its practical importance. During the past three decades, optimization strategies for JSSP ranging from exact algorithms (mathematical programming) to approximation algorithms (metaheuristics) have been proposed [1]. Exact methods which are based on exhaustive enumeration and decomposition method guarantee global convergence and have been successfully applied to small instances. But, for the moderate and large scale instances, they require very high computational time to be either ineffective or inefficient [2]. Therefore, a lot of researchers focused their attention on approximation methods. Metaheuristics is one of the approximation methods that were proposed in the literatures to deal with JSSP which include tabu search (TS) [3], simulated annealing (SA) [4], genetic algorithm (GA) [5], and discrete artificial bee colony (DABC) [6].

In recent years, since the first use of GA based algorithm to solve the JSSP proposed by Davis [7], various GA strategies are introduced to increase the efficiency of GA to find the optimal or near optimal solutions for JSSP [8]. In the GA strategies, hybridization of GA with local search methods provides good results in solving the problems where GA capitalizes on the strength of the local search in locating the optimal or near optimal solutions. For example, Gonçalves et al. [9] and Zhang et al. [10] embedded the local search procedure of Nowicki and Smutnicki [11] into GA due to the effectiveness of the local search that increases the performance of GA. Qing-Dao-Er-Ji and Wang [12] proposed new crossover operator and mutation operator together with local search in improving local search ability of GA.

Additionally, the structure of the GA can be modified and enhanced to reduce the problems often encountered in GA. Yusof et al. [13] implemented a migration operator in GA by using parallelization of GA (PGA) to find the near optimal solutions. Watanabe et al. [14] proposed a GA with search area adaption and a modified crossover operator for adapting to the structure of the solutions space. Ripon et al. [15] embedded heuristic method in the crossover to reduce the tail redundancy of chromosome. Particularly,

recombination operators, especially crossover operators, play important roles in the structure of GA.

Crossover between two parents is traditionally adopted in GA [8] for JSSP but the GA can be modified accordingly to suit the problem at hand including selecting several numbers of parents for the crossover operation which is known as multiparents crossover.

The application of multiparents recombination can be found in different research areas. Mühlenbein and Voigt [16] proposed gene pool recombination (GPR) in solving discrete domain problems. Eiben and van Kemenade [17] introduced the diagonal crossover as the generalization of uniform crossover in GA for numerical optimization problems. Wu et al. [18] proposed multiparents orthogonal recombination to determine the identity of an unknown image contour. The crossover operators that were used in those areas show the good search ability of the operator but are very much problem dependent.

The above literatures indicated the ascendency of multiparents crossover over two parents' crossover. Although multiparents crossover has been used in different fields, to the best of our knowledge, only limited numbers are applied to combinatorial scheduling problems and none has been proposed for JSSP. In particular, Eiben et al. [19] proposed multiparent for the adjacency based crossover (ABC) and Ting et al. [20] developed multiparent extension of partially mapped crossover (MPPMX) for the travelling salesman problems. Although the experimental results point out that ABC of multiparents has no tangible benefit, MPPMX shows significant improvement in the use of multiparents in crossover. In other words, one would expect that, by biasing the recombination operator, the performance of the GA would improve.

In this paper, we propose extended precedence preservative crossover (EPPX) as a multiparent crossover. EPPX is based on the precedence preservative crossover (PPX) proposed by Bierwirth et al. [21]. Because of its ability to preserve the phenotypical properties of the schedules. EPPX as crossover operator will retain this advantage in the GA. EPPX is used in GA in conjunction with neighborhood search to solve JSSP. The rest of the paper is organized as follows. JSSP and the different types of schedules are described in detail in the next section. In Section 3, we present our approach to solve the JSSP: chromosome representation, schedule generation procedure, neighborhood search procedure, and GA with multiparents crossover. Section 4 provides experimental results and analysis. The conclusions are drawn in Section 5.

2. Problem Definition

2.1. Job Shop Scheduling Problem (JSSP). The JSSP can be defined as a set of n jobs that need to be processed on a set of m machines. A job consists of a set of operations J, where O_{ij} represents the jth ($1 \le j \le J$) operation of the ith ($1 \le i \le n$) job. The technological requirement for each operation processing time is denoted as p_{ij} and a set of machines is denoted by M_k ($1 \le k \le m$).

Table 1: Example for 3-job and 3-machine problem.

	Job	Operation routing		
		1	2	3
Processing time	1	3	3	2
	2	1	5	3
	3	3	2	3
Machine sequence	1	M1	M2	M3
	2	M1	M3	M2
	3	M2	M1	M3

Precedence constraint of the JSSP is defined as [22] operation jth must finish before operation jth + 1 in the job. A job can visit a machine once only. Only one operation at a time for one time is allowed to be processed in a machine. It is assumed that the delay time for the job transfer machine will be neglected and operation allocation for machine will be predefined. Preemption of operations is not allowed. There are no precedence constraints among the operations of different jobs.

The main objective of JSSP is to find the minimum makespan for the scheduling. The finish time of job i with last operation, J, is represented by F_{iJ}. The time for the whole schedule to complete or the makespan is also the maximum finish time of a set of the jobs i. Therefore, the makespan is expressed as follows:

$$\text{Makespan} = \max\left(F_{iJ}\right). \tag{1}$$

Let $G(k)$ be the set of operations being processed in machine k and let

$$X_{O_{ij},k} = \begin{cases} 1 & \text{if } O_{ij} \text{ has been assigned to machine } k \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The conceptual model of the JSSP is shown as below:

$$\text{Minimize}\left\{\max\left(F_{iJ}\right)\right\} \tag{3}$$

$$F_{ij} \le F_{ij+1} - p_{ij+1}, \quad j = 1, 2, \ldots J, \quad \forall i \tag{4}$$

$$\sum_{O_{ij} \epsilon G(k)} X_{O_{ij},k} \le 1, \quad \forall k. \tag{5}$$

The objective function represented by (3) minimizes the maximum finish time in the set of the jobs i and therefore minimizes the makespan. Equation (4) satisfies precedence relationships between operations and (5) imposes that an operation can only be assigned to a machine at a time.

Table 1 shows an example of 3 jobs and 3 machines and their sequences for JSSP.

2.2. Type of Schedules. Three types of feasible schedule are considered: semiactive, active, and nondelay schedule [22]. In a semiactive schedule, there are no operations that can be started earlier without altering the sequences of the operations. In a semiactive schedule the makespan may often be reduced by shifting an operation to the left without delaying

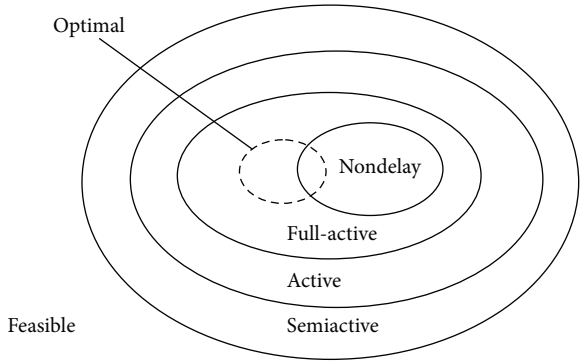


Figure 1: Relationship of the schedules.

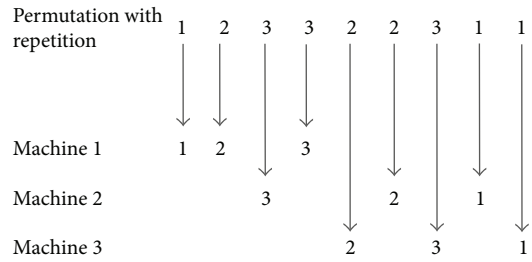


Figure 2: Permutation with repetition representation for 3 jobs and 3 machines.

other jobs. Reassignment of operations is called a permissible left shift. An active schedule is a schedule with no more permissible left shifts. Therefore, the set of active schedules is a subset of semiactive schedules. In a nondelay schedule, no machine is kept idle at a time when it could begin processing other operations and hence the set of nondelay schedules is a subset of active schedules. In addition the search space can be further reduced by implementing a full-active schedule introduced by Zhang et al. [10]. A full-active schedule is defined as a schedule where there is no more permissible left or right shift. Optimal solution of the scheduling always lies in the full-active schedule. Figure 1 illustrates the interaction of the schedules.

In order to generate the full-active schedule, we employ a scheduling approach called iterative forward-backward pass [23] in Section 3.1.2 which performs a kind of local search that can be used to introduce heuristic improvement into genetic search.

3. Hybrid Genetic Algorithm

GA is a stochastic search optimization technique that mimics the evolutionary processes in biological systems. This approach begins with a population, which represents a set of potential solutions in the search space. Each individual in the population is assigned a value by GA according to a problem specific objective function. The individuals will attempt to combine the good features in each individual in the population using a reproduction operator step such as crossover or mutation in order to construct individuals which are better suited than previous individuals. Through this evolution process, individuals that are less fit tend to be replaced by fitter individuals to generate a new population which eventually the desired optimal solutions will be found.

3.1. Schedule Generator Procedure

3.1.1. Chromosome Representation and Decoding. Representation of JSSP in chromosome is classified by Cheng et al. [22] into two approaches: direct and indirect. The direct approach directly encodes and decodes the representation into schedule while the indirect approach needs a schedule builder to encode and decode the chromosome into the schedule. The indirect approach is adopted in this study to represent the individual.

The chromosome in this paper is represented as operation based representation. In this representation, i represents the number of jobs $i = 1, 2, 3 \ldots, n$ and is repeated according the total number of operations in the job. Figure 2 illustrates the representation of 3 jobs and 3 machines. The chromosome is represented as [1 2 3 3 2 2 3 1 1], where jobs 1, 2, and 3 are represented as numbers 1, 2, and 3 in the chromosome, respectively. The number of occurrences in each chromosome depends on the number of operations required. The chromosome is scanned from left to right and at the jth occurrence of a job number refers to the jth operation in the technological sequence of this job. The chromosome created is always feasible and legal.

A scheduling can be built by constructing a schedule builder that performs a simple local search to decode the genes of the chromosome from left to right to a list of ordered operations. The first operation in the list is scheduled first, then the second operation, and so on. The operation will always be shifted to the left until time is equal to zero or inserted into a blank time interval between operations to find the earliest completion time. The process is repeated until all operations are scheduled. A schedule generated by the procedure can be guaranteed to be an active schedule [22]. Figure 3(a) illustrates the scheduling encoded by following operation sequences in the chromosome [1 1 3 $\cdots$]. Applying the process will enable the job to find possible earlier start time before being appended as last operation in the machine (Figure 3(b)) and the chromosome encoded is transformed to [1 3 1 $\cdots$]. In this representation, two or more chromosomes decoded may be translated into an identical schedule.

3.1.2. Schedule Generation Procedure. The procedure used to construct full-active schedule is based on a scheduling scheme called iterative forward-backward pass. This approach was proposed by Lova et al. [23] which had been shown to produce significant improvement in reducing makespan in the other field of scheduling problems.

The algorithm described in Section 3.1.1 uses a forward pass approach to generate an active schedule with the makespan $\max(F_{iJ})$ in the schedule in which the operations are able to shift left until time is equal to zero. Backward pass is a reverse process of the forward pass where the operations

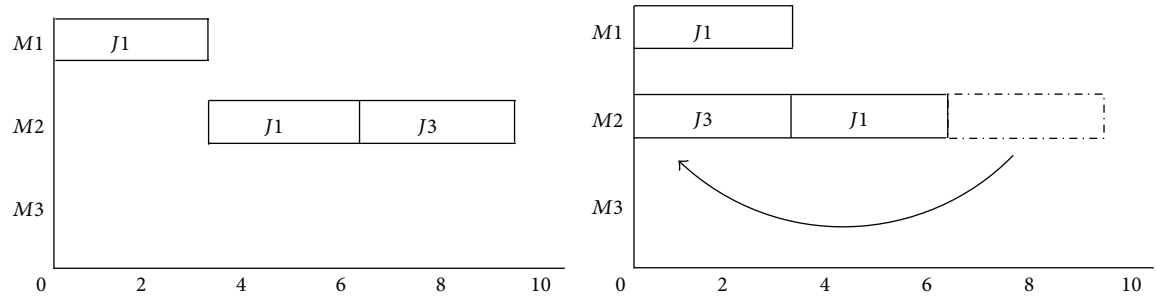


Figure 3: Permissible left shift for semiactive schedule.

in a schedule start from end of the schedule and end at the beginning of the schedule $\min(F_{i1})$, in which the operations are able to shift right until time is equal to $\max(F_{iJ})$.

The iterative forward-backward pass approach can be described with the following steps.

Step 1. An active schedule chromosome is generated by forward pass with maximum makespan $\max(F_{iJ})$.

Step 2. Apply backward pass on the chromosome from Step 1. Chromosome is scanned starting from right to left to generate a schedule with start time $\max(F_{iJ})$ (Figure 4). Through the right shifting in the schedule, we can obtain the beginning of the schedule $\min(F_{i1})$ and the makespan of this schedule is given as

$$BC_{\max} = \max\left(F_{iJ}\right) - \min\left(F_{i1}\right). \quad (6)$$

$BC_{\max}$ and $\min(F_{i1})$ denote makespan and minimum time, respectively, in this new schedule. The new schedule is encoded into a new chromosome with makespan $BC_{\max}$.

Step 3. If $BC_{\max} < \max(F_{iJ})$, the makespan obtained from the schedule in Step 2 is less than the makespan in the schedule of Step 1, there is improvement of the schedule makespan, and then the new chromosome is used in Step 1; otherwise the active schedule chromosome and the makespan generated by forward pass are maintained. Steps 1 and 2 are repeated until there is no further improvement on the schedule (Figure 5).

In this iterative function, the makespan of both processes is mutually restricted and hence the makespan of new solution generated either is lesser or remains unchanged. The last schedule generated by forward pass is equivalent to the last schedule generated by backward pass with the same makespan and they are sharing the same critical path.

Figure 6 shows the steps where each chromosome generated by the GA employed the schedule generator procedure. The procedure starts from generating the full-active schedule by schedule generation procedure and then applying the neighborhood search to improve the schedule obtained. After improvement of the schedule ends, the corresponding quality of the makespan is obtained. The neighborhood search is presented in the next section.

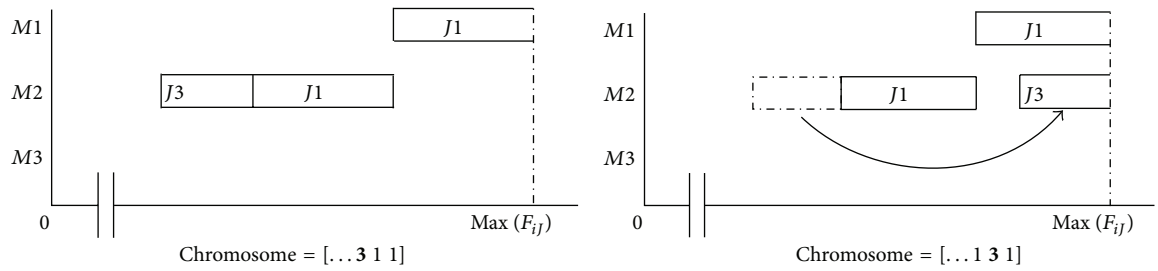
3.1.3. Neighborhood Search Procedure. Reduction of the search space does not guarantee the optimal solution will be found. Therefore we use a neighborhood search as an exploitation mechanism to decrease the makespan. This mechanism is restricted to searching the possible solutions in a critical path that consists of longest sequences of operation in a schedule. Swapping the operations on the path by using neighborhood search significantly reduces the total length of the makespan [11].

Instead of the swap operation which is determined deterministically as in [11], we modify the operation such that we chose the operations to be swapped randomly in a critical block. The neighborhood search starts with the identification of the critical path in the schedule generated by the scheduling process. Operations on the critical path are called critical operations. A critical block consists of a maximal sequence of adjacent critical operations that are processed on the same machine [11]. Our neighborhood is defined as the random swap between two jobs in a critical block that contains two or more operations. No swap is made if the critical block contains only one operation.

All possible moves of the operations are predetermined (Figure 7). A swap of the operations is accepted if it improves the makespan; otherwise the operations remain unchanged. Once the swap is accepted, the new critical path must be identified. The procedure is repeated and stops if there is no swap that can improve the makespan. Pseudocode for the neighborhood search is presented in Algorithm 1.

3.2. Hybrid GA. In our proposed hybrid GA, the search methods will be based on intensification and diversification mechanisms. The neighborhood search acts as an intensification mechanism that exploits the better solution in an individual and GA functions as diversification mechanism that explores the search space to provide different individuals for the local search. Structure of the hybrid GA that is based on a standard GA may be represented by the pseudocode in Algorithm 2.

Population is initialized randomly with a set of parameters and a particular chromosome which is often referred to as an individual. Each individual is evaluated by a fitness function and the parents selected for the recombination, which favors fitter individuals. The selected chromosomes



(a) Before backward pass

(b) After backward pass

Figure 4: Backward pass.

```
while New solution accepted = true
New solutions accepted = false
Determine the critical path, critical block in New schedule
List out the possible swaps of the operations
    for        k = 1 to total of possible swaps do
               Swap a pair of operations
               New schedule generated and makespan recalculated (New makespan)
               if        New makespan < Current makespan
                                 Current makespan = New Makespan
                                 New solution accepted = true
               end if
    end for
end while
```

Algorithm 1: Pseudocode for neighborhood search.

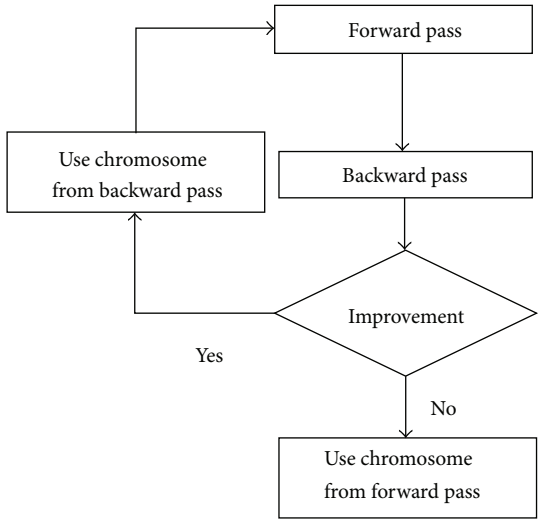


Figure 5: Iterative forward-backward pass.

are recombined using mechanisms of crossover and mutation to produce offspring. These offspring are then evaluated and then elitism strategy is applied to the new offspring with best fitness to replace the worst individuals in the previous population to generate a new population. Termination criterions of the GA in JSSP are set to terminate once the GA had achieved the optimal solution (if have) or maximum number of generations is reached.

3.2.1. Fitness and Selection Method. In this paper, we use nonlinear ranking to rank the evaluated chromosomes and each chromosome competes with the others and the selected chromosome survives to the next generation based on the fitness value (objective function). Chromosome with greater fitness indicates the greater probability to survive. The highest ranking chromosome in a population is considered the best solution. It is noted that the lowest makespan is given the highest ranking.

Stochastic universal sampling (SUS) is applied to select the parents for recombination in our GA. SUS is one of the selection methods that are often used in practice because they have less stochastic noise and have a constant selection pressure [24]. This fitness based proportionate selection will choose the chromosomes process with minimum spread and zero bias. SUS uses N equally spaced pins on wheel, where N is the number of selections required. The population is shuffled randomly and a single random number in the range $[0 - \sum f_i/N]$ is generated in which $\sum f_i$ represents the sum of all the fitness values of the individuals in the population.

3.2.2. Proposed Extended Precedence Preservative Crossover (EPPX). Generally in natural biological system the reproduction takes place between two parents (bisexual) or in a single

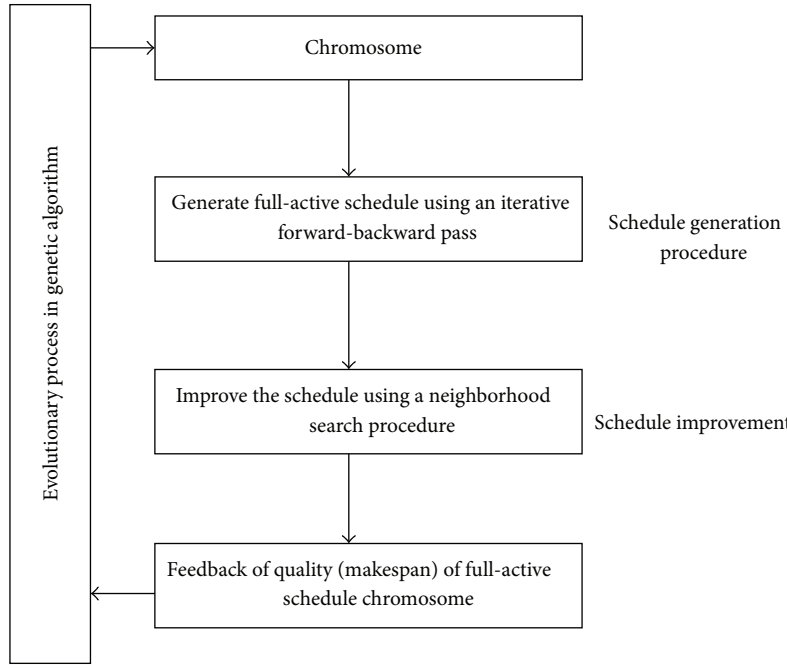


Figure 6: Architecture of schedule generation procedure.

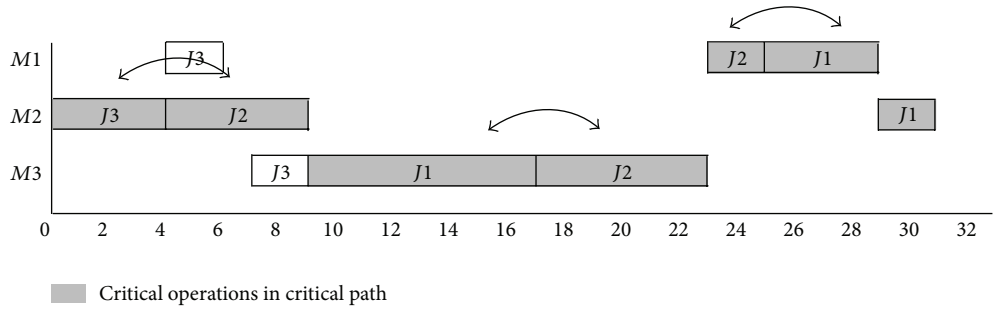


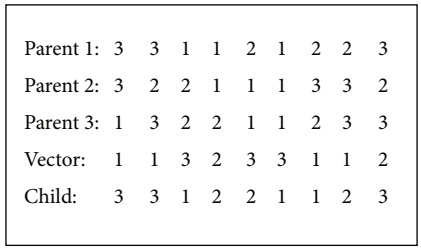
Figure 7: Critical path, critical operations, and possible operations swaps.

```
Initialize population
Evaluation
while termination criterion = false
        Selection
        Crossover
        Mutation
        Evaluation
        Reinsertion
End while
```

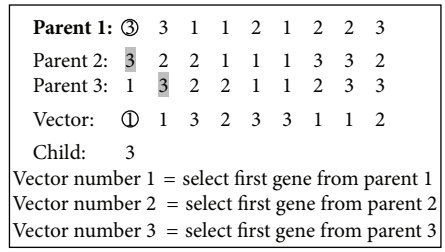
Algorithm 2: Pseudocode for a standard GA.

parent (asexual). However from the computational perspective, there is no restriction on the number of parents to use. Therefore some of the multiparents crossover operators are extended from the two-parent crossover operators [17–20] for recombination. As mentioned before, our proposed EPPX is an extension of PPX. A crossover mask in the form of a vector is generated randomly to determine in which parent the genes, specified in the mask, are to be selected for recombination. The multiparents will then recombine into a single offspring (Figure 8(a)). Starting from the first element on the mask vector, number 1 in the first element of the mask vector indicates that the first gene in that parent 1 is selected. In general, the mask vector indicates from which parent the element is to be selected. In this example, the selected job (job 3) is selected and eliminated in the other parents (Figures 8(b) and 8(c)). The second element in the mask indicates that the first element (after deletion) is to be selected also from parent 1 (Figure 8(c)). The third element in the mask shows that the first element in parent 3 is selected (Figure 8(d)). The process continues until all the elements in the mask have been examined. Algorithm 3 outlines the algorithm for multiparents crossover for JSSP.

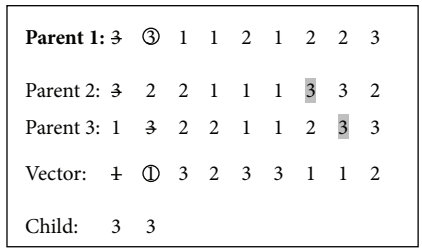
3.2.3. Mutation. Mutation is a genetic operator, analogous to the biological mutation, which is used to maintain genetic



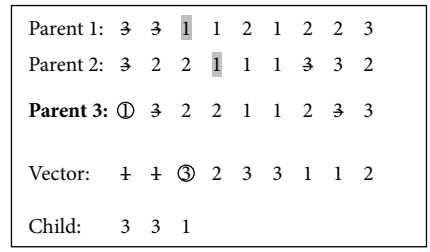
(a) Example of EPPX



(b) First step—vector number 1; the first gene in parent 1 is selected and the same job from the other parents is removed



(c) Second step—previous selected gene will be deleted, first gene (after deletion) in parent 1 is selected, and the job from the other parents are removed



(d) Repeat—the process will continue until at the end of the vector

Figure 8: EPPX crossover.

```
Crossover vector generated randomly
Select three parents: → S1, S2, and S3
    for  k = 1 to length of the chromosome do
    Select vector number by position kth starting from the left element in the vector
        case
            Vector number 1:
                Choose first gene at left most S1
                Search same job number at left most in S2 and S3
                Remove the first gene in S1 and gene searched in S2 and S3
            Vector number 2:
                Choose first gene at left most S2
                Search same job number at left most in S1 and S3
                Remove the first gene in S2 and gene searched in S1 and S3
            Vector number 3:
                Choose first gene at left most S3
                Search same job number at left most in S1 and S2
                Remove the first gene in S3 and gene searched in S1 and S2
        end case
        Selected gene insert to new chromosome by sequence from left to right
    end for
```

Algorithm 3: Pseudocode for EPPX (3 jobs and 3 machines).

diversity from one generation of a population of chromosomes to the next. In this study, the mutation is applied by selecting two genes in different positions and different jobs inside the same chromosome to be swapped. The process will be repeated if two genes selected are at the same position or the same job. Figure 9 illustrates the swapping of the two genes in the chromosome.

4. Experiment Setup and Results

4.1. Experimental Setup. To test the performance of our hybrid GA, we consider the benchmark from four classes of different JSSP test problems: instances FT06, FT10, and FT20 from Fisher and Thompson [25], instances ORB01 to ORB10 from Applegate and Cook [26], instances ABZ5 to

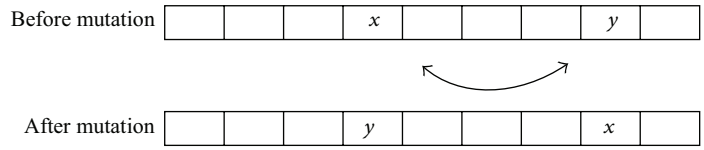


Figure 9: Mutation by swapping two genes in the chromosome.

Table 2: Output for different crossover rates and mutation rates.

	Mut_01	Mut_02	Mut_03	Mut_04	Mut_05	Mut_06	Mut_07	Mut_08	Mut_09	Mut_10	Average	Relative error (%)
Crs_10	971.96	978.20	972.64	968.06	969.28	972.34	971.58	965.46	963.24	968.12	970.09	4.31
Crs_9	967.38	975.94	965.82	968.24	970.30	965.36	963.44	961.14	961.00	960.32	965.89	3.86
Crs_8	971.54	968.00	967.24	965.92	966.10	964.28	964.58	958.92	958.84	959.12	964.45	3.70
Crs_7	973.94	968.32	969.26	963.72	963.48	961.58	958.46	957.84	959.02	958.94	963.46	3.60
Crs_6	972.96	968.32	969.30	965.26	964.02	965.10	964.08	960.16	959.90	959.58	964.87	3.75
Crs_5	971.88	975.96	975.60	969.12	962.58	967.36	960.92	961.60	961.04	957.10	966.32	3.90

Crs_10 represents crossover rate at 1.0, Crs_9 represents crossover rate at 0.9, and so on.
Mut_01 represents mutation rate at 0.1, Mut_02 represents mutation rate at 0.2, and so on.

ABZ9 from Adams et al. [27], and instances YN1 to YN4 from Yamada and Nakano [28]. The instances that we use were downloaded from http://people.brunel.ac.uk, OR-Library of Brunel University.

MATLAB 7.11 R2010b is used to develop the GA and the simulations are run using workstation Intel Xeon 12 GB RAM with 2.4 GHz processor. The population size comprises 100 individuals for problems FT06, FT10, ABZ5, ABZ6, and ORB01–ORB10. In large sized problems (FT20, ABZ7–ABZ9, and YN1–YN4), their population sizes are increased to 150. The number of parents for EPPX ranges from 3 to 10. Crossover rate and mutation rate are set to 0.7 and 1.0, respectively. After the recombination process, elitism strategy is applied; 10% of the fittest new offspring replaces 10% of the worst individuals in the previous population to generate a new population. Each instance is run for 50 times for different numbers of parents to find out their best solution available.

4.1.1. Maximum Number of Generations. In the multiparent crossover, the parents will be recombined to generate one child. Therefore, the relationship of different numbers of parents with different total solutions (offspring) for a population exists. Thus the total solutions are defined as

$$\text{total solutions} = \frac{\text{MAXGen} \times \text{population size}}{\text{number of parents}}. \quad (7)$$

The maximum number of generations is denoted as MAXGen. To be fair, the total number of generations is adjusted to make sure that different numbers of parents for recombination generate approximately the same number of total solutions by referring to (8). Total solutions for instances FT06, FT10, ABZ5 and ABZ6, and ORB01–ORB10 after recombination process are set at around 5000 schedules. In addition, problems FT20, ABZ7–ABZ9, and YN1–YN4 are set at around 10,000 schedules. Among the multiparents crossover, three-parent crossover operator is used as reference because we consider them as the starting point of multiparents recombination (more than two parents). Consider

$$\text{MAXGen} = \frac{\text{total solutions} \times \text{number of parents}}{\text{populations size}}. \quad (8)$$

4.1.2. Crossover Rate. In our cases, due to the different problems sizes and different numbers of parents for recombination, we need to fix crossover rate and mutation rate to use them for all numbers of parents. Instance FT10 is selected for the parameters testing because it is considered as a difficult problem. Among the multiparents crossovers, three-parent crossovers are used because we consider them as the starting point of multiparents recombination (more than two parents).

The dependencies between the crossover and mutation rates are tested by the GA. The crossover rates are set from 1.0 to 0.5 with varied mutation rates from 0.1 to 1.0. Each case (example: crossover rate = 1.0, mutation rate = 0.1) will be run for 100 times and the average of each case will be figured out. The relative errors are calculated by computing the difference between the average solutions for each crossover rate and the optimal solution of FT10 (930).

In Table 2, the relative error for crossover rate at 0.7 appears as the lowest value compared to the other crossover rates. The frequencies of optimal solutions for FT10 at the crossover rate 0.7 that obtain from the simulation had shown the highest occurrences as indicated in Figure 10. Thus, it is reasonable for us to use the crossover rate at 0.7 to compare to other values.

4.1.3. Mutation Rate. In the multiparents crossover application, especially in the JSSP, there is a lack of information about the mutation rate values. Hence, we try to find the suitable mutation rate for our GA. Due to the inconsistencies of the results between the crossover rates and mutation rates we obtained from the simulation, Figure 11 plotted the best fit line for the problems. All lines for the different crossover

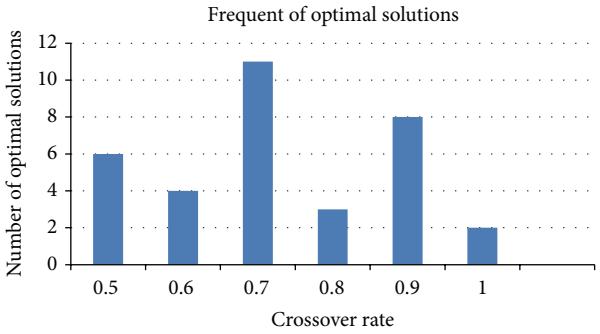


Figure 10: Frequency of optimal solutions appears (FT10) at different crossover rates.

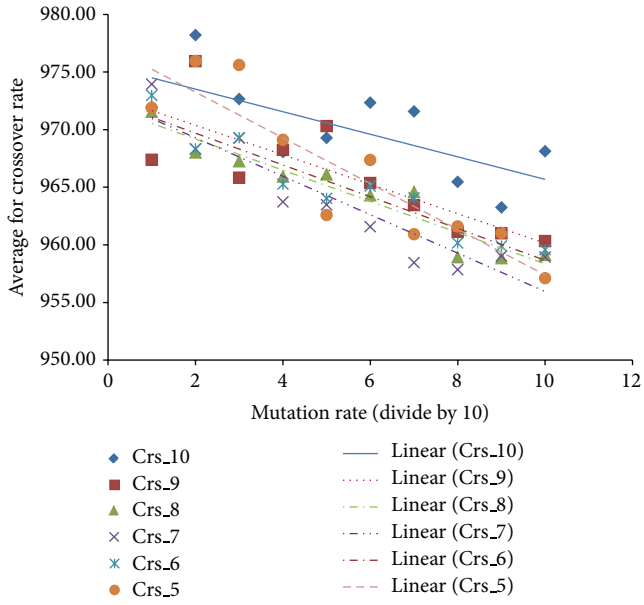


Figure 11: Best fit line for crossover with different mutation rates.

rates are decreasing from left to right which means that the average solutions will get better when the mutation rates increase. Thus, we conclude that the last mutation rate (1.0) performs the best when applied in this GA and it will be used as parameter for other instances and problems.

4.2. Results and Discussions

4.2.1. Comparison with the Other Permutation Crossover Operator That uses Two Parents. Table 3 lists the GAs using permutation crossover operator with various hybridization or modification that had been applied on the FT problem. The GAs in the list (except EPPX) use only two parents for crossover operation and the results vary because of different strategies of hybridization. Initially the earlier papers [21, 29] was unable to obtain the optimal solutions for FT10 using the proposed crossover operators. Then later by hybridization or modification using this method, the optimal results of FT10 can be reached, including our EPPX. EPPX is able to perform well in the FT20 as compared to the GP-GA and IPPX with the deviation of 1.12%. The acceptable ranges of comparable GAs are up to 7% [21]. Tested results show that EPPX is able to get the solution within these values and is considered applicable for solving the JSSP. Instance FT06 can be solved easily with the number of parents used for crossover ranging from 3 to 10 parents. For difficult problems such as FT10, EPPX also performs well because it is able to get the optimum solution with the number of parents of 3 and 5 in crossover operation. As the variable of JSSP increases (instance FT20), a deviation between optimal solution and EPPX which obtains the best solution of 1178 by using 6 parents in EPPX's crossover operation occurs.

4.2.2. Comparison with Different Algorithms. ORB problem contains ten 10 × 10 instances. To illustrate the effectiveness, the proposed multiparent crossover is compared with the other algorithms such as synergistically combining small and large neighborhood schemes (SLENP) [31], parallel genetic algorithms (PGA) [13] with two-parent crossover, and greedy randomized adaptive search procedure (GRASP) [32]. The relative errors (REs) are calculated based on deviation between BKS and the best solution found (Best). In EPPX, different numbers of parents are used for recombination. The numbers of parents (MP) that achieve the best solutions are recorded in the table.

Table 4 states the name, size (number of jobs x number of machines), the best known solutions (BKS), the best solutions found (Best), multiparents that obtain the best solutions (MP), and relative error (RE) for each test instance. The RE is calculated from the gap between Best and BKS in percentage. The effectiveness of the proposed algorithm is analyzed by referring to the total relative error, total RE, in the last lane. In the ORB problem, EPPX found the BKS in 5 instances. The best solutions found are located in the different numbers of parents for different instances. Total RE provided by EPPX is lower than GRASP but higher than SLENP and PGA. This shows that EPPX performs averagely among these methods.

ABZ problem contains five instances with the sizes 10×10 and 20 × 15. ABZ8 and ABZ9 and no optimal solutions have been known. Table 5 shows the detailed results of comparison for ORB instances. EPPX is able to perform better as compared to PGA and GRASP with lower total RE.

YN problem consists of four instances with size of 20 × 20 that are still open problems. EPPX is compared with TSSA and PGA in Table 6 because algorithm GRASP did not provide the results of each instance.

4.2.3. The Performance Evaluation of EPPX. FT, ORB, ABZ, and YN problems are used to evaluate the performance of our hybrid GA which incorporates EPPX. Tables 3–6 show that, in 9 out of 22 cases, the EPPX has been able to find the best known solution for the corresponding benchmark instance. Table 4 shows that our results are competitive when compared to PGA achieving the same solutions in 5 out of 10 instances. However when the problem size increases (Tables 5 and 6) the performances of our algorithm are better with smaller relative error compared to PGA and GRASP. The computational results indicated that EPPX is capable of

Table 3: Results for FT06, FT10, and FT20 with n jobs $\times$ m machines.

Author(s)	Year	Crossover operator	FT06 (6 × 6)	FT10 (10 × 10)	FT20 (20 × 5)
Optimum			55	930	1165
Gen et al. [29]	1994	Partial schedule exchange crossover	55	962	1175
Bierwirth et al. [21]	1996	Generalized permutation GP-GA	55	936	1181
Park et al. [30]	2003	Parallel genetic algorithm PGA	55	930	1173
Ripon et al. [15]	2011	Improved precedence preservation crossover IPPX	55	930	1180
	2014	Multiparents crossover EPPX	55	930	1178

Table 4: Results for ORB problems.

Instances	Size	BKS	EPPX			SLENP		PGA		GRASP	
			Best	RE	MP	Best	RE	Best	RE	Best	RE
ORB01	10 × 10	1059	1077	1.70	9	1059	0.00	1060	0.09	1070	1.04
ORB02	10 × 10	888	889	0.11	3, 4, 6, 7, 9	888	0.00	889	0.11	889	0.11
ORB03	10 × 10	1005	1022	1.69	7	1005	0.00	1020	1.49	1021	1.59
ORB04	10 × 10	1005	1005	0.00	5, 9	1005	0.00	1005	0.00	1031	2.59
ORB05	10 × 10	887	890	0.34	5	887	0.00	889	0.23	891	0.45
ORB06	10 × 10	1010	1021	1.09	4	1010	0.00	1013	0.30	1013	0.30
ORB07	10 × 10	397	397	0.00	10	397	0.00	397	0.00	397	0.00
ORB08	10 × 10	899	899	0.00	4	899	0.00	899	0.00	909	1.11
ORB09	10 × 10	934	934	0.00	10	934	0.00	934	0.00	945	1.18
ORB10	10 × 10	944	944	0.00	3–7, 9	944	0.00	944	0.00	953	0.95
	Total RE			4.93			0.00		2.22		9.32

Table 5: Results for ABZ problems.

Instances	Size	BKS	EPPX			SLENP		PGA		GRASP	
			Best	RE	MP	Best	RE	Best	RE	Best	RE
ABZ5	10 × 10	1234	1234	0.00	3	1234	0.00	1236	0.16	1238	0.32
ABZ6	10 × 10	943	943	0.00	7, 8	943	0.00	943	0.00	947	0.42
ABZ7	20 × 15	656	683	4.12	4	662	0.91	685	4.42	723	10.21
ABZ8	20 × 15	665	701	5.41	3	668	0.45	704	5.86	729	9.62
ABZ9	20 × 15	678	712	5.01	4	688	1.47	723	6.64	758	11.80
	Total RE			14.54			2.83		17.08		32.37

Table 6: Results for YN problems.

Instances	Size	BKS	EPPX			SLENP		PGA	
			Best	RE	MP	Best	RE	Best	RE
YN1	20 × 20	884	911	3.05	7	892	0.90	933	5.54
YN2	20 × 20	907	941	3.75	4	911	0.44	944	4.08
YN3	20 × 20	892	928	4.04	4	900	0.90	928	4.04
YN4	20 × 20	968	1011	4.44	6	982	1.45	1018	5.17
	Total RE			15.28			3.69		18.82

adapting to more difficult and larger size problems in comparison with PGA that only uses two parents for crossover. The best solutions found are located in the different numbers of parents for different instances indicated in column MP in all tables.

Table 7 displays the average computational times for different numbers of parents for each instance. Optimal solutions for easy problem (FT06) can be found in shorter time while, as for harder problem, the algorithm needs more time to find out the near optimal solutions. It can be seen that our algorithms require on average less than 15 seconds CPU time. The CPU times of EPPX, PGA, and GRASP are calculated per iteration meanwhile SLENP is based on the time per run.

The computational time for EPPX and PGA which are based on genetic algorithm is large for harder problems.

Table 7: Average computational time for different number of parents.

Instances	Size	Computational time (in second)										
		EPPX (number of parents)								SLENP	PGA	GRASP
		3	4	5	6	7	8	9	10			
FT 06	6 × 6	**0.02**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	0.01	0.76	0.70
FT 10	10 × 10	**0.93**	0.72	**0.57**	0.46	0.41	0.37	0.33	0.30	9.22	1.54	2.90
FT 20	20 × 5	**1.19**	0.89	0.73	0.61	0.52	0.45	0.40	0.38	2.73	1.52	4.30
ORB01	10 × 10	1.02	0.78	0.64	0.52	0.45	0.38	**0.37**	0.33	5.96	1.56	2.90
ORB02	10 × 10	**0.92**	**0.70**	0.57	**0.46**	**0.41**	0.35	**0.32**	0.29	0.48	1.56	3.80
ORB03	10 × 10	1.07	0.82	0.66	0.55	**0.48**	0.40	0.38	0.34	7.42	1.56	3.10
ORB04	10 × 10	0.93	0.73	**0.59**	0.47	0.41	0.36	**0.33**	0.30	7.58	1.56	3.10
ORB05	10 × 10	0.98	0.81	**0.60**	0.49	0.43	0.37	0.34	0.30	12.09	1.56	2.80
ORB06	10 × 10	1.11	**0.80**	0.64	0.53	0.46	0.40	0.37	0.33	9.17	1.56	3.10
ORB07	10 × 10	0.90	0.70	0.55	0.46	0.40	0.35	0.32	**0.29**	0.28	1.56	3.20
ORB08	10 × 10	1.01	**0.75**	0.63	0.51	0.46	0.39	0.35	0.33	6.02	1.56	3.10
ORB09	10 × 10	0.92	0.72	0.57	0.47	0.41	0.36	0.33	**0.30**	0.51	1.56	3.10
ORB10	10 × 10	**0.98**	**0.74**	**0.59**	**0.48**	**0.42**	0.36	**0.33**	0.30	0.18	1.56	2.90
ABZ5	10 × 10	**0.84**	0.64	0.51	0.41	0.36	0.31	0.29	0.26	3.55	1.78	0.30
ABZ6	10 × 10	0.79	0.61	0.53	0.43	**0.38**	**0.34**	0.31	0.28	0.15	1.56	3.10
ABZ7	20 × 15	6.17	**4.77**	3.78	3.29	2.79	2.42	2.09	1.97	64.82	7.36	17.40
ABZ8	20 × 15	**6.45**	4.76	3.85	3.21	2.77	2.52	2.05	1.93	55.94	7.28	18.20
ABZ9	20 × 15	5.83	**4.38**	3.53	2.97	2.49	2.19	1.96	1.79	35.82	7.32	17.10
YN1	20 × 20	12.19	9.32	7.56	6.46	**5.39**	4.66	4.13	3.79	40.04	7.60	—
YN2	20 × 20	11.72	**8.97**	7.32	6.00	5.18	4.48	3.95	3.80	62.31	7.60	—
YN3	20 × 20	14.09	**10.43**	8.56	7.13	6.14	5.28	4.71	4.45	42.18	7.60	—
YN4	20 × 20	10.52	7.75	6.38	**5.30**	4.51	3.89	3.45	3.26	56.05	7.60	—

However the large computational time displayed by SLENP and GRASP which are based on single solution is larger when compared to EPPX and PGA. PGA is expected to perform better in terms of computational time because the algorithm is run in parallel. Based on the calculation per iteration, EPPX is more efficient compared to PGA and GRASP because it requires less time to do the calculation.

The highlighted time in the EPPX shows the computational time of the result achieved by different number of parents. Multiparents are recombined to produce a single child resulting in a decreasing rate of computational time with the increasing number of parents. It is proven that increasing the number of parents in the EPPX may reduce the calculation time to obtain the optimal result in some instances.

5. Conclusion

This paper presents a hybrid genetic algorithm with multiparent crossover, EPPX, for the job shops scheduling problem. The hybrid GA combines genetic algorithm with neighborhood search in which GA explores the population while the neighborhood search exploits the individual solution to find better solution. The chromosome represented by operation based representation is used to generate a full-active schedule through iterative forward-backward pass which can further reduce the search space.

In the experimental results, EPPX using multiparent is able to get the solutions within the acceptable range of GA values. Results show that the best solutions are obtained from different numbers of parents for crossover; thus it is proven that GA is not restricted to two-parent crossover in order to find the best solution. The number of parents used in EPPX and GA is dependent on the problem and it may be observed that the best solutions for different instances are produced by different numbers of parents.

In future works, suitable ranges for number of parents for crossover are needed to determine and solve JSSP. It is also anticipated that the efficiency of GA can be increased by embedding other local searches into GA. Further research is necessary to reduce the computational time as a way to improve the efficiency of hybrid genetic algorithm with multiparent crossover for JSSP.

Conflict of Interests

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

Acknowledgment

This work has been supported by University of Malaya Research Grant Scheme UMRG-RG116-10AFR.

References

[1] A. Jones, L. C. Rabelo, and A. T. Sharawi, "Survey of job shop scheduling techniques," in *Wiley Encyclopedia of Electrical and Electronics Engineering*, John Wiley & Sons, New York, NY, USA, 1999.

[2] S. Ólafsson, "Metaheuristics," in *Handbooks in Operations Research and Management Science*, vol. 13, pp. 633–654, 2006.

[3] C. Zhang, P. Li, Z. Guan, and Y. Rao, "A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem," *Computers & Operations Research*, vol. 34, no. 11, pp. 3229–3242, 2007.

[4] S. Z. Song, J. J. Ren, and J. X. Fan, "Improved simulated annealing algorithm used for job shop scheduling problems," in *Advances in Electrical Engineering and Automation*, vol. 139, pp. 17–25, Springer, Berlin, Germany, 2012.

[5] Z. Yaqin, L. Beizhi, and W. Lv, "Study on job-shop scheduling with multi-objectives based on genetic algorithms," in *Proceedings of the International Conference on Computer Application and System Modeling (ICCASM '10)*, pp. v10294–v10298, IEEE, October 2010.

[6] M. Yin, X. Li, and J. Zhou, "An efficient job shop scheduling algorithm based on artificial bee colony," *Scientific Research and Essays*, vol. 6, no. 12, pp. 2578–2596, 2011.

[7] L. Davis, "Job shop scheduling with genetic algorithms," in *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, vol. 140, 1985.

[8] R. Cheng, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies," *Computers & Industrial Engineering*, vol. 36, no. 2, pp. 343–364, 1999.

[9] J. F. Gonçalves, J. J. Mendes, and M. G. C. Resende, "A hybrid genetic algorithm for the job shop scheduling problem," *European Journal of Operational Research*, vol. 167, no. 1, pp. 77–95, 2005.

[10] C. Zhang, Y. Rao, and P. Li, "An effective hybrid genetic algorithm for the job shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 39, no. 9-10, pp. 965–974, 2008.

[11] E. Nowicki and C. Smutnicki, "A fast taboo search algorithm for the job shop problem," *Management Science*, vol. 42, no. 6, pp. 797–813, 1996.

[12] R. Qing-Dao-Er-Ji and Y. Wang, "A new hybrid genetic algorithm for job shop scheduling problem," *Computers and Operations Research*, vol. 39, no. 10, pp. 2291–2299, 2012.

[13] R. Yusof, M. Khalid, G. T. Hui, S. Md Yusof, and M. F. Othman, "Solving job shop scheduling problem using a hybrid parallel micro genetic algorithm," *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 5782–5792, 2011.

[14] M. Watanabe, K. Ida, and M. Gen, "A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem," *Computers & Industrial Engineering*, vol. 48, no. 4, pp. 743–752, 2005.

[15] K. S. N. Ripon, N. H. Siddique, and J. Torresen, "Improved precedence preservation crossover for multi-objective job shop scheduling problem," *Evolving Systems*, vol. 2, no. 2, pp. 119–129, 2011.

[16] H. Mühlenbein and H. M. Voigt, "Gene pool recombination in genetic algorithms," in *Proceedings of the Metaheuristics Conference*, Kluwer Academic Publishers, Norwell, Mass, USA, 1995.

[17] A. E. Eiben and C. H. van Kemenade, "Diagonal crossover in genetic algorithms for numerical optimization," *Control and Cybernetics*, vol. 26, no. 3, pp. 447–466, 1997.

[18] A. Wu, P. W. M. Tsang, T. Y. F. Yuen, and L. F. Yeung, "Affine invariant object shape matching using genetic algorithm with multi-parent orthogonal recombination and migrant principle," *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 282–289, 2009.

[19] A. E. Eiben, P.-E. Raué, and Z. Ruttkay, "Genetic algorithms with multi-parent recombination," in *Parallel Problem Solving from Nature—PPSN III*, vol. 866 of *Lecture Notes in Computer Science*, pp. 78–87, 1994.

[20] C.-K. Ting, C.-H. Su, and C.-N. Lee, "Multi-parent extension of partially mapped crossover for combinatorial optimization problems," *Expert Systems with Applications*, vol. 37, no. 3, pp. 1879–1886, 2010.

[21] C. Bierwirth, D. C. Mattfeld, and H. Kopfer, "On permutation representations for scheduling problems," in *Parallel Problem Solving from Nature—PPSN IV*, vol. 1141 of *Lecture Notes in Computer Science*, pp. 310–318, Springer, Berlin, Germany, 1996.

[22] R. Cheng, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms—I. Representation," *Computers & Industrial Engineering*, vol. 30, no. 4, pp. 983–997, 1996.

[23] A. Lova, C. Maroto, and P. Tormos, "Multicriteria heuristic method to improve resource allocation in multiproject scheduling," *European Journal of Operational Research*, vol. 127, no. 2, pp. 408–424, 2000.

[24] T. Blickle and L. Thiele, "A comparison of selection schemes used in evolutionary algorithms," *Evolutionary Computation*, vol. 4, no. 4, pp. 361–394, 1996.

[25] H. Fisher and G. L. Thompson, "Probabilistic learning combinations of local jobshop scheduling rules," in *Industrial Scheduling*, J. F. Muth and G. L. Thompson, Eds., pp. 225–251, Prentice Hall, Englewood Cliffs , NJ, USA, 1963.

[26] D. Applegate and W. Cook, "Computational study of the job-shop scheduling problem," *ORSA Journal on Computing*, vol. 3, no. 2, pp. 149–156, 1991.

[27] J. Adams, E. Balas, and D. Zawack, "The shifting bottleneck procedure for job shop scheduling," *Management Science*, vol. 34, no. 3, pp. 391–401, 1988.

[28] T. Yamada and R. Nakano, "A genetic algorithm applicable to large-scale job-shop problems," in *Proceedings of the 2nd International Workshop on Parallel Problem Solving from Nature (PPSN '92)*, R. Manner and B. Manderick, Eds., pp. 281–290, Brussels, Belgium, 1992.

[29] M. Gen, Y. Tsujimura, and E. Kubota, "Solving job-shop scheduling problems by genetic algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. Humans, Information and Technology*, vol. 2, pp. 1577–1582, October 1994.

[30] B. J. Park, H. R. Choi, and H. S. Kim, "A hybrid genetic algorithm for the job shop scheduling problems," *Computers & Industrial Engineering*, vol. 45, no. 4, pp. 597–613, 2003.

[31] M. Amirghasemi and R. Zamani, "A synergetic combination of small and large neighborhood schemes in developing an effective procedure for solving the job shop problem," *SpringerPlus*, vol. 3, no. 1, 2014.

[32] S. Binato, W. J. Hery, D. M. Loewenstern, and M. G. C. Resende, "A GRASP for job shop scheduling," in *Essays and Surveys in Metaheuristics*, pp. 59–79, Springer, 2001.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 745732, 14 pages
http://dx.doi.org/10.1155/2015/745732

Research Article

Stability and Bogdanov-Takens Bifurcation of an SIS Epidemic Model with Saturated Treatment Function

Yanju Xiao,[1] Weipeng Zhang,[1] Guifeng Deng,[2] and Zhehua Liu[1]

[1]*School of Mathematics and Statistics, Northeast Normal University, Changchun, Jilin 130024, China*
[2]*School of Mathematics and Information, Shanghai Lixin University of Commerce, Shanghai 201620, China*

Correspondence should be addressed to Weipeng Zhang; wpzhang808@163.com

Received 11 July 2014; Revised 18 October 2014; Accepted 28 October 2014

Academic Editor: Shifei Ding

This paper introduces the global dynamics of an SIS model with bilinear incidence rate and saturated treatment function. The treatment function is a continuous and differential function which shows the effect of delayed treatment when the rate of treatment is lower and the number of infected individuals is getting larger. Sufficient conditions for the existence and global asymptotic stability of the disease-free and endemic equilibria are given in this paper. The first Lyapunov coefficient is computed to determine various types of Hopf bifurcation, such as subcritical or supercritical. By some complex algebra, the Bogdanov-Takens normal form and the three types of bifurcation curves are derived. Finally, mathematical analysis and numerical simulations are given to support our theoretical results.

1. Introduction

In the mathematical modeling of epidemic transmission, there are several factors that substantially affect the dynamical behavior of the models. Incidence rate functions are seen as a major factor in producing the rich dynamics of epidemic models in many literatures (see [1–25]). In most classical models of epidemics, the incidence rate is taken to be mass action incidence with bilinear interaction, that is, βIS, where β is the probability of transmission per contact and S and I represent the number of susceptible and infected individuals, respectively. Epidemic models with such bilinear incidence rates usually have at most one endemic equilibrium, and then the diseases will be eradicated if the basic reproduction number is less than one and will persist otherwise. Besides, there are also many other types of incidence rate functions, such as nonlinear incidence rate, standard incidence rate, and saturated incidence rate. Recently, there are many studies that have demonstrated the nonlinear incidence rate which is one of the key factors that induce periodic oscillations in epidemic models (see [1, 2, 9, 10, 15]). Moreover, Liu et al. [12] introduced a nonlinear incidence rate of the form

$$f(I)S = \frac{\beta I^p S}{1 + \alpha I^q}, \tag{1}$$

where βI^p means the infection force of the disease and $1/(1+\alpha I^q)$ measures the inhibition effect from the behavioral change of the susceptible individuals when the number of infective individuals increases. So we can see that the bilinear incidence rate βSI is a special case of (1) with $p = 1$ and $\alpha = 0$ or $q = 0$. Furthermore, Wang and Ruan in [16] studied the global dynamics of an SIRS model with the incidence function $f(I) = \beta I^2/(1 + \alpha I^2)$; that is, $p = q = 2$; they also showed that the SIRS epidemic model undergoes a Bogdanov-Takens bifurcation, that is, saddle-node, Hopf, and homoclinic bifurcations. To have a better understanding of the dynamics of the system, Tang et al. [17] calculated higher order Lyapunov values of the weak focus and reduced the system to a universal unfolding form for a cusp of codimension 3, and they gave the bifurcation surfaces

and displayed all limit cycles and monoclinic loops of order up to 2. Wei and Cui explored an SIS epidemic model with standard incidence rate in [19]. They took the incidence rate form $f(I)S = \beta IS/(I + S)$ and showed the dynamics and backward bifurcation of the SIS epidemic model.

Recently, in order to prevent and control the spread of the infectious diseases such as measles, tuberculosis, and flu, many mathematicians (see [7, 11, 16, 20, 23, 25–29]) have begun to investigate the role of treatment functions in epidemiological models. In some classical epidemic models, the treatment function is an important method to decrease the spread of the epidemiological diseases. Generally speaking, the treatment function of the infective individuals is assumed to be proportional to the number of the infective individuals. But every community should have a maximal capacity for the treatment of a disease and the resources for treatment should be very large. Therefore, it is very important to adopt a suitable treatment function. In [16], Wang and Ruan introduced a constant treatment function of diseases in an SIR model; that is,

$$T(I) = \begin{cases} r, & I > 0, \\ 0, & I = 0. \end{cases} \tag{2}$$

This means that they use the maximal treatment capacity to cure infective individuals so that the disease can be eradicated. They also found that the model undergoes saddle-node bifurcation, Hopf bifurcation, and Bogdanov-Takens bifurcation. Further, a piecewise linear treatment function was considered in [20]; that is,

$$T(I) = \begin{cases} kI, & 0 \le I \le I_0, \\ m, & I > I_0, \end{cases} \tag{3}$$

where $m = kI_0$ and k and I_0 are positive constants. This means that the treatment rate is proportional to the number of the infective individuals when the capacity of treatment has not been reached; otherwise it takes the maximal capacity of treatment kI_0. By considering the above treatment function, Wang [20] found that a backward bifurcation takes place in an SIR epidemic model. In [30], J. C. Eckalbar and W. L. Eckalbar constructed an SIR epidemic model with a quadratic treatment function; that is, $T(I) = \max\{rI - gI^2, 0\}$, $r, g > 0$. They found that the system has as many as four equilibria, with possible bistability, backward bifurcations, and limit cycles.

Recently, saturated treatment function has been widely applied in many epidemic models. In particular, in paper [23], Zhang and Liu took a continuous and differentiable saturated treatment function $T(I) = rI/(1 + \alpha I)$, where $r > 0$, $\alpha \ge 0$. r stands for the cure rate and α measures the extent of the effect of the infected being delayed for treatment. We can see that the treatment function $T(I) \sim rI$ when I is small enough, whereas $T(I) \sim r/\alpha$ when I is large enough. It is more realistic and it has the convenience of being continuous and differential compared to the previous ones. Furthermore, the authors in [23] found that $R_0 = 1$ is a critical threshold for disease eradication when this delayed effect for treatment is weak and a backward bifurcation will take place when this effect is strong. So, it is really important to adequately stress the interesting connection recently established between the choice of saturated treatment functions in epidemic models and the occurrence of backward bifurcation in the related system dynamics. In fact, recently, saturated-type treatment functions have been indicated as responsible for the occurrence of backward bifurcations for SIR [20, 23], for SIS [19, 21], and for SEIR [28, 29] models, supporting the general circumstance that saturated-type treatments can be one of the causes of backward bifurcations in epidemic models. In particular, in [29], such connection has also been shown and validated in a specific concrete disease-control setting.

In the real world, some infectious diseases do not confer immunity. Such infections do not have a recovered state and individuals become susceptible again after infection. This type of disease can be modeled by the SIS type. So the SIS epidemic model has been adopted by many mathematicians (see [8, 11, 18, 19, 21, 25]). And SIS models are appropriate for some bacterial agent diseases such as meningitis, plague, and venereal diseases and for protozoan agent diseases such as malaria and sleeping sickness.

Motivated by the above points, we will consider the following SIS model with bilinear incidence rate and saturated treatment function:

$$\begin{aligned} \frac{dS}{dt} &= A - dS - \lambda SI + \varepsilon I + \frac{rI}{1 + \alpha I}, \\ \frac{dI}{dt} &= \lambda SI - (d + \varepsilon + \mu) I - \frac{rI}{1 + \alpha I}, \end{aligned} \tag{4}$$

where S and I denote the numbers of susceptible and infective individuals, respectively. Positive constant A is the recruitment rate of the population. Positive constant d is the nature death rate of population. The bilinear incidence rate is λSI, where λ is positive. Positive constant ε is the natural recovery rate of infective individuals. Positive constant μ is the disease-related death rate. The saturated treatment function $h(I) \triangleq rI/(1 + \alpha I)$, where r is positive and α is nonnegative.

This paper focuses on the detailed dynamics analysis of the model (4). The local stability of these equilibria is investigated, which enables us to classify the types of model equilibria (e.g., attractor, saddle, or repeller). We show that the system has backward bifurcation and Bogdanov-Takens bifurcation (i.e., Hopf bifurcation, saddle-node bifurcation, and homoclinic bifurcation) under some certain conditions. Finally, the three bifurcation curves and the complicated global bifurcation phase portraits are derived by applying the Bogdanov-Takens normal form and the corresponding parameters which satisfy the conditions that ensure Bogdanov-Takens bifurcation exists.

The organization of this paper is as follows. In Section 2, we study the existence and local stability of equilibria and backward bifurcation. In Section 3, we investigate the global stability of the model. In Section 4, we give the supercritical and subcritical bifurcation under two different conditions in system (4). In Section 5, we show that the system (4) undergoes Bogdanov-Takens bifurcation under some certain conditions. In Section 6, some numerical simulations are displayed in detail. We close with a discussion in Section 7 on our mathematical results and epidemiological implications.

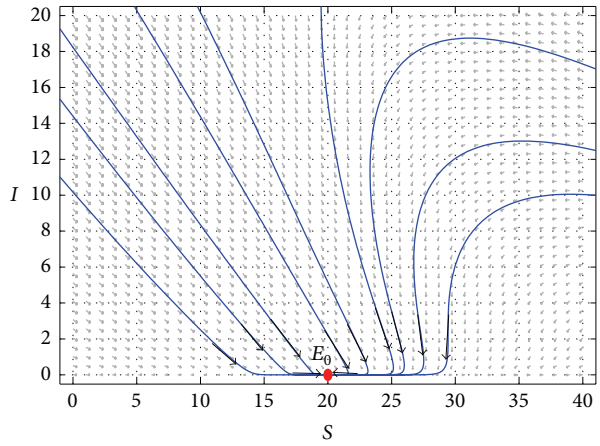


Figure 1: The disease-free equilibrium E_0 is locally asymptotically stable when $R_0 < 1$, with the parameter values $A = 2$, $d = 0.1$, $k = 0.1$, $\mu = 0.01$, $\lambda = 0.01$, $\alpha = 1$, $r = 2$, and $\varepsilon = 0.1$.

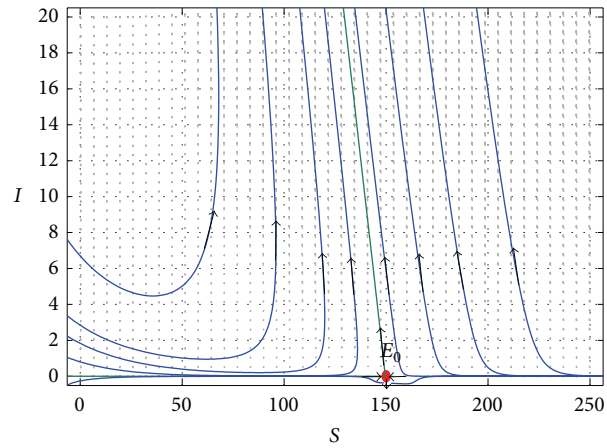


Figure 2: The disease-free equilibrium is unstable when $R_0 > 1$, with the parameter values $A = 15$, $d = 0.1$, $k = 0.1$, $\mu = 0.01$, $\lambda = 0.01$, $\alpha = 1$, $r = 0.8$, and $\varepsilon = 0.1$.

2. Equilibria and Backward Bifurcation

2.1. Disease-Free Equilibrium. Obviously, system (4) has a disease-free equilibrium $E_0 = (A/d, 0)$. The Jacobian matrix of (4) at E_0 is

$$M(E_0) = \begin{pmatrix} -d & \dfrac{-\lambda A}{d} + \varepsilon + r \\ 0 & \dfrac{\lambda A}{d} - (d + \varepsilon + \mu) - r \end{pmatrix}. \tag{5}$$

By using the next generation matrix of (4), we get the basic reproduction number $R_0 = \lambda A/d(d + \varepsilon + r + \mu)$. $M(E_0)$ has negative eigenvalues if $\lambda A/d - (d + \varepsilon + \mu) - r < 0$. Then we have the following result.

Theorem 1. *The disease-free equilibrium E_0 is locally asymptotically stable when $R_0 < 1$ (see Figure 1) and is unstable when $R_0 > 1$ (see Figure 2).*

2.2. Endemic Equilibria. An endemic equilibrium always satisfies

$$\begin{aligned} A - dS - \lambda SI + \varepsilon I + \frac{rI}{1 + \alpha I} &= 0, \\ \lambda SI - (d + \varepsilon + \mu) I - \frac{rI}{1 + \alpha I} &= 0. \end{aligned} \tag{6}$$

In view of $dS + (d + \mu)I = A$, we get $S = (A - (d + \mu)I)/d$ and substitute it into the second equation of (6). When $I \neq 0$, we obtain

$$\lambda \left[\frac{A - (d + \mu) I}{d} \right] - (d + \varepsilon + \mu) - \frac{r}{1 + \alpha I} = 0. \tag{7}$$

Then we have an equation of the form

$$aI^2 + bI + c = 0, \tag{8}$$

with

$$\begin{aligned} a &= \alpha\lambda (d + \mu), \\ b &= \lambda (d + \mu) + \alpha d (d + \varepsilon + \mu) - \alpha\lambda A, \\ c &= d (d + \varepsilon + r + \mu) - \lambda A. \end{aligned} \tag{9}$$

This equation may admit positive solution

$$I_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \qquad I_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}. \tag{10}$$

Obviously, if $R_0 = 1$, then $c = 0$, if $R_0 > 1$, then $c < 0$, and if $R_0 < 1$, then $c > 0$. From (8), it is obvious that we have the following results.

Theorem 2. *The following results hold.*

(H_1) *Let $\alpha = 0$. Equation (8) is a linear equation with a unique solution $I = -c/b$. Then the system (4) has a unique endemic equilibrium when $R_0 > 1$ and has no endemic equilibrium when $R_0 \leq 1$.*

(H_2) *Let $\alpha > 0$. If $b > 0$, system (4) has a unique endemic equilibrium when $R_0 > 1$ and no endemic equilibrium when $R_0 \leq 1$.*

(H_3) *Let $\alpha > 0$. If $b < 0$, system (4) has a unique endemic equilibrium when $R_0 \geq 1$, no endemic equilibrium when $R_0 < R_0^*$, and two endemic equilibria E_1 and E_2 when $R_0^* \leq R_0 < 1$. When $R_0 = R_0^*$ and $E_1 = E_2$, one has $b^2 - 4ac = 0$ which is equivalent to*

$$\begin{aligned} R_0 &= \frac{4a\lambda A}{b^2 + 4a\lambda A} \\ &= \frac{4\lambda^2 \alpha (d + \mu) A}{\left[\lambda (d + \mu) + \alpha d (d + \varepsilon + \mu) - \alpha\lambda A\right]^2 + 4\lambda^2 \alpha (d + \mu) A}. \end{aligned} \tag{11}$$

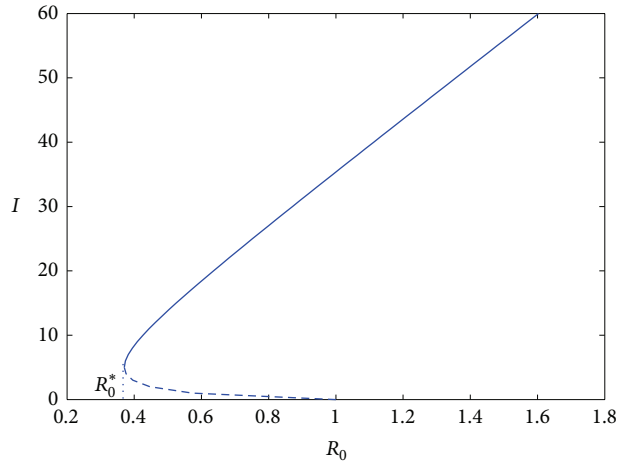


FIGURE 3: The figure of infective sizes at equilibria versus R_0 when $\alpha = 1$, $d = 0.1$, $\mu = 0.01$, $\lambda = 0.05$, $r = 2$, and $\varepsilon = 0.1$.

We define the right-hand side of (11) as

$$R_0^* \triangleq \frac{4\lambda^2\alpha(d+\mu)A}{\left[\lambda(d+\mu)+\alpha d(d+\varepsilon+\mu)-\alpha\lambda A\right]^2+4\lambda^2\alpha(d+\mu)A}. \tag{12}$$

Therefore, according to the qualitative approach recently proposed by [31] which is based on the analysis of the equilibria curve in the neighboring of the critical threshold $R_0 = 1$, we have the following theorem.

Theorem 3. *If $\alpha > 0$, $b < 0$, then system (4) has a backward bifurcation at $R_0 = 1$ (see Figure 3).*

In order to verify the bifurcation curve (the graph of I as a function of R_0) in Figure 3, we think of r as a variable with the other parameters as constant. Through implicit differentiation of (8) with respect to r, we get

$$(2aI+b)\frac{dI}{dr} = -d < 0. \tag{13}$$

From (13), we know that the sign of dI/dr is opposite to that of $2aI + b$. And from the definition of R_0 we know that R_0 decreases when r increases. It implies that the bifurcation curve has positive slope at equilibrium values with $2aI+b > 0$ and negative slope at equilibrium values with $2aI + b < 0$. If there is no backward bifurcation at $R_0 = 1$, then the unique endemic equilibrium for $R_0 > 1$ satisfies

$$2aI + b = \sqrt{b^2 - 4ac} > 0, \tag{14}$$

and the bifurcation curve has positive slope at all points where $I > 0$. If there is a backward bifurcation at $R_0 = 1$, then there is an interval on which there are two endemic equilibria given by

$$2aI + b = \pm\sqrt{b^2 - 4ac}. \tag{15}$$

The bifurcation curve has negative slope at the smaller one and positive slope at the larger one. Thus the bifurcation curve is as shown in Figure 3.

Under the conditions of Theorem 3, if a backward bifurcation takes place, we can see from Figure 3 that there is a critical value R_0^* at the turning point. In this case, the disease will not die out when $R_0 < 1$. However, the disease will die out when $R_0 < R_0^*$. Therefore, the critical value R_0^* can be taken as a new threshold for the control of the disease.

In the following, we give an explicit criterion of a backward bifurcation at $R_0 = 1$.

For convenience, we define

$$\alpha_0 := \frac{\lambda(d+\mu)}{dr}. \tag{16}$$

Corollary 4. *When $\alpha > \alpha_0$, system (4) has a backward bifurcation at $R_0 = 1$.*

Proof. When $R_0 \le 1 \Leftrightarrow c \ge 0$,

$$\lambda A \le d(d+\varepsilon+\mu+r). \tag{17}$$

The condition $b < 0$ is equivalent to

$$\lambda(d+\mu)+\alpha d(d+\varepsilon+\mu) < \alpha\lambda A. \tag{18}$$

From (17) and (18), we get $\lambda(d+\mu)+\alpha d(d+\varepsilon+\mu) < \alpha d(d+\varepsilon+r+\mu)$, which reduces to

$$\alpha > \frac{\lambda(d+\mu)}{dr} \triangleq \alpha_0. \tag{19}$$

It means that α is big enough to lead a backward bifurcation with two endemic equilibria when $R_0 < 1$. Therefore, the proof is complete. □

Next, we consider the local stability of the unique endemic equilibrium when $R_0 > 1$.

Theorem 5. *When $R_0 > 1$ and $0 \le \alpha < \lambda/r$, the unique endemic equilibrium E^* is locally asymptotically stable.*

Proof. Firstly, from Theorem 2, we can know that system (4) has a unique endemic equilibrium E^* when $R_0 > 1$. Moreover, the Jacobian matrix of system (4) is

$$M = \begin{pmatrix} -d-\lambda I & -\lambda S+\varepsilon+\dfrac{r}{1+\alpha I}-\dfrac{r\alpha I}{(1+\alpha I)^2} \\ \lambda I & \lambda S-(d+\varepsilon+\mu)-\dfrac{r}{1+\alpha I}+\dfrac{r\alpha I}{(1+\alpha I)^2} \end{pmatrix}. \tag{20}$$

From the second equation of (6), we have

$$-\lambda S+\varepsilon+\frac{r}{1+\alpha I} = -d-\mu. \tag{21}$$

From (21), the Jacobian matrix M reduces to

$$M = \begin{pmatrix} -d-\lambda I & -d-\mu-\dfrac{r\alpha I}{(1+\alpha I)^2} \\ \lambda I & \dfrac{r\alpha I}{(1+\alpha I)^2} \end{pmatrix}. \tag{22}$$

We obtain

$$\det(M) = \frac{I}{(1+\alpha I)^2}\left[\lambda(d+\mu)(1+\alpha I)^2 - r\alpha d\right]. \quad (23)$$

In fact, there is $\lambda(d+\mu)(1+\alpha I)^2 > \lambda d$. Since $R_0 > 1$ and $0 \le \alpha < \lambda/r$, we have

$$\lambda(d+\mu)(1+\alpha I)^2 - r\alpha d > \lambda d - r\alpha d > 0. \quad (24)$$

So we get $\det(M) > 0$. The trace of M is given by

$$\operatorname{tr}(M) = \frac{1}{(1+\alpha I)^2}\left[-(d+\lambda I)(1+\alpha I)^2 + r\alpha I\right]. \quad (25)$$

In the same way as the above calculation of (24), we have

$$-(d+\lambda I)(1+\alpha I)^2 + r\alpha I < -\lambda I + r\alpha I < 0. \quad (26)$$

So we get $\operatorname{tr}(M) < 0$. The proof is complete. □

Now we consider the case that there are two endemic equilibria E_1 and E_2; let M_i be the Jacobian matrix at E_i, $i = 1, 2$.

Theorem 6. *The endemic equilibrium E_1 is a saddle whenever it exists.*

Proof. Since $I_1 = (-b - \sqrt{b^2 - 4ac})/2a$ and $\Delta = b^2 - 4ac$, we have $I_1 = (-b - \sqrt{\Delta})/2a$. Thus

$$\begin{aligned}
&\det(M_1)\\
&= \frac{I_1}{(1+\alpha I_1)^2}\left[\lambda(d+\mu)(1+\alpha I_1)^2 - r\alpha d\right]\\
&\triangleq \frac{I_1}{(1+\alpha I_1)^2}\psi(I_1).
\end{aligned} \quad (27)$$

From (H_3) of Theorem 2, we can know that if $\alpha > 0$, $b < 0$, and $R_0^* < R_0 < 1$, then E_1 exists. Hence, we can get $\psi(0) = \lambda(d+\mu) - r\alpha d < 0$ and $\psi'(I_1) = 2\lambda(d+\mu)(1+\alpha I_1)\alpha > 0$. It follows that there exists a unique $I^* > 0$ such that

$$\begin{aligned}
&\psi(I_1) = 0, \quad \text{when } I_1 = I^*,\\
&\psi(I_1) < 0, \quad \text{when } 0 < I_1 < I^*,\\
&\psi(I_1) > 0, \quad \text{when } I_1 > I^*,
\end{aligned} \quad (28)$$

where

$$I^* = \sqrt{\frac{rd}{\alpha\lambda(d+\mu)}} - \frac{1}{\alpha}. \quad (29)$$

On the other hand,

$$\begin{aligned}
I_1 &= -\frac{b}{2a} - \frac{\sqrt{\Delta}}{2a}\\
&= \frac{\alpha\lambda A - \lambda(d+\mu) - \alpha d(d+\varepsilon+\mu)}{2\alpha\lambda(d+\mu)} - \frac{\sqrt{\Delta}}{2a}\\
&= \sqrt{\frac{rd}{\alpha\lambda(d+\mu)}} - \frac{1}{\alpha} + \frac{1}{\alpha} - \sqrt{\frac{rd}{\alpha\lambda(d+\mu)}}\\
&\quad + \frac{\alpha\lambda A - \lambda(d+\mu) - \alpha d(d+\varepsilon+\mu)}{2\alpha\lambda(d+\mu)} - \frac{\sqrt{\Delta}}{2a}\\
&= I^* + \Big(\Big[\lambda(d+\mu) + \alpha\lambda A - 2\sqrt{rd\alpha\lambda(d+\mu)}\\
&\qquad - \alpha d(d+\varepsilon+\mu)\Big] - \sqrt{\Delta}\Big) \times \left(2\alpha\lambda(d+\mu)\right)^{-1},
\end{aligned} \quad (30)$$

$$\begin{aligned}
\Delta &= \left\{\lambda(d+\mu) - \left[\alpha\lambda A - \alpha d(d+\varepsilon+\mu)\right]\right\}^2\\
&\quad - 4\alpha\lambda(d+\mu)\left[d(d+\varepsilon+r+\mu) - \lambda A\right]\\
&= \left\{\lambda(d+\mu) + \left[\alpha\lambda A - \alpha d(d+\varepsilon+\mu)\right]\right.\\
&\qquad \left. - 2\sqrt{rd\alpha\lambda(d+\mu)}\right\}^2 - 4rd\alpha\lambda(d+\mu)\\
&\quad - 4\lambda(d+\mu)\left[\alpha\lambda A - \alpha d(d+\varepsilon+\mu)\right]\\
&\quad + 4\sqrt{rd\alpha\lambda(d+\mu)}\left\{\lambda(d+\mu) + \left[\alpha\lambda A - \alpha d(d+\varepsilon+\mu)\right]\right\}\\
&\quad - 4\alpha\lambda(d+\mu)\left[-\lambda A + d(d+\varepsilon+r+\mu)\right]\\
&\triangleq \left\{\lambda(d+\mu) + \left[\alpha\lambda A - \alpha d(d+\varepsilon+\mu)\right]\right.\\
&\qquad \left. - 2\sqrt{rd\alpha\lambda(d+\mu)}\right\}^2 + P,
\end{aligned} \quad (31)$$

where

$$\begin{aligned}
P = &-8rd\alpha\lambda(d+\mu) + 4\sqrt{rd\alpha\lambda(d+\mu)}\\
&\times\left\{\lambda(d+\mu) + \left[\alpha\lambda A - \alpha d(d+\varepsilon+\mu)\right]\right\}.
\end{aligned} \quad (32)$$

In the following we will show that $P > 0$. Since $R_0^* < R_0$, we have

$$\begin{aligned}
&\frac{4\alpha\lambda(d+\mu)}{\left[\lambda(d+\mu) + \alpha d(d+\varepsilon+\mu) - \alpha\lambda A\right]^2 + 4\alpha\lambda^2(d+\mu)A}\\
&\quad < \frac{1}{d(d+\varepsilon+r+\mu)};
\end{aligned} \quad (33)$$

that is,

$$\begin{aligned}&\left[\lambda(d+\mu)+\alpha d(d+\varepsilon+\mu)-\alpha\lambda A\right]^2\\&\quad>4\alpha\lambda(d+\mu)d(d+\varepsilon+r+\mu)-4\alpha\lambda^2(d+\mu)A.\end{aligned}\tag{34}$$

Therefore,

$$\begin{aligned}&\{\lambda(d+\mu)+[\alpha\lambda A-\alpha d(d+\varepsilon+\mu)]\}^2\\&\quad=\{\lambda(d+\mu)-[\alpha\lambda A-\alpha d(d+\varepsilon+\mu)]\}^2\\&\qquad+4\lambda(d+\mu)[\alpha\lambda A-\alpha d(d+\varepsilon+\mu)]\\&\quad>4\alpha\lambda(d+\mu)d(d+\varepsilon+r+\mu)-4\alpha\lambda^2(d+\mu)A\\&\qquad+4\lambda(d+\mu)[\alpha\lambda A-\alpha d(d+\varepsilon+\mu)]\\&\quad=4rd\alpha\lambda(d+\mu).\end{aligned}\tag{35}$$

Obviously, we have $P > 0$. From (30), one has $I_1 < I^*$. So we get $\det(M_1) < 0$. Hence the endemic equilibrium E_1 is a saddle. The proof is complete. □

In order to explore the stability of the endemic equilibrium E_2, define

$$\begin{aligned}m_1&:=\left(2d\alpha a^2+\lambda a^2-\varepsilon\alpha a^2-ac\lambda\alpha^2\right)\\&\quad-b\left(2\lambda\alpha a+d\alpha^2 a-b\lambda\alpha^2\right),\\m_2&:=a^2d-c\left(2\lambda\alpha a+d\alpha^2 a-b\lambda\alpha^2\right).\end{aligned}\tag{36}$$

Theorem 7. *If $\eta > 0$, then endemic equilibrium E_2 is locally asymptotically stable; if $\eta < 0$, then endemic equilibrium E_2 is unstable, where $\eta := 2am_2 + m_1(\sqrt{b^2-4ac}-b)$.*

Proof. Consider

$$\begin{aligned}\det(M_2)&=\frac{I_2}{(1+\alpha I_2)^2}\left[\lambda(d+\mu)(1+\alpha I_2)^2-\varepsilon\alpha d\right]\\&=\frac{I_2}{(1+\alpha I_2)^2}\times\psi(I_2).\end{aligned}\tag{37}$$

By carrying out arguments similar to that of Theorem 6, we have $I_2 > I^*$. Therefore, $\det(M_2) > 0$. In addition, we have

$$\begin{aligned}\operatorname{tr}(M_2)&=-\frac{(d+\lambda I_2)(1+\alpha I_2)^2-\varepsilon\alpha I_2}{(1+\alpha I_2)^2}\\&=-\frac{\lambda\alpha^2 I_2^3+\left(2\lambda\alpha+d\alpha^2\right)I_2^2+(2d\alpha+\lambda-\varepsilon\alpha)I_2+d}{(1+\alpha I_2)^2},\end{aligned}\tag{38}$$

and then $\operatorname{sgn}(\operatorname{tr}(M_2)) = -\operatorname{sgn}(G(I_2))$, where

$$G(x)=\lambda\alpha^2x^3+\left(2\lambda\alpha+d\alpha^2\right)x^2+(2d\alpha+\lambda-\varepsilon\alpha)x+d.\tag{39}$$

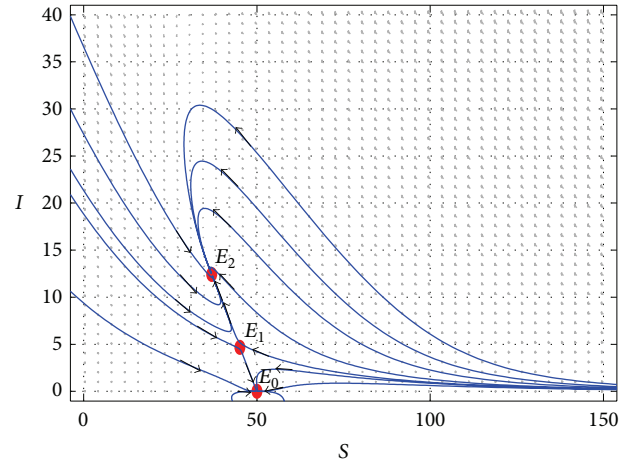


Figure 4: One region of disease persistence and one region of disease extinction when $A = 10$, $\alpha = 1$, $d = 0.1$, $\mu = 0.01$, $\lambda = 0.01$, $r = 0.1$, and $\varepsilon = 0.1$.

Using the expression of $m_1 = (2d\alpha a^2 + \lambda a^2 - \varepsilon\alpha a^2 - ac\lambda\alpha^2) - b(2\lambda\alpha a + d\alpha^2 a - b\lambda\alpha^2)$ and $m_2 = a^2 d - c(2\lambda\alpha a + d\alpha^2 a - b\lambda\alpha^2)$, one has

$$G(I_2)=\left(aI_2^2+bI_2+c\right)\varphi_0+\frac{m_1I_2+m_2}{a^2},\tag{40}$$

where φ_0 is a first degree polynomial of I_2. Since $aI_2^2+bI_2+c = 0$, $\operatorname{sgn}(\operatorname{tr}(M_2)) = -\operatorname{sgn}(G(I_2)) = -\operatorname{sgn}(m_1I_2 + m_2)$. From the expression of I_2, we have

$$\begin{aligned}&\operatorname{sgn}(m_1I_2+m_2)\\&\quad=\operatorname{sgn}\left(2am_2+m_1\left(\sqrt{b^2-4ac}-b\right)\right)\triangleq\operatorname{sgn}(\eta).\end{aligned}\tag{41}$$

Thus, E_2 is locally asymptotically stable if $\eta > 0$ and E_2 is unstable if $\eta < 0$. The proof is complete. □

From the above discussion, we have the following conclusion. If two endemic equilibria E_1 and E_2 exist, the stable manifolds of the saddle E_1 split R_+^2 into two regions. The disease is persistent in the upper region and dies out in the lower region (see Figure 4).

3. Global Analysis

Firstly, we consider the global stability of the disease-free equilibrium E_0. Let $N = S + I$ be the total population size. Now we note that the equation for total population is given by $dN/dt = A - dS - (d+\mu)I \le A - dN$. It follows that $\lim_{t\to+\infty} N(t) \le A/d$. Let

$$\mathfrak{R}=\left\{(S,I)\in R_+^2:S+I\le\frac{A}{d}\right\}\tag{42}$$

which is positive invariant with respect to system (4).

Theorem 8. *If $R_0 < R_0^*$, the disease-free equilibrium $E_0(A/d, 0)$ is globally asymptotically stable; that is, the disease dies out.*

Proof. Suppose $R_0 < R_0^*$. From the (H_3) of Theorem 2, we know that the model has no endemic equilibrium. From the corollary of Poincaré-Bendixson theorem [32], we know that there is no periodic orbit in $\mathfrak{R}$ as there is a disease-free equilibrium in $\mathfrak{R}$. Since $\mathfrak{R}$ is a bounded positively invariant region and E_0 is the only equilibrium in $\mathfrak{R}$, the local stability of E_0 implies that every solution initiating in $\mathfrak{R}$ approaches E_0. Thus, the disease-free equilibrium E_0 is globally asymptotically stable. The proof is complete. □

Now we analyze the global dynamics of the endemic equilibrium when $R_0 > 1$.

Theorem 9. *If $R_0 > 1$ and $0 \le \alpha < \lambda/r$, the system (4) has no limit cycles.*

Proof. We use Dulac theorem to exclude the limit cycle. Let

$$\begin{aligned} P(S,I) &= A - dS - \lambda SI + \varepsilon I + \frac{rI}{1+\alpha I}, \\ Q(S,I) &= \lambda SI - (d+\varepsilon+\mu) I - \frac{rI}{1+\alpha I}, \end{aligned} \tag{43}$$

and take the Dulac function

$$D = \frac{1}{I}. \tag{44}$$

According to $0 \le \alpha < \lambda/r$, we can get

$$\begin{aligned} &\frac{\partial (PD)}{\partial S} + \frac{\partial (QD)}{\partial I} \\ &= -\frac{d}{I} - \lambda + \frac{\alpha r}{(1+\alpha I)^2} \\ &< -\frac{d}{I} - \lambda + \frac{\lambda}{(1+\alpha I)^2} \\ &= \frac{1}{I(1+\alpha I)^2} \left\{ -d(1+\alpha I)^2 - \lambda I \left[(1+\alpha I)^2 - 1 \right] \right\} \\ &< 0. \end{aligned} \tag{45}$$

Hence, the system (4) has no limit cycles. The proof is complete. □

Therefore, we obtain the global result of the unique endemic equilibrium.

Theorem 10. *If $R_0 > 1$ and $0 \le \alpha < \lambda/r$, the unique endemic equilibrium E^* is globally asymptotically stable (see Figure 5).*

4. Hopf Bifurcation

In this section, we study the Hopf bifurcation of system (4). From the above discussion, we know that there is no closed orbit surrounding E_0 or E_1 because the S-axis is invariant with respect to system (4) and E_1 is always a saddle. Therefore, Hopf bifurcation can only occur at E_2. Set

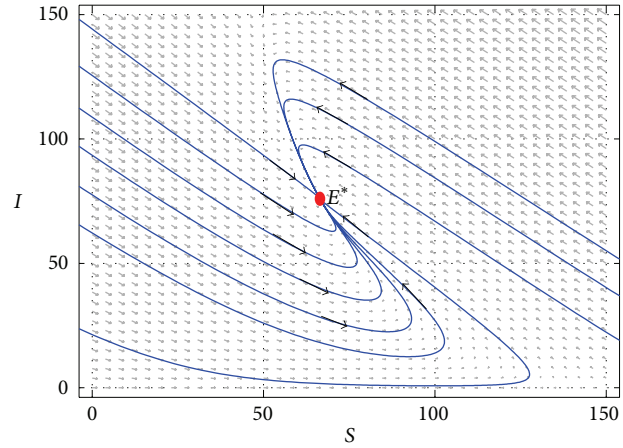


Figure 5: The unique endemic equilibrium E^* is globally asymptotically stable when $A = 15$, $\alpha = 0.01$, $d = 0.1$, $\mu = 0.01$, $\lambda = 0.01$, $r = 0.8$, and $\varepsilon = 0.1$.

$$\begin{aligned} \sigma = &\left[-\lambda\alpha - \frac{c_1 (c_2 + 2c_4)}{a_{12}} \right] D^4 \\ &+ \left[-\lambda\alpha a_{11}^2 - 2\lambda\alpha a_{11} a_{12} + \left(c_2 - \frac{2a_{11}c_1}{a_{12}} \right) \right. \\ &\times \left(a_{11}c_2 - \frac{a_{11}^2 c_1}{a_{12}} \right) \\ &- \left(\frac{2a_{11}^2 c_1}{a_{12}} - a_{11}c_2 + 2c_4 a_{11} - c_3 a_{12} \right) \\ &\left. \times \left(c_4 - \frac{a_{11}c_1}{a_{12}} \right) \right] D^2 \\ &+ a_{11} \left(\frac{a_{11}^2 c_1}{a_{12}} - a_{11}c_2 + c_4 a_{11} - c_3 a_{12} \right) \\ &\times (a_{12}c_3 - 2a_{11}c_4 - a_{11}c_2), \end{aligned} \tag{46}$$

where a_{ij} $(i, j = 1, 2)$, c_k $(k = 1, 2, 3, 4)$, and D are defined by (50) and (52).

Theorem 11. *System (4) undergoes a Hopf bifurcation if $\eta = 0$. Moreover, if $\sigma < 0$, there is a family of stable period orbits of system (4) as η decreases from 0; that is, a supercritical Hopf bifurcation occurs; if $\sigma > 0$, there is a family of unstable period orbits of system (4) as η increases from 0; that is, a subcritical Hopf bifurcation occurs.*

Proof. The proof of Theorem 7 shows that $\mathrm{tr}(M_2) = 0$ if and only if $\eta = 0$, and $\det(M_2) > 0$ when E_2 exists. Therefore, the eigenvalues of M_2 are a pair of pure imaginary roots if and only if $\eta = 0$. The direct calculations show that

$$\left. \frac{d(\mathrm{tr}(M_2))}{d\eta} \right|_{\eta=0} = -\frac{1}{2a^3 (1+\alpha I_2)^2} < 0. \tag{47}$$

By Theorem 3.4.2 in [24], $\eta = 0$ is the Hopf bifurcation point for system (4).

To be concise in notations, rescale (4) by $\tau = t/(1+\alpha I)$. For simplicity, we still use t instead of τ. Then we obtain

$$\begin{aligned}\frac{dS}{dt} &= (A - dS - \lambda SI + \varepsilon I)(1+\alpha I) + rI,\\ \frac{dI}{dt} &= \lambda SI(1+\alpha I) - (d+\varepsilon+\mu) I(1+\alpha I) - rI.\end{aligned} \tag{48}$$

Let $x = S - S_2$ and $y = I - I_2$; then (48) becomes

$$\begin{aligned}\frac{dx}{dt} &= a_{11}x + a_{12}y + c_1y^2 + c_2xy - \lambda\alpha xy^2,\\ \frac{dy}{dt} &= a_{21}x + a_{22}y + c_3xy + c_4y^2 + \lambda\alpha xy^2,\end{aligned} \tag{49}$$

where

$$\begin{aligned}a_{11} &= -(d+\lambda I_2)(1+\alpha I_2),\\ a_{12} &= (-\lambda S_2 + r)(1+\alpha I_2)\\ &\quad + \alpha(A - dS_2 - \lambda S_2 I_2 + rI_2) + \varepsilon,\\ a_{21} &= \lambda I_2(1+\alpha I_2),\\ a_{22} &= [\lambda S_2 - (d+r+\mu)](1+\alpha I_2)\\ &\quad + \alpha[\lambda S_2 I_2 - (d+r+\mu) I_2] - \varepsilon,\\ c_1 &= \alpha(-\lambda S_2 + r),\\ c_2 &= \alpha(-d - \lambda I_2) - \lambda(1+\alpha I_2),\\ c_3 &= \lambda(1+\alpha I_2) + \lambda I_2\alpha,\\ c_4 &= [\lambda S_2 - (d+r+\mu)]\alpha.\end{aligned} \tag{50}$$

Let E^* denote the origin of x-y plane. Since $E_2 = (S_2, I_2)$ satisfies (6), we obtain

$$\begin{aligned}&\det(M(E^*))\\ &= -(d+\lambda I_2)\alpha\varepsilon I_2 + (d+\mu)\left(1+\alpha I_2^2\right)^2 \lambda I_2 + \alpha\lambda\varepsilon I_2^2\\ &= I_2 \times \psi(I_2).\end{aligned} \tag{51}$$

From the proof of Theorem 7, it follows that $\psi(I_2)$ is always positive. It is easy to verify that $a_{11} + a_{22} = 0$ if and only if $\eta = 0$. Set

$$D = \sqrt{\det(M(E^*))} \tag{52}$$

and let $u = -x$ and $v = (a_{11}/D)x + (a_{12}/D)y$; then the normal form of (48) for Hopf bifurcation reads

$$\begin{aligned}\frac{du}{dt} &= -Dv + f(u,v),\\ \frac{dv}{dt} &= Du + g(u,v),\end{aligned} \tag{53}$$

where

$$\begin{aligned}f(u,v) &= \left(\frac{a_{11}}{a_{12}}c_2 - \frac{a_{11}^2}{a_{12}^2}c_1\right)u^2 - \frac{D^2c_1}{a_{12}^2}v^2\\ &\quad + \left(\frac{Dc_2}{a_{12}} - 2\frac{Da_{11}c_1}{a_{12}^2}\right)uv - \frac{\lambda\alpha a_{11}^2}{a_{12}^2}u^3\\ &\quad - 2\frac{Da_{11}\lambda\alpha}{a_{12}^2}u^2v - \frac{D^2\lambda\alpha}{a_{12}^2}uv^2,\\ g(u,v) &= \frac{a_{11}}{D}\left(\frac{a_{11}^2}{a_{12}^2}c_1 - \frac{a_{11}}{a_{12}}c_2 + \frac{a_{11}}{a_{12}}c_4 - c_3\right)u^2\\ &\quad + \left(\frac{Da_{11}c_1}{a_{12}^2} + \frac{Dc_4}{a_{12}}\right)v^2\\ &\quad + \left(2\frac{a_{11}^2}{a_{12}^2}c_1 - \frac{a_{11}}{a_{12}}c_2 + 2\frac{a_{11}}{a_{12}}c_4 - c_3\right)uv\\ &\quad + \frac{\lambda\alpha a_{11}^2}{a_{12}D}\left(\frac{1}{a_{12}} - 1\right)u^3 + \frac{2\lambda\alpha a_{11}}{a_{12}}\left(\frac{a_{11}}{a_{12}} - 1\right)u^2v\\ &\quad + \frac{D\lambda\alpha}{a_{12}}\left(\frac{a_{11}}{a_{12}} - 1\right)uv^2.\end{aligned} \tag{54}$$

Now, we evaluate the first Lyapunov coefficient Γ of system (4) as follows:

$$\begin{aligned}\Gamma &= \frac{1}{16}[f_{uuu} + f_{uvv} + g_{uuv} + g_{vvv}] + \frac{1}{16D}\\ &\quad \times [f_{uv}(f_{uu} + f_{vv}) - g_{uv}(g_{uu} + g_{vv}) - f_{uu}g_{uu} + f_{vv}g_{vv}],\end{aligned} \tag{55}$$

where f_{uv} denotes $(\partial^2 f/\partial u\partial v)(0,0)$ and so forth. Then

$$\begin{aligned}\Gamma &= \frac{1}{8a_{12}^2D^2}\left\{\left[-\lambda\alpha - \frac{c_1(c_2+2c_4)}{a_{12}}\right]D^4\right.\\ &\quad + \left[-\lambda\alpha a_{11}^2 - 2\lambda\alpha a_{11}a_{12} + \left(c_2 - \frac{2a_{11}c_1}{a_{12}}\right)\right.\\ &\quad \times \left(a_{11}c_2 - \frac{a_{11}^2c_1}{a_{12}}\right)\\ &\quad - \left(\frac{2a_{11}^2c_1}{a_{12}} - a_{11}c_2 + 2c_4a_{11} - c_3a_{12}\right)\\ &\quad \left.\times\left(c_4 - \frac{a_{11}c_1}{a_{12}}\right)\right]D^2\\ &\quad + a_{11}\left(\frac{a_{11}^2c_1}{a_{12}} - a_{11}c_2 + c_4a_{11} - c_3a_{12}\right)\\ &\quad \left.\times(a_{12}c_3 - 2a_{11}c_4 - a_{11}c_2)\right\}\\ &= \frac{\sigma}{8a_{12}^2D^2}.\end{aligned} \tag{56}$$

Obviously, the sign of Γ is determined by σ. By Theorem 3.4.2 and (3.4.11) in [24], the rest of the claims in Theorem 11 are valid. The proof is complete. □

5. Bogdanov-Takens Bifurcations

The purpose of this section is to study the Bogdanov-Takens bifurcation of (4). Now, we assume the following:

(A_1) $b^2 - 4ac = 0$,

(A_2) $2ad^2 + b\lambda\mu = 0$.

Then (6) admits a unique positive equilibrium $E^* = (S^*, I^*)$, where

$$I^* = -\frac{b}{2a}, \qquad S^* = \frac{A - (d+\mu) I^*}{d}. \tag{57}$$

The Jacobian matrix of system (4) at E^* is

$$M^* = \begin{pmatrix} -d - \lambda I^* & -\lambda S^* + \varepsilon + \dfrac{r}{(1+\alpha I^*)^2} \\ \lambda I^* & \lambda S^* - (d+\varepsilon+\mu) - \dfrac{r}{(1+\alpha I^*)^2} \end{pmatrix}. \tag{58}$$

By (58), we have

$$\begin{aligned} \det(M^*) &= \frac{I^*}{(1+\alpha I^*)^2}\left[(d+\mu)\lambda(1+\alpha I^*)^2 - rd\alpha\right] \\ &= 0, \end{aligned} \tag{59}$$

because of

$$\begin{aligned} (1+\alpha I^*)^2 &= \frac{4a^2 - 4a\alpha b + \alpha^2 b^2}{4a^2} \\ &= \frac{4a^2 - 4a\alpha b + \alpha^2 4ac}{4a^2} \\ &= \frac{a - \alpha b + \alpha^2 c}{a} \\ &= \frac{\alpha^2 rd}{a}. \end{aligned} \tag{60}$$

Furthermore, (A_2) implies that

$$\operatorname{tr}(M^*) = 0. \tag{61}$$

Thus, (A_1) and (A_2) imply that the Jacobian matrix has a zero eigenvalue with multiplicity 2. This suggests that (4) may admit a Bogdanov-Takens singularity.

Theorem 12. *Suppose that (A_1), (A_2), and $2b_1 + b_4 \neq 0$ hold. Then the endemic equilibrium $E^* = (S^*, I^*)$ of (4) is a cusp of codimension 2; that is, it is a Bogdanov-Takens singularity. Here, b_1 and b_4 are defined by (65).*

Proof. Using the transformation of $x = I - I^*$ and $y = S - S^*$, system (4) becomes

$$\begin{aligned} \frac{dx}{dt} &= a_1 x + a_2 y + \lambda xy + \widehat{a}_{11} x^2 - P_1(x), \\ \frac{dy}{dt} &= -\frac{a_1^2}{a_2} x - a_1 y - \lambda xy - \widehat{a}_{11} x^2 + P_1(x), \end{aligned} \tag{62}$$

where $P_1(x)$ is a smooth function of x at least of the third order and

$$\begin{aligned} a_1 &= \lambda S^* - (d+\varepsilon+\mu) - \frac{r}{(1+\alpha I^*)^2} > 0, \\ a_2 &= \lambda I^* > 0, \\ \widehat{a}_{11} &= \frac{r\alpha}{(1+\alpha I^*)^3} > 0. \end{aligned} \tag{63}$$

Set $X = x$, $Y = a_1 x + a_2 y$. Then (62) is transformed into

$$\begin{aligned} \frac{dX}{dt} &= Y + b_1 X^2 + b_2 XY + Q_1(X), \\ \frac{dY}{dt} &= b_3 X^2 + b_4 XY + Q_2(X), \end{aligned} \tag{64}$$

where $Q_i(X)$ are smooth functions of X at least of the third order and

$$\begin{aligned} b_1 &= \widehat{a}_{11} - \frac{\lambda a_1}{a_2}, \\ b_2 &= \frac{\lambda}{a_2}, \\ b_3 &= a_1 \widehat{a}_{11} - \frac{\lambda a_1^2}{a_2} + a_1\lambda - a_2 \widehat{a}_{11}, \\ b_4 &= \frac{\lambda a_1}{a_2} - \lambda. \end{aligned} \tag{65}$$

In order to obtain the canonical normal form, we perform the transformation of variables by

$$u = X - \frac{b_2}{2} X^2, \qquad v = Y + b_1 X^2. \tag{66}$$

Then, we obtain

$$\begin{aligned} \frac{du}{dt} &= v + R_1(u), \\ \frac{dv}{dt} &= b_3 u^2 + (2b_1 + b_4) uv + R_2(u), \end{aligned} \tag{67}$$

where $R_i(u)$ are smooth functions of u at least of the third order. Note that $b_3 < 0$ and $2b_1 + b_4 \neq 0$. It follows from [33–35] that (4) admits a Bogdanov-Takens bifurcation. □

In the following, we will find the versal unfolding in terms of the original parameters in (4). In this way, we will know the approximate saddle-node, Hopf, and homoclinic bifurcation

curves. We choose A and d as bifurcation parameters. Fix $\lambda = \lambda_0$, $\varepsilon = \varepsilon_0$, $r = r_0$, $\alpha = \alpha_0$, and $\mu = \mu_0$. Let $A = A_0 + \theta_1$ and $d = d_0 + \theta_2$, where θ_1 and θ_2 are parameters which vary in a small neighborhood of the origin.

Suppose that $A = A_0$, $d = d_0$, $\lambda = \lambda_0$, $\varepsilon = \varepsilon_0$, $r = r_0$, $\alpha = \alpha_0$, and $\mu = \mu_0$ satisfy (A_1) and (A_2). Consider the following system:

$$\begin{aligned}\frac{dI}{dt} &= \lambda_0 SI - \left(d_0 + \theta_2 + \varepsilon_0 + \mu_0\right) I - \frac{r_0 I}{1 + \alpha_0 I},\\ \frac{dS}{dt} &= A_0 + \theta_1 - \left(d_0 + \theta_2\right) S - \lambda_0 SI + \varepsilon_0 I + \frac{r_0 I}{1 + \alpha_0 I}.\end{aligned} \tag{68}$$

By the transformations of $x = I - I^*$ and $y = S - S^*$, system (68) becomes

$$\begin{aligned}\frac{dx}{dt} &= -\theta_2 I^* + \widehat{c}_1 x + \widehat{c}_2 y + c_{11} x^2 + \lambda_0 xy - w_1(x),\\ \frac{dy}{dt} &= \left(\theta_1 - \theta_2 S^*\right) + \widehat{c}_3 x + \widehat{c}_4 y - c_{11} x^2 - \lambda_0 xy + w_1(x),\end{aligned} \tag{69}$$

where $w_1(x)$ is a smooth function of x at least of the third order and

$$\begin{aligned}\widehat{c}_1 &= \lambda_0 S^* - \left(d_0 + \theta_2 + \varepsilon_0 + \mu_0\right) - \frac{r_0}{\left(1 + \alpha_0 I^*\right)^2},\\ \widehat{c}_2 &= \lambda_0 I^*,\\ \widehat{c}_3 &= -\lambda_0 S^* + \varepsilon_0 + \frac{r_0}{\left(1 + \alpha_0 I^*\right)^2},\\ \widehat{c}_4 &= -\left(d_0 + \theta_2\right) - \lambda_0 I^*,\\ c_{11} &= \frac{r_0 \alpha_0}{\left(1 + \alpha_0 I^*\right)^3}.\end{aligned} \tag{70}$$

Making the change of variables $X = x$, $Y = -\theta_2 I^* + \widehat{c}_1 x + \widehat{c}_2 y + c_{11} x^2 + \lambda_0 xy - w_1(x)$ and rewriting X, Y as x and y, respectively, we have

$$\begin{aligned}\frac{dx}{dt} &= y,\\ \frac{dy}{dt} &= e_0 + e_1 x + e_2 y + e_{11} x^2 + e_{12} xy + e_{22} y^2 + w_2(x, y, \theta),\end{aligned} \tag{71}$$

where $\theta = (\theta_1, \theta_2)$, $w_2(x, y, \theta)$ is a smooth function of x, y, θ_1, and θ_2 at least of the third order, and

$$\begin{aligned}e_0 &= \widehat{c}_2\left(\theta_1 - \theta_2 S^*\right) + \widehat{c}_4 \theta_2 I^*,\\ e_1 &= \lambda_0\left(\theta_1 - \theta_2 S^*\right) + \widehat{c}_2 \widehat{c}_3 - \widehat{c}_1 \widehat{c}_4 - \lambda_0 \theta_2 I^*,\\ e_2 &= \widehat{c}_1 + \widehat{c}_4 + \lambda_0 \frac{\theta_2 I^*}{\widehat{c}_2},\\ e_{11} &= \lambda_0 \widehat{c}_3 - \widehat{c}_4 c_{11} - c_{11} \widehat{c}_2 + \widehat{c}_1 \lambda_0,\\ e_{12} &= -\lambda_0 + 2c_{11} + \lambda_0 \frac{-\widehat{c}_1 \widehat{c}_2 - \lambda_0 \theta_2 I^*}{\widehat{c}_2^2},\\ e_{22} &= \frac{\lambda_0}{\widehat{c}_2}.\end{aligned} \tag{72}$$

Next, introduce a new time variable τ by $dt = (1 - (\lambda_0/\widehat{c}_2)x)d\tau$. Rewriting τ as t, we obtain

$$\begin{aligned}\frac{dx}{dt} &= y\left(1 - \frac{\lambda_0}{\widehat{c}_2} x\right),\\ \frac{dy}{dt} &= \left(1 - \frac{\lambda_0}{\widehat{c}_2} x\right)\\ &\quad \times \left(e_0 + e_1 x + e_2 y + e_{11} x^2 + e_{12} xy + e_{22} y^2 + w_2(x, y, \theta)\right).\end{aligned} \tag{73}$$

Let $X = x$, $Y = y(1 - (\lambda_0/\widehat{c}_2)x)$ and rename X and Y as x and y; we have

$$\begin{aligned}\frac{dx}{dt} &= y,\\ \frac{dy}{dt} &= e_0 + g_1 x + e_2 y + g_{11} x^2 + g_{12} xy + w_3(x, y, \theta),\end{aligned} \tag{74}$$

where $\theta = (\theta_1, \theta_2)$, $w_3(x, y, \theta)$ is a smooth function of x, y, θ_1, and θ_2 at least of the third order, and

$$\begin{aligned}g_1 &= -2e_0 \frac{\lambda_0}{\widehat{c}_2} + e_1,\\ g_{11} &= e_{11} - 2\frac{e_1 \lambda_0}{\widehat{c}_2} + \frac{e_0 \lambda_0^2}{\widehat{c}_2^2},\\ g_{12} &= e_{12} - \frac{e_2 \lambda_0}{\widehat{c}_2}.\end{aligned} \tag{75}$$

Now, we assume that $g_{11} \neq 0$ and $g_{12} \neq 0$ when λ_i are small. Set $x = X - e_2/g_{12}$ and rewrite X as x; we can get

$$\begin{aligned}\frac{dx}{dt} &= y,\\ \frac{dy}{dt} &= f_0 + f_1 x + g_{11} x^2 + g_{12} xy + w_4(x, y, \theta),\end{aligned} \tag{76}$$

where $\theta = (\theta_1, \theta_2)$, $w_4(x, y, \theta)$ is a smooth function of x, y, θ_1, and θ_2 at least of the third order, and

$$\begin{aligned}f_0 &= e_0 - \frac{g_1 e_2}{g_{12}} + \frac{g_{11} e_2^2}{g_{12}^2},\\ f_1 &= g_1 - \frac{2g_{11} e_2}{g_{12}}.\end{aligned} \tag{77}$$

Making the final change of variables by $X = g_{12}^2 x/g_{11}$, $Y = g_{12}^3 y/g_{11}^2$, and $\tau = g_{11}t/g_{12}$ and then denoting them again by x, y, and t, respectively, we obtain

$$\frac{dx}{dt} = y,$$
$$\frac{dy}{dt} = \tau_1(\theta_1,\theta_2) + \tau_2(\theta_1,\theta_2)x + x^2 + xy + w_5(x,y,\theta), \tag{78}$$

where $\theta = (\theta_1,\theta_2)$ and $w_5(x,y,\theta)$ is a smooth function of x, y, θ_1, and θ_2 at least of the third order.

We substitute values of $\alpha_0 = 1$, $\lambda_0 = 1/4$, $d_0 = 1/10$, $\varepsilon_0 = 1/10$, $\mu_0 = 1/100$, $A_0 = 537/500$, and $r_0 = 55/8$ for the above system and these values satisfy conditions (A_1) and (A_2). And we obtain the following equations:

$$\begin{aligned}\tau_1(\theta_1,\theta_2) &= \frac{f_0 g_{12}^4}{g_{11}^3} \\ &= -\frac{237291605}{85184}\theta_1 + \frac{25485118377}{851840}\theta_2 \\ &\quad + O\left(|\theta_1,\theta_2|^2\right), \\ \tau_2(\theta_1,\theta_2) &= \frac{f_1 g_{12}^2}{g_{11}^2} \\ &= -\frac{172225}{1936}\theta_1 + \frac{3983253}{3872}\theta_2 + O\left(|\theta_1,\theta_2|^2\right),\end{aligned} \tag{79}$$

where $g_{11} = -11/500 - (1/16)\theta_1 + (341/800)\theta_2 + (1/4)\theta_2^2 < 0$ and $g_{12} = -83/200 + (1/4)\theta_2 < 0$ in a small neighborhood of $(\theta_1,\theta_2) = (0,0)$. Let

$$J = \begin{pmatrix} \dfrac{\partial\tau_1}{\partial\theta_1} & \dfrac{\partial\tau_1}{\partial\theta_2} \\ \dfrac{\partial\tau_2}{\partial\theta_1} & \dfrac{\partial\tau_2}{\partial\theta_2} \end{pmatrix}. \tag{80}$$

And after simple calculation we obtain that

$$\det(J)|_{\lambda=0} = -\frac{16839398748825}{82458112} \neq 0. \tag{81}$$

Thus, τ_1 and τ_2 are regular maps in a small neighborhood of $(\theta_1,\theta_2) = (0,0)$. By the Bogdanov and Takens bifurcation theorems [36], we obtain the following conclusion.

Theorem 13. *Suppose that A_0, d_0, λ_0, ε_0, r_0, α_0, and μ_0 satisfy (A_1), (A_2), $g_{11} \neq 0$, and $g_{12} \neq 0$ when θ_i are small. Then (4) admits the following bifurcation behavior.*

(1) *There is a saddle-node bifurcation curve* $SN = \{(\theta_1,\theta_2) : 4f_0 g_{11} = f_1^2 + o(|(\theta_1,\theta_2)|^2),\ f_1 \neq 0\} = \{(\theta_1,\theta_2) : (5907/6250)\theta_2 - (11/125)\theta_1 + (2313/400)\theta_1\theta_2 - (5/16)\theta_1^2 - (1028637/40000)\theta_2^2 + o(|(\theta_1,\theta_2)|^2) = 0,\ (47991/16600)\theta_2 - (1/4)\theta_1 + (25/83)\theta_1\theta_2 - (13645/13778)\theta_2^2 + o(|(\theta_1,\theta_2)|^2) \neq 0\}$.

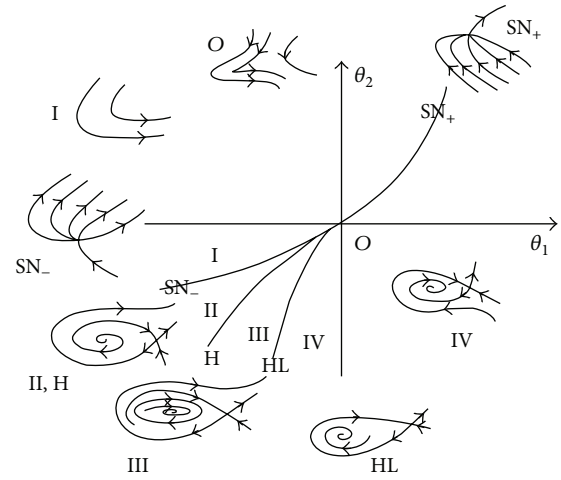


Figure 6: The bifurcation set and the corresponding phase portraits for system (68).

(2) *There is a Hopf bifurcation curve* $H = \{(\theta_1,\theta_2) : f_0 + o(|(\theta_1,\theta_2)|^2) = 0,\ f_1 < 0\} = \{(\theta_1,\theta_2) : -(537/50)\theta_2 + \theta_1 + (50/83)\theta_1\theta_2 - (74667/6889)\theta_2^2 + o(|(\theta_1,\theta_2)|^2) = 0,\ (47991/16600)\theta_2 - (1/4)\theta_1 + (25/83)\theta_1\theta_2 - (13645/13778)\theta_2^2 + o(|(\theta_1,\theta_2)|^2) < 0\}$.

(3) *There is a homoclinic bifurcation curve* $HL = \{(\theta_1,\theta_2) : 25g_{11}f_0 + 6f_1^2 = o(|(\theta_1,\theta_2)|^2),\ f_1 < 0\} = \{(\theta_1,\theta_2) : (5907/1000)\theta_2 - (11/20)\theta_1 + (611979/33200)\theta_1\theta_2 - (19/16)\theta_1^2 - (16075834839/275560000)\theta_2^2 + o(|(\theta_1,\theta_2)|^2) = 0,\ (47991/16600)\theta_2 - (1/4)\theta_1 + (25/83)\theta_1\theta_2 - (13645/13778)\theta_2^2 + o(|(\theta_1,\theta_2)|^2) < 0\}$.

6. Numerical Simulations

When $\alpha = 1$, $\lambda = 1/4$, $d = 1/10$, $\varepsilon = 1/10$, $\mu = 1/100$, $A = 537/500$, and $r = 55/8$, $\sigma > 0$. By applying PPLANE8 and Photoshop software, the (θ_1,θ_2)-plane near the origin is divided into 4 regions by these bifurcation curves as shown in Figure 6. Fix $\theta_1 < 0$ and decrease θ_2 from 0; our conclusions are summarized as follows.

(a) When (θ_1, θ_2) lies in region I, there is no positive equilibrium, which implies that the positive orbits of (4) meet the positive S-axis in finite time, and therefore the disease disappears.

(b) There is a saddle-node point when (θ_1, θ_2) lies on curve SN.

(c) When (θ_1, θ_2) crosses SN into region II, two positive equilibria which are an unstable focus and a saddle appear.

(d) When the parameters lie on the curve H, there is also a saddle and an unstable focus. An unstable limit cycle appears with the parameters crossing H into III.

(e) A homoclinic cycle appears as the parameters passing III into HL through the homoclinic bifurcation.

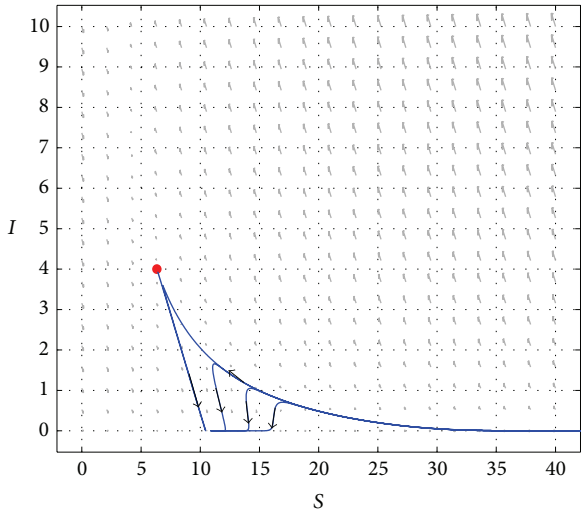


Figure 7: When $(\theta_1, \theta_2) = (0, 0)$, the unique positive equilibrium is a cusp of codimension 2.

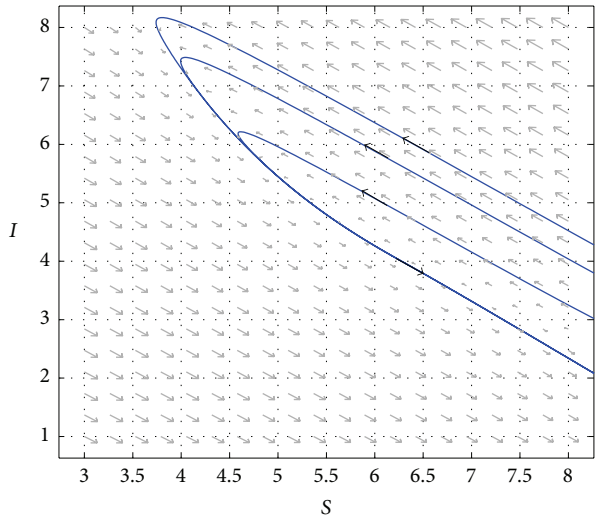


Figure 8: When $(\theta_1, \theta_2) = (-0.1, -0.0093)$ lies in region I, there is no positive equilibrium.

(f) The homoclinic loop breaks with the parameter crossing HL into IV, and a saddle and a stable focus appear.

Furthermore, using PPLANE8, some numerical simulations of system (68) are depicted in Figures 7–10. When $(\theta_1, \theta_2) = (0, 0)$, that is $(A_0, d_0) = (537/500, 1/10)$, there is a unique positive equilibrium $(S^*, I^*) = (317/50, 4)$, which is a cusp of codimension 2 (Figure 7). When $(\theta_1, \theta_2) = (-0.1, -0.0093)$ lies in region I, there is no positive equilibrium (Figure 8). When $(\theta_1, \theta_2) = (-0.02, -0.001863798)$, there is a homoclinic loop (Figure 9). When $(\theta_1, \theta_2) = (-0.02, -0.0020)$ lies in region IV, the homoclinic loop is broken, and there is a stable focus and a saddle (Figure 10).

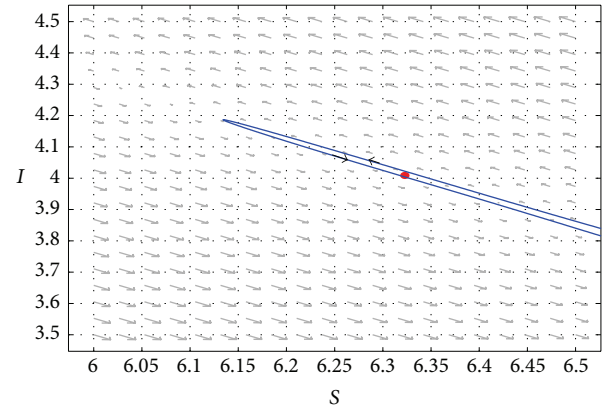


Figure 9: When $(\theta_1, \theta_2) = (-0.02, -0.001863798)$, there is a homoclinic loop.

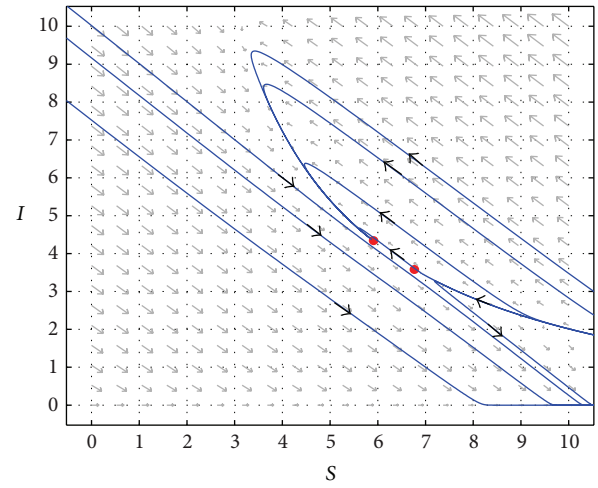


Figure 10: When $(\theta_1, \theta_2) = (-0.02, -0.0020)$ lies in region IV, there is a stable focus and a saddle.

7. Discussion

In this paper, we focus on the bifurcation analysis of an SIS epidemic model with bilinear incidence rate and saturated treatment. Generally speaking, in many epidemic models, the basic reproduction number, which is the key concept in epidemiology, can be decreased below unity to eradicate the disease. However, in our model, the basic reproduction number below 1 is not enough to eradicate the disease. According to our analysis in this paper, we demonstrate not only the global stability of the disease-free equilibrium when $R_0 < R_0^*$ and the local asymptotic stability of the endemic equilibrium E_2 when $R_0^* < R_0 < 1$, but also the global stability of the unique endemic equilibrium E^* when $R_0 > 1$. Moreover, it has been shown in Corollary 4 that backward bifurcations occur if the effect of the infected being delayed for treatment is strong. That is to say, we should get prompt treatment for patients. Through Figure 4, we can see that there is a region such that the disease will persist if the initial position lies in the region and disappear if the initial position lies outside this region. So, in order to restrain the spread

of the disease, governments should take timely measures to control the initial infected individuals in a lower level. In addition, from Theorem 8, we can see that the disease will die out if R_0 is small enough. Therefore, from the definition of R_0, one knows that we can decrease the incidence rate λ and increase the cure rate r so as to eradicate the diseases or to make them controlled in a lower endemic steady state.

The stability analysis of the model equilibria enables us to completely analyze their local bifurcation behavior, such as Hopf, saddle-node, and Bogdanov-Takens bifurcation. By computing the first Lyapunov coefficient, we can determine that the Hopf bifurcation is a supercritical Hopf bifurcation or a subcritical Hopf bifurcation. We also show that, under assumptions (A_1) and (A_2), the model undertakes Bogdanov-Takens bifurcation; that is, there are saddle-node bifurcation, Hopf bifurcation, and homoclinic bifurcation in the system. The normal norm of the Bogdanov-Takens bifurcation is derived in Section 5 which is very helpful to obtain the three kinds of bifurcation curves. By analytical techniques, A and d are chosen as bifurcation parameters and other parameters are fixed, and we easily get a clear picture about the rich dynamics behaviors of our model. Through studying the bifurcations of the SIS epidemic model, we are suggesting that, in order to eradicate the disease, more medical facilities as well as medical professionals are needed and the medical standard needs to be improved as well.

Conflict of Interests

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

Acknowledgments

The research was supported by the National Natural Science Foundation of China (no. 11001041, no. 10926105, no. 11101283, no. 11271065, and no. 11201360), SRFDP (no. 200802001008), and Scientific Research Innovation Project of Shanghai Municipal Education Commission (12YZ173).

References

[1] M. E. Alexander and S. M. Moghadas, "Periodicity in an epidemic model with a generalized non-linear incidence," *Mathematical Biosciences*, vol. 189, no. 1, pp. 75–96, 2004.

[2] M. E. Alexander and S. M. Moghadas, "Bifurcation analysis of SIRS epidemic model with generalized incidence," *SIAM Journal on Applied Mathematics*, vol. 65, no. 5, pp. 1794–1816, 2005.

[3] J. Arino, C. C. McCluskey, and P. van den Driessche, "Global results for an epidemic model with vaccination that exhibits backward bifurcation," *SIAM Journal on Applied Mathematics*, vol. 64, no. 1, pp. 260–276, 2003.

[4] W. R. Derrick and P. van den Driessche, "Homoclinic orbits in a disease transmission model with nonlinear incidence and nonconstant population," *Discrete and Continuous Dynamical Systems B*, vol. 3, no. 2, pp. 299–309, 2003.

[5] F. Brauer, "Backward bifurcations in simple vaccination models," *Journal of Mathematical Analysis and Applications*, vol. 298, no. 2, pp. 418–431, 2004.

[6] K. P. Hadeler and P. van den Driessche, "Backward bifurcation in epidemic control," *Mathematical Biosciences*, vol. 146, no. 1, pp. 15–35, 1997.

[7] Z. X. Hu, S. Liu, and H. Wang, "Backward bifurcation of an epidemic model with standard incidence rate and treatment rate," *Nonlinear Analysis: Real World Applications*, vol. 9, no. 5, pp. 2302–2312, 2008.

[8] J. Hui and D. M. Zhu, "Global stability and periodicity on SIS epidemic models with backward bifurcation," *Computers & Mathematics with Applications*, vol. 50, no. 8-9, pp. 1271–1290, 2005.

[9] Y. Jin, W. D. Wang, and S. W. Xiao, "An SIRS model with a nonlinear incidence rate," *Chaos, Solitons & Fractals*, vol. 34, no. 5, pp. 1482–1497, 2007.

[10] G. H. Li and W. D. Wang, "Bifurcation analysis of an epidemic model with nonlinear incidence," *Applied Mathematics and Computation*, vol. 214, no. 2, pp. 411–423, 2009.

[11] X. Z. Li, W. S. Li, and M. Ghosh, "Stability and bifurcation of an SIS epidemic model with treatment," *Chaos, Solitons amp& Fractals*, vol. 42, no. 5, pp. 2822–2832, 2009.

[12] W. M. Liu, S. A. Levin, and Y. Iwasa, "Influence of nonlinear incidence rates upon the behavior of SIRS epidemiological models," *Journal of Mathematical Biology*, vol. 23, no. 2, pp. 187–204, 1986.

[13] X. B. Liu and L. J. Yang, "Stability analysis of an SEIQV epidemic model with saturated incidence rate," *Nonlinear Analysis: Real World Applications*, vol. 13, no. 6, pp. 2671–2679, 2012.

[14] M. Martcheva and H. R. Thieme, "Progression age enhanced backward bifurcation in an epidemic model with superinfection," *Journal of Mathematical Biology*, vol. 46, no. 5, pp. 385–424, 2003.

[15] S. G. Ruan and W. D. Wang, "Dynamical behavior of an epidemic model with a nonlinear incidence rate," *Journal of Differential Equations*, vol. 188, no. 1, pp. 135–163, 2003.

[16] W. D. Wang and S. G. Ruan, "Bifurcation in an epidemic model with constant removal rate of the infectives," *Journal of Mathematical Analysis and Applications*, vol. 291, no. 2, pp. 775–793, 2004.

[17] Y. L. Tang, D. Q. Huang, S. G. Ruan, and W. N. Zhang, "Coexistence of limit cycles and homoclinic loops in a SIRS model with a nonlinear incidence rate," *SIAM Journal on Applied Mathematics*, vol. 69, no. 2, pp. 621–639, 2008.

[18] P. van den Driessche and J. Watmough, "A simple SIS epidemic model with a backward bifurcation," *Journal of Mathematical Biology*, vol. 40, no. 6, pp. 525–540, 2000.

[19] J. J. Wei and J. A. Cui, "Dynamics of SIS epidemic model with the standard incidence rate and saturated treatment function," *International Journal of Biomathematics*, vol. 5, no. 3, Article ID 1260003, 18 pages, 2012.

[20] W. D. Wang, "Backward bifurcation of an epidemic model with treatment," *Mathematical Biosciences*, vol. 201, no. 1-2, pp. 58–71, 2006.

[21] Z. W. Wang, "Backward bifurcation in simple SIS model," *Acta Mathematicae Applicatae Sinica*, vol. 25, no. 1, pp. 127–136, 2009.

[22] L. Xue and C. Scoglio, "The network level reproduction number for infectious diseases with both vertical and horizontal transmission," *Mathematical Biosciences*, vol. 243, no. 1, pp. 67–80, 2013.

[23] X. Zhang and X. N. Liu, "Backward bifurcation of an epidemic model with saturated treatment function," *Journal of Mathematical Analysis and Applications*, vol. 348, no. 1, pp. 433–443, 2008.

[24] J. Guckenheimer and P. J. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Fields*, Springer, New York, NY, USA, 1996.

[25] X. Zhang and X. N. Liu, "Backward bifurcation and global dynamics of an SIS epidemic model with general incidence rate and treatment," *Nonlinear Analysis: Real World Applications*, vol. 10, no. 2, pp. 565–575, 2009.

[26] M. A. Safi, A. B. Gumel, and E. H. Elbasha, "Qualitative analysis of an age-structured SEIR epidemic model with treatment," *Applied Mathematics and Computation*, vol. 219, no. 22, pp. 10627–10642, 2013.

[27] L. H. Zhou and M. Fan, "Dynamics of an SIR epidemic model with limited medical resources revisited," *Nonlinear Analysis: Real World Applications*, vol. 13, no. 1, pp. 312–324, 2012.

[28] X. Y. Zhou and J. A. Cui, "Analysis of stability and bifurcation for an SEIR epidemic model with saturated recovery rate," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 11, pp. 4438–4450, 2011.

[29] D. Lacitignola, "Saturated treatments and measles resurgence episodes in South Africa: a possible linkage," *Mathematical Biosciences and Engineering: MBE*, vol. 10, no. 4, pp. 1135–1157, 2013.

[30] J. C. Eckalbar and W. L. Eckalbar, "Dynamics of an epidemic model with quadratic treatment," *Nonlinear Analysis: Real World Applications*, vol. 12, no. 1, pp. 320–332, 2011.

[31] F. Brauer, "Backward bifurcations in simple vaccination/treatment models," *Journal of Biological Dynamics*, vol. 5, no. 5, pp. 410–418, 2011.

[32] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential Equations, Dynamical Systems, and an Introduction to Chaos*, Academic Press, New York, NY, USA, 2003.

[33] R. Bogdanov, "Bifurcations of a limit cycle for a family of vector fields on the plan," *Selecta Mathematica*, vol. 1, pp. 373–388, 1981.

[34] R. Bogdanov, "Versal deformations of a singular point on the plan in the case of zero eigen-values," *Selecta Mathematica: Sovietica*, vol. 1, pp. 389–421, 1981.

[35] F. Takens, "Forced oscillations and bifurcation," in *Applications of Global Analysis I*, vol. 3, pp. 1–59, Communications of the Mathematical Institute, Rijksuniversiteit Utrecht, 1974.

[36] Y. A. Kuznetsov, *Elements of Applied Bifurcation Theory*, Springer, New York, NY, USA, 2nd edition, 1998.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 765985, 6 pages
http://dx.doi.org/10.1155/2015/765985

Research Article

Identification of CTQs for Complex Products Based on Mutual Information and Improved Gravitational Search Algorithm

Huaqiang Wang,[1] Lijun Liang,[2] Zhanwen Niu,[1] and Zhen He[1]

[1]*Department of Industrial Engineering, Tianjin University, Tianjin 300072, China*
[2]*College of Management, Tianjin University of Traditional Chinese Medicine, Tianjin 300073, China*

Correspondence should be addressed to Zhanwen Niu; zw.niu@163.com

Received 31 July 2014; Revised 13 November 2014; Accepted 13 November 2014

Academic Editor: Shifei Ding

The identification of CTQs for complex products is the first step to implement quality control. To improve the efficiency and accuracy of CTQs identification, we propose a novel hybrid approach based on mutual information and improved gravitational search algorithm, which has advantages of filter and wrapper. At first, the information relevance and redundancy are measured by mutual information. Then, the improved gravitational search algorithm is used to search the CTQs. Experimentation is carried out using 2 UCI data sets, and the classification capability of CTQs is tested by SVM and tenfold cross validation. The results show that the presented method is verified to be effective and practically applicable.

1. Introduction

The quality characteristics of complex products with complex structure, high-tech, and components highly integrated are high-dimensional characteristics. It is not economically or logistically feasible to control or monitor all of the quality characteristics for the high-dimensional data. Of all these characteristics, some are critical to quality characteristics (CTQs) which determine the quality of the product while others are redundant or insignificant [1–4]. Therefore, identifying the CTQs is the key to monitor, analyze, and improve the quality of complex products. In this paper, we propose a novel hybrid approach based on mutual information and improved gravitational search algorithm (MIGSA) for CTQs identification; see Algorithm 1. The rest of this paper is structured as follows. Section 2 introduces the literature review of CTQs identification. Some basic concepts on mutual information, gravitational search algorithm, and rough sets theory are described briefly in Section 3. The fundamentals of the proposed hybrid approach to CTQs identification of complex products are described in Section 4. The experimental results and discussion are presented in Section 5. Finally, conclusions are given in Section 6.

2. Literature Reviews of CTQs

According to the literatures, the research on CTQs identification can be divided into two sections: CTQs in design phase and CTQs in manufacture process.

For CTQs of design, the design team identifies the voice of the customer (VOC) and determines the CTQs from engineering characteristics based on the relationship of VOCs and engineering characteristics. The typical method of CTQs identification is quality function deployment (QFD) [5, 6]. QFD provides a means of translating customer requirements into the appropriate technical requirements for each stage of product development and production [7]. For example, Zhang et al. established a multiple CTQs optimization model by using QFD technology and successfully extracted various influencing factors for multiple quality characteristics [8]. He et al. put forward an approach to CTQs decomposition from customer requirements into critical technical parameters based on the relational tree [6]. Thornton distinguishes importance degree of obtained characteristics through adding process data [4, 9]. Rowe presented a methodology compatible with Design for Six Sigma (DFSS) for constructing comprehensive

Input: data set D; d, the number of characteristics;
$F = \{X_1, X_2, \ldots, X_d\}$, the set of characteristics; $S = \varnothing$; C, the set of classes;
Values of parameters in GSA: N, the size of agents;
$T_{\max}$, the maximum iteration; T_s, a constant; P_0, the initial positions of agents. V_0, initial velocities of agents; $\text{Fitness}_{\max}$, the best fitness value.
Output: CTQs
Steps:
for $i = 1$ to d, **do**
Compute $I(X_i; C)$
end
Select the characteristic X_i with maximum $I(X_i; C)$
$S = S \cup \{X_i\}$
$F = F - \{X_i\}$
$t = 1$
while $t \leq k$ **do**
for each characteristic $X_j \in F$, do
Compute $J(X_j)$
end
Select the next characteristic X_j with the maximum $J(X_j)$.
$S = S \cup \{X_j\}$
$F = F - \{X_j\}$
$t = t + 1$
end
Initialize P_0
$T = 0$
while $T \leq T_{\max}$ do
Evaluate the fitness value for each agent and find the best fitness. If the best value is met, then stop. Otherwise, update agents' position by (8). If the best global position is no change for T_s consecutive generations, use improving strategies to generate new agents.
$T = T + 1$
end

Algorithm 1: MIGSA.

statistical design and process control specifications for CTQs [10].

During manufacture process, the CTQs can be identified by using sequential experimental design. Shen and Wan proposed controlled sequential factorial design (CSFD) for discrete-event simulation experiments [11]. Rout and Mittal used combined array design of experiment approach to screen the factors influencing the performance of manipulator [12]. Mathieu and Marguet presented an integrated method for CTQs identification based on assembly directed graph [13]. Whitney proposed the concept of a data flow chain (DFC), which was used to analyze the effect factors of CTQs [14]. Variation mode and effect analysis (VMEA) was used in identification of noise factors that caused CTQs to fluctuate and risk coefficient was used to measure the importance of them [15]. Wang et al. proposed CTQs identification method in multistage manufacturing process by combining the partial least squares regression (PLSR) method with the state space model [16]. Wu presented an approach to optimizing the correlated multiple quality characteristics based on the modified double-exponential desirability function [17].

The above methods have been applied successfully in many fields. However, if the number of factors is large (say more than 30) and the output dimension is relatively high, it is hard to obtain the expected characteristics reduction by using the above methods.

According to the characteristic of the manufacturing process, Yan et al. constructed the relationship between the quality characteristics and the class of each product sample and then used information gain (IG) methodology to identify the CTQs of high dimensional complex products [18]. CTQs identification in complex products can be regarded as a feature selection problem. Data mining approach can be used to solve the problem. Yan et al. [1] used ReliefF algorithm to identify CTQs in complex products. The method is verified feasible, but the classification accuracy of the results is low (nearly 70%).

The method MIGSA proposed in this paper merges the merits of efficiency of both filter and high accuracy wrapper.

The following sections provide a more detailed description of the approach.

3. Preliminaries

3.1. Mutual Information. In information theory, the uncertainty of random variables can be measured by the entropy [19]. Let X be random variables with discrete values; its entropy $H(X)$ is defined as

$$H(X) = -\sum_{x \in X} p(x) \log p(x), \tag{1}$$

where $p(x) = \Pr\{X = x\}$ is the probability density function of X. Then, let X and Y be two discrete random variables; their joint probability density function is $p(x, y)$; then, the joint entropy $H(X, Y)$ of them is

$$H(X, Y) = -\sum_{y \in Y} \sum_{x \in X} p(x, y) \log p(x, y). \tag{2}$$

Conditional entropy $H(X \mid Y)$ is used to describe the uncertainty reduction of variable X when variable Y is known. It is defined as

$$H(X \mid Y) = -\sum_{y \in Y} \sum_{x \in X} p(x, y) \log p(x \mid y). \tag{3}$$

The mutual information $I(X; Y)$ is defined to measure the common information of two variables X and Y:

$$I(X; Y) = -\sum_{y \in Y} \sum_{x \in X} p(x, y) \log \frac{p(x, y)}{p(x)\, p(y)}. \tag{4}$$

From the above definition, the high value of $I(X; Y)$ means that the two variables X and Y are closely related; otherwise, the two variables are not closely related when the value is small; specially, they are totally unrelated when $I(X; Y) = 0$.

According to (1)–(4), the relation between the entropy and mutual information can be described in

$$I(X; Y) = H(X) - (X \mid Y). \tag{5}$$

3.2. Gravitational Search Algorithm. The gravitational search algorithm (GSA) is a recently proposed heuristic search algorithm by Rashedi et al. [20], which has been inspired by the Newtonian laws of gravity and motion. As a new stochastic population-based heuristic optimization tool, the GSA algorithm provides an iterative method that simulates mass interactions and moves through a multidimensional search space in the influence of gravitation. In the GSA algorithm, agents are considered as objects and their performance is measured by their masses. The GSA algorithm is introduced as follows [20, 21].

For a system with n agents, the position of the ith agent is defined as

$$P_i = \left(p_i^1, \ldots, p_i^d, \ldots, p_i^h\right) \quad \text{for } i = 1, 2, \ldots, n, \tag{6}$$

where p_i^d presents the position of the ith agent in the dth dimension and h is the dimension of the search space.

The velocity $v_i^d(t)$ and position $p_i^d(t)$ of the ith agent in the dth dimension can be updated using (7) and (12):

$$v_i^d(t+1) = \text{rand}_i \times v_i^d(t) + a_i^d(t), \tag{7}$$

$$p_i^d(t+1) = p_i^d(t) + v_i^d(t+1), \tag{8}$$

where rand_i is uniform random data in the interval $[0, 1]$, which is utilized to give a randomized characteristic to the search. The acceleration $a_i^d(t)$ can be calculated as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)}. \tag{9}$$

$F_i^d(t)$ is the total force exerted on agent i in the dth dimension and $M_i(t)$ is the mass of the agent i at time t.

The force acting on agent i from the agent j at time t is defined as

$$F_{ij}^d(t) = G(t) \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon} \left(p_j^d(t) - p_i^d(t)\right), \tag{10}$$

where $G(t)$ is gravitational constant at time t, $G(t) = G_0(1 - t/T)$, ε is a small constant, and $R_{ij}(t)$ is the distance between agents i and j. Then, $F_i^d(t)$ and $M_i(t)$ can be calculated:

$$F_i^d(t) = \sum_{j=1, j \neq i}^{n} \text{rand}_j F_{ij}^d(t), \tag{11}$$

$$q_i(t) = \frac{\text{fit}_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)}, \tag{12}$$

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^{n} q_j(t)}, \tag{13}$$

where rand_j is a random number in the interval $[0, 1]$. $\text{fit}_i(t)$ is the fitness value of the agent i at time t. $\text{best}(t)$ and $\text{worst}(t)$ are the best and worst $\text{fit}_i(t)$ at time t, respectively.

According to the difference of the position updating, GSA can be divided into continuous GSA (CGSA) and binary GSA (BGSA). In the binary algorithm, the position updating means a switching between "0" and "1" values. In the implementation of the BGSA, a large value of velocity must provide a high probability of hanging the position of the mass with respect to its previous position and a small value of the velocity must provide a small probability of changing the position. So, v_i^d can be transferred into a probability function $S(v_i^d)$ as follows:

$$S\left(v_i^d(t)\right) = \left|\tanh\left(v_i^d(t)\right)\right|. \tag{14}$$

Then, the agents will move according to the following rule:

$$x_i^d(t+1) = \begin{cases} 1 - p_i^d(t), & \text{rand} < S\left(v_i^d(t+1)\right), \\ p_i^d(t), & \text{else.} \end{cases} \tag{15}$$

4. The Approach to CTQs Identification

4.1. Characteristics Ranking by Mutual Information. In information theory, mutual information is used to quantitatively analyze the relationship between two characteristics or between a characteristic and a class variable. There are two subsets about the characteristics, one is the already selected subset S; the other is unselected subset F. Among all the characteristics in F, the characteristic x_i, which has the largest information about classes C that not provided by the already selected characteristics, can be selected into S [19]. The mutual information $J(X_i)$ of characteristic X_i can be estimated as follows [22]:

$$J(X_i) = NI(C; X_i) - \frac{1}{|S_{i-1}|} \sum_{X_s \in S_{i-1}} \frac{I(X_s, X_i)}{\min(H(X_s), H(X_i))}, \tag{16}$$

where $NI(C; X_i) = I(C; X_i)/\log_2(|\Omega_C|)$ and X_s is the characteristic of already selected subset.

4.2. Improved GSA. We use the GSA method to identify the CTQs for complex products. It has been proved that GSA is a better optimization algorithm than PSO and GA in most cases [20]. However, similar to other intelligent algorithms, GSA has a limitation of premature convergence. In order to overcome the drawback, we use opposition-based learning and immune strategy to improve GSA (IGSA).

The concept of opposition-based learning was introduced by Tizhoosh [23]. The main idea behind opposition-based learning is the simultaneous consideration of an estimate and its corresponding opposite estimate in order to achieve a better approximation for the current candidate solution [24]. Let $x \in R$ be a real number defined on a certain interval: $x \in [a, b]$. The opposite number $\tilde{x}$ is defined as follows:

$$\tilde{x} = a + b - x. \tag{17}$$

Analogously, the opposite number in a multidimensional case can be defined.

Immune algorithm (IA) is a kind of optimization method based on the characteristics of the biological immune system. It is proposed by Farmer, Packard, and Perelson in 1986, when they discussed links between immune system and other artificial intelligence methods. It has the capability to control a complex system [25] and has been applied in many fields. In the IA, antigen represents the problem to be solved. An antibody is generated where each member represents a candidate solution. Affinity is the fitness of an antibody to the antigen. The key of IA is how to generate antibodies [26].

Immune strategy can effectively solve the problem of population diversity and improve the convergence speed through immune recognition and immune memory. In the immune strategy, affinity is used to describe the information contained in an antibody [25]. The affinity function can be calculated as follows:

$$A_i = \frac{1}{\sum_{j=1}^{M} \sqrt{\sum_{k=1}^{D} \left(P_{g_{ik}} - P_{g_{jk}}\right)}}, \tag{18}$$

where D is the dimension, M is the number of antibodies, and P_g is the best position. Then the affinity of all memory antibodies with best position is calculated.

4.3. Implementation of Hybrid Approach

4.3.1. Representation of Position. Assume that there are L total characteristics; there will be 2^L kinds of characteristic subsets which are different from each other. If each agent takes one characteristic subset, the agent's position can be represented as binary bit strings of length L; every bit represents a characteristic; 1 means the corresponding characteristic is selected while 0 means the characteristic is not selected.

4.3.2. Fitness Function. The CTQs subset should not only have a small length but also have a high classification quality [27]. So the fitness function can be defined as follows:

$$\text{Fitness} = \alpha \cdot \gamma_B(D) + \beta \cdot \frac{|C| - |B|}{|C|}, \tag{19}$$

where $\gamma_B(D)$ is the classification quality of condition attribute set B relative to decision D in rough set theory and $|B|$ is the length of selected characteristic subset. $|C|$ is the total number of characteristics. α and β are two parameters corresponding to the importance of classification quality and subset length, $\alpha \in [0, 1]$ and $\beta = 1 - \alpha$. The high α assures that the best position is at least a real rough set reduction. We can calculate the quality of each position by the formula. The goal is to maximize fitness values [27].

4.3.3. Hybrid Approach (MIGSA). The proposed method has two major parts. The first one is ranking the characteristics by mutual information, and the second one is finding the optimal subset of characteristics by improved GSA based on the result of the first part. Let k be the number of the characteristics that we select by the mutual information, $k < d$. Then, the characteristics dimension is reduced through the selection.

5. Experiments

For our experiments, we implement the hybrid approach for CTQs identification in Matlab 7.10.0. We test the algorithm through two datasets from UCI machine learning repository: SECOM dataset and Statlog (LS) dataset. The brief information about them is given in Table 1. SECOM dataset is the data about semi-conductor manufacturing process. There are two classes in 1567 instances including 104 fails, and they have some missing values. Statlog (LS) dataset is the data about landsat satellite. There are 6 classes in 6435 instances. The instance number of each class is 1533, 703, 1358, 626, 707, and 1508, respectively. The dataset of SECOM is used to test the proposed method's capability, and Statlog (LS) dataset is used to compare with other methods in literatures.

Before the beginning of hybrid approach, data preprocessing, which contains missing values preprocessing and data balance, should be done first. Then mutual information was calculated between characteristics and classes C.

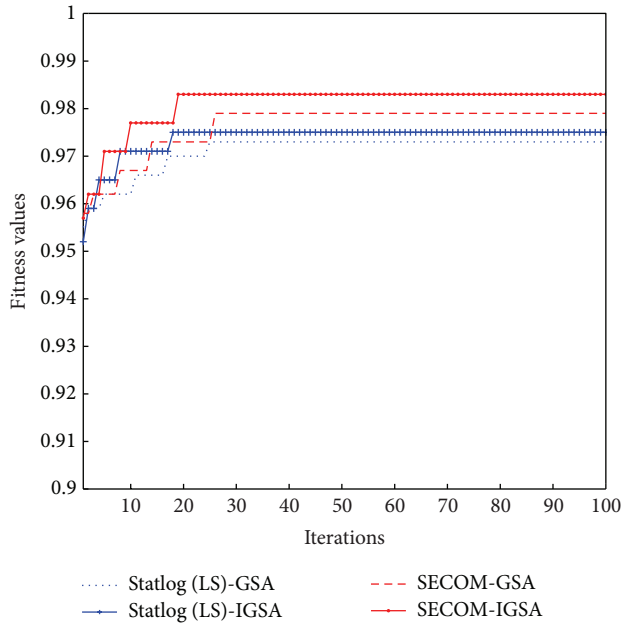


Figure 1: Evolution process of the global best on 2 datasets.

Table 1: Information of 2 datasets.

Dataset	Instances	Characteristics	Class
SECOM	1567	591	2
Statlog (LS)	6435	36	6

Table 2: Parameters of algorithm used in the experiment.

G_0	α	β	Max iterations	Population (N)
100	0.9	0.1	100	30

After that, characteristics are selected to optimize through improved GSA. The program is terminated when the algorithm reaches the stopping criterion. The parameters of configuration of GSA are given in Table 2.

Figure 1 is the process of global best on SECOM and Statlog (LS) by IGSA and GSA. From Figure 1, we find that the IGSA overcomes the premature convergence of GSA. Then, SVM is used as the training procedure, and the classification accuracy of CTQs is estimated by tenfold cross validation. The results are given in Table 3. From Table 3, we can find that the number of CTQs is 11, and the classification is 88.3% for SECOM dataset; the number of CTQs is 5, and the classification is 84.9% for the Statlog (LS) dataset.

In order to prove the proposed method's capability, it is compared with four algorithms (MRPSO, BPSO, CBPSO, and ReliefF) in the literature [28] from two dimensions: number of CTQs and accuracy using Statlog (LS). The results are listed in Table 4. From Table 4, we can find that the accuracy of CTQs obtained by 5 methods is similar. However, the numbers of CTQs are significantly different. Of the four existing methods, the best is MRPSO with 85.24 percent accuracy and 16 CTQs, while the accuracy of CTQs obtained by our proposed method MIGSA can reach 84.9 percent and only needs 5 CTQs. Hence, the proposed approach is an efficient method of CTQs identification.

Table 3: Information of 2 datasets.

Dataset	Number of CTQs	Accuracy (%)
SECOM	11	88.3
Statlog (LS)	5	84.9

Table 4: Number of CTQs and accuracy with different methods for dataset landsat satellite.

	MRPSO	BPSO	CBPSO	ReliefF	MIGSA
Number of CTQs	16	22	25	34	**5**
Accuracy (%)	**85.24**	83.15	83.4	82.03	84.9

6. Conclusions

In order to solve the identification of CTQs for complex products, we propose a hybrid approach based on mutual information and GSA. Due to premature convergence that often happens on GSA, we improve the algorithm through opposition learning and immune algorithm. At first, we compare the improved method IGSA with the original method GSA, and the results of experiment show that IGSA has a strong search capability. Then, MIGSA is compared with 4 methods in the literature; the experimental results show that it can reduce CTQs dimension greatly. From experiments, it can be said that MIGSA is an effective method to identify CTQs for complex products.

Conflict of Interests

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

Acknowledgments

This work is financially supported by the National Science Fund for Distinguished Young Scholars of China (no. 71225006) and National Natural Science Foundation of China (nos. 71071107 and 70931004). The authors thank the editor and reviewers for their insightful comments and suggestions.

References

[1] W. Yan, Z. He, and W. Tian, "The application of ReliefF algorithm for identifying CTQ in complex products," in *Proceedings of the 2nd IEEE International Conference on Emergency Management and Management Sciences (ICEMMS '11)*, pp. 459–463, Beijing, China, August 2011.

[2] M. Hobday and H. Rush, "Technology management in complex product systems (CoPS): ten questions answered," *International Journal of Technology Management*, vol. 17, no. 6, pp. 618–638, 1999.

[3] L. Y. Zheng, C. A. McMahon, L. Li, L. Ding, and J. Jamshidi, "Key characteristics management in product lifecycle management: a survey of methodologies and practices," *Proceedings*

of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, vol. 222, no. 8, pp. 989–1008, 2008.

[4] A. C. Thornton, "Mathematical framework for the key characteristic process," *Research in Engineering Design*, vol. 11, no. 3, pp. 145–157, 1999.

[5] N. Alp, M. Yaworsky, and P. Kazemersky, "The identification of critical-to-quality (CTQ) characteristics by using the quality function deployment (QFD)," in *Proceedings of National Conference of the American-Society-for-Engineering-Management*, pp. 402–406, Huntsville, Ala, USA, October 2001.

[6] Y. He, X. Tang, and W. Chang, "Technical decomposition approach of critical to quality characteristics for product design for six sigma," *Quality and Reliability Engineering International*, vol. 26, no. 4, pp. 325–339, 2010.

[7] L. P. Sullivan, "Quality function deployment," *Quality Progress*, vol. 19, no. 6, pp. 39–50, 1986.

[8] G.-B. Zhang, X.-L. Ren, L.-K. Liu, and H.-Y. Ge, "Multiple key quality characteristics optimization model for manufacturing process," *Computer Integrated Manufacturing Systems, CIMS*, vol. 16, no. 6, pp. 1286–1291, 2010.

[9] A. C. Thornton, "Quantitative selection of variation reduction plans," *Journal of Mechanical Design*, vol. 122, no. 2, pp. 185–193, 2000.

[10] P. G. Rowe, "Setting statistical specifications for critical to quality characteristics," *International Journal of Six Sigma and Competitive Advantage*, vol. 2, no. 1, pp. 89–103, 2006.

[11] H. Shen and H. Wan, "Controlled sequential factorial design for simulation factor screening," *European Journal of Operational Research*, vol. 198, no. 2, pp. 511–519, 2009.

[12] B. K. Rout and R. K. Mittal, "Screening of factors influencing the performance of manipulator using combined array design of experiment approach," *Robotics and Computer: Integrated Manufacturing*, vol. 25, no. 3, pp. 651–666, 2009.

[13] L. Mathieu and B. Marguet, "Integrated design method to improve producibility based on product key characteristics and assembly sequences," *CIRP Annals—Manufacturing Technology*, vol. 50, no. 1, pp. 85–88, 2001.

[14] D. E. Whitney, "The role of key characteristics in the design of mechanical assemblies," *Assembly Automation*, vol. 26, no. 4, pp. 315–322, 2006.

[15] P. Johansson, A. Chakhunashvili, S. Barone, and B. Bergman, "Variation mode and effect analysis: a practical tool for quality improvement," *Quality and Reliability Engineering International*, vol. 22, no. 8, pp. 865–876, 2006.

[16] N. Wang, J. C. Xu, and J. F. Yang, "Identifying method of key quality characteristics in multistage manufacturing process," *Computer Integrated Manufacturing Systems*, vol. 19, no. 4, pp. 888–895, 2013.

[17] F.-C. Wu, "Optimization of correlated multiple quality characteristics using desirability function," *Quality Engineering*, vol. 17, no. 1, pp. 119–126, 2005.

[18] W. Yan, Z. He, W. Tian, and S. He, "Complex products critical-to-quality characteristics identification based on IG," *Industrial Engineering and Management*, vol. 17, no. 1, pp. 70–74, 2012.

[19] J. Huang, Y. Cai, and X. Xu, "A hybrid genetic algorithm for feature selection wrapper based on mutual information," *Pattern Recognition Letters*, vol. 28, no. 13, pp. 1825–1844, 2007.

[20] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[21] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "BGSA: binary gravitational search algorithm," *Natural Computing*, vol. 9, no. 3, pp. 727–745, 2010.

[22] L. T. Vinh, S. Lee, Y.-T. Park, and B. J. d'Auriol, "A novel feature selection method based on normalized mutual information," *Applied Intelligence*, vol. 37, no. 1, pp. 100–120, 2012.

[23] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA '05)*, pp. 695–701, November 2005.

[24] B. Shaw, V. Mukherjee, and S. P. Ghoshal, "A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems," *International Journal of Electrical Power & Energy Systems*, vol. 35, no. 1, pp. 21–33, 2012.

[25] C.-J. Lin, "An efficient immune-based symbiotic particle swarm optimization learning algorithm for TSK-type neuro-fuzzy networks design," *Fuzzy Sets and Systems*, vol. 159, no. 21, pp. 2890–2909, 2008.

[26] M. M. El-Sherbiny and R. M. Alhamali, "A hybrid particle swarm algorithm with artificial immune learning for solving the fixed charge transportation problem," *Computers and Industrial Engineering*, vol. 64, no. 2, pp. 610–620, 2013.

[27] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459–471, 2007.

[28] X. Yao, X.-D. Wang, Y.-X. Zhang, and W. Quan, "A maximum relevance minimum redundancy hybrid feature selection algorithm based on particle swarm optimization," *Control and Decision*, vol. 28, no. 3, pp. 413–423, 2013.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 836925, 15 pages
http://dx.doi.org/10.1155/2015/836925

Research Article

Construction Example for Algebra System Using Harmony Search Algorithm

FangAn Deng,[1] Shouheng Tuo,[1] Longquan Yong,[1] and Tao Zhou[2]

[1]*School of Mathematics and Computer Science, Shaanxi University of Technology, Hanzhong 723000, China*
[2]*School of Science, Ningxia Medical University, Yinchuan, Ningxia 750004, China*

Correspondence should be addressed to Shouheng Tuo; tuo_sh@126.com

Received 24 June 2014; Accepted 15 September 2014

Academic Editor: Shifei Ding

The construction example of algebra system is to verify the existence of a complex algebra system, and it is a NP-hard problem. In this paper, to solve this kind of problems, firstly, a mathematical optimization model for construction example of algebra system is established. Secondly, an improved harmony search algorithm based on NGHS algorithm (INGHS) is proposed to find as more solutions as possible for the optimization model; in the proposed INGHS algorithm, to achieve the balance between exploration power and exploitation power in the search process, a global best strategy and parameters dynamic adjustment method are present. Finally, nine construction examples of algebra system are used to evaluate the optimization model and performance of INGHS. The experimental results show that the proposed algorithm has strong performance for solving complex construction example problems of algebra system.

1. Introduction

Algebra system is one of the broad parts of mathematics, together with number theory, geometry, and analysis. It has wide applications in astronomy, biology, construction, computer science, and so on. Many scientific researchers focus on its studies. However, with the rapid development of science and technology, more and more complex algebra systems emerge in large numbers, so the algebra system researchers meet a lot of difficult problems. One of them is to construct appropriate examples for the algebra system so as to prove the existence of the algebra system. This is because there is the tremendous computing workload to construct appropriate examples that are satisfied with the operations of the algebra system. In the current study of algebra system, the construction examples of algebra system are often manually achieved. However, for a multielement complex algebra system, the construction examples are very difficult. In order to understand the construction example of algebra system better, we introduce its construction by an example of $N(2, 2, 0)$ algebra system.

Above all, we begin this section by introducing a simplified definition of $N(2, 2, 0)$ algebra system [1, 2].

Definition 1. An $N(2, 2, 0)$ algebra is a system $(S, *, \Delta, 0)$ where S is nonempty set, 0 is a constant element of S, and $*$ and Δ are two binary operations on S, obeying the following axioms. For all $x, y, z \in S$,

(F_1) $x * (y \Delta z) = z * (x * y)$,

(F_2) $(x \Delta y) * z = y * (x * z)$,

(F_3) $0 * x = x$.

For Definition 1, there are two works for researchers to do.

(1) Give some examples to prove the existence of this algebra.

(2) How many solutions are satisfying the three conditions (F_1), (F_2), and (F_3) simultaneously?

Example 2. Let $S = \{a_1, a_2, \ldots, a_n\}$ be a finite set of distinct elements. We define operations $*$ and Δ by the following equations, respectively:

$$\begin{array}{c|cccc} * & a_1 & a_2 & \cdots & a_n \\ \hline a_1 & x_{11} & x_{12} & \cdots & x_{1n} \\ a_2 & x_{21} & x_{22} & \cdots & x_{2n} \\ & & & \cdots & \\ a_n & x_{n1} & x_{n2} & \cdots & x_{nn} \end{array} \quad (1)$$

$$\begin{array}{c|cccc} \Delta & a_1 & a_2 & \cdots & a_n \\ \hline a_1 & y_{11} & y_{12} & \cdots & y_{1n} \\ a_2 & y_{21} & y_{22} & \cdots & y_{2n} \\ & & & \cdots & \\ a_n & y_{n1} & y_{n2} & \cdots & y_{nn} \end{array} \quad (2)$$

In (1) and (2), $a_i * a_j = x_{ij} \in S$, $a_i \Delta a_j = y_{ij} \in S$ $(i = 1, 2, \ldots, n; j = 1, 2, \ldots, n)$.

Next we consider

(F_1) $a_i * (a_j \Delta a_k) = a_k * (a_i * a_j) \Leftrightarrow a_i * (y_{jk}) = a_k * (x_{ik})$,

(F_2) $(a_i \Delta a_j) * a_k = a_j * (a_i * a_k) \Leftrightarrow (y_{ij}) * a_k = a_j * (x_{ik}) \Leftrightarrow a_i * (y_{jk}) = a_k * (x_{ik})$,

(F_3) $a_1 * a_i = a_i \Leftrightarrow x_{1i} = a_i$,

where $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, n$; $k = 1, 2, \ldots, n$.

For this algebra system, researchers hope to construct some examples that satisfy the conditions (F_1), (F_2), and (F_3) and to prove the existence of the algebra. However, the calculated amount of this work is so large that computing cost is typically high. Like Example 2, each variable x_{ij} or y_{ij} $(i = 1, 2, \ldots, n; j = 1, 2, \ldots, n)$ can take n different values $(a_1, a_2, \ldots, a_n)$; the combinatorial number with the repetition for all x_{ij} and y_{ij} is n^{2n^2}. Obviously it is a NP-hard problem. It will result in the combination blast when the dimension increases. So the enumeration method can hardly be used to construct an example for the algebra system $(S, *, \Delta, 0)$.

Therefore, for this kind of NP-hard problem, this paper proposes an intelligent metaheuristic algorithm to get a set of $(x_{11}, x_{12}, \ldots, x_{1n}; \ldots; x_{n1}, x_{n2}, \ldots, x_{nn})$ and $(y_{11}, y_{12}, \ldots, y_{1n}; \ldots; y_{n1}, y_{n2}, \ldots, y_{nn})$ quickly that satisfy conditions (F_1), (F_2), and (F_3).

2. The Optimization Model of $N(2, 2, 0)$ Algebra System

For the construction example of an $N(2, 2, 0)$ algebra system, we can consider it as an optimized constraint satisfaction problem (OCSP) that satisfies conditions (F_1), (F_2), and (F_3). Then it can be solved by the intelligent optimization algorithm.

To predigest the solving process and promote its practicability, we turn this problem into nonconstrained optimization problem. The optimization model is as follows:

$$\min F(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left[f_1(a_i, a_j, a_k) + f_2(a_i, a_j, a_k) + f_3(a_i)\right], \quad (3)$$

where

$$\begin{aligned} \mathbf{X} &= (x_{11}, x_{12}, \ldots, x_{1n}; \ldots; x_{n1}, x_{n2}, \ldots, x_{nn}) \\ &\quad \text{is algebra } (S, *, 0); \\ \mathbf{Y} &= (y_{11}, y_{12}, \ldots, y_{1n}; \ldots; y_{n1}, y_{n2}, \ldots, y_{nn}) \\ &\quad \text{is algebra } (S, \Delta, 0); \end{aligned} \quad (4)$$

for any a_i, a_j, and a_k $(i = 1, 2, \ldots, n; j = 1, 2, \ldots, n; k = 1, 2, \ldots, n)$ in $S = \{a_1, a_2, \ldots, a_n\}$,

$$a_i * a_j = x_{ij} \in S, \qquad a_i \Delta a_j = y_{ij} \in S,$$

$$\begin{aligned} f_1(a_i, a_j, a_k) &= \begin{cases} 0, & a_i * (a_j \Delta a_k) = a_k * (a_i * a_j) \\ 1, & a_i * (a_j \Delta a_k) \neq a_k * (a_i * a_j), \end{cases} \\ f_2(a_i, a_j, a_k) &= \begin{cases} 0, & (a_i \Delta a_j) * a_k = a_j * (a_i * a_k) \\ 1, & (a_i \Delta a_j) * a_k \neq a_j * (a_i * a_k), \end{cases} \\ f_3(a_i) &= \begin{cases} 0, & 0 * a_i = a_i \\ 1, & 0 * a_i \neq a_i. \end{cases} \end{aligned} \quad (5)$$

The functions f_1, f_2, and f_3 are to transform the constraint satisfaction problem to a nonconstraint optimization problem by adding the penalty factors. When f_1, f_2, and f_3 meet conditions (F_1), (F_2), and (F_3), respectively, the objective values of each functions are 0; otherwise, a penalty factor 1 is as the objective value.

For this problem, there are multiple solutions that meet conditions (F_1), (F_2), and (F_3). Therefore, sometimes researchers hope to get as many solutions as possible. To resolve the multisolution problem, we introduce tube table (tube table: TT) to store the obtained solutions and to avoid duplication of solutions. The definite means are as follows:

$$f_4(\mathbf{X}) = \begin{cases} 0 & \mathbf{X} \notin \text{TT} \\ n & \mathbf{X} \in \text{TT}, \end{cases} \quad (6)$$

where TT is the tube table. f_4 is defined as a penalty function. If $\mathbf{X}$ is not in tube table TT, f_4 is equal to 0. Otherwise, f_4 is assigned a penalty value n.

Under this condition, the objective function can be expressed as

$$\min F(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left[f_1(a_i, a_j, a_k) + f_2(a_i, a_j, a_k) + f_3(a_i)\right] + f_4(\mathbf{X}). \quad (7)$$

It is clear that, for this optimization problem, the objective value of optimal solution is $\min F^* = 0$.

In order to solve the nonconstrained optimization problem, we propose a new harmony search (HS) algorithm which is a big improvement for classical HS algorithm [1] and NGHS [3] method. We called it improved NGHS (INGHS).

3. Classical Harmony Search (HS) and NGHS

Classical harmony search (HS) is derivative-free metaheuristic algorithm [4, 5]. It mimics the improvisation process of music players and attains the ideal harmony by adjusting the HM. HS algorithm has the same feature as genetic algorithm (GA). The HS algorithm is good at identifying the new regions of the searching space in a reasonable time; however, it has difficulties performing a local search for numerical applications [3, 6–8].

So, several variants of HS have been proposed to improve the performance of the HS algorithm [3, 6–14], such as local-best harmony search algorithm with dynamic subpopulations (DLHS) [6], self-adaptive global best harmony search algorithm (SGHS) [7], intelligent tuned harmony search (ITHS) [8] algorithm, and exploratory power of the harmony search (EHS) [3] algorithm. In addition, Zou et al. presented a novel global harmony search algorithm for unconstrained problems (NGHS) [9, 10]. The literature [11] presents the recent advances in HS algorithm. Tuo and Yong proposed HSTL methods for large scale optimization problem and presented an improved harmony search with chaos (HSCH) [12], as well as other variants [13–15] of HS and some applications [16–19] of HS.

Consider an optimization model as follows:

$$\begin{aligned} &\min \quad f(\mathbf{X}), \quad \mathbf{X} = (x_1, x_2, \ldots, x_D) \\ &\text{subject to} \quad x_i \in \left[x_i^L, x_i^U\right], \quad i = 1, 2, \ldots, D, \end{aligned} \tag{8}$$

where $f(\mathbf{X})$ is the optimization objective function, $\mathbf{X}$ is the solution vector that consists of D decision variables x_i ($i = 1, 2, \ldots, D$), and $[x_i^L, x_i^U]$ is feasible range of values for decision variable x_i.

In the HS algorithm, $\mathbf{X}$ represents the harmony and $f(\mathbf{X})$ denotes the melody of harmony $\mathbf{X}$. HS uses three rules (memory consideration, pitch adjustments, and randomization) to optimize the harmony memories (HM) composed of HMS harmony vectors.

3.1. The Classical Harmony Search (HS) Algorithm. The steps in the procedure of classical harmony search algorithm are as follows.

Step 1 (initialize the harmony memory). The harmony memory (HM) consists of HMS harmony vectors. Each harmony vector is generated from a uniform distribution in the feasible space, as

$$\begin{aligned} x_i^j &= x_i^L + r \cdot \left(x_i^U - x_i^L\right), \\ i &= 1, 2, \ldots, D; \quad j = 1, 2, \ldots, \mathrm{HMS}, \end{aligned} \tag{9}$$

where D and HMS represent the number of decision variables and the size of harmony memory, respectively, and r denotes a uniform distribution random number between 0 and 1, as

$$\mathrm{HM} = \begin{bmatrix} \mathbf{X}^1 \\ \mathbf{X}^2 \\ \vdots \\ \mathbf{X}^{\mathrm{HMS}} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_D^1 \\ x_1^2 & x_2^2 & \cdots & x_D^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{\mathrm{HMS}} & x_2^{\mathrm{HMS}} & \cdots & x_D^{\mathrm{HMS}} \end{bmatrix}. \tag{10}$$

Step 2 (improvise a new harmony via three rules). There are three rules that can be used to improvise a new harmony vector $\mathbf{X}^{\mathrm{new}}$.

(a) Memory Consideration. A decision variable value of the harmony vector will be adopted by choosing from the harmony memory with probability HMCR.

(b) Pitch Adjustment. Get a component randomly from an adjacent value of one decision variable of a harmony vector with probability PAR.

(c) Random Generation. Generate a component randomly in the feasible region with probability 1-HMCR.

The improvisation procedure of a new harmony vector works as Algorithm 2.

A trial harmony vector $\mathbf{X}^{\mathrm{new}}$ is generated in Step 2. Next, in Step 3, it will be decided whether to survive.

Step 3 (select operator). Get the worst harmony vector $\mathbf{X}^{\mathrm{worst}}$ from the HM (see Algorithm 1).

Step 4 (check stopping criterion). If the stopping criterion (maximum function evaluation times: maxFEs) is satisfied, computation is terminated. Otherwise, Steps 2 and 3 are repeated.

3.2. The NGHS Algorithm. In the novel global harmony search (NGHS) algorithm [9], three significant parameters, harmony memory considering rate (HMCR), fret width (FW), and pitch adjusting rate (PAR), are excluded from NGHS and a random selection rate (p_m) is included in the NGHS. In Step 3, NGHS works as Algorithm 3, where x_i^{best} and x_i^{worst}, respectively, represent indexes of the best harmony and the worst harmony in HM. rand(0, 1) is a uniformly generated random number in [0, 1], and it should set parameter $p_m = 2/D$ for the 0-1 knapsack problem and set parameter $p_m = 0.01 \sim 0.1$ for continuous optimal problem.

3.3. The Proposed HS Method. In this section, we proposed a new harmony search algorithm which is improved based on classical HS algorithm and NGHS method, so we call the proposed algorithm INGHS.

Since the HS algorithm origination, it has been applied to many practical optimization problems. However, for large scale optimization problems, classical HS has slow convergence and low precision, which is due to the fact that because a new decision variable value in harmony memory (HM) can

If fitness ($\mathbf{X}^{\text{new}}$) is be**tt**er than fitness $\left(\mathbf{X}^{\text{worst}}\right)$ **then**
$\mathbf{X}^{\text{worst}} = \mathbf{X}^{\text{new}}$; ($\mathbf{X}^{\text{worst}}$ is the worst harmony vector in HM).
EndIf

Algorithm 1

For $i = 1$ to D **do**
If rand(0, 1) < HMCR **then**
$x_i^{\text{new}} = x_i^j$, $j \in U\{1, 2, \ldots, \text{HMS}\}$
If rand(0, 1) < PAR **then**
$x_i^{\text{new}} = x_i^{\text{new}} \pm \text{rand}() \times \text{FW}(i)$; **//FW: fretwidth**
$x_i^{\text{new}} = \min\left(\max\left(x_i^{\text{new}}, x_i^L\right), x_i^U\right)$
Endif
Else
$x_i^{\text{new}} = x_i^L + \left(x_i^U - x_i^L\right) \times \text{rand}()$
Endif
End For

Algorithm 2: The improvisation procedure of new harmony vector by HS.

For $i = 1$ to D **do**
$x_i^r = 2 \times x_i^{\text{best}} - x_i^{\text{worst}}$
$x_i^r = \min\left(\max\left(x_i^r, x_i^L\right), x_i^U\right)$
$x_i^{\text{new}} = x_i^{\text{worst}} + \text{rand}(0, 1) \times \left(x_i^r - x_i^{\text{worst}}\right)$.
If rand(0, 1) $\le p_m$ %random mutation
$x_i^{\text{new}} = x_i^L + \left(x_i^U - x_i^L\right) \times \text{rand}(0, 1)$
EndIf
EndFor

Algorithm 3: The NGHS modifies improvisation step.

be generated only by using pitch adjustment and randomization strategies during the search procedure, the memory consideration rule is only used to adjust the existing decision variable values according to the harmony memory (HM). Thus HS can maintain a strong performance of exploration, but not a good performance of exploitation, and, in the later stage of search, it is characterized by slow convergent. Therefore, for solving the large scale optimization problem, the key is to balance the global exploration performance and the local exploitation ability.

Because the construction example of the algebra is a large scale high-dimensional optimization problem, to achieve the most satisfactory optimization performance by applying the HS algorithm to a given problem, we adopt four optimization strategies and dynamical parameter control method to balance the global exploration power and the local exploitation ability.

It is of very importance between the convergence and the diversity in order to improve the efficiency of the search. In the classical HS algorithm, a new harmony is generated in Step 2. After the selecting operation in Step 3, the population variance may increase or decrease. With a high population variance, the diversity and exploration power will increase, and in the same time the convergence and the exploitation power will decrease accordingly. Conversely, with a low population variance, the convergence and the exploitation power will increase [8]; the diversity and the exploration power will decrease. So it is significant how to keep the balance between the convergence and the diversity. Classical HS algorithm loses its ability easily at the later evolution process [3], because of improvising a new harmony from HM with a high HMCR and local adjusting with PAR. And HM diversity decreases gradually from the early iteration to the last. However, in HS algorithm, a low HMCR employed will increase the probability (1-HMCR) of random select in search space; the exploration power will improve, but the local search ability and the exploitation accuracy cannot be improved by the single pitch adjusting strategy.

To overcome the inherent weaknesses of HS, in this section, we present INGHS method to construct example for algebra. The INGHS algorithm works as Algorithm 4 and Figure 1.

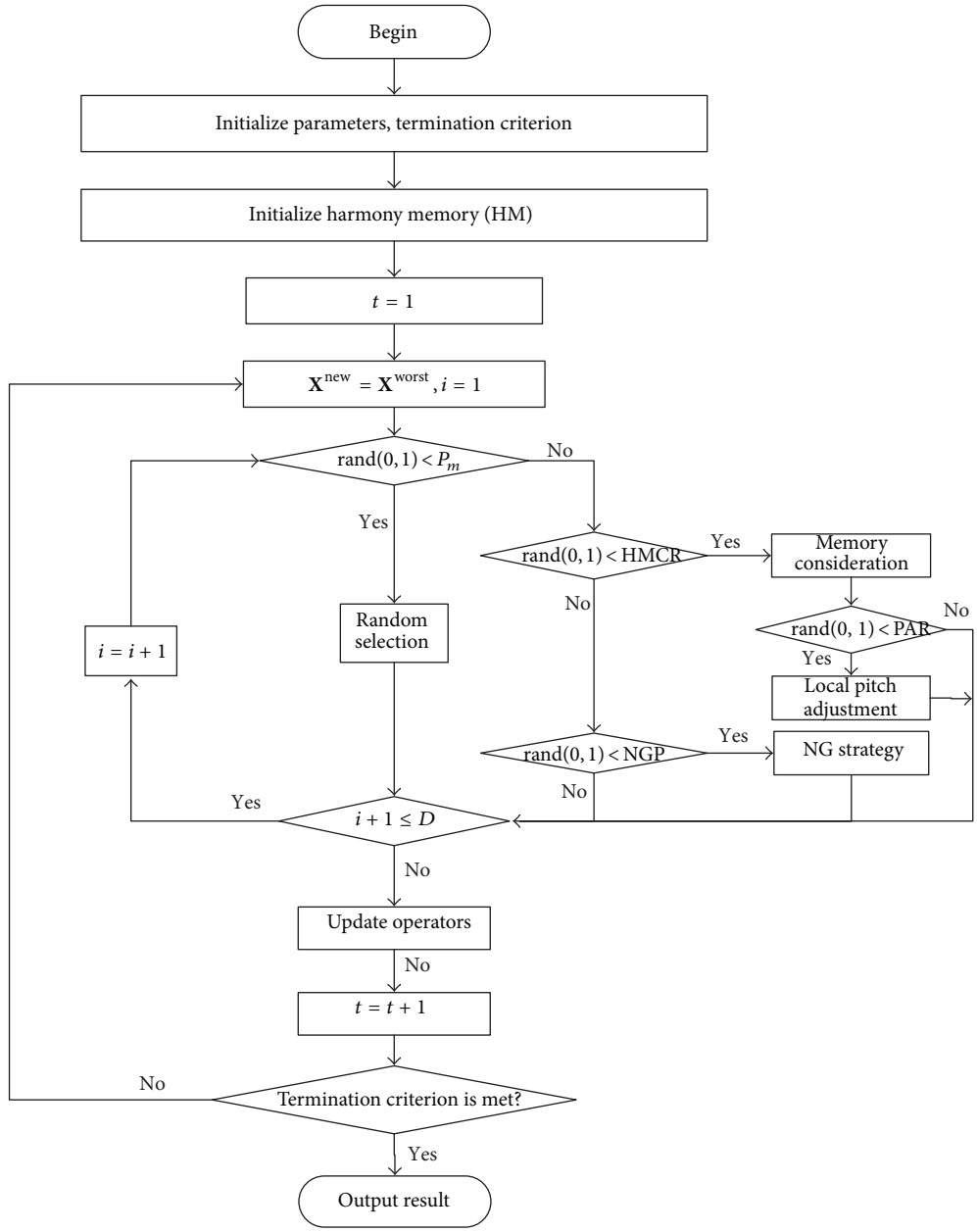


Figure 1: The flow chart of the proposed algorithm.

In Algorithm 4, to resolve the discrete optimization problems, the function Round(x) is used to round each element of x to the nearest integer.

(1) Novel Global (NG) Best Strategy. In NGHS algorithm, the novel global best strategy is very effective to explore the global best solution. So, in the proposed method, the novel global best strategy is adopted (see Algorithm 5), where the best and worst, respectively, represent indexes of the best harmony and the worst harmony in HM.

(2) Parameters Dynamically Changed. To balance the exploration and exploitation power of the INGHS algorithm efficiently, HMCR, PAR, FW, and NGP parameters are dynamically adapted to a suitable range with the increase of generations. Equation (11) shows the dynamic change of HMCR, PAR, and NGP, respectively,

$$\begin{aligned} \mathrm{HMCR} &= \mathrm{HMCR}_{\min} + \left(\mathrm{HMCR}_{\max} - \mathrm{HMCR}_{\min}\right) \\ &\quad \times \left(\frac{t}{T_{\max}}\right), \\ \mathrm{NGP} &= \mathrm{NGP}_{\min} + \left(\mathrm{NGP}_{\max} - \mathrm{NGP}_{\min}\right) \times \left(\frac{t}{T_{\max}}\right)^{3}, \\ \mathrm{PAR} &= \mathrm{PAR}_{\max} - \frac{\left(\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min}\right) \times t}{T_{\max}} \end{aligned} \tag{11}$$

(see [8]).

$\mathbf{X}^{\text{new}} = \mathbf{X}^{\text{worst}}$ *%%% worst is the index of the worst harmony in HM*
For i = 1 to D
If rand**(0,1)** < $\mathbf{P_m}$
% Random playing: randomly select any pitch within bounds
$x_i^{\text{new}} = \text{Round}\left(x_i^L + \left(x_i^U - x_i^L\right) \times \text{rand}(0,1)\right)$
Else
If rand**(0,1)** < **HMCR**
% Memory considering: randomly select a note stored in HM
$x_i^{\text{new}} = x_i^a;\ a \in U(1,2,\ldots,\text{HMS})$
If rand**(0,1)** < **PAR**
% Pitch adjusting: randomly adjust the pitch slightly
$x_i^{\text{new}} = \text{Round}(x_i^{\text{new}} \pm \text{rand}(0,1) \times \text{FW}(i))$
EndIf
ElseIf rand**(0,1)** < **NGP** **%**
$x_r = \text{Round}\left(2x_i^{\text{best}} - x_i^{\text{worst}}\right)$
$x_i^{\text{new}} = \text{Round}\left(x_i^{\text{worst}} + \text{rand}\,(0,1) \times \left(x_r - x_i^{\text{worst}}\right)\right)$
EndIf
EndIf
EndFor *% Finished improvising a new harmony*

Algorithm 4: The pseudocode of the proposed algorithm INGHS.

If rand$(0,1)$ < NGP (*NGP is the rate of choosing the novel global best strategy*)
$x_r = \text{Round}\left(2x_i^{\text{best}} - x_i^{\text{worst}}\right)$
$x_i^{\text{new}} = \text{Round}\left(x_i^{\text{worst}} + \text{rand}\,(0,1) \times \left(x_r - x_i^{\text{worst}}\right)\right)$
End

Algorithm 5

It can be seen that the parameter HMCR gradually increased from $\text{HMCR}_{\min}$ to $\text{HMCR}_{\max}$ linearly and the parameter NGP increased with low velocity in the early stage and it increased sharply in the final stage. That is because, in the beginning, in order to explore the global optimal solution, the harmony consideration rules and NG strategies are carried out with a smaller probability, and, in later stage, INGHS methods begin to focus on the local exploitation that needs a high probability to employ the NG strategy and harmony consideration rules. The benefits of doing so can get more opportunities to reinforce the global exploration by strengthening disturbance in the early stage and can acquire high precision solution by carrying local intensification search in the later stage. For the same reason, FW is decreased gradually in order to reduce perturbation step size step by step, and the variation of PAR from 0.55 to 0.3 is to reduce the probability of pitch adjustment.

3.4. The Construction Example for $N(2,2,0)$ Algebra System with the Proposed HS Method. For an $N(2,2,0)$ algebra system, there are multiple solutions that meet the conditions. Sometimes we need to obtain multiple solutions. So we adopt the proposed HS algorithm and tube table technology to resolve the multisolution problem. Its flow chart is as shown in Figure 2 and in Pseudocode 1.

3.5. Effect of Parameters HMCR and NGP on INGHS Performance. In this section, we determine the effect of parameters $\text{HMCR}_{\max}$, $\text{HMCR}_{\min}$, $\text{NGP}_{\max}$, and $\text{NGP}_{\min}$ on the performance of the proposed algorithm INGHS. The experiments are investigated for algebra system $(S, *, \Delta, 0)$, where $S = \{a, b, c, d, e, f, g\}$.

(1) Investigation for $\text{NGP}_{\max}$ and $\text{NGP}_{\min}$ (initialize $\text{HMCR}_{\max} = 0.9$, $\text{HMCR}_{\min} = 0.6$). Let $\text{NGP}_{\min}$ change from 0.1 to 0.6; then, for each $\text{NGP}_{\min}$, let $\text{NGP}_{\max} = 0.5$ to 1. So there are $6 \times 6 = 36$ pairs of parameters ($\text{NGP}_{\min}$, $\text{NGP}_{\max}$). For each pair of parameters ($\text{NGP}_{\min}$, $\text{NGP}_{\max}$), we execute the proposed algorithm (INGHS) $|S|^4$ ($7^4 = 2401$) times and then record the number of solutions obtained. The result is shown in Figure 3.

(2) Investigation for $\text{HMCR}_{\max}$ and $\text{HMCR}_{\min}$ (initialize $\text{NGP}_{\max} = 0.5$, $\text{NGP}_{\min} = 0.15$). Let $\text{HMCR}_{\min} = 0.2$ to 0.7 and let $\text{HMCR}_{\max} = 0.5$ to 1, respectively. For each pair of parameters ($\text{HMCR}_{\min}$, $\text{HMCR}_{\max}$), the proposed algorithm INGHS is performed $|S|^4$ ($7^4 = 2401$) times and then we record the number of solutions obtained. The number of solutions obtained is shown in Figure 4.

It can be seen from Figure 3 that most solutions can be obtained when $\text{NGP}_{\min} = 0.2$ and $\text{NGP}_{\max} = 0.5$.

Set TT **as empty**
For $L = 1$ to Max Loop times (ML)
  Run the proposed HS algorithm to get a best solution $\mathbf{X}^{\text{new}}$;
  If the objective function value $F(\mathbf{X}^{\text{new}}) == 0$
    Put $\mathbf{X}^{\text{new}}$ into tube table TT.
  EndIf
EndFor

Pseudocode 1

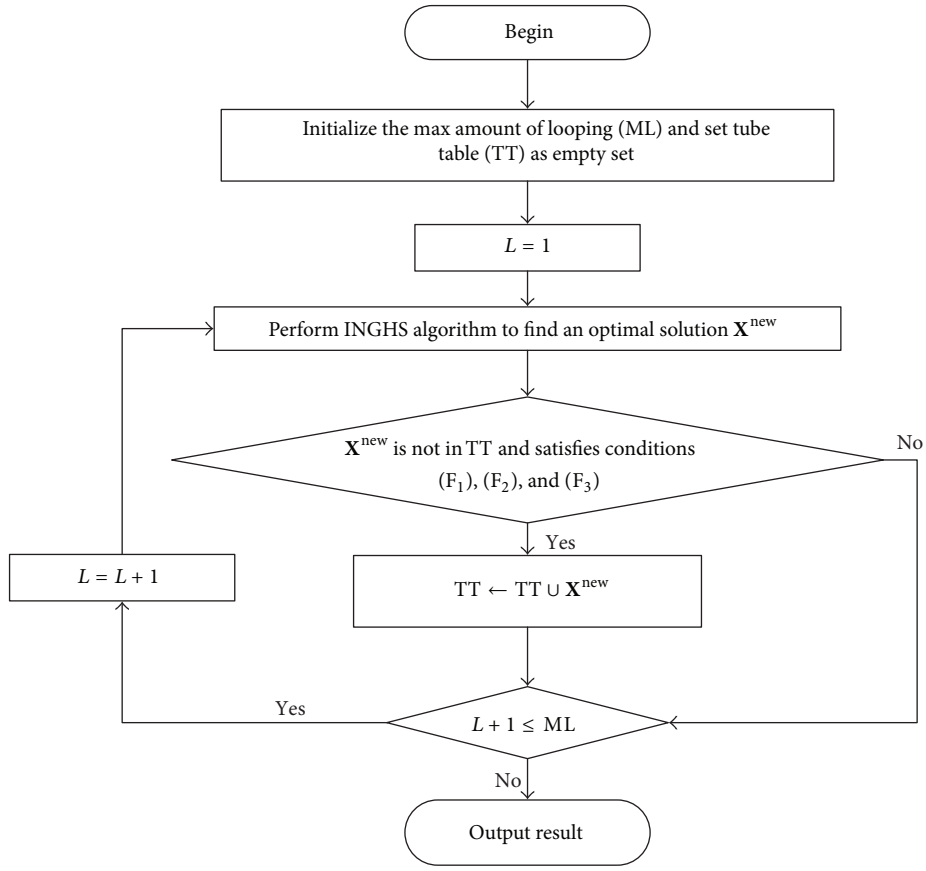


Figure 2: The flow chart for construction example of algebra system based on INGHS algorithm.

From Figure 4, we find that most solutions are found when $\text{HMCR}_{\text{min}} = 0.6$ and $\text{HMCR}_{\text{max}} = 0.9$.

From the above, the proposed algorithm has better performance when it is set as $\text{HMCR}_{\text{min}} = 0.6$, $\text{HMCR}_{\text{max}} = 0.9$, $\text{NGP}_{\text{min}} = 0.2$, and $\text{NGP}_{\text{max}} = 0.5$.

4. Computational Experiments and Results

In this section, we have tested the proposed algorithm over a set of $N(2, 2, 0)$ algebra system construction examples problems.

4.1. The Proposed Algorithm for Solving the Construction Examples of Algebra System. Definition 1 is chosen as the optimization objective algebra system. For Definition 1, we, respectively, set the following:

(1) $S = \{a, b\}$;

(2) $S = \{a, b, c\}$;

(3) $S = \{a, b, c, d\}$;

(4) $S = \{a, b, c, d, e\}$;

(5) $S = \{a, b, c, d, e, f\}$;

(6) $S = \{a, b, c, d, e, f, g\}$;

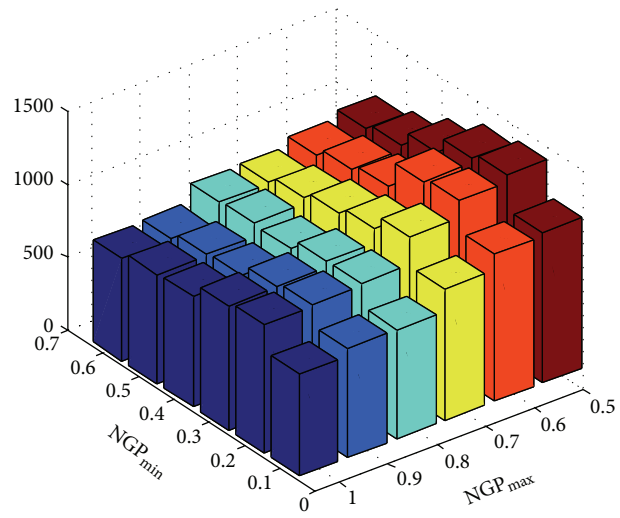


Figure 3: Variation of number of solutions with change in NGP_{max} and NGP_{min}.

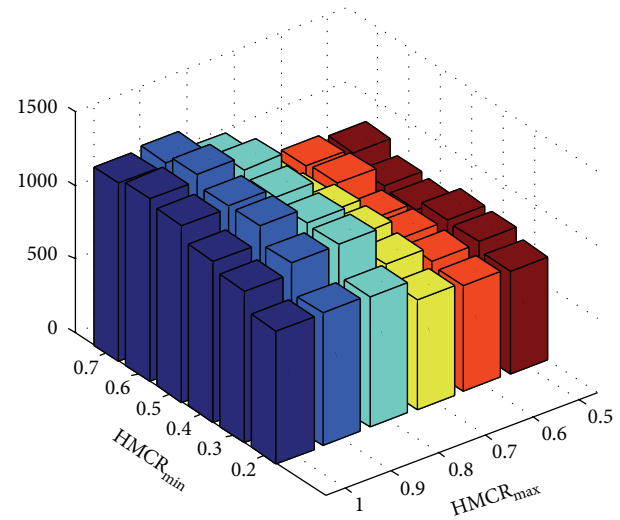


Figure 4: Variation of number of solutions with change in $HMCR_{max}$ and $HMCR_{min}$.

(7) $S = \{a, b, c, d, e, f, g, h\}$;

(8) $S = \{a, b, c, d, e, f, g, h, i\}$;

(9) $S = \{a, b, c, d, e, f, g, h, i, j\}$; $n = |S|$ represents the number of set S, $D = 2n^2$.

Then we use the proposed algorithm to construct the examples for the algebra system on S.

In order to deal with the optimization problems easily, we replace the elements $a, b, c, d, e, f, g, h, i$, and j with 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10, respectively. So, for each S, the upper limit of variables about optimization problems is set as $|S|$ and the lower limit is set as 1.

4.2. The Parameters Setting for Algorithms. In the experiment test, we set the following parameter values: $T_{max} = 500 \times D$; for NGHS algorithm, HMS = 10, $p_m = \min(0.01, 2/D)$; for the proposed algorithm (INGHS), HMS = 10, $HMCR_{max} = 0.9$, $HMCR_{max} = 0.6$, $NGP_{max} = 0.55$, $NGP_{min} = 0.15$, $PAR_{max} = 0.8$, $PAR_{min} = 0.25$, $FW_{max} = n/2$, $FW_{min} = n/10$, and $p_m = \min(0.01, 2/D)$.

When $n = 2$ and $n = 3$, there are 3 and 16 solutions for the $N(2, 2, 0)$ algebra system, respectively. So we set max looping times (ML) = 3 and 16, respectively. When $n > 3$ and $n < 9$, we do not know how many solutions are there for the algebra system. So we set $ML = n^4$. However, with the increasing of n, there are much more solutions, and computer's running time is very long. So we set ML = 8000 and 5000 when $n = 9$ and $n = 10$.

4.3. The Experiment Results and Analysis. In order to evaluate the performance of the proposed algorithm INGHS, we compared its success rate of finding the solutions with HS, HSTL, and NGHS algorithms in the same conditions. The results are shown in Table 1.

When $S = \{a, b\}$ ($n = 2$), the solutions, as shown in (12a)–(12c), are obtained by using the proposed algorithm. And when $S = \{a, b, c\}$ ($n = 3$), the solutions obtained by INGHS method are shown in (13a)–(13p). For other instances ($n > 3$), due to the existence of many solutions, we only present one solution for each instance. All is shown in (14)–(20).

The first solution on $n = 2$ is as follows:

$*$	a	b
a	a	b
b	a	b

Δ	a	b
a	a	a
b	b	b

(12a)

The second solution on $n = 2$ is as follows:

$*$	a	b
a	a	b
b	b	a

Δ	a	b
a	a	b
b	b	a

(12b)

The third solution on $n = 2$ is as follows:

$*$	a	b
a	a	b
b	b	b

Δ	a	b
a	a	b
b	b	b

(12c)

The first solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	b	b
c	c	b	c

Δ	a	b	c
a	a	b	c
b	b	b	b
c	c	b	c

(13a)

The second solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	b	b
c	a	b	c

Δ	a	b	c
a	a	b	a
b	b	b	b
c	c	b	c

(13b)

Table 1: The success rate comparison of the proposed method with HS and NGHS algorithm.

n	The number of solutions	HS			HSTL			NGHS			INGHS		
		Max looping times (ML)	Success times	Success rate	Max looping times (ML)	Success times	Success rate	Max looping times (ML)	Success times	Success rate	Max looping times (ML)	Success times	Success rate
2	3	3	3	**100.00%**	3	3	**100.00%**	3	3	**100.00%**	3	3	**100.00%**
3	16	16	15	93.75%	16	16	**100.00%**	16	16	**100.00%**	16	16	**100.00%**
4	Unknown	256	127	49.61%	256	148	57.81%	256	127	49.61%	256	162	**63.28%**
5	Unknown	625	485	77.60%	625	600	96.00%	625	579	92.64%	625	601	**96.16%**
6	Unknown	1296	1016	78.40%	1296	1258	97.07%	1296	1243	95.91%	1296	1271	**98.07%**
7	Unknown	2401	1834	76.38%	2401	2212	92.13%	2401	2119	88.25%	2401	2217	**92.34%**
8	Unknown	4096	79	1.93%	4096	3364	82.13%	4096	3174	77.49%	4096	3383	**82.59%**
9	Unknown	8000	0	0.00%	8000	5763	72.04%	8000	4768	59.60%	8000	6151	**76.89%**
10	Unknown	5000	0	0.00%	5000	3451	69.02%	5000	3251	65.02%	5000	3749	**74.98%**

The third solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	b	b
c	b	b	b

Δ	a	b	c
a	a	b	b
b	b	b	b
c	c	b	b

(13c)

The fourth solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	a	b	c
c	c	c	c

Δ	a	b	c
a	a	a	c
b	b	b	c
c	c	c	c

(13d)

The fifth solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	b	b
c	c	b	b

Δ	a	b	c
a	a	b	c
b	b	b	b
c	c	b	b

(13e)

The sixth solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	b	b
c	c	b	a

Δ	a	b	c
a	a	b	c
b	b	b	b
c	c	b	a

(13f)

The seventh solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	a	b	c
c	a	b	c

Δ	a	b	c
a	a	b	c
b	a	b	c
c	a	b	c

(13g)

The eighth solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	c	b
c	c	b	c

Δ	a	b	c
a	a	b	c
b	b	c	b
c	c	b	c

(13h)

The ninth solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	b	c
c	b	b	c

Δ	a	b	c
a	a	b	c
b	b	b	c
c	b	b	c

(13i)

The tenth solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	c	b	c
c	c	b	c

Δ	a	b	c
a	a	b	c
b	c	b	c
c	c	b	c

(13j)

The eleventh solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	c	c
c	c	c	c

Δ	a	b	c
a	a	b	c
b	b	c	c
c	c	c	c

(13k)

The twelfth solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	b	c
c	c	c	c

Δ	a	b	c
a	a	b	c
b	b	b	c
c	c	c	c

(13l)

The thirteenth solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	b	c
c	c	c	b

Δ	a	b	c
a	a	b	c
b	b	b	c
c	c	c	b

(13m)

The fourteenth solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	a	c
c	c	c	c

Δ	a	b	c
a	a	b	c
b	b	a	c
c	c	c	c

(13n)

The fifteenth solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	c	c	c
c	c	c	c

Δ	a	b	c
a	a	c	c
b	b	c	c
c	c	c	c

(13o)

The sixteenth solution on $n = 3$ is as follows:

$*$	a	b	c
a	a	b	c
b	b	c	a
c	c	a	b

Δ	a	b	c
a	a	b	c
b	b	c	a
c	c	a	b

(13p)

One of the solutions on $n = 4$ ($S = \{a, b, c, d\}$) is as follows:

$*$	a	b	c	d
a	a	b	c	d
b	a	b	c	d
c	d	c	c	d
d	d	c	c	d

Δ	a	b	c	d
a	a	a	d	d
b	b	b	c	c
c	c	c	c	c
d	d	d	d	d

(14)

One of the solutions on $n = 5$ ($S = \{a, b, c, d, e\}$) is as follows:

$*$	a	b	c	d	e
a	a	b	c	d	e
b	b	b	d	d	b
c	c	d	d	d	c
d	d	d	d	d	d
e	a	b	c	d	e

Δ	a	b	c	d	e
a	a	b	c	d	a
b	b	b	d	d	b
c	c	d	d	d	c
d	d	d	d	d	d
e	e	b	c	d	e

(15)

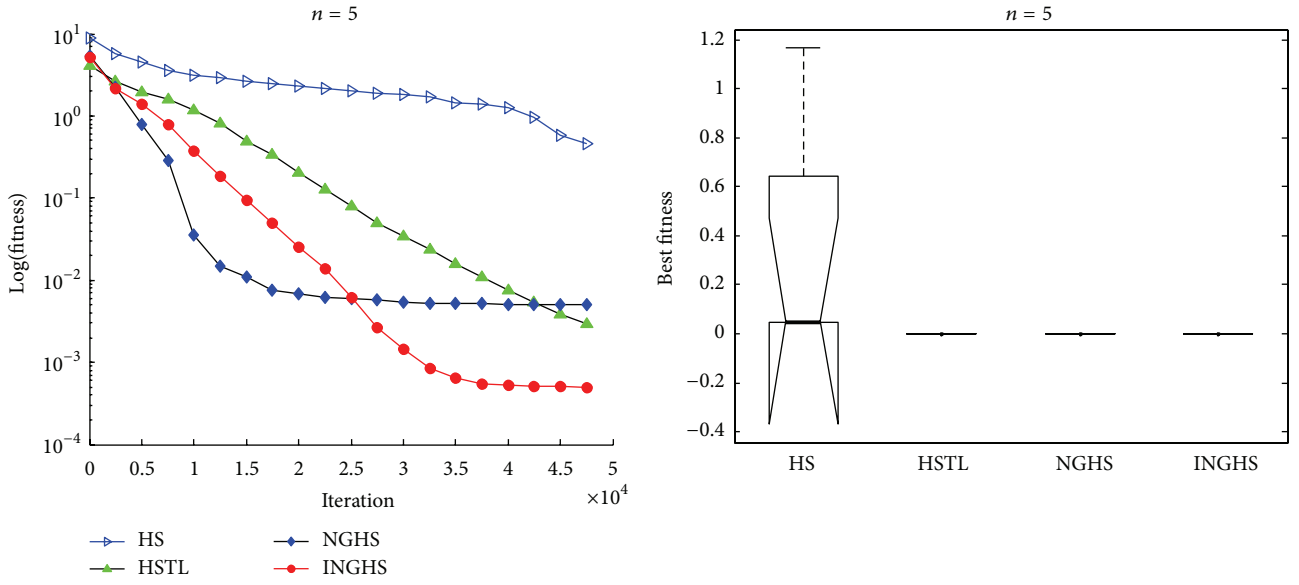


Figure 5: The convergence curve and the box plot when $n = 5$.

One of the solutions on $n = 6$ ($S = \{a, b, c, d, e, f\}$) is as follows:

$*$	a	b	c	d	e	f
a	a	b	c	d	e	f
b	a	b	c	d	e	f
c	c	c	c	d	c	c
d	d	d	d	d	d	d
e	e	e	c	d	e	e
f	a	b	c	d	e	f

Δ	a	b	c	d	e	f
a	a	a	c	d	e	a
b	b	b	c	d	e	b
c	c	c	c	d	c	c
d	d	d	d	d	d	d
e	e	e	c	d	e	e
f	f	f	c	d	e	f

(16)

One of the solutions on $n = 7$ ($S = \{a, b, c, d, e, f, g\}$) is as follows:

$*$	a	b	c	d	e	f	g
a	a	b	c	d	e	f	g
b	a	b	c	d	e	f	g
c	a	b	c	d	e	f	g
d	d	e	d	d	e	f	e
e	d	e	d	d	e	f	e
f	f	f	f	f	f	f	f
g	d	e	d	d	e	f	e

Δ	a	b	c	d	e	f	g
a	a	a	a	d	d	f	d
b	b	b	b	e	e	f	e
c	c	c	c	d	d	f	d
d	d	d	d	d	d	f	d
e	e	e	e	e	e	f	e
f	f	f	f	f	f	f	f
g	g	g	g	e	e	f	e

(17)

One of the solutions on $n = 8$ ($S = \{a, b, c, d, e, f, g, h\}$) is as follows:

$*$	a	b	c	d	e	f	g	h
a	a	b	c	d	e	f	g	h
b	a	b	c	d	e	f	g	h
c	c	c	c	c	c	c	c	c
d	d	d	c	c	c	c	d	d
e	e	e	c	c	e	f	e	e
f	c	c	c	c	c	c	c	c
g	a	b	c	d	e	f	g	h
h	a	b	c	d	e	f	g	h

Δ	a	b	c	d	e	f	g	h
a	a	a	c	d	e	c	a	a
b	b	b	c	d	e	c	b	b
c	c	c	c	c	c	c	c	c
d	d	d	c	c	c	c	d	d
e	e	e	c	c	e	c	e	e
f	f	f	c	c	f	c	f	f
g	g	g	c	d	e	c	g	g
h	h	h	c	d	e	c	h	h

(18)

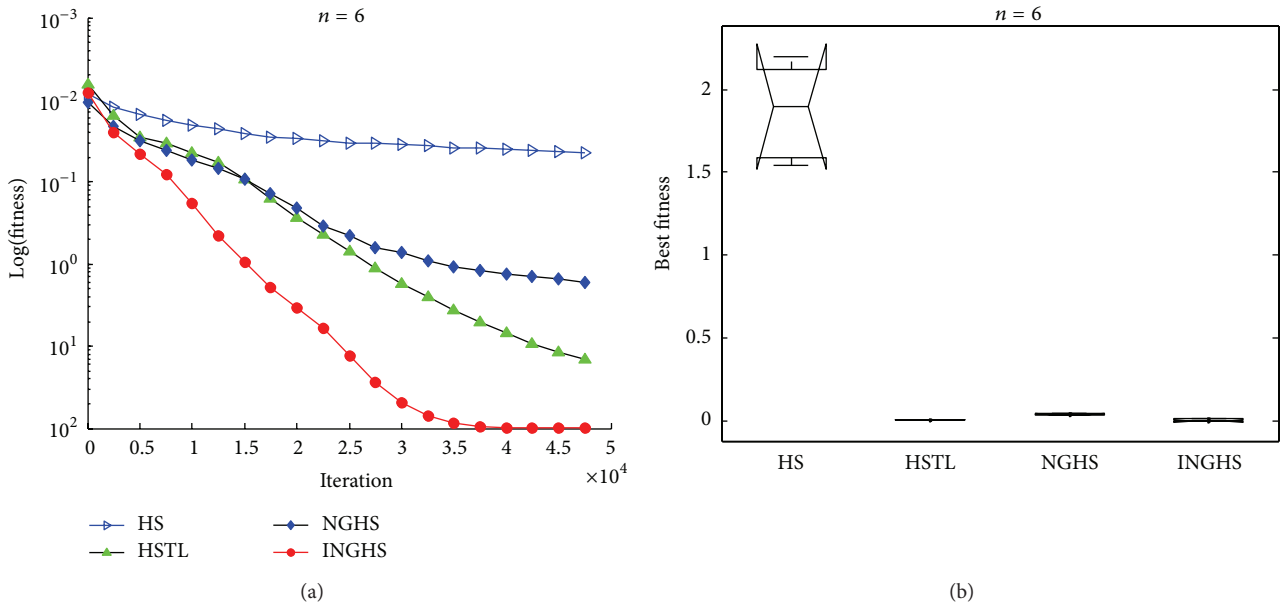


FIGURE 6: The convergence curve and the box plot when $n = 6$.

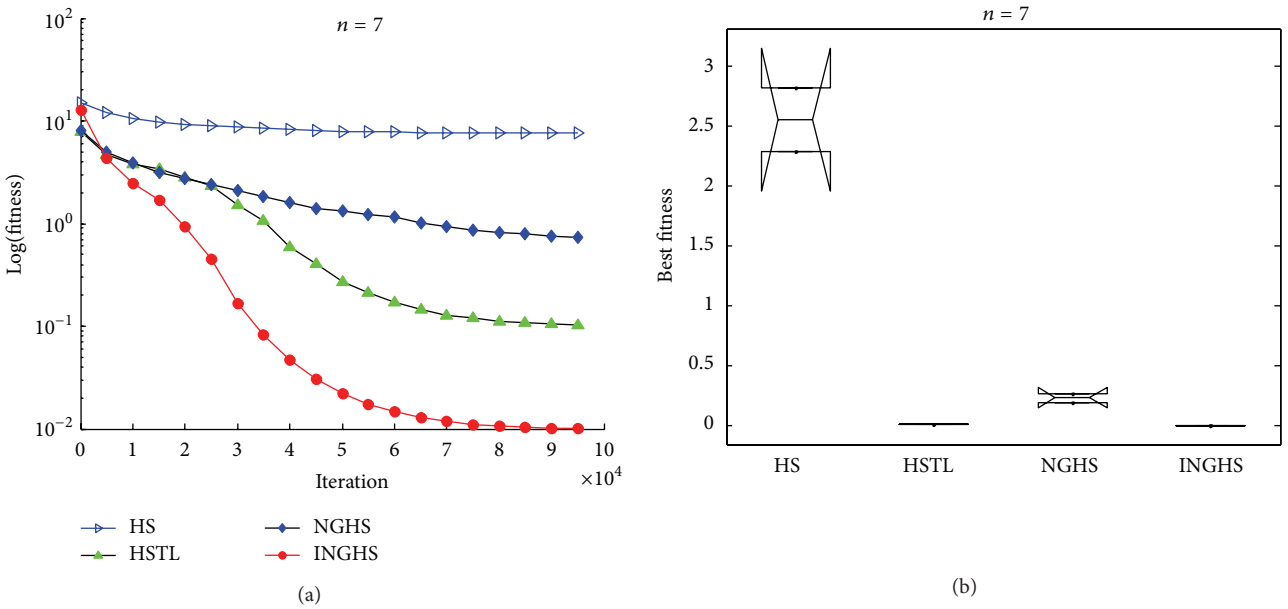


FIGURE 7: The convergence curve and the box plot when $n = 7$.

One of the solutions on $n = 9$ ($S = \{a, b, c, d, e, f, g, h, i\}$) is as follows:

$*$	a	b	c	d	e	f	g	h	i
a	a	b	c	d	e	f	g	h	i
b	b	b	b	b	b	b	b	b	b
c	c	b	c	b	e	e	g	b	c
d	d	b	b	b	b	d	b	b	d
e	c	b	c	b	e	e	g	b	c
f	a	b	c	d	e	f	g	h	i
g	b	b	b	b	b	b	b	b	b
h	b	b	b	b	b	b	b	b	b
i	a	b	c	d	e	f	g	h	i

Δ	a	b	c	d	e	f	g	h	i
a	a	b	c	d	c	a	b	b	a
b	b	b	b	b	b	b	b	b	b
c	c	b	c	b	c	c	b	b	c
d	d	b	b	b	b	d	b	b	d
e	e	b	e	b	e	e	b	b	e
f	f	b	e	d	e	f	b	b	f
g	g	b	g	b	g	g	b	b	g
h	h	b	b	b	b	h	b	b	h
i	i	b	c	d	c	i	b	b	i

(19)

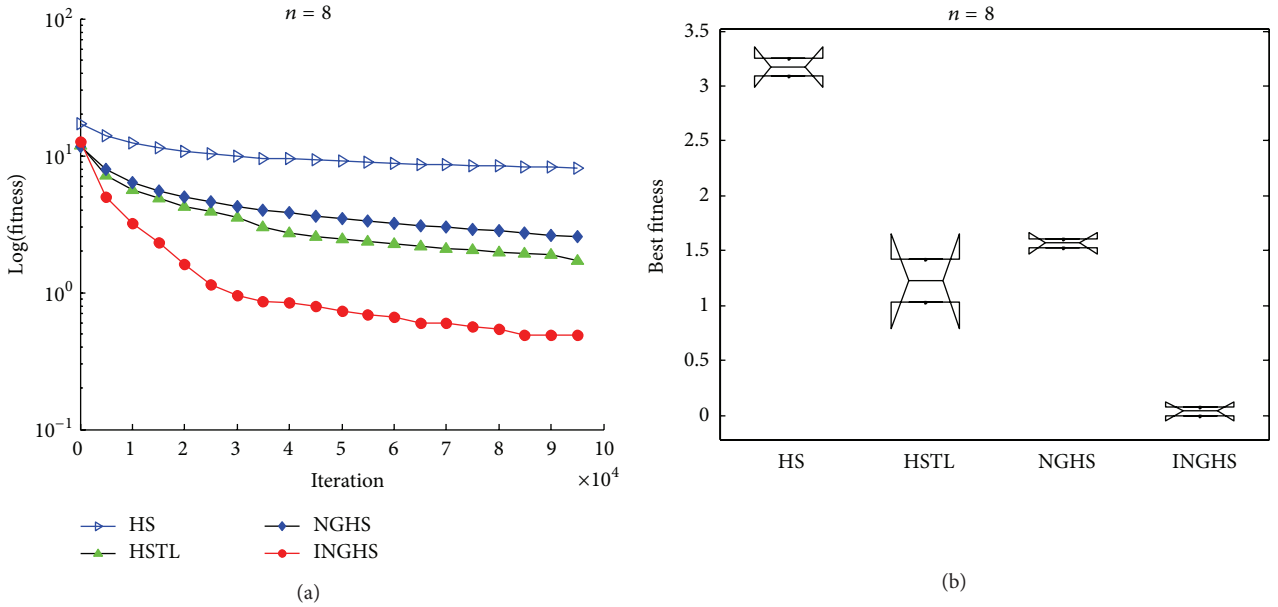


Figure 8: The convergence curve and the box plot when $n = 8$.

One of the solutions on $n = 10$ ($S = \{a, b, c, d, e, f, g, h, i, j\}$) is as follows:

$*$	a	b	c	d	e	f	g	h	i	j
a	a	b	c	d	e	f	g	h	i	j
b	e	e	c	e	e	e	g	e	e	e
c	e	e	c	e	e	e	g	e	e	e
d	d	e	c	f	e	e	g	e	e	e
e	e	e	c	e	e	e	g	e	e	e
f	f	e	c	e	e	e	g	e	e	e
g	e	e	c	e	e	e	g	e	e	e
h	h	b	c	e	e	e	g	h	e	e
i	i	e	c	e	e	e	g	e	e	e
j	e	e	c	e	e	e	g	e	e	e

Δ	a	b	c	d	e	f	g	h	i	j
a	a	e	e	d	e	f	e	h	i	e
b	b	e	e	e	e	e	e	b	e	e
c	c	c	c	c	c	c	c	c	c	c
d	d	e	e	f	e	e	e	e	e	e
e	e	e	e	e	e	e	e	e	e	e
f	f	e	e	e	e	e	e	e	e	e
g	g	g	g	g	g	g	g	g	g	g
h	h	e	e	e	e	e	e	h	e	e
i	i	e	e	e	e	e	e	e	e	e
j	j	e	e	e	e	e	e	e	e	e

(20)

From Table 1, it can be seen obviously that the success rate of all instances of the proposed method is higher than HS, HSTL, and NGHS.

For instance, $n = 6, 7, 8, 9$, and 10, and Figures 5, 6, 7, 8, 9, and 10 show the convergence curves of three algorithms (HS, NGHS, and INGHS). It is evident from Figures 5–10 that the proposed algorithm is better than HS and NGHS for all instances. The convergence curve of the proposed algorithm can maintain falling until it finds the best solution. When $n > 6$, the convergence curve of INGHS method falls faster than HS, HSTL, and NGHS algorithms, and it keeps declining actively until the best solution is found.

5. Conclusions

This paper has investigated nine construction example problems of algebra system and converted the construction example problems into optimization problems. A novel harmony search algorithm (INGHS) is proposed to solve the problems. Global best strategy and dynamic parameters adjustment are employed in INGHS. The experimental results on 9 instances of algebra system demonstrate that the proposed algorithm is more effective than HS, HSTL, and NGHS algorithms. Further research will investigate the INGHS algorithm to solve the combinatorial optimization problems and some practical optimization problems.

Conflict of Interests

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

Acknowledgments

The authors would like to thank all the editors and referees for their valuable comments. This work was supported by the National Natural Science Foundation of China under Grant no. 11401357 and no. 81160183, the Scientific Research Program funded by Shaanxi Provincial Education Department

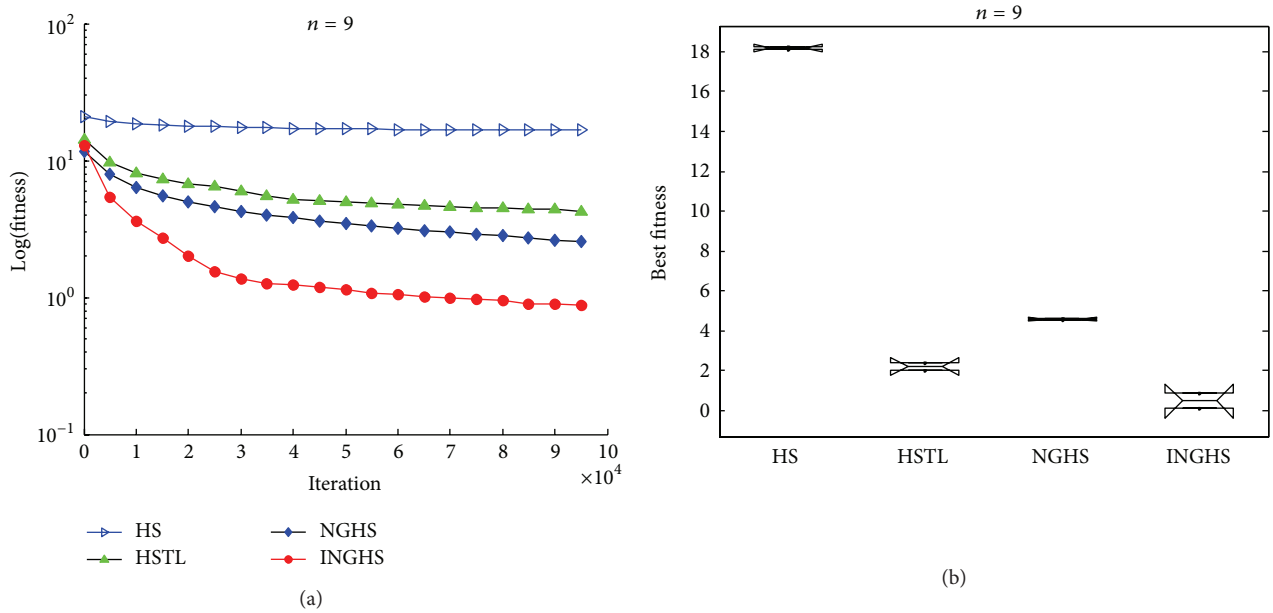


Figure 9: The convergence curve and the box plot when $n = 9$.

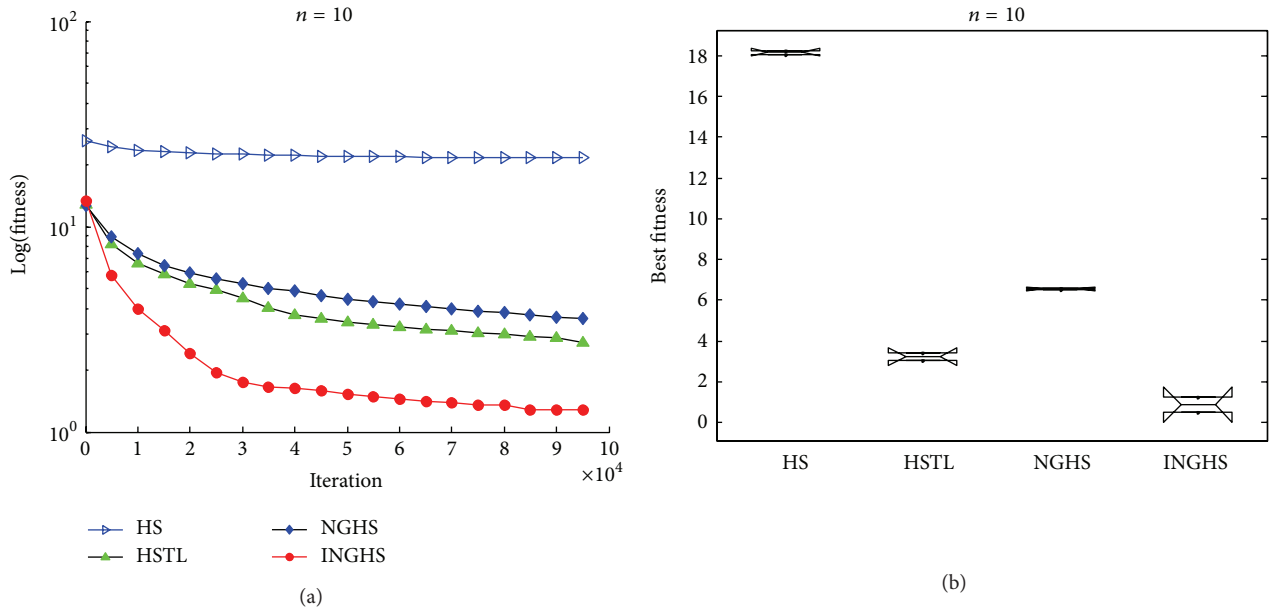


Figure 10: The convergence curve and the box plot when $n = 10$.

under Grant no. 14JK1141, and Hanzhong Administration of Science & Technology under Grant no. 2013hzzx-39.

References

[1] F. Deng and Y. Xu, "On the $N(2,2,0)$ algebra," *Journal of Southwest Jiao Tong University*, vol. 31, no. 4, pp. 457–653, 1996.

[2] F. Deng and L. Yong, "Regular semigroups of an $N(2,2,0)$ algebra," *Journal of Advances in Mathematics*, vol. 41, no. 6, pp. 665–671, 2012.

[3] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, "Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 1, pp. 89–106, 2011.

[4] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[5] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, 2005.

[6] Q.-K. Pan, P. N. Suganthan, J. J. Liang, and M. F. Tasgetiren, "A local-best harmony search algorithm with dynamic subpopulations," *Engineering Optimization*, vol. 42, no. 2, pp. 101–117, 2010.

[7] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, "A self-adaptive global best harmony search algorithm for continuous optimization problems," *Applied Mathematics and Computation*, vol. 216, no. 3, pp. 830–848, 2010.

[8] P. Yadav, R. Kumar, S. K. Panda, and C. S. Chang, "An intelligent tuned harmony search algorithm for optimisation," *Information Sciences*, vol. 196, pp. 47–72, 2012.

[9] D. Zou, L. Gao, J. Wu, and S. Li, "Novel global harmony search algorithm for unconstrained problems," *Neurocomputing*, vol. 73, no. 16–18, pp. 3308–3318, 2010.

[10] D. Zou, L. Gao, J. Wu, S. Li, and Y. Li, "A novel global harmony search algorithm for reliability problems," *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 307–316, 2010.

[11] Z. W. Geem, *Recent Advances in Harmony Search Algorithm*, Springer, New York, NY, USA, 2010, https://sites.google.com/a/hydroteq.com/www/HS_Structure. pdf.

[12] S. Tuo and L. Yong, "Improved harmony search algorithm with chaos," *Journal of Computational Information Systems*, vol. 8, no. 10, pp. 4269–4276, 2012.

[13] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643–656, 2008.

[14] A. Kattan and R. Abdullah, "A dynamic self-adaptive harmony search algorithm for continuous optimization problems," *Applied Mathematics and Computation*, vol. 219, no. 16, pp. 8542–8567, 2013.

[15] L. Wang, R. Yang, Y. Xu, Q. Niu, P. M. Pardalos, and M. Fei, "An improved adaptive binary harmony search algorithm," *Information Sciences*, vol. 232, pp. 58–87, 2013.

[16] H. Sarvari and K. Zamanifar, "Improvement of harmony search algorithm by using statistical analysis," *Artificial Intelligence Review*, vol. 37, no. 3, pp. 181–215, 2012.

[17] M. Hadwan, M. Ayob, N. R. Sabar, and R. Qu, "A harmony search algorithm for nurse rostering problems," *Information Sciences*, vol. 233, pp. 126–140, 2013.

[18] N. Sinsuphan, U. Leeton, and T. Kulworawanichpong, "Optimal power flow solution using improved harmony search method," *Applied Soft Computing Journal*, vol. 13, no. 5, pp. 2364–2374, 2013.

[19] H. Wang, X. Yuan, Y. Wang, and Y. Yang, "Harmony search algorithm-based fuzzy-PID controller for electronic throttle valve," *Neural Computing and Applications*, vol. 22, no. 2, pp. 329–336, 2013.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 706897, 9 pages
http://dx.doi.org/10.1155/2015/706897

Research Article

Developing a Robust Surrogate Model of Chemical Flooding Based on the Artificial Neural Network for Enhanced Oil Recovery Implications

Mohammad Ali Ahmadi

Department of Petroleum Engineering, Ahwaz Faculty of Petroleum Engineering, Petroleum University of Technology, P.O. Box 63431, Ahwaz 6198144471, Iran

Correspondence should be addressed to Mohammad Ali Ahmadi; ahmadi6776@yahoo.com

Received 10 June 2014; Revised 5 October 2014; Accepted 13 October 2014

Academic Editor: Shifei Ding

Application of chemical flooding in petroleum reservoirs turns into hot topic of the recent researches. Development strategies of the aforementioned technique are more robust and precise when we consider both economical points of view (net present value, NPV) and technical points of view (recovery factor, RF). In current study many attempts have been made to propose predictive model for estimation of efficiency of chemical flooding in oil reservoirs. To gain this end, a couple of swarm intelligence and artificial neural network (ANN) is employed. Also, lucrative and high precise chemical flooding data banks reported in previous attentions are utilized to test and validate proposed intelligent model. According to the mean square error (MSE), correlation coefficient, and average absolute relative deviation, the suggested swarm approach has acceptable reliability, integrity and robustness. Thus, the proposed intelligent model can be considered as an alternative model to predict the efficiency of chemical flooding in oil reservoir when the required experimental data are not available or accessible.

1. Introduction

The oil and gas upstream industries are recently encountered with the difficulties and challenges of dealing with hydrocarbon resources whose productions with conventional technologies are following an upward trend of technical limitations. It is because of achieving the stage of decline phase by most of oilfields around the world. Therefore, how to postpone the abandonment of reservoirs has tuned into the priority of researchers in the worldwide. Their researches normally highlight the concept of great necessities for inventions of new techniques, normally classified as tertiary oil recovery methods, having abilities of maintaining the economic production rate [1–3].

Chemical enhanced oil recovery approaches as one of the most effective subsets of tertiary methods are known as a key to unlock the exploitation of referred resources. Different methods for this process have been developed, such as polymer, surfactant/polymer (SP), and alkaline/surfactant/polymer (ASP) flooding. These methods are applied to increase the rate of oil production through focusing on both lowering the interfacial tension and reducing the water mobility. In more details, it has enormously been declared in previous literatures that in order to design, manage, and run a chemical enhanced oil recovery operation it is highly required to set very expensive and time-consuming but precise experimental procedures which their generated results must be gained to plan effectively the process of injecting chemical materials [4–9].

The laboratorial generated outputs are then used to conclude two parameters, recovery factor (RF) and net present value (NPV), which are used to evaluate the performance of the chemical flooding which is one of the most popular methods of chemical enhanced oil recovery. Having knowledge about these two parameters is essentially vital to make decisions if it is beneficial to run the referred operation. Unfortunately, there are no global methods to interpret simultaneously both aforementioned factors although there are numerous numbers of different software and numerical or

Table 1: Statistical analysis of the implemented chemical flooding data samples [9].

Parameter	Unit	Type	Min.	Max.	Average	Standard deviation
Surfactant slug size	PV	Input	0.097	0.259	0.177	0.072
Surfactant concentration	Vol. fraction	Input	0.005	0.03	0.017	0.011
Polymer concentration in surfactant slug	wt.%	Input	0.1	0.25	0.177	0.067
Polymer drive size	PV	Input	0.324	0.648	0.482	0.144
Polymer concentration in polymer drive	wt.%	Input	0.1	0.2	0.148	0.044
K_v/K_h ratio	—	Input	0.01	0.25	0.129	0.107
Salinity of polymer drive	Meq/mL	Input	0.3	0.4	0.349	0.045
Recovery factor (RF)	%	Output	14.82	56.99	39.67	9.24
Net present value (NPV)	$ MM	Output	1.781	7.229	4.45	1.53

analytical methods which are capable of making very precise quantitative decisions about the amount of one of the RF or NPV [10–12].

Hence, there is a great need in oilfield for having access to a solution or model which can predict the amount of these two parameters at the same time. The major aim of current study is to execute new kind of artificial intelligence approaches to suggest robust and accurate predictive method to forecast efficiency of the chemical flooding through petroleum reservoirs. To gain successfully this referred goal, hybridization of artificial neural network and particle swarm optimization (PSO) was executed on the previous literature data bases. The integrity and performance of the proposed predictive approaches in estimating recovery factor (RF) and net present value (NPV) from the literature are described in details.

2. Data Gathering

The data utilized throughout this research have been gathered from previous attentions [9] in which chemical flooding had been simulated in Benoist sand reservoir, by executing UTCHEM simulator. That reservoir has been produced under primary and secondary processes over fifty years. The original dataset contained 202 data. Each data had 7 inputs: surfactant slug size, surfactant concentration in surfactant slug, polymer concentration in surfactant slug, polymer drive size, and polymer concentration in polymer drive, K_v/K_h ratio, and salinity of polymer drive. In addition, the outputs were RF and NPV. The ranges of implemented data banks are reported in Table 1 [9].

3. Artificial Neural Network and Particle Swarm Optimization

Artificial neural network (ANN) includes simple nodes, named as neurons, which are bonded to each other to construct a network model. Indeed, the biological nervous systems can be simulated with the ANN system, somehow. Characterization of an ANN model is normally performed through three ways including (a) certain patterns between various layers, (b) connection between input and output via activation function, and (c) updating the interconnection weights through training process [13–24].

In fact, the main purpose of an ANN model is to determine target function through internal computation during the training phase if the values of input variables are provided. The most common type of ANN is the multilayer feed forward neural network which is made up of group of interconnected neurons organized in the form of layers: input layer, hidden layer(s), and output layer where each layer comprises a group of neurons as presented in Figure 1. This network is strictly an acyclic type since signals propagate only in a forward direction from the input neurons to the output neurons and no signals are allowed to be fed-back among the neurons. The number of neurons in the input and output layers is decided by the number of input and output variables that are planned for the predictive tool. However, the optimal number of neurons in hidden layer(s) is a strong function of nonlinearity and dimensionality of the problem under study [13–24].

The artificial neuron is the fundamental part of the neural networks. Each artificial neuron—excluding neurons at the input layer—takes and processes inputs gathered from other neurons. Given further information, each artificial neuron is a mathematical information-processing unit. The processed information is presented at the output end of the neuron. Figure 2 addresses the procedure in which an artificial neuron treats the data and information entered in the model. Each input signal (a_k) is primarily multiplied by the corresponding weight value (w_{kj}) and the resultant products are summed up to generate a total weight in the form of $w_{j1}a_1 + w_{j2}a_2 + \cdots + w_{jm}a_m$. The sum of the weighted inputs and the bias ($S_j = \sum_{k=1}^{m} w_{jk} \cdot a_k + b_j$) forms the input to the activation function, φ. An activation function processes this sum and gives out the output, o_j. Indeed, the resulting sum is processed by a neuron activation function to obtain the ultimate output of the neuron as follows [13–26]:

$$o_j = \varphi\left(S_j\right) = \varphi\left(\sum_{k=1}^{m} w_{jk} \cdot a_k + b_j\right). \quad (1)$$

This output will be the input signal for the neurons in the following layer. The linear (purelin) transfer, tan-sigmoid (tansig) activation, and log-sigmoid (logsig) activation functions are mostly employed in the practical cases with applications in science and engineering disciplines. The corresponding

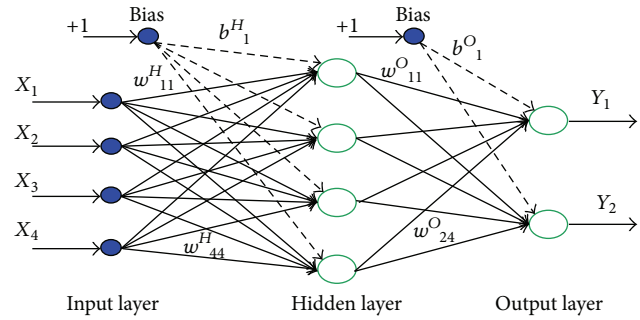


FIGURE 1: Architecture of multilayer feed forward neural network. The symbol w^H_{qr} denotes the synaptic weight between the output of the rth neuron in the hidden layer and the input of the qth neuron in output layer. The symbol b^H_q denotes the bias of the qth neuron in hidden layer. The superscript O stands for output layer.

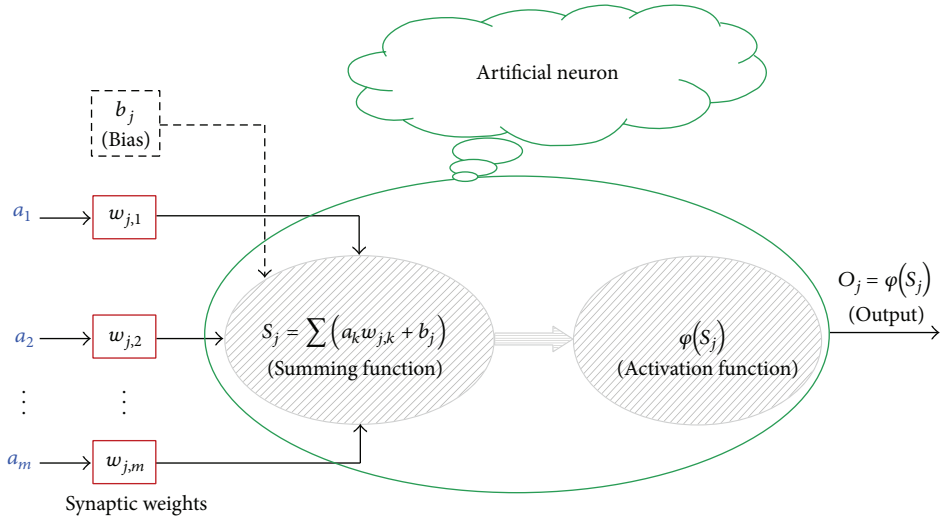


FIGURE 2: Information processing by an artificial neuron.

relationships for these functions are defined, respectively, by (2)–(4), as given below [13–27]:

$$\varphi(s) = s, \tag{2}$$

$$\varphi(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}, \tag{3}$$

$$\varphi(s) = \frac{1}{1 + e^{-s}}. \tag{4}$$

The weight factors are generally considered as the adaptive parameters in the network to obtain the strength of the input signals. A bias is characterized with a weight which is not responsible for connecting an input of two neurons to an output. A particular level of a neuron output signal is represented by a set of bias that does not depend on the input signals. The weight factors and biases are tuned during the course of training phase such that the network is able to forecast the accurate target parameter for a given set of inputs. There are a number of training algorithms with different methodologies in the context of intelligence system. A variety of optimization tools such as particle swarm optimization (PSO) [15, 18, 19], genetic algorithm (GA) [21], hybrid genetic algorithm and particle swarm optimization (HGAPSO) [13, 16], unified particle swarm optimization (UPSO) [14], and imperialist competitive algorithm (ICA) [17, 20, 23] for weight training of neural networks have been used.

Kennedy [27] introduced the PSO as a strong stochastic optimization technique which simulates the social manners of birds within a group, based on population concept. It searches for an optimum solution by iteratively updating a swarm of particles.

The model originally includes a group of random particles (solutions). A random velocity is attributed to each candidate particle which flies within the problem space. The solutions consist of memory and try to attain the best position or/and fitness. This parameter is symbolized by "p_{best}" that is linked only to a specific particle. The model also retains the best fitness, known as "g_{best}," which is found among the entire solutions (particles) in the swarm. The candidate particle that obtains this fitness is the global best in the population [25–28]. In the current study, a particle's fitness is calculated through determination of the network output for every point in the training part and then computing the sum of squares of the resultant errors (MSE) for performance evaluation.

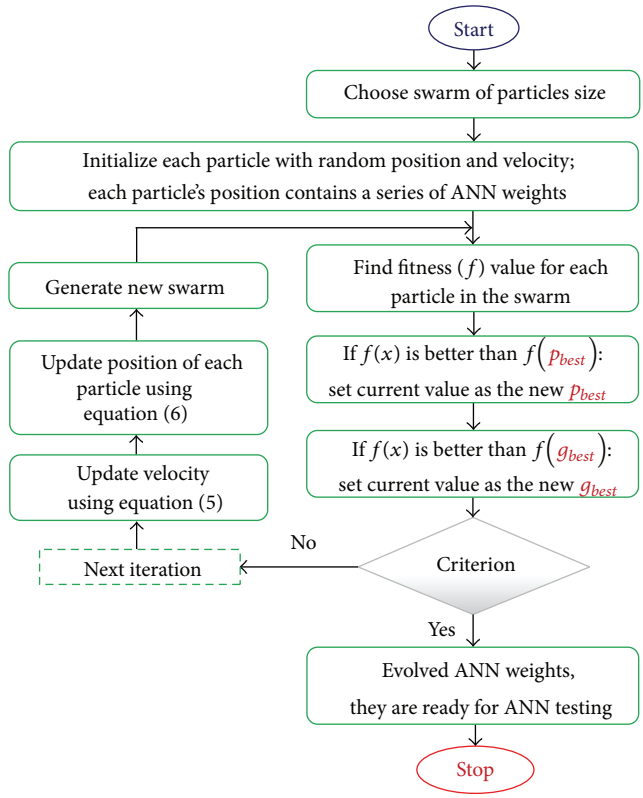


Figure 3: PSO-based algorithm flowchart in optimization of the weights of ANN.

The basic PSO theory involves variation of each particle velocity toward its p_{best} and g_{best} locations at each time interval. The particles' new velocity and position are updated according to the following equations [13–28]:

$$v_i^{n+1} = \omega v_i^n + c_1 r_1^n \left[x_{i,p_{\text{best}}}^n - x_i^n\right] + c_2 r_2^n \left[x_{g_{\text{best}}}^n - x_i^n\right], \quad (5)$$

$$x_i^{n+1} = x_i^n + v_i^{n+1}, \quad (6)$$

where v_i^n and v_i^{n+1} are velocities of particle i at iterations n and $n + 1$; x_i^n and x_i^{n+1} are positions of particle i at iterations n and $n + 1$; ω represents the inertia weight that directs the exploitation and exploration of the search space as it continuously updates velocity; c_1 and c_2 are termed as cognition and social components, respectively. They are considered as the acceleration constants which alter the velocity of a solution in the direction of p_{best} and g_{best} [13–28]; and r_1^n and r_2^n refer to the two random variables uniformly distributed in the interval of $[0, 1]$.

Herein, PSO algorithm has been used in evolving weights of multilayer feed forward neural network. In this case, a particle's position at any iteration is described as a particle whose coordinates are connection weights. The vectors of weights for each particle i will be called x_i. Throughout the training process the above equations (equations (5) and (6)) will customize the network weights until a criterion is met. In this case, a lower MSE, as a sufficiently good fitness, is achieved; nevertheless, a maximum number of iterations are used to terminate the iterative search process if no improvement is observed over a number of consecutive generations in an appropriate time. The flowchart of the PSO-based training algorithm for the ANN is shown in Figure 3.

The PSO utilizes a random procedure in the search space of the problem such that particles in the population are directed toward optimum positions but not in or between optimal areas [27]. Thus, PSO can be used to train neural networks with nondifferentiable (even discontinuous) neurons activation functions. It can be also implemented in cases where gradient or error information is not accessible. PSO is easy to implement and there are few parameters to be adjusted. However, the uniqueness of the algorithm lies in the dynamic interactions among the particles that turn it into a social-psychological model of knowledge management [27].

4. Results and Discussion

According to the study accomplished by Cybenko [29], a network that consists of only one single hidden layer has the ability to approximate nearly any kind of nonlinear function. However, determination of the ideal number of neurons in the hidden layer is a challenging task; few neurons will not give adequate precision and too many hidden neurons may lead to overfitting. It means that the training data might be fitted adequately; however considerable oscillations between the points are noticed in the fitting curve, resulting in poor

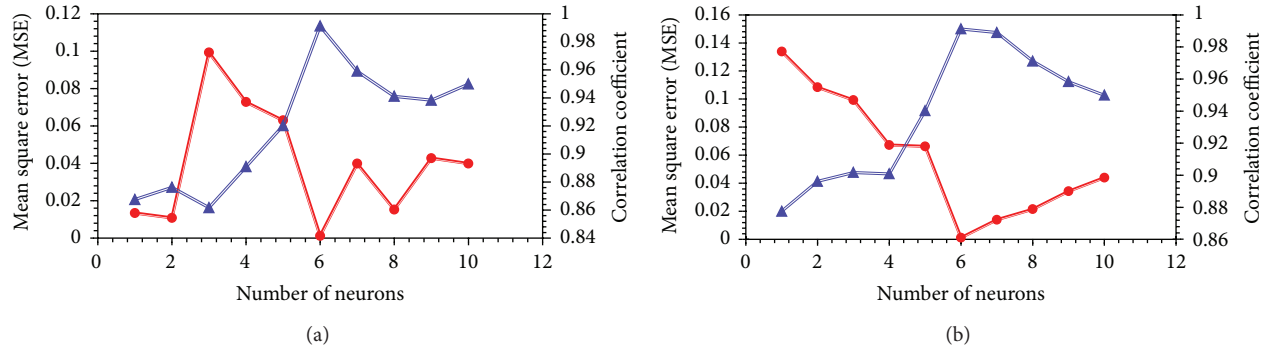


Figure 4: Effect of number of hidden neuron on PSO-ANN accuracy of (a) recovery and (b) NPV predictions in terms of MSE and R-squared.

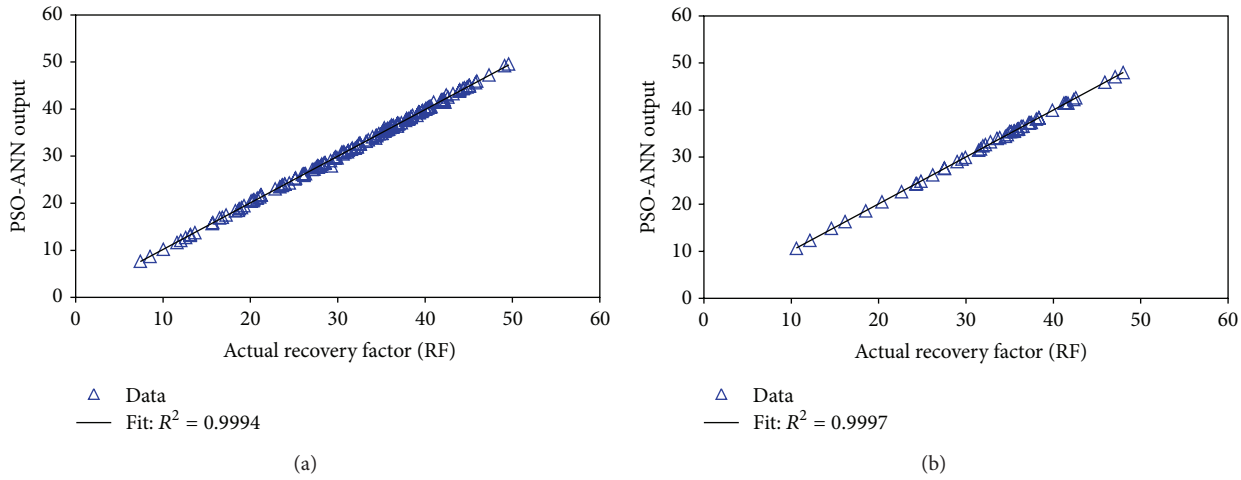


Figure 5: Performance plot of the suggested network model for determining recovery factor of chemical flooding owing to correlation coefficient (R^2): (a) training phase and (b) testing phase.

interpolation and extrapolation. The network performance is evaluated as demonstrated in Figure 4, when different number of neurons is tested. A smart model with one hidden layer (including just one neuron) was primarily built in the current study to predict the recovery factor and net present value (NPV) of chemical flooding in oil reservoirs. Prediction accuracy was further analyzed by an increase in the number of neurons to 10 to decide on the most precise technique. As clear from the results demonstrated in Figure 4, a 3-6-1 architecture (6 neurons in the hidden layer, 3 neurons in the input layer, and one neuron in output layer) offers the best model for recovery factor and net present value (NPV) prediction in terms of MSE and R^2, since the optimum structure achieved by the trial and error procedure has a very low mean squared error of MSE = 0.0012 and a satisfactory coefficient of determination of R^2 = 0.9996, on the basis of comparison between the predicted and real data.

The generated results of the proposed intelligent approach are depicted through Figures 5 to 10. The existing contrasts between suggested intelligent approach and related recovery factor (RF) of the chemical flooding in oil reservoir in the regression plot have been depicted in Figure 5. As shown in Figure 5 which is a graphical and scatter presentation of the PSO-ANN results versus corresponding determined recovery factor (RF) data, the PSO-ANN outputs lie over the line $Y = X$, the fact that indicates the identity of outputs gained from suggested PSO-ANN model and relevant recovery factor data samples. To serve better understanding about generated results of the proposed PSO-ANN model, the comparison between gained recovery factor from the addressed model and real recovery factor data versus relevant data index has been illustrated in Figure 6. As illustrated in Figure 6, the obtained results of proposed model are as close as possible to real recovery factor (RF) data samples. To put it another way, the outputs of the PSO-ANN approach have the same behaviour as actual data do. The high considerable level of efficiency and accuracy related to the PSO-ANN approach in prediction of the recovery factor dataset of chemical flooding has once again been certified in Figure 6. Moreover, the robustness of the PSO-ANN has been demonstrated in terms of the relative deviations of PSO-ANN model outputs from corresponding determined recovery factor data in Figure 7. As could be observed in Figure 7, the highest deviations of the suggested approach results are subjected to the early boundary of recovery factor data samples. 5% is the maximum degree of relative deviation shown in Figure 7.

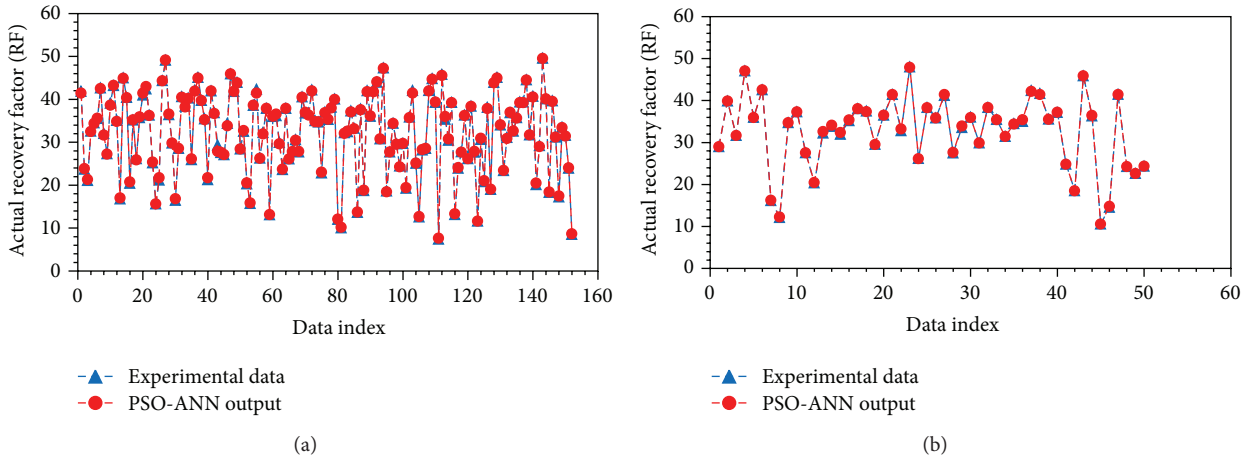


Figure 6: Comparison between suggested network model and recovery factor versus relevant data index: (a) training phase and (b) testing phase.

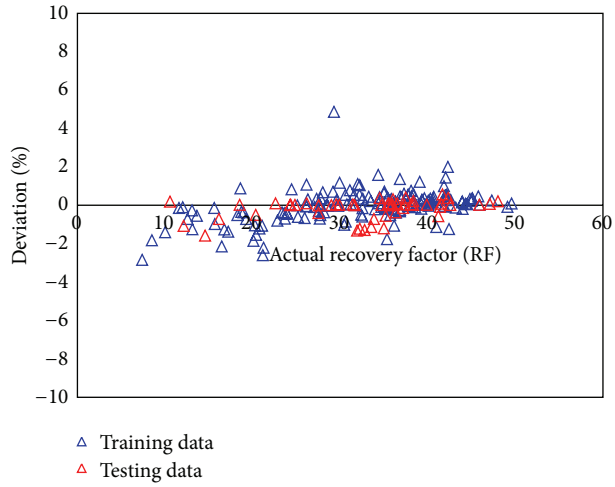


Figure 7: Relative error distribution of the proposed approach versus actual recovery factor (RF).

The draw parallel between our proposed intelligent PSO-ANN model results and related net present value (NPV) of the chemical flooding in oil reservoir in the regression plot has been shown in Figure 8. As shown in Figure 8 which is a graphical and scatter presentation of the PSO-ANN results versus corresponding determined net present value (NPV) data, the PSO-ANN outputs lie over the line $Y = X$, the fact that indicates the identity of outputs gained from suggested PSO-ANN model and relevant net present value (NPV) data samples. The comparison between generated net present value (NPV) from the addressed approach and real net present value (NPV) data versus relevant data index has been shown in Figure 9. As illustrated in Figure 9, the obtained results of proposed model are as close as possible to net present value (NPV) data samples. To put it another way, the outputs of the PSO-ANN approach have the same behaviour as actual data do. Furthermore, the effectiveness of the proposed intelligent model has been depicted in terms of the relative deviations of PSO-ANN model outputs from corresponding indicated net present value (NPV) data in Figure 10. As can be seen from Figure 10, the highest deviations of the suggested approach results are subjected to the early boundary of net present value (NPV) data. 6% is the maximum degree of relative deviation depicted in Figure 10.

Table 2: Statistical parameters of the proposed approaches in prediction of efficiency of chemical flooding in oil reservoirs.

	PSO-ANN	
	RF	NPV
Correlation coefficient (R^2)	0.9997	0.9996
Mean square error (MSE)	0.0012	0.0015
Mean absolute error (MAE)	0.098	0.0206

The performance efficiency of the selected network is assessed using the various error analysis parameters. Table 2 tabulates the PSO-ANN accuracy in terms of correlation coefficient (R), coefficient of determination (R^2), mean absolute

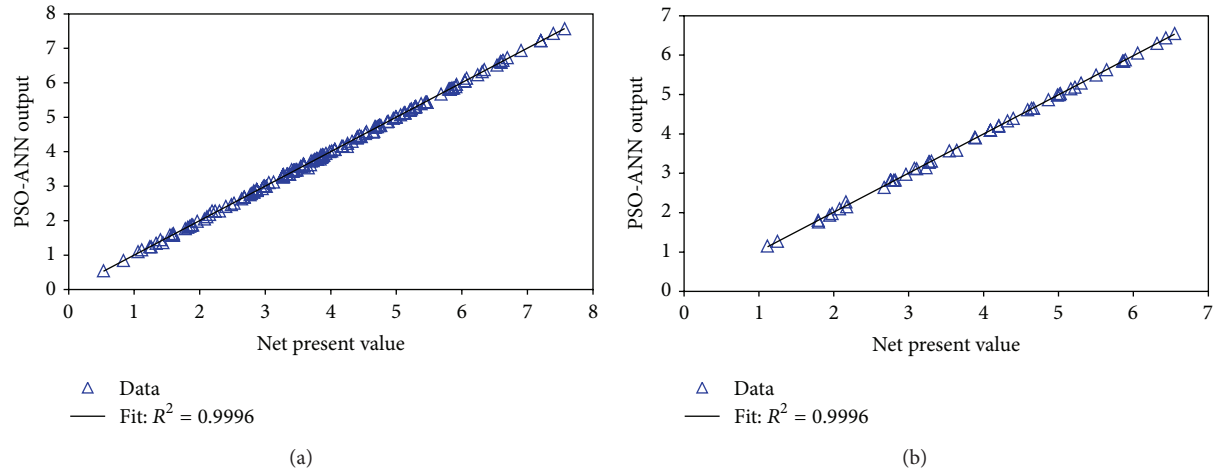


Figure 8: Performance plot of the suggested network model for determining net present value (NPV) of chemical flooding owing to correlation coefficient (R^2): (a) training phase and (b) testing phase.

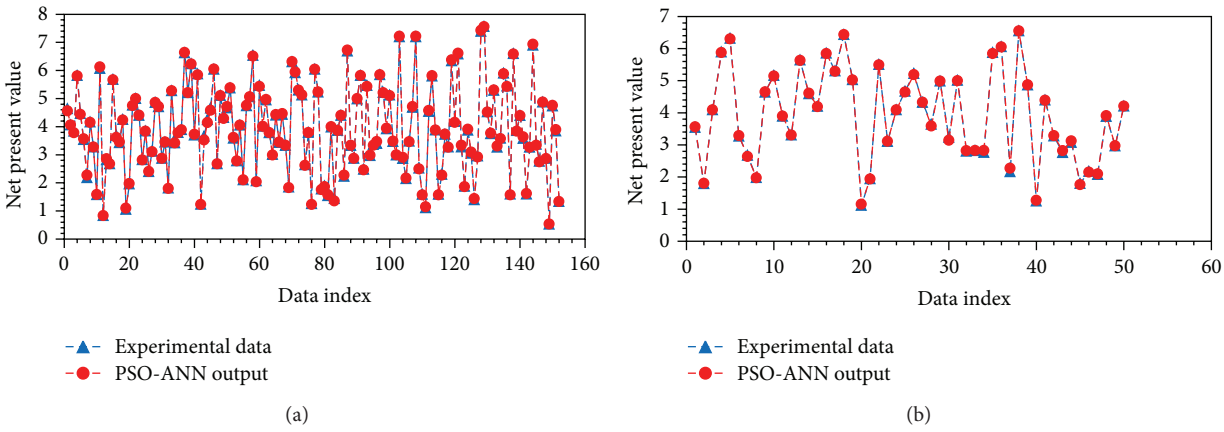


Figure 9: Comparison between suggested network model and net present value (NPV) versus relevant data index: (a) training phase and (b) testing phase.

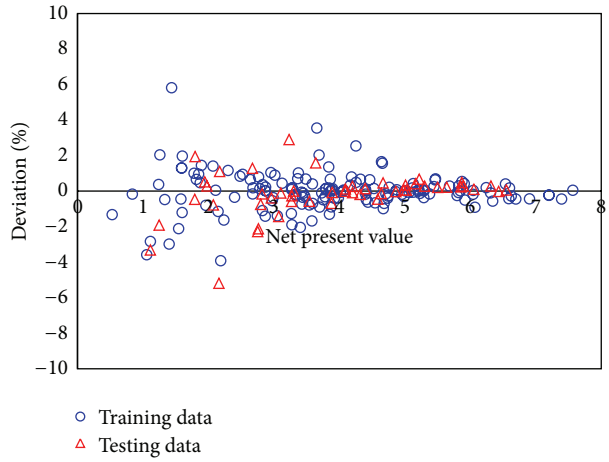


Figure 10: Relative error distribution of the proposed approach versus actual net present value (NPV).

error (MAE), and mean squared error (MSE), which are defined as follows:

$$R = \frac{\sum_{i=1}^{N}\left(y^{T}{}_{i} - \overline{y^{T}}\right)\left(y^{P}{}_{i} - \overline{y^{P}}\right)}{\sqrt{\sum_{i=1}^{N}\left(y^{T}{}_{i} - \overline{y^{T}}\right)^{2}\sum_{i=1}^{N}\left(y^{P}{}_{i} - \overline{y^{P}}\right)^{2}}},$$

$$R^{2} = 1 - \frac{\sum_{i=1}^{N}\left(y^{T}{}_{i} - y^{P}{}_{i}\right)^{2}}{\sum_{i=1}^{N}\left(y^{T}{}_{i} - \overline{y^{T}}\right)^{2}}, \tag{7}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y^{T}{}_{i} - y^{P}{}_{i}\right|,$$

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y^{T}{}_{i} - y^{P}{}_{i}\right)^{2},$$

in which N represents the total number of data points including either training, testing, or whole data set (input and output pairs), $y^{T}{}_{i}$ refers to the actual value at the sampling point i, $y^{P}{}_{i}$ is the ith output of the model, and $\overline{y^{T}}$ and $\overline{y^{P}}$ stand for the average magnitudes of the actual and predicted data, respectively.

5. Conclusions

Owing to the gained results of this contribution the following major conclusions can be drawn.

(1) Adequate agreement between gain dew point pressure from the developed intelligent model and corresponding real recovery factor/net present value (NPV) values is observed. In other words, the conventional approaches fail to monitor real recovery factor/net present value (NPV) of chemical flooding dedicated to the gained statistical criteria such as mean square error (MSE) and correlation coefficient.

(2) The evolved intelligent network model for monitoring real recovery factor/net present value (NPV) of chemical flooding is user friendly, fast, and cheap for implementation. Moreover, it is very useful and user friendly for evolving the accuracy and robustness of the commercial reservoir simulators like ECLIPSE and computer modelling group (CMG) software for enhanced oil recovery (EOR) from oil reservoirs.

Nomenclature

Abbreviations

ANN: Artificial neural network
BP: Back propagation
GA: Genetic algorithm
HGAPSO: Hybrid genetic algorithm and particle swarm optimization
ICA: Imperialist competitive algorithm
MAE: Mean absolute error (MAE)
MSE: Mean squared error (MSE)
PSO: Particle swarm optimization
PSO: Particle swarm optimization
R^2: Coefficient of determination
UPSO: Unified particle swarm optimization.

Variables

$\overline{y^{P}}$: The average of the predicted data
$\overline{y^{T}}$: The average of the actual data
b_j: Bias
c_1: Cognition component
c_2: Social components
N: The total number of data points
o_j: Output
$r_1{}^{n}$ and $r_2{}^{n}$: Two random numbers
S_j: Sum of interconnection weights
v_i: Velocity of particle i
W_{ji}: Interconnection weights in network model
x_i: Position of particle i
$y_i{}^{P}$: The ith output of the model
$y_i{}^{T}$: The actual at the sampling point i.

Greek Letters

δ: Absolute relative error
φ: Activation function
ω: The inertia weight.

Conflict of Interests

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

References

[1] B. Shaker Shiran and A. Skauge, "Enhanced oil recovery (EOR) by combined low salinity water/polymer flooding," *Energy and Fuels*, vol. 27, no. 3, pp. 1223–1235, 2013.

[2] S. Strand, T. Puntervold, and T. Austad, "Effect of temperature on enhanced oil recovery from mixed-wet chalk cores by spontaneous imbibition and forced displacement using seawater," *Energy and Fuels*, vol. 22, no. 5, pp. 3222–3225, 2008.

[3] H. Zhang, M. Dong, and S. Zhao, "Which one is more important in chemical flooding for enhanced court heavy oil recovery, lowering interfacial tension or reducing water mobility?" *Energy & Fuels*, vol. 24, no. 3, pp. 1829–1836, 2010.

[4] A. Z. Abidin, T. Puspasari, and W. A. Nugroho, "Polymers for enhanced oil recovery technology," *Procedia Chemistry*, vol. 4, pp. 11–16, 2012.

[5] M. A. Ahmadi, Y. Arabsahebi, S. R. Shadizadeh, and S. Shokrollahzadeh Behbahani, "Preliminary evaluation of mulberry leaf-derived surfactant on interfacial tension in an oil-aqueous system: EOR application," *Fuel*, vol. 117, pp. 749–755, 2014.

[6] P. Luo, Y. Zhang, and S. Huang, "A promising chemical-augmented WAG process for enhanced heavy oil recovery," *Fuel*, vol. 104, pp. 333–341, 2013.

[7] A. Mandal, A. Samanta, A. Bera, and K. Ojha, "Characterization of oil-water emulsion and its use in enhanced oil recovery,"

Industrial and Engineering Chemistry Research, vol. 49, no. 24, pp. 12756–12761, 2010.

[8] A. Sabhapondit, A. Borthakur, and I. Haque, "Water soluble acrylamidomethyl propane sulfonate (AMPS) copolymer as an enhanced oil recovery chemical," *Energy and Fuels*, vol. 17, no. 3, pp. 683–688, 2003.

[9] M. S. Karambeigi, R. Zabihi, and Z. Hekmat, "Neuro-simulation modeling of chemical flooding," *Journal of Petroleum Science and Engineering*, vol. 78, no. 2, pp. 208–219, 2011.

[10] M. A. Ahmadi and S. R. Shadizadeh, "Implementation of a high-performance surfactant for enhanced oil recovery from carbonate reservoirs," *Journal of Petroleum Science and Engineering*, vol. 110, pp. 66–73, 2013.

[11] G. Chen, X. Wang, Z. Liang et al., "Tontiwachwuthikul P: simulation of CO_2-oil Minimum Miscibility Pressure (MMP) for CO_2 Enhanced Oil Recovery (EOR) using neural networks," *Energy Procedia*, vol. 37, pp. 6877–6884, 2013.

[12] N. Loahardjo, X. Xie, and N. R. Morrow, "Oil recovery by sequential waterflooding of mixed-wet sandstone and limestone," *Energy & Fuels*, vol. 24, no. 9, pp. 5073–5080, 2010.

[13] M. Ali Ahmadi and M. Golshadi, "Neural network based swarm concept for prediction asphaltene precipitation due to natural depletion," *Journal of Petroleum Science and Engineering*, vol. 98-99, pp. 40–49, 2012.

[14] M. A. Ahmadi, "Neural network based unified particle swarm optimization for prediction of asphaltene precipitation," *Fluid Phase Equilibria*, vol. 314, pp. 46–51, 2012.

[15] S. Zendehboudi, M. A. Ahmadi, L. James, and I. Chatzis, "Prediction of condensate-to-gas ratio for retrograde gas condensate reservoirs using artificial neural network with particle swarm optimization," *Energy & Fuels*, vol. 26, no. 6, pp. 3432–3447, 2012.

[16] M. Ali Ahmadi, S. Zendehboudi, A. Lohi, A. Elkamel, and I. Chatzis, "Reservoir permeability prediction by neural networks combined with hybrid genetic algorithm and particle swarm optimization," *Geophysical Prospecting*, vol. 61, no. 3, pp. 582–598, 2013.

[17] S. Zendehboudi, M. A. Ahmadi, O. Mohammadzadeh, A. Bahadori, and I. Chatzis, "Thermodynamic investigation of asphaltene precipitation during primary oil production: laboratory and smart technique," *Industrial and Engineering Chemistry Research*, vol. 52, no. 17, pp. 6009–6031, 2013.

[18] S. Zendehboudi, M. A. Ahmadi, A. Bahadori, A. Shafiei, and T. Babadagli, "A developed smart technique to predict minimum miscible pressure—EOR implications," *Canadian Journal of Chemical Engineering*, vol. 91, no. 7, pp. 1325–1337, 2013.

[19] M. A. Ahmadi and S. R. Shadizadeh, "New approach for prediction of asphaltene precipitation due to natural depletion by using evolutionary algorithm concept," *Fuel*, vol. 102, pp. 716–723, 2012.

[20] M. A. Ahmadi, "Prediction of asphaltene precipitation using artificial neural network optimized by imperialist competitive algorithm," *Journal of Petroleum Exploration and Production Technology*, vol. 1, no. 2–4, pp. 99–106, 2011.

[21] S. Zendehboudi, A. R. Rajabzadeh, A. Bahadori et al., "Connectionist model to estimate performance of steam-assisted gravity drainage in fractured and unfractured petroleum reservoirs: enhanced oil recovery implications," *Industrial and Engineering Chemistry Research*, vol. 53, no. 4, pp. 1645–1662, 2014.

[22] M. A. Ahmadi, M. Ebadi, and S. M. Hosseini, "Prediction breakthrough time of water coning in the fractured reservoirs by implementing low parameter support vector machine approach," *Fuel*, vol. 117, part A, pp. 579–589, 2014.

[23] M. A. Ahmadi, M. Ebadi, A. Shokrollahi, and S. M. J. Majidi, "Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir," *Applied Soft Computing Journal*, vol. 13, no. 2, pp. 1085–1098, 2013.

[24] M. A. Ahmadi and M. Ebadi, "Evolving smart approach for determination dew point pressure through condensate gas reservoirs," *Fuel*, vol. 117, part B, pp. 1074–1084, 2014.

[25] M. A. Ahmadi, R. Soleimani, and A. Bahadori, "A computational intelligence scheme for prediction equilibrium water dew point of natural gas in TEG dehydration systems," *Fuel*, vol. 137, pp. 145–154, 2014.

[26] M. A. Ahmadi, M. Ebadi, and A. Yazdanpanah, "Robust intelligent tool for estimating dew point pressure in retrograded condensate gas reservoirs: application of particle swarm optimization," *Journal of Petroleum Science and Engineering*, 2014.

[27] J. Kennedy, "Particle swarm: social adaptation of knowledge," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '97)*, pp. 303–308, Indianapolis, Ind, USA, April 1997.

[28] M.-A. Ahmadi, M. Masumi, R. Kharrat, and A. H. Mohammadi, "Gas analysis by in situ combustion in heavy-oil recovery process: experimental and modeling studies," *Chemical Engineering & Technology*, vol. 37, no. 3, pp. 409–418, 2014.

[29] G. V. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 620797, 8 pages
http://dx.doi.org/10.1155/2015/620797

Research Article

Formalization of Human Categorization Process Using Interpolative Boolean Algebra

Vladimir Dobrić,[1] Darko Kovačević,[1] Bratislav Petrović,[1] Dragan Radojević,[2] and Pavle Milošević[1]

[1] *Faculty of Organizational Sciences, University of Belgrade, Jove Ilića 154, 11000 Belgrade, Serbia*
[2] *Institute Mihajlo Pupin, University of Belgrade, Volgina 15, 11000 Belgrade, Serbia*

Correspondence should be addressed to Vladimir Dobrić; vdobric@gmail.com

Received 1 August 2014; Accepted 21 September 2014

Academic Editor: Shifei Ding

Since the ancient times, it has been assumed that categorization has the basic form of classical sets. This implies that the categorization process rests on the Boolean laws. In the second half of the twentieth century, the classical theory has been challenged in cognitive science. According to the prototype theory, objects belong to categories with intensities, while humans categorize objects by comparing them to prototypes of relevant categories. Such categorization process is governed by the principles of perceived world structure and cognitive economy. Approaching the prototype theory by using truth-functional fuzzy logic has been harshly criticized due to not satisfying the complementation laws. In this paper, the prototype theory is approached by using structure-functional fuzzy logic, the interpolative Boolean algebra. The proposed formalism is within the Boolean frame. Categories are represented as fuzzy sets of objects, while comparisons between objects and prototypes are formalized by using Boolean consistent fuzzy relations. Such relations are directly constructed from a Boolean consistent fuzzy partial order relation, which is treated by Boolean implication. The introduced formalism secures the principles of categorization showing that Boolean laws are fundamental in the categorization process. For illustration purposes, the artificial cognitive system which mimics human categorization activity is proposed.

1. Introduction

Categorization is the process in which ideas and objects are recognized, differentiated, and understood [1]. Humans and other organisms consider objects and events as members of categories. Such cognitive activity is automatic and effortless. Categories structure our knowledge about the world. The ability to categorize is fundamental to any natural or artificial cognitive system. Since the time of ancient Greece, it has been assumed that an object can belong or cannot belong to a category and that humans categorize by identifying necessary and sufficient conditions for an object to belong to the category. All objects satisfying such conditions are equivalent with respect to that category. This can be formalized by using classical set theory. The meanings of the categories can be explained by the operations of classical logic. Such formalization implies that the categorization process is governed by the laws of Boolean algebra, among which the laws of thought are fundamental. This view on the categorization is called the classical view [2–4].

In the second half of the twentieth century, the classical theory has been challenged in cognitive science. From this point of view, the categorization is based on gradation. This means that objects can belong to categories with intensities. Two fundamental principles of categorization have been proposed in [5]. According to the principle of perceived world structure, the perceived world comes as structured information. In accordance with the principle of cognitive economy, the task of the categorization is to provide maximum information about the world with the least cognitive effort. This is possible thanks to gradation. In the classical case, two objects can be differentiated if one belongs, while the other one does not belong, to the relevant category. When more objects have to be differentiated from each other,

more categories have to be considered in the process of cognition, which requires more cognitive effort. In the case of gradation, an infinite number of objects can be differentiated inside one category. This implies that the categorization based on gradation can provide maximum information about the world with the least cognitive effort.

According to the gradation (prototype) theory, humans categorize objects by comparing them to prototypes of relevant categories [5–7]. As in [8], we use the term prototype in the most general sense. Under this term, we assume either an abstract summary of a category, or a set of exemplars—actual category members pulled from memory. Since the classical theory has been approached by using classical logic/set theory, the prototype theory should be approached by using fuzzy logic/set theory. In the conventional fuzzy logic [9], the laws of complements are not taken as axioms. This has been harshly criticized in cognitive science [10], as such laws have been considered fundamental from the time of Aristotle. According to [11], the laws of complements cannot be satisfied in the conventional fuzzy logic because it follows the principle of truth functionality. According to this principle, the value of any logical expression can be calculated by using the values of its components. The principle is valid in classical Boolean algebra, but it breaks the Boolean frame in the case of gradation. Fuzzy logic, which secures the laws of thought and the Boolean frame, has been proposed in [12]. In opposition to conventional fuzzy logics, this approach is based on the fundamental principle of structure functionality.

In this paper, the Boolean consistent approach to gradation—interpolative Boolean algebra—is used in an attempt at the formalization of human categorization activity. It is assumed that this innate human activity follows the prototype theory. The proposed formalism is within the Boolean frame, securing the principles of categorization. This shows that Boolean laws are universal in categorization—valid in the classical and the prototype theory. Such laws can be considered as the fundamental laws governing the categorization process. For illustration purposes, the simple artificial cognitive system which mimics human categorization ability is proposed.

2. Interpolative Boolean Algebra

Interpolative Boolean algebra is the Boolean consistent fuzzy logic [11, 12]. From the mathematical point of view, fuzzy logic rests on the principle of incompatibility [13]. According to this principle, increase in problem complexity diminishes our ability to solve the problem by using classical mathematical approaches (based on classical logic/set theory). According to [14], fuzzy logic brings a drastic reduction of complexity immanent to classical approaches when dealing with real-world problems, and fuzzy approaches offer more expressive power with less complexity in comparison to the classical ones. From the cognitive science point of view, fuzzy logic is an attempt at the formalization of remarkable human capabilities [15]. According to [15–17], fuzzy logic is inspired by the brain's crucial ability to manipulate perceptions and is essential for complex problems in artificial intelligence, such as the pattern recognition problem. This is the problem approached in this paper.

Fuzzy logic has been challenged in mathematics and cognitive science since its introduction. The main drawback of conventional fuzzy logics, based on the truth functionality principle, is the fact that they are not in the Boolean frame [14]. From the point of view of mathematics, this implies that the classical results cannot be directly generalized. For example, a basic concept of preference modeling is a preference structure. In the classical case, a preference structure can be constructed from a reflexive preference relation. In the case of gradation, this result is not possible in conventional fuzzy logics [18]. In cognitive science, fuzzy logic has been harshly criticized due to not satisfying the laws of complements [19–21]. Such laws have been considered as the fundamental laws of rational thinking since the ancient times. The laws of thought must apply without exception to any subject matter of thought. Boolean laws cannot be secured in conventional fuzzy logics as they are based on the truth functionality principle. In contrast to these fuzzy logics, interpolative Boolean algebra is based on the principle of structure functionality [12, 22]. According to this principle, logical expressions have vector nature. The truth functionality is valid in the classical case as attention is reduced to only one vector component. In the fuzzy case, all vector components have to be used for Boolean consistent calculations. The structure vector of any logical expression can be calculated by using the structures of expression's components. The structure functionality is the fundamental Boolean principle as it secures the Boolean frame in any value realization of the algebra (binary, multivalued, or fuzzy).

Formally, the finite or atomic Boolean algebra $\mathrm{BA}(\Omega) = P(P(\Omega))$ is generated by a set of free variables $\Omega = \{x_1, x_2, \ldots, x_m\}$. Set $P(\Omega)$ is the power set of Ω. For any $x_i, x_j, x_k \in \Omega$, Boolean laws are given by the following expressions.

Associativity:

$$\begin{aligned} x_i \vee \left(x_j \vee x_k\right) &= \left(x_i \vee x_j\right) \vee x_k, \\ x_i \wedge \left(x_j \wedge x_k\right) &= \left(x_i \wedge x_j\right) \wedge x_k. \end{aligned} \tag{1}$$

Commutativity:

$$\begin{aligned} x_i \vee x_j &= x_j \vee x_i, \\ x_i \wedge x_j &= x_j \wedge x_i. \end{aligned} \tag{2}$$

Distributivity:

$$\begin{aligned} x_i \wedge \left(x_j \vee x_k\right) &= \left(x_i \wedge x_j\right) \vee \left(x_i \wedge x_k\right), \\ x_i \vee \left(x_j \wedge x_k\right) &= \left(x_i \vee x_j\right) \wedge \left(x_i \vee x_k\right). \end{aligned} \tag{3}$$

Identity:

$$\begin{aligned} x_i \vee \underline{0} &= x_i, \qquad x_i \wedge \overline{1} = x_i, \\ x_i \wedge \underline{0} &= \underline{0}, \qquad x_i \vee \overline{1} = \overline{1}. \end{aligned} \tag{4}$$

Idempotence:

$$x_i \vee x_i = x_i, \qquad x_i \wedge x_i = x_i. \tag{5}$$

Absorption:

$$\begin{aligned} x_i \wedge \left(x_i \vee x_j\right) &= x_i, \\ x_i \vee \left(x_i \wedge x_j\right) &= x_i. \end{aligned} \tag{6}$$

Complementation:

$$x_i \wedge \neg x_i = \underline{0}, \qquad x_i \vee \neg x_i = \overline{1}. \tag{7}$$

De Morgan laws:

$$\begin{aligned} \neg\left(x_i \wedge x_j\right) &= \neg x_i \vee \neg x_j, \\ \neg\left(x_i \vee x_j\right) &= \neg x_i \wedge \neg x_j. \end{aligned} \tag{8}$$

Here $\underline{0}$ and $\overline{1}$ are the smallest and the biggest elements of $\mathrm{BA}(\Omega)$.

Atomic elements $\alpha_S(x_1, x_2, \ldots, x_m)$ of $\mathrm{BA}(\Omega)$ are defined by the following expression:

$$\alpha_S\left(x_1, x_2, \ldots, x_m\right) = \bigwedge_{x_i \in S} x_i \bigwedge_{x_j \in \Omega \setminus S} \neg x_j. \tag{9}$$

Here $S \in P(\Omega)$.

Atomic elements have the following properties:

$$\begin{aligned} &\alpha_i\left(x_1, x_2, \ldots, x_m\right) \cap \alpha_j\left(x_1, x_2, \ldots, x_m\right) \\ &\quad = \begin{cases} \underline{0}, & i \neq j \\ \alpha_i\left(x_1, x_2, \ldots, x_m\right), & i = j \end{cases} \\ &\bigcup_{S \in P(\Omega)} \alpha_S\left(x_1, x_2, \ldots, x_m\right) = \overline{1}. \end{aligned} \tag{10}$$

Any element of Boolean algebra can be defined as union of relevant atoms. Which atoms are relevant for (included in) $\varphi(x_1, x_2, \ldots, x_m) \in \mathrm{BA}(\Omega)$ is defined by element's structure function $\sigma_\varphi(\alpha_S)$. Structure function is defined by the following expression:

$$\sigma_\varphi\left(\alpha_S\right) = \begin{cases} 1, & \alpha_S\left(x_1, x_2, \ldots, x_m\right) \cap \varphi\left(x_1, x_2, \ldots, x_m\right) \\ & \quad = \alpha_S\left(x_1, x_2, \ldots, x_m\right) \\ 0, & \alpha_S\left(x_1, x_2, \ldots, x_m\right) \cap \varphi\left(x_1, x_2, \ldots, x_m\right) = \underline{0}. \end{cases} \tag{11}$$

Here $S \in P(\Omega)$.

Any element $\varphi(x_1, x_2, \ldots, x_m) \in \mathrm{BA}(\Omega)$ can be defined by the following expression:

$$\varphi\left(x_1, x_2, \ldots, x_m\right) = \bigcup_{S \in P(\Omega) \mid \sigma_\varphi(\alpha_S) = 1} \alpha_S\left(x_1, x_2, \ldots, x_m\right). \tag{12}$$

Information on which atoms are relevant for $\varphi(x_1, x_2, \ldots, x_m) \in \mathrm{BA}(\Omega)$ is stored in structure vector defined by the following expression:

$$\vec{\sigma}_\varphi = \left[\sigma_\varphi\left(\alpha_S\right) \mid S \in P\left(\Omega\right)\right]^T. \tag{13}$$

The structure of any combined element of $\mathrm{BA}(\Omega)$ can be directly calculated from the structures of its component:

$$\begin{aligned} \vec{\sigma}_{\phi \wedge \psi} &= \vec{\sigma}_\phi \wedge \vec{\sigma}_\psi, \\ \vec{\sigma}_{\phi \vee \psi} &= \vec{\sigma}_\phi \vee \vec{\sigma}_\psi, \\ \vec{\sigma}_{\neg\phi} &= \neg\vec{\sigma}_\phi = \vec{1} - \vec{\sigma}_\phi. \end{aligned} \tag{14}$$

The value $v(\varphi(x_1, x_2, \ldots, x_m)) \in [0, 1]$ of any $\varphi(x_1, x_2, \ldots, x_m) \in \mathrm{BA}(\Omega)$ is given by corresponding Boolean polynomial, which is the sum of values of relevant atomic elements:

$$v\left(\varphi\left(x_1, x_2, \ldots, x_m\right)\right) = \sum_{S \in P(\Omega) \mid \sigma_\varphi(S) = 1} v\left(\alpha_S\left(x_1, x_2, \ldots, x_m\right)\right). \tag{15}$$

The value $v(\alpha_S(x_1, x_2, \ldots, x_m)) \in [0, 1]$ of atomic element $\alpha_S(x_1, x_2, \ldots, x_m) \in \mathrm{BA}(\Omega)$ is given by corresponding atomic Boolean polynomial:

$$v\left(\alpha_S\left(x_1, x_2, \ldots, x_m\right)\right) = \bigotimes_{x_i \in S} v\left(x_i\right) \bigotimes_{x_j \in \Omega \setminus S} \left(1 - v\left(x_j\right)\right). \tag{16}$$

Here $v(x_i) \in [0, 1]$ is the fuzzy value of $x_i \in \Omega$, $S \in P(\Omega)$ and $\otimes : [0, 1] \times [0, 1] \to [0, 1]$ is a generalized product operator which can be any T-norm.

For example, for $\Omega = \{x_i, x_j\}$ the generalized product operator has the following property:

$$\begin{aligned} \max\left(v\left(x_i\right) + v\left(x_j\right) - 1, 0\right) &\leq v\left(x_i\right) \otimes v\left(x_j\right) \\ &\leq \min\left(v\left(x_i\right), v\left(x_j\right)\right). \end{aligned} \tag{17}$$

The value of any $\varphi(x_1, x_2, \ldots, x_m) \in \mathrm{BA}(\Omega)$ can be represented as the scalar product of two vectors:

$$v\left(\varphi\left(x_1, x_2, \ldots, x_m\right)\right) = \vec{\sigma}_\varphi \vec{\alpha}. \tag{18}$$

Here $\vec{\sigma}_\varphi$ is the structure vector of $\varphi(x_1, x_2, \ldots, x_m) \in \mathrm{BA}(\Omega)$ and $\vec{\alpha}$ is the vector of atomic Boolean polynomials.

Vector of atomic Boolean polynomials $\vec{\alpha}$ is defined by the following expression:

$$\vec{\alpha} = \left[v\left(\alpha_S\left(x_1, x_2, \ldots, x_m\right)\right) \mid S \in P\left(\Omega\right)\right]^T. \tag{19}$$

According to the principle of structure functionality, Boolean consistent fuzzy value of any element of BA(Ω) can be calculated by the following rules [22]:

$$\begin{aligned}
&v\left(\varphi_i\left(x_1,\ldots,x_m\right)\wedge\varphi_j\left(x_1,\ldots,x_m\right)\right)\\
&\quad=_{\text{def}} v\left(\varphi_i\left(x_1,\ldots,x_m\right)\right)\otimes v\left(\varphi_j\left(x_1,\ldots,x_m\right)\right),\\
&v\left(\varphi_i\left(x_1,\ldots,x_m\right)\vee\varphi_j\left(x_1,\ldots,x_m\right)\right)\\
&\quad=_{\text{def}} v\left(\varphi_i\left(x_1,\ldots,x_m\right)\right)+v\left(\varphi_j\left(x_1,\ldots,x_m\right)\right)\\
&\qquad-v\left(\varphi_i\left(x_1,\ldots,x_m\right)\right)\otimes v\left(\varphi_j\left(x_1,\ldots,x_m\right)\right),\\
&v\left(\neg\varphi_i\left(x_1,\ldots,x_m\right)\right)=_{\text{def}} 1-v\left(\varphi_i\left(x_1,\ldots,x_m\right)\right),\\
&v\left(x_i\wedge x_j\right)=_{\text{def}}\begin{cases}v\left(x_i\right)\otimes v\left(x_j\right) & i\neq j,\\ v\left(x_i\right), & i=j,\end{cases}\\
&v\left(x_i\vee x_j\right)=_{\text{def}} v\left(x_i\right)+v\left(x_j\right)-v\left(x_i\right)\otimes v\left(x_j\right),\\
&v\left(\neg x_i\right)=_{\text{def}} 1-v\left(x_i\right).
\end{aligned}\tag{20}$$

Here $\varphi_i(x_1,\ldots,x_m),\varphi_j(x_1,\ldots,x_m)\in \mathrm{BA}(\Omega)$, $v(\varphi_i(x_1,\ldots,x_m)),v(\varphi_j(x_1,\ldots,x_m))\in[0,1]$, $x_1,\ldots,x_m\in\Omega$, $v(x_1),\ldots,v(x_m)\in[0,1]$.

Software tool for calculations of Boolean consistent fuzzy values of elements of BA(Ω) is proposed in [23].

Generalized product $\otimes:[0,1]\times[0,1]\to[0,1]$ can be any operator that satisfies the following properties.

Commutativity:

$$v\left(x_i\right)\otimes v\left(x_j\right)=v\left(x_i\right)\otimes v\left(x_j\right).\tag{21}$$

Associativity:

$$\left(v\left(x_i\right)\otimes v\left(x_j\right)\right)\otimes v\left(x_k\right)=v\left(x_i\right)\otimes\left(v\left(x_j\right)\otimes v\left(x_k\right)\right).\tag{22}$$

Monotonicity:

$$v\left(x_i\right)\le v\left(x_j\right)\Longrightarrow v\left(x_i\right)\otimes v\left(x_k\right)\le v\left(x_j\right)\otimes v\left(x_k\right).\tag{23}$$

Boundary:

$$v\left(x_i\right)\otimes 1=v\left(x_i\right).\tag{24}$$

Nonnegativity:

$$\underset{x_i\in S}{\otimes} v\left(x_i\right)\underset{x_j\in\Omega\setminus S}{\otimes}\left(1-v\left(x_j\right)\right)\ge 0.\tag{25}$$

Here $\Omega=\{x_1,x_2,\ldots,x_m\}$, $S\in P(\Omega)$.

In the following section, the prototype theory will be formalized by using interpolative Boolean algebra.

3. Formalization of Human Categorization Process

According to the prototype theory, humans categorize objects by comparing them to prototypes of relevant categories. Prototypes or cognitive reference points are formed in the process of cognition by considering observed (actual) members of the categories. The prototype theory is based on the principle of perceived world structure and the principle of cognitive economy. According to these principles, categories with graded structure "spring" to mind whenever objects or events are considered in the process of cognition. This provides maximum information about the world with the least cognitive effort. It is assumed in this paper that the categorization rests on cognitive acts of ordering of perceived objects and prototypes of relevant categories. The ordering must be relevant to a cognitive task at hand, requiring the least cognitive effort. From the information provided by such ordering, maximum information about the world can be acquired in the process of cognition. This way, the principles of categorization are secured.

Formally, an object considered in the process of cognition will be represented by using vector $[A_1(x),\ldots,A_n(x)]$, where $A_i(x)\in[0,1]$, $i=1,\ldots,n$, are the properties of object x. Set X is the set of objects for categorization. Set Y is the set of prototypes of relevant categories. The prototype of a category is an object or a set of objects, statistically derived or selected from the observed category members. For example, the prototype of category "C" can be represented as an object $y_C\in Y$, whose properties are the averages of properties of all category members:

$$\begin{aligned}
y_C&=\frac{1}{|C|}\sum_{z\in C} z=\left[\frac{1}{|C|}\sum_{z\in C}A_1(z),\ldots,\frac{1}{|C|}\sum_{z\in C}A_n(z)\right]\\
&=\left[A_1\left(y_C\right),\ldots,A_n\left(y_C\right)\right].
\end{aligned}\tag{26}$$

Here $y_C\in[0,1]^n$ is the prototype of category C, $z\in[0,1]^n$ is the member of C, and $|C|$ is the cardinality of C.

Alternatively, the prototype of a category can be defined as the set of the most representative objects of the category (according to some criteria), or as the set which contains all members of the category. Approach to categorization, according to which humans categorize objects by comparing them to actual category members, is called the exemplar approach [24, 25]. The categorization based on exemplars is more flexible but less economical, in comparison to the first approach. Considering exemplars in the process of cognition secures better categorization of "atypical" objects but requires more cognitive effort.

The ordering of objects and prototypes will be formalized by using a Boolean consistent fuzzy partial order relation defined over $X\cup Y$. This reflexive, antisymmetric and transitive binary (two-place) fuzzy relation will be called the primitive relation and will be treated by using Boolean implication. The fuzzy partial order relation can be defined by the following expression:

$$\begin{aligned}
&(x\le y)\\
&=v\left(\phi_i\left(A_1(x),\ldots,A_n(x)\right)\Longrightarrow\phi_i\left(A_1(y),\ldots,A_n(y)\right)\right)\\
&=v\left(\neg\phi_i\left(A_1(x),\ldots,A_n(x)\right)\vee\phi_i\left(A_1(y),\ldots,A_n(y)\right)\right)
\end{aligned}$$

$$= 1 - v\left(\phi_i\left(A_1(x), \ldots, A_n(x)\right)\right) + \min\left(v\left(\phi_i\left(A_1(x), \ldots, A_n(x)\right)\right), v\left(\phi_i\left(A_1(y), \ldots, A_n(y)\right)\right)\right). \quad (27)$$

Here $(x \leq y) \in [0,1]$ is a fuzzy partial order relation (the proof is straightforward), $x, y \in [0,1]^n$, $x \in X$, $y \in Y$, $\phi_i(A_1(x), \ldots, A_n(x))$ can be any Boolean function, $v(\phi_i(A_1(x), \ldots, A_n(x))) \in [0,1]$ is the function's value, and $1 - v(\phi_i(A_1(x), \ldots, A_n(x))) + \min(v(\phi_i(A_1(x), \ldots, A_n(x))), v(\phi_i(A_1(y), \ldots, A_n(y))))$ is the Boolean polynomial which corresponds to Boolean implication $\phi_i(A_1(x), \ldots, A_n(x)) \Rightarrow \phi_i(A_1(y), \ldots, A_n(y))$.

Less complex primitive relations can be defined by the following expressions:

$$\begin{aligned}(x \leq y) &= \sum_{i=1,\ldots,n} w_i v\left(A_i(x) \Longrightarrow A_i(y)\right) \\ &= \sum_{i=1,\ldots,n} w_i v\left(\neg A_i(x) \vee A_i(y)\right) \\ &= \sum_{i=1,\ldots,n} w_i \left(1 - A_i(x) + \left(\min A_i(x), A_i(y)\right)\right), \\ (x \leq y) &= v\left(\underset{i=1,\ldots,n}{\wedge}\left(A_i(x) \Longrightarrow A_i(y)\right)\right) \\ &= v\left(\underset{i=1,\ldots,n}{\wedge}\left(\neg A_i(x) \vee A_i(y)\right)\right) \\ &= \underset{i=1,\ldots,n}{\otimes} v\left(\neg A_i(x) \vee A_i(y)\right) \\ &= \underset{i=1,\ldots,n}{\otimes}\left(1 - A_i(x) + \min\left(A_i(x), A_i(y)\right)\right).\end{aligned} \quad (28)$$

Here $w_i \in [0,1]$, $\sum_{i=1,\ldots,n} w_i = 1$, the generalized product operator ($\otimes$) can be any t-norm, and $v(A_i(x) \Rightarrow A_i(y)) = 1 - A_i(x) + \min(A_i(x), A_i(y))$ is the Boolean polynomial which corresponds (gives value) to Boolean implication $A_i(x) \Rightarrow A_i(y) = \neg A_i(x) \vee A_i(y)$.

Pseudological expressions, such as the first expression in (28), are introduced in [26].

Comparisons between object $x \in X$ and prototype $y \in Y$ will be formalized on the basis of primitive relation $(x \leq y) \in [0,1]$ and its inverse $(x \geq y) \in [0,1]$, $x, y \in [0,1]^n$, $(x \geq y) = (x \leq y)^{-1}$. By using the information provided by these relations, Boolean algebra of fuzzy relations $\mathrm{BA}(\Omega) = P(P(\Omega))$ can be generated, where $P(\Omega)$ is the power set of $\Omega = \{(x \leq y), (x \geq y)\}$. Elements of this algebra have semantics (meanings) thanks to the principle of structure functionality. Such elements formalize the cognitive acts of comparisons between objects and prototypes, on the basis of which the objects are categorized. Proposed formalization of prototype theory secures the principles of categorization. The cognitive effort (in this case computational effort) is minimized as the artificial process of categorization is only based on the primitive relation. At the same time, maximum information about the world can be directly derived from the information provided by the primitive relation. This implies that an artificial cognitive system based on the proposed formalism can acquire maximum information about the world with the least computational effort. In the rest of this section, elements of the Boolean algebra of fuzzy relations are defined. Detailed information on Boolean consistent fuzzy relations can be found in [27].

Atomic elements (atomic relations) of $\mathrm{BA}(\Omega)$ are the following.

Reflexive, symmetric and transitive *fuzzy equivalence* (or *similarity*) relation:

$$(x = y) = (x \leq y) \wedge (x \geq y) = (x \leq y) \otimes (x \geq y). \quad (29)$$

Irreflexive, asymmetric and transitive *fuzzy strict order* relations:

$$\begin{aligned}(x > y) &= \neg(x \leq y) \wedge (x \geq y) = (1 - (x \leq y)) \otimes (x \geq y) \\ &= (x \geq y) - (x \leq y) \otimes (x \geq y), \\ (x < y) &= (x \leq y) \wedge \neg(x \geq y) = (x \leq y) \otimes (1 - (x \geq y)) \\ &= (x \leq y) - (x \leq y) \otimes (x \geq y).\end{aligned} \quad (30)$$

Irreflexive, symmetric and transitive *fuzzy incomparability* relation:

$$\begin{aligned}(x <> y) &= \neg(x \leq y) \wedge \neg(x \geq y) \\ &= (1 - (x \leq y)) \otimes (1 - (x \geq y)) \\ &= 1 - (x \geq y) - (x \leq y) + (x \leq y) \otimes (x \geq y).\end{aligned} \quad (31)$$

In these equations, operator of generalized product $\otimes$ can be any T-norm. According to the structure functionality principle, other Boolean consistent fuzzy relations/elements of $\mathrm{BA}(\Omega)$ can be defined as unions (sums) of relevant atomic relations. *Fuzzy partial order* relations are defined by the following expressions:

$$\begin{aligned}(x \leq y) &= (x < y) \vee (x = y) = (x < y) + (x = y) \\ &= (x \leq y) - (x \leq y) \otimes (x \geq y) + (x \leq y) \otimes (x \geq y) \\ (x \geq y) &= (x > y) \vee (x = y) = (x > y) + (x = y) \\ &= (x \geq y) - (x \leq y) \otimes (x \geq y) + (x \leq y) \otimes (x \geq y).\end{aligned} \quad (32)$$

Complements of fuzzy partial order relations are defined by the following expressions:

$$\begin{aligned}\neg(x \leq y) &= (x <> y) \vee (x > y) = (x <> y) + (x > y) \\ &= 1 - (x \geq y) - (x \leq y) + (x \leq y) \otimes (x \geq y) \\ &\quad + (x \geq y) - (x \leq y) \otimes (x \geq y) = 1 - (x \leq y),\end{aligned}$$

$$\begin{aligned}\neg(x \geq y) &= (x <> y) \vee (x < y) = (x <> y) + (x < y)\\ &= 1 - (x \geq y) - (x \leq y) + (x \leq y) \otimes (x \geq y)\\ &\quad + (x \leq y) - (x \leq y) \otimes (x \geq y) = 1 - (x \geq y).\end{aligned} \tag{33}$$

Another reflexive, symmetric and transitive *fuzzy similarity* relation is defined by the following expression:

$$\begin{aligned}(x \Longleftrightarrow y) &= (x = y) \vee (x <> y) = (x = y) + (x <> y)\\ &= (x \leq y) \otimes (x \geq y) + 1 - (x \geq y) - (x \leq y)\\ &\quad + (x \leq y) \otimes (x \geq y)\\ &= 1 - (x \geq y) - (x \leq y) + 2(x \leq y) \otimes (x \geq y).\end{aligned} \tag{34}$$

The complement of this relation is the *fuzzy dissimilarity* relation, defined by the following expression:

$$\begin{aligned}(x \veebar y) &= \neg(x \Longleftrightarrow y) = (x < y) \vee (x > y)\\ &= (x < y) + (x > y)\\ &= (x \leq y) - (x \leq y) \otimes (x \geq y) + (x \geq y)\\ &\quad - (x \leq y) \otimes (x \geq y)\\ &= (x \leq y) + (x \geq y) - 2(x \leq y) \otimes (x \geq y).\end{aligned} \tag{35}$$

Complements of fuzzy strict order relations are defined by the following expression:

$$\begin{aligned}\neg(x < y) &= (x <> y) \vee (x > y) \vee (x = y)\\ &= (x <> y) + (x > y) + (x = y)\\ &= 1 - (x \geq y) - (x \leq y) + (x \leq y) \otimes (x \geq y)\\ &\quad + (x \geq y) - (x \leq y) \otimes (x \geq y)\\ &\quad + (x \leq y) \otimes (x \geq y)\\ &= 1 - (x \leq y) + (x \leq y) \otimes (x \geq y) = 1 - (x < y),\\ \neg(x > y) &= (x <> y) \vee (x < y) \vee (x = y)\\ &= (x <> y) + (x < y) + (x = y)\\ &= 1 - (x \geq y) - (x \leq y) + (x \leq y) \otimes (x \geq y)\\ &\quad + (x \leq y) - (x \leq y) \otimes (x \geq y)\\ &\quad + (x \leq y) \otimes (x \geq y)\\ &= 1 - (x \geq y) + (x \leq y) \otimes (x \geq y) = 1 - (x > y).\end{aligned} \tag{36}$$

The *complement of a fuzzy incomparability relation (relatedness or comparability relation)* is defined by the following expression:

$$\begin{aligned}\neg(x <> y) &= (x < y) \vee (x > y) \vee (x = y)\\ &= (x < y) + (x > y) + (x = y)\\ &= (x \leq y) - (x \leq y) \otimes (x \geq y) + (x \geq y)\\ &\quad - (x \leq y) \otimes (x \geq y) + (x \leq y) \otimes (x \geq y)\\ &= (x \leq y) + (x \geq y) - (x \leq y) \otimes (x \geq y)\\ &= 1 - (x <> y).\end{aligned} \tag{37}$$

Another *fuzzy dissimilarity* relation, complement of equivalence relation given in (29), is defined by the following expression:

$$\begin{aligned}(x \neq y) &= \neg(x = y) = (x < y) \vee (x > y) \vee (x <> y)\\ &= (x < y) + (x > y) + (x <> y)\\ &= (x \leq y) - (x \leq y) \otimes (x \geq y) + (x \geq y)\\ &\quad - (x \leq y) \otimes (x \geq y) + 1 - (x \geq y) - (x \leq y)\\ &\quad + (x \leq y) \otimes (x \geq y)\\ &= 1 - (x \leq y) \otimes (x \geq y) = 1 - (x = y).\end{aligned} \tag{38}$$

On the basis of information provided by the Boolean algebra of fuzzy relations, objects can be easily categorized in the artificial process of cognition. For example, the categorization process in which objects are recognized on the basis of their resemblance to prototypes can be formalized by using any of aforementioned similarity (or dissimilarity) relations. This is demonstrated in the following section.

4. Illustrative Example

In this section, the simple artificial cognitive system which mimics human categorization activity is constructed. The system categorizes objects on the basis of their similarities to prototypes of relevant categories. This will be demonstrated by using iris flower data set taken from [28]. The data set contains information on 150 four-dimensional objects, which belong to one of the three categories: I_{Se}, I_{Ve}, and I_{Vi}—Iris setosa, Iris versicolor and Iris virginica, respectively. There are 50 objects in each category. Such objects can be defined by the following expression:

$$o = [sl(o), sw(o), pl(o), pw(o)]. \tag{39}$$

Here $o \in R^4$ is a physical object; $sl(o)$, $sw(o)$, $pl(o)$, and $pw(o)$ are the physical properties of object o—sepal length, sepal width, petal length, and petal width, respectively.

The artificial cognitive system maps the physical object $o \in O$ into "mental object" x by using the following expression:

$$x = \left[\frac{sl(o)}{\max_{o_i \in O}(sl(o_i))}, \frac{sw(o)}{\max_{o_i \in O}(sw(o_i))}, \frac{pl(o)}{\max_{o_i \in O}(pl(o_i))}, \frac{pw(o)}{\max_{o_i \in O}(pw(o_i))} \right] = [SL(x), SW(x), PL(x), PW(x)]. \quad (40)$$

Here $x \in [0,1]^4$ is an object considered in the process of cognition, $SL(x), SW(x), PL(x), PW(x) \in [0,1]$ are the properties of object x, and O is the set of objects or the domain of cognition.

Information on category membership is provided to the artificial cognitive system for 105 randomly selected objects. Set X is the set of the remaining 45 objects (for categorization); set Y is the set of prototypes for the three relevant categories. The task of the system is to categorize objects, based on their similarities to prototypes. The system builds prototypes on the basis of observed category members, that is, by using 105 randomly selected objects. For illustration purposes, prototypes are defined in two ways. In the first case, the prototype of category C is the object $y_C \in Y$, whose properties are the averages of properties of all category members:

$$y_C = \frac{1}{|C|}\sum_{z \in C} z = \left[\frac{1}{|C|}\sum_{z \in C} SL(z), \frac{1}{|C|}\sum_{z \in C} SW(z), \frac{1}{|C|}\sum_{z \in C} PL(z), \frac{1}{|C|}\sum_{z \in C} PW(z) \right] = [SL(y_C), SW(y_C), PL(y_C), PW(y_C)]. \quad (41)$$

Here $y_C \in [0,1]^4$ is the prototype of category C, $z \in [0,1]^4$ is the member of C, and $|C|$ is the cardinality of C.

In the second case, all observed members of a category are used as the prototype of that category. This is the exemplar approach. In both approaches, similarity between object $x \in X$ and prototype (or exemplar) $y_C \in Y$ of category C is defined by the following expression:

$$(x \Longleftrightarrow y_C) = 1 - (x \le y_C) - (x \ge y_C) + 2(x \le y_C) * (x \ge y_C). \quad (42)$$

Here $(x \Leftrightarrow y_C) \in [0,1]$ is the Boolean consistent fuzzy similarity relation defined in (34); $(x \le y_C) \in [0,1]$ and $(x \ge y_C) \in [0,1]$ are Boolean consistent fuzzy partial order relations defined in (28).

Boolean consistent fuzzy partial order relations are defined by the following expressions:

$$\begin{aligned}(x \le y_C) &= (1 - SL(x) + \min(SL(x), SL(y_C))) \\ &\quad * (1 - SW(x) + \min(SW(x), SW(y_C))) \\ &\quad * (1 - PL(x) + \min(PL(x), PL(y_C))) \\ &\quad * (1 - PW(x) + \min(PW(x), PW(y_C))), \\ (x \ge y_C) &= (x \le y_C)^{-1} \\ &= (1 - SL(y_C) + \min(SL(y_C), SL(x))) \\ &\quad * (1 - SW(y_C) + \min(SW(y_C), SW(x))) \\ &\quad * (1 - PL(y_C) + \min(PL(y_C), PL(x))) \\ &\quad * (1 - PW(y_C) + \min(PW(y_C), PW(x))).\end{aligned} \quad (43)$$

In the first case, the artificial cognitive system assigns an object to a category whose prototype is the most similar to the object. This is defined by the following expression:

$$C = \arg \max_{C \in \{I_{Se}, I_{Ve}, I_{Vi}\}} (x \Longleftrightarrow y_C). \quad (44)$$

Here $x \in X$, $y_C \in Y$ is the prototype of category $C \in \{I_{Se}, I_{Ve}, I_{Vi}\}$, and $x, y_C \in [0,1]^4$.

In the second case, an object is assigned to a category with the highest average similarity between the object and all category members (exemplars). This is defined by the following expression:

$$C = \arg \max_{C \in \{I_{Se}, I_{Ve}, I_{Vi}\}} \left(\frac{1}{|C|} \sum_{y_C \in C} (x \Longleftrightarrow y_C) \right). \quad (45)$$

Here $y_C \in [0,1]^4$ is the exemplar of $C \in \{I_{Se}, I_{Ve}, I_{Vi}\}$, and $|C|$ is the cardinality of category C.

The categorization process was executed 50 times. In each execution, 70 percent of randomly selected objects are used by the system for prototype formation. The remaining 30 percent of objects are then categorized. The accuracy of the categorization is measured as the average of the percentages of successfully categorized objects in each execution. In the first case, 95.56 percent of objects are successfully categorized. In the second case, 96.67 percent of objects are successfully categorized. As we can see, the artificial cognitive system based on the exemplar approach achieves slightly better results. The same results would have been obtained if the system had been constructed by using the Boolean consistent fuzzy dissimilarity relation defined in (35).

5. Conclusion

Categorization is a fundamental activity for any natural or artificial cognitive system. Formalization of this activity in accordance with the prototype theory of categorization is proposed. This is accomplished by using Boolean consistent

fuzzy logic—the interpolative Boolean algebra. The proposed formalism is within the Boolean frame, securing the principles of categorization—perceived world structure and cognitive economy. Accordingly, an artificial cognitive system based on the proposed formalism can acquire maximum information about the world with the least computational effort. The universality of Boolean laws in the classical and the prototype theory shows that such laws are fundamental in the categorization process. The proposed artificial cognitive system achieves notable results in the considered example.

Conflict of Interests

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

References

[1] H. Cohen and C. Lafebre, *Handbook of Categorization in Cognitive Science*, Elsevier, New York, NY, USA, 2005.

[2] J. S. Bruner, J. J. Goodnow, and G. A. Austin, *A Study of Thinking*, John Wiley & Sons, New York, NY, USA, 1956.

[3] G. P. Goodwin and P. N. Johnson-Laird, "Mental models of Boolean concepts," *Cognitive Psychology*, vol. 63, no. 1, pp. 34–59, 2011.

[4] J. Feldman, "A catalog of Boolean concepts," *Journal of Mathematical Psychology*, vol. 47, no. 1, pp. 75–89, 2003.

[5] E. Rosch and B. B. Lloyd, "Principles of categorization," in *Cognition and Categorization*, pp. 27–48, Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1978.

[6] J. A. Hampton, "Concepts as prototypes," *Psychology of Learning and Motivation: Advances in Research and Theory*, vol. 46, pp. 79–113, 2006.

[7] R. M. Nosofsky and S. R. Zaki, "Exemplar and prototype models revisited: response strategies, selective attention, and stimulus generalization," *Journal of Experimental Psychology: Learning Memory and Cognition*, vol. 28, no. 5, pp. 924–940, 2002.

[8] E. Rosch, "Slow lettuce: categories, concepts, fuzzy sets, and logical deduction," in *Concepts and Fuzzy Logic*, pp. 89–120, The MIT Press, 2011.

[9] L. A. Zadeh, "Fuzzy sets," *Information and Computation*, vol. 8, no. 3, pp. 338–352, 1965.

[10] L. A. Zadeh, "A note on prototype theory and fuzzy sets," *Cognition*, vol. 12, no. 3, pp. 291–297, 1982.

[11] D. G. Radojevic, "Fuzzy set theory in the Boolean frame," *International Journal of Computers Communications & Control*, vol. 3, no. 3, pp. 121–131, 2008.

[12] D. G. Radojević, "New [0,1]-valued logic: a natural generalization of Boolean logic," *Yugoslav Journal of Operations Research*, vol. 10, no. 2, pp. 185–216, 2000.

[13] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, pp. 28–44, 1973.

[14] D. Radojević, "Real-valued realizations of boolean algebras are a natural frame for consistent fuzzy logic," *Studies in Fuzziness and Soft Computing*, vol. 299, pp. 559–565, 2013.

[15] L. A. Zadeh, "Is there a need for fuzzy logic?" *Information Sciences*, vol. 178, no. 13, pp. 2751–2779, 2008.

[16] L. A. Zadeh, "From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions," *IEEE Transactions on Circuits and Systems. I. Fundamental Theory and Applications*, vol. 46, no. 1, pp. 105–119, 1999.

[17] L. A. Zadeh, "A new direction in AI—toward a computational theory of perceptions," *AI Magazine*, vol. 22, no. 1, pp. 73–84, 2001.

[18] J. Fodor and M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[19] D. N. Osherson and E. E. Smith, "On the adequacy of prototype theory as a theory of concepts," *Cognition*, vol. 9, no. 1, pp. 35–58, 1981.

[20] H. Kamp and B. Partee, "Prototype theory and compositionality," *Cognition*, vol. 57, no. 2, pp. 129–191, 1995.

[21] P. N. Johnson-Laird, *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*, Harvard University Press, Cambridge, Mass, USA, 1983.

[22] D. Radojević, "Real sets as consistent Boolean generalization of classical sets," in *From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence*, pp. 150–171, Editing House of Romanian Academy, Bucharest, Romania, 2008.

[23] P. Milošević, B. Petrović, D. Radojević, and D. Kovačević, "A software tool for uncertainty modeling using Interpolative Boolean algebra," *Knowledge-Based Systems*, vol. 62, pp. 1–10, 2014.

[24] R. M. Nosofsky, "Exemplar-based approach to relating categorization, identification, and recognition," in *Multidimensional Models of Perception and Cognition*, pp. 363–393, Lawrence Erlbaum Associates, New York, NY, USA, 1992.

[25] D. L. Medin and M. M. Schaffer, "Context theory of classification learning," *Psychological Review*, vol. 85, no. 3, pp. 207–238, 1978.

[26] D. Radojević, "Logical aggregation based on interpolative boolean algebra," *Mathware & Soft Computing*, vol. 15, no. 1, pp. 125–141, 2008.

[27] D. Radojević, "Interpolative relations and interpolative preference structures," *Yugoslav Journal of Operations Research*, vol. 15, no. 2, pp. 171–189, 2005.

[28] K. Bache and M. Lichman, *UCI Machine Learning Repository*, 2013, http://archive.ics.uci.edu/ml.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 231394, 7 pages
http://dx.doi.org/10.1155/2015/231394

Research Article

A Hybrid Least Square Support Vector Machine Model with Parameters Optimization for Stock Forecasting

Jian Chai,[1] Jiangze Du,[2] Kin Keung Lai,[1,2] and Yan Pui Lee[3]

[1] *International Business School, Shaanxi Normal University, Xian 710062, China*
[2] *Department of Management Sciences, City University of Hong Kong, Hong Kong*
[3] *School of Business, Tung Wah College, Hong Kong*

Correspondence should be addressed to Kin Keung Lai; mskklai@cityu.edu.hk

Received 30 May 2014; Accepted 20 August 2014

Academic Editor: Shifei Ding

This paper proposes an EMD-LSSVM (empirical mode decomposition least squares support vector machine) model to analyze the CSI 300 index. A WD-LSSVM (wavelet denoising least squares support machine) is also proposed as a benchmark to compare with the performance of EMD-LSSVM. Since parameters selection is vital to the performance of the model, different optimization methods are used, including simplex, GS (grid search), PSO (particle swarm optimization), and GA (genetic algorithm). Experimental results show that the EMD-LSSVM model with GS algorithm outperforms other methods in predicting stock market movement direction.

1. Introduction

Stock market is one of the most sophisticated and challenging financial markets since many factors affect its movement, including government policy, global economic situation, investors' expectations, and even correlations with other markets [1]. References [2, 3] described financial time series as essentially noisy, dynamic, and deterministically chaotic data sequences. Hence, a precise prediction of stock index movement can help investors make decisions to take or shed positions in the stock market at the right time and make profits. Many works have been published by researchers to maximize investment profits and minimize risk. Therefore, predicting stock market is quite important and significant.

Neural networks have been successfully applied in forecasting of financial time series during the past two decades [4–6]. Neural networks are general function approximations which can approximate many nonlinear functions regardless of the properties of time series data [7]. Besides, neural networks are able to learn dynamic systems which make them a more powerful tool for studying financial time series compared with traditional models [8–10]. However, there are a couple of weaknesses when neural networks are used in forecasting financial time series. For instance, when the typical back-propagation neural network is applied, a huge number of parameters are required to be controlled for. This makes the solution unstable and causes overfitting. The overfitting problem results in poor performance and becomes a critical issue for researchers.

Accordingly, [11] proposed a support vector machine (SVM) model. According to [12–14], there are two advantages of using SVM rather than neural networks. One is that SVM has a better performance in terms of generalization. Unlike the empirical risk minimization principle in traditional neural networks, SVM reduces generalization error bounds based on the structural risk minimization principle. SVM seeks to achieve an optimal structure through finding out a balance between generalization errors and Vapnik-Chervonenkis (VC) confidence interval. Another advantage is that SVM prevents the model from getting stuck into local minima.

Since the introduction of SVM, it has been developed rapidly in the real world. There are mainly two ways for applying SVM: one is classification and the other is regression. For classification, [15] constructed a SVM based model to accurately evaluate the consumers' credit score and solve

classification problems. Also, SVM is widely used in the area of forecasting. Reference [16] used SVM to predict the direction of daily stock price in the Korea composite stock price index (KOSPI). More recently, [17] applies the Support Vector Regression to forecast the Nikki 225 opening index and TAIEX closing index after detecting and removing the noise by independent component analysis (ICA).

However, the performance of SVM mainly depends on the input data and is sensitive to parameters. Recent empirical studies have demonstrated that properties of the model performance are influenced by two aspects: low level of signal to noise ratio (SNR) and instability of model specification during the estimation process. For example, [18] investigates the hyperparameters selection for support vector machine with different noise distributions to compare the model performance. Moreover, [19] applied wavelet to denoise the bearing vibration signals by improving the SNR and then figure out the best model according to the performances of ANN and SVM.

To improve the classification and forecasting accuracy, several researchers including [20, 21] have proved that the combined classifying and forecasting models perform better than any individual model. Also, [22] showed that the ensemble empirical model decomposition (EEMD) can be integrated with extreme learning machine (ELM) to an effective forecasting model for computer products sales. In this paper, we propose a hybrid EMD-LSSVM (empirical mode decomposition least squares support vector machine) with different parameters optimization algorithms. The experimental results prove that the EMD-LSSVM model has a better performance than the WD-LSSVM (wavelet denoising least squares support vector machine) model. Firstly, we use the empirical mode decomposition and wavelet denoising algorithm to deal with the original input data. Secondly, parameters of SVM are optimized by different methods, including simplex, grid search (GS), particle swarm optimization (PSO), and genetic algorithm (GA). Results from empirical studies show that the hybrid model EMD-LSSVM with GS parameter optimization outperforms the other model.

2. EMD-LSSVM Model and WD-LSSVM

2.1. Empirical Mode Decomposition (EMD). References [23, 24] proposed empirical mode decomposition (EMD) which decomposes data series into a number of intrinsic mode functions (IMFs). It was designed for nonstationary and nonlinear data sets. In order to apply EMD, time series data set must satisfy the following two conditions.

(1) The sum of local maxima and local minima must equate to the total number of zero crossings or the difference between them is 1. In other words, for every local maxima and local minima, there must be one zero crossing following up.

(2) The local average is zero, which means that mean value of the upper envelope (defined by local maxima) and lower envelope (defined by local minima) must be zero.

Thus, if a function is an IMF, it represents a signal symmetric to local mean zero. An IMF is a simple oscillatory mode which is more general than the simple harmonic function and the frequency and amplitude of the IMF can be variable. Then, data series $x(t)$ $(t = 1, 2, \ldots, n)$ can be decomposed by the following sifting procedure.

(1) Find all local maxima and minima in $x(t)$. Then use the cubic spline line to connect all local maxima to generate upper envelope $x_{\text{up}}(t)$ and connect all local minima to generate lower envelop $x_{\text{low}}(t)$.

(2) According to the upper and lower envelopes obtained in Step (1), calculate the envelope mean $m_1(t)$:

$$m_1(t) = \frac{\left(x_{\text{up}}(t) + x_{\text{low}}(t)\right)}{2}. \tag{1}$$

(3) Data series $x(t)$ minus envelope mean $m_1(t)$ gives the first component $d_1(t)$:

$$d_1(t) = x(t) - m_1(t). \tag{2}$$

(4) Check if $d_1(t)$ satisfies the IMF requirements; if $d_1(t)$ does not satisfy them, go back to Step (1) and replace $x(t)$ with $d_1(t)$ to conduct the second sifting procedure; that is, $d_2(t) = d_1(t) - m_2(t)$. Repeat the sifting procedure k times $d_k(t) = d_{k-1}(t) - m_k(t)$ until the following stop criterion is satisfied:

$$\sum_{t=1}^{T} \frac{\left[d_j(t) - d_{j+1}(t)\right]^2}{d_j^2(t)} < \text{SC}, \tag{3}$$

where SC is the stopping condition. Normally, it is set between 0.2 and 0.3. Then, we get the first IMF component; that is, $c_1(t) = d_k(t)$.

(5) Subtract first IMF component $c_1(t)$ from data sets $x(t)$ and get the residual $r_1(t) = x(t) - c_1(t)$.

(6) Treat $r_1(t)$ as the new data series and repeat Steps (1) to (5). Then get the new residual $r_2(t)$. In this way, after repeating n times, we get

$$\begin{aligned} r_2(t) &= r_1(t) - c_2(t), \\ r_3(t) &= r_2(t) - c_3(t), \\ &\vdots \\ r_n(t) &= r_{n-1}(t) - c_n(t). \end{aligned} \tag{4}$$

When the residual $r_n(t)$ becomes a monotonic function, the data sets cannot be decomposed anymore. The whole EMD is completed. The original date series can be described as the combination of n IMF components and a mean trend $r_n(t)$; that is,

$$x(t) = \sum_{j=1}^{n} c_j(t) + r_n(t). \tag{5}$$

In this way, the original data series $x(t)$ can be decomposed into n IMFs and a mean trend function. Then, we use the IMFs for instantaneous frequency analysis.

The traditional Fourier transform decomposes a data series into a number of sine or cosine waves for the analysis. However, the EMD technique decomposes the data series into several sinusoid-like signals with variable frequencies and a mean trend function. The EMD has several advantages. First, this method is relatively easy to understand and is also widely applied since it avoids complex mathematical algorithms. Secondly, EMD is suitable to deal with nonlinear and nonstationary data series. Thirdly, EMD is more suitable for analysing data series with trends such as weather and economic data. Finally, EMD is able to find the residual which reveals the data series trends [25–27].

2.2. Wavelet Denoising Algorithm. While the traditional Fourier analysis can only remove noise of certain patterns over the entire time horizon, wavelet analysis can deal with multiscales and more detailed data and is more suitable for financial time series. Wavelets are continuous functions which satisfy the unit energy and admissibility condition in

$$C_\varphi = \int_0^\infty \frac{|\varphi(f)|}{f} df < \infty, \qquad \int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1, \tag{6}$$

where φ is the Fourier transform of frequency f. ψ is the wavelet transform.

The continuous wavelet function can orthogonally transform the original data into subdata series in the wavelet domain. Consider

$$W(u,s) = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) dt, \tag{7}$$

where u is the dilation parameter and s is the translation parameter.

The wavelet synthesis rebuilds the original data series, guaranteed by the properties of orthogonal transformation in

$$x(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^{\infty} W(u,s)\, \psi_{u,s}(t)\, du \frac{ds}{s^2}. \tag{8}$$

In wavelet analysis, the denoising technique separates the data and noise from the original data sets by selecting a threshold. The raw data series are first decomposed into some data subsets. Then, based on a certain strategy of selecting the threshold, the boundary between noises and data is set. Depending on the boundary, smaller data points are eliminated and the remaining data are handled by setting certain thresholds. Finally, these denoised data sets are rebuilt from the decomposed data points [28].

2.3. LSSVM in Function Estimation. This section shows the basic theory of the least squares support vector machine. The support vector methodology has been used mainly in two areas, that is, classification and function estimation. Considering regression in the set of function $f(x) = \omega^T \varphi(x) + b$ with given training data inputs $x_k \in R^n$ and outputs $y_k \in R$, we apply $\varphi(x)$ to map x_k from R^n to R^{n_h}. Notice that $\varphi(x)$ can be of infinite dimensional and is defined only implicitly. Also, vector ω can also be infinite dimensional. Thus, the optimization problem becomes

$$\begin{aligned}
\min \quad & J_P(\omega, \xi, \xi^*) = \frac{1}{2}\omega^T\omega + c\sum_{k=1}^{N}(\xi + \xi^*), \\
\text{s.t.} \quad & y_k - \omega^T\varphi(x_k) - b \le \varepsilon + \xi_k, \quad k = 1, \ldots, N, \\
& \omega^T\varphi(x_k) + b - y_k \le \varepsilon + \xi_k^*, \quad k = 1, \ldots, N, \\
& \xi_k, \xi_k^* \ge 0, \quad k = 1, \ldots, N.
\end{aligned} \tag{9}$$

The constant $c > 0$ defines the tolerance of deviations from the desired ε accuracy. It defines the weight of the regularization term empirical risk. The larger the c is, the more important it is for the empirical risk to grow, compared with the regularization term. ε is called the tube size and represents the accuracy required in training data points.

By introducing Lagrange multipliers $\alpha, \alpha^*, \eta, \eta^* \ge 0$, we obtain the Lagrangian for this problem. Consider

$$\begin{aligned}
& L(\omega, b, \xi_k, \xi_k^*; \alpha, \alpha^*, \eta, \eta^*) \\
& \quad = \frac{1}{2}\omega^T\omega + c\sum_{k=1}^{N}(\xi + \xi^*) \\
& \qquad - \sum_{k=1}^{N}\alpha_k\left(\varepsilon + \xi_k - y_k + \omega^T\varphi(x_k) + b\right) \\
& \qquad - \sum_{k=1}^{N}\alpha_k^*\left(\varepsilon + \xi_k^* + y_k - \omega^T\varphi(x_k) - b\right) \\
& \qquad - \sum_{k=1}^{N}(\eta_k\xi_k + \eta_k^*\xi_k^*).
\end{aligned} \tag{10}$$

The reason of introducing another Lagrange multiplier α_k^* is that there are other slack variables ξ_k, ξ_k^*. By maximizing the Lagrangian

$$\max_{\alpha,\alpha^*,\eta,\eta^*} \min_{\omega,b,\xi_k,\xi_k^*} L(\omega, b, \xi_k, \xi_k^*; \alpha, \alpha^*, \eta, \eta^*), \tag{11}$$

we obtain

$$\begin{aligned}
& \frac{\partial L}{\partial \omega} = 0 \longrightarrow \omega = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)\varphi(x_k), \\
& \frac{\partial L}{\partial b} = 0 \longrightarrow \sum_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0, \\
& \frac{\partial L}{\partial \xi_k} = 0 \longrightarrow c - \alpha_k - \eta_k = 0, \\
& \frac{\partial L}{\partial \xi_k^*} = 0 \longrightarrow c - \alpha_k^* - \eta_k^* = 0.
\end{aligned} \tag{12}$$

Table 1: Input and output variables.

Category	Input variables selection	Output variable	Sample	Data number
Training and validation testing	CSI 300, USDX, SHIBOR, REPO, CDS, PE, M2/mkt cap, short-mid not/mkt cap, New Loan/mkt cap	CSI 300	05/01/2009–23/08/2011 24/08/2011–20/01/2012	643 100

Then we obtain the following dual problem:

$$\max_{\alpha,\alpha^*} \quad J_D(\alpha,\alpha^*) = -\frac{1}{2}\sum_{k=1}^{N}(\alpha_k-\alpha_k^*)(\alpha_l-\alpha_l^*)K(x_k,x_l) - \varepsilon\sum_{k=1}^{N}(\alpha_k+\alpha_k^*) + \sum_{k=1}^{N} y_k(\alpha_k-\alpha_k^*),$$
$$\text{s.t.} \quad \sum_{k=1}^{N}(\alpha_k-\alpha_k^*)=0, \quad \alpha_k,\alpha_k^*\in[0,c]. \tag{13}$$

Here we use the kernel function $K(x_k,x_l)=\varphi(x_k)^T\varphi(x_l)$ for $k,l=1,\ldots,N$. Then the function estimation becomes

$$f(x)=\sum_{k=1}^{N}(\alpha_k-\alpha_k^*)K(x_k,x_l)+b, \tag{14}$$

where α_k,α_k^* are solutions of the above quadratic programming problem and b is obtained from the complementarity of KKT conditions. It is obvious that the decision function is determined by the support vectors in which coefficients $(\alpha_k-\alpha_k^*)$ are not zero. In practice, a larger ε results in a smaller number of support vectors and thus the sparser of the solution. Also, the larger the ε is, the worse the accuracy of training points will be. Hence, ε can be applied to control the balance between closeness to training data and sparseness of the solution.

Kernel function can be obtained by seeking the function which satisfies Mercer's condition. Here are some popular kernel functions [14, 29, 30]:

linear: $K(x,x_k)=x^Tx_k$;

polynomial: $K(x,x_k)=(x^Tx_l+1)^d$, where d is the degree of the polynomial kernel;

RBF kernel: $K(x,x_k)=\exp(-\|x-x_k\|^2/\sigma^2)$, where σ^2 is the bandwidth of the Gaussian kernel.

Parameters of the kernel function define the structure of the high dimensional feature space $\varphi(x)$ and also control the accuracy of the final solution. Thus, they should be selected carefully.

3. Empirical Study

3.1. Data Description. The CSI 300 is chosen for empirical analysis and to examine the performance of the proposed model. This index comprises 179 stocks from Shanghai stock exchange and 121 stocks from Shenzhen stock exchange and is managed by the China Securities Index Company Ltd.

Most researchers have chosen international indices in the past, including S & P 500, NIKKEI 225, NASDAQ, DAX, and gold price as input variables. They have examined the cross relationship between stock market index and macroeconomic variables. The potential input variables that can be used for forecasting model mainly consist of the gross domestic product (GDP), gross national product (GNP), short-term interest rate (ST), long-term interest rate (LT), and term structure of interest rate (TS) [1, 31, 32].

Although China has overtaken Japan to become the world's second largest economy and the Chinese stock market has developed into one of the most important markets in the global economy, Chinese consumption capacity is limited in the domestic market. The movement of the stock market has a close relationship with the money available of the investors, which is determined by the money supply and the interest rate. Considering that the Chinese stock market is affected by the global economic situation as well as the domestic economic development, we choose US Dollar Index (USDX), Shanghai Interbank Offered Rate (SHIBOR), P/E ratio (PE), money supply (M2), repurchase agreement (REPO), China CNY Monthly New Loan, market capitalization of the 300 publicly traded companies (mkt cap), People's Bank 5-year CDS, and short-mid note as input variables.

The lag of input variables is 3 days. We use the daily data to predict the CSI 300 index by nonlinear SVM regression. Since M2, short-mid note, and New Loan are published once a month, we transform these variables into daily variables by dividing them by a daily variable. We divided all data sets into two sections and used the first section as the training part to find the optimal parameters for the LSSVM and avoid overfitting by training and validating the model. The other section is used for testing. As shown in Table 1, we choose nine variables as the input variables and one variable as the output variable including 643 daily data from May 1, 2009, to August 23, 2011, to train the parameters in the model. Once we obtain these parameters, we use the same input and output variables from August 24, 2011, to January 20, 2012, including 100 daily data to examine the performance of different model in the testing part.

In the hybrid wavelet denoising least squares support vector machine model (WD-LSSVM), we first denoise the CSI 300 index with wavelet denoising technique. As shown in Figure 1, the original data, which is depicted in the upper part of the figure, is packed with irrelevant noise. Then the wavelet denoising algorithm is applied to reduce the noise in the upper figure of Figure 1. The denoised data is depicted in the lower part of Figure 1 and it is clear that the denoised data

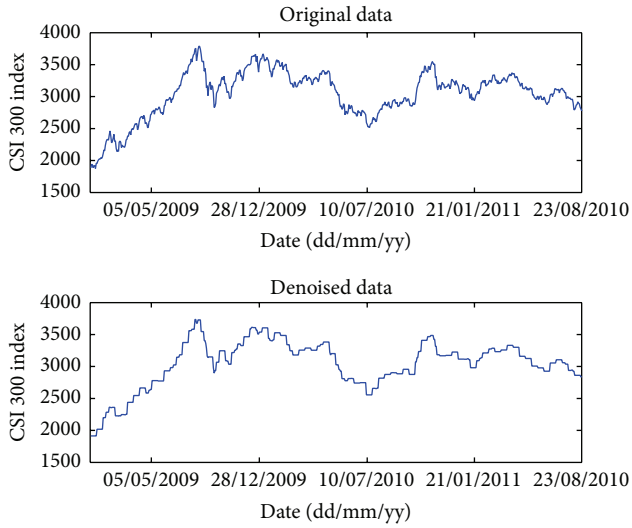


Figure 1: The original and denoised daily CSI 300 index.

Table 2: Parameters setting for simplex, GS, GA, and PSO.

Method	Parameter setting
Simplex	Chi: 2, Gamma: 0.5, Rho: 1, Sigma: 0.5
GS	TolX: 0.001, maxFunEvals: 70, grain: 7, zoomfactor: 5
GA	Sizepop: 20, maxgen: 200, $c$min: 0, $c$max: 100, $g$min: 0, $g$max: 1000
PSO	Sizepop: 20, maxgen: 200, $c$min: 0, $c$max: 100, $g$min: 0, $g$max: 1000, k: 0.6

Table 3: Performance metrics and their calculations.

Metrics	Calculation
NMSE	$\text{NMSE} = \frac{\sum_{i=1}^{n}(a_i - p_i)^2}{\delta^2 n}$, $\delta^2 = \frac{\sum_{i=1}^{n}(a_i - \bar{a})^2}{n-1}$
MAPE	$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{a_i - p_i}{p_i}\right\rvert \times 100\%$
HR	$\text{HR} = \frac{\sum_{i=1}^{n} d_i}{n}$, $d_i = \begin{cases} 1 & \text{if } (a_i - a_{i-1})(p_i - p_{i-1}) \geq 0 \\ 0 & \text{otherwise} \end{cases}$

Table 4: Results of eight different forecasting models.

Model	NMSE	MAPE	HR
EMD-LSSVM (simplex)	0.0253	0.79222%	77.7778%
EMD-LSSVM (GS)	0.0245	0.78834%	79.798%
EMD-LSSVM (GA)	2.5749	9.0641%	42.4242%
EMD-LSSVM (PSO)	9.4471	18.2733%	40.404%
WD-LSSVM (simplex)	0.0521	1.1772%	65.6566%
WD-LSSVM (GS)	0.0609	1.1357%	61.6162%
WD-LSSVM (GA)	0.0657	1.2997%	62.6263%
WD-LSSVM (PSO)	0.0910	1.5072%	63.6364%

can better reveal the trend of the index. Also, in both EMD-LSSVM and WD-LSSVM models, we preprocess the input data by scaling to the range of $[0, 1]$ to prevent small numbers in the data sets from being overshadowed by large numbers, resulting in loss of information.

3.2. Optimization Methods and Parameters Setting. In both EMD-LSSVM models and WD-LSSVM, we try four kinds of search methods, that is, simplex, GS, GA, and PSO. In the simplex method, we define the parameters of expanding (Chi), contracting (Gamma), reflecting (Rho), and shrinking (Sigma) and get the optimal parameters for SVM through iteration until the stopping criteria is satisfied. Also, by calculating the objective function, we can get all points in the grids, which are related to the range and the unit grid search size.

The optimal parameters can be obtained from the point which has the lowest cost. Another effective method to solve optimization problem is the genetic algorithm. The first step of this method is to randomly select parents from the population. Then, parents produce children continuously. Step by step, the population eventually develops and optimal solution can be obtained when the stopping criteria are met. The PSO algorithm works by moving the candidate solution (particles) within the given search range. These particles are moved by the best known positions of particles and the entire swarm in the search space. When the particles arrive at a better position, they guide the swarm to move. The procedure is repeated until the stopping criteria are satisfied. In our experiment, Table 2 shows the setting of each optimization method.

3.3. Performance Criteria. We evaluate the performance of these models using three measurement methods, that is, normalized mean squared error (NMSE), mean absolute percentage error (MAPE), and the hitting ratio (HR) (Table 3). NMSE and MAPE are designed to measure the deviation of predicted value from the actual value; smaller values of NMSE and MAPE indicate better performance of the model. In the stock market, smaller values of MAPE and NMSE are able to control investment risk. We also introduce hitting rate to evaluate the model since the HR reveals accuracy of prediction of the CSI 300, which is valuable for individual and institutional traders.

3.4. Experiment Results. The experiments explore four parameter selection methods in both EMD-LSSVM and WD-LSSVM. Results of the experiments are as in Table 4. From the results, we can see that the hybrid model EMD-LSSVM with GS parameter optimization method not only has the smallest NMSE and MAPE but also gets the best hitting rate, which means it outperforms the other model with different parameter search methods.

From the experiment results, we can draw three conclusions.

(1) For overall accuracy, the EMD-LSSVM (GS) is the best approach, followed by EMD-LSSVM (simplex), WD-LSSVM (simplex), WD-LSSVM (PSO), WD-LSSVM (GA), and WD-LSSVM (GS). Hitting rates of the other approaches are below 60%. Prediction accuracy of all methods is also related to the chosen sample. So it is difficult to identify which model is the best and performs the best. However, tests based on the same sample may help us identify which is the best model.

(2) According to the experiments, the PSO and GA need more computational time to obtain the best parameters for the model compared with simplex and GS optimization methods. Although the PSO and GA algorithm are relatively more complex than the other two methods, they do not perform better than GS and simplex.

(3) Another interesting finding is that thresholds of the denoising algorithm also influence the performance of the model. When the threshold is too large, useful information in the data gets damaged. Besides, a small threshold makes the denoising process insignificant for handling noise. Therefore, we argue that the performance of the wavelet denoising algorithm is sensitive to the estimation method of the threshold level.

4. Conclusion

We have examined the use of the hybrid EMD-LSSVM and WD-LSSVM models to predict financial time series by four different parameters selection methods in this paper. The study shows that the hybrid EMD-LSSVM model provides a better way to forecast financial time series compared with WD-LSSVM. The key findings contain two aspects. First, empirical mode decomposition can serve as a potential tool for removing noise from original data during the modeling process and improving the prediction accuracy. Second, we compare four kinds of search methods for parameters in the experiments. The results show that the EMD-LSSVM with GS parameter optimization method provides the best performance. Use of the GS algorithm reduces the computation time and improves the prediction accuracy of the model for forecasting financial time series.

Future research in this direction mainly includes gaining better understanding of the relationship between optimal loss function, noise distribution, and the number of training samples. In this paper, we only consider applying different algorithm to denoise the original data without considering the distribution of the noise. The research on the density of noise which will be reduced for the SVM model will attract the effort of us. Moreover, another interesting research direction is to figure out the minimum number of samples based on which a theoretically optimal loss function will indeed have superior generalization performance.

Conflict of Interests

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

References

[1] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers and Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.

[2] J. W. Hall, "Adaptive selection of U.S. stocks with neural nets," in *Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets*, G. J. Deboeck, Ed., John Wiley & Sons, New York, NY, USA, 1994.

[3] Y. S. Abu-Mostafa and A. F. Atiya, "Introduction to financial forecasting," *Applied Intelligence*, vol. 6, no. 3, pp. 205–213, 1996.

[4] W. Cheng, L. Wagner, and C.-H. Lin, "Forecasting the 30-year US treasury bond with a system of neural networks," *Journal of Computational Intelligence in Finance*, vol. 4, pp. 10–16, 1996.

[5] R. Sharda and R. B. Patil, "A connectionist approach to time series prediction: an empirical test," in *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance*, R. R. Trippi and E. Turban, Eds., Irwin Professional Publishing, Chicago, Ill, USA, 1996.

[6] J. R. van Eyden, *The Application of Neural Networks in the Forecasting of Share Prices*, Finance and Technology Publishing, Haymarket, Va, USA, 1996.

[7] I. Kaastra and M. S. Boyd, "Forecasting futures trading volume using neural networks," *Journal of Futures Markets*, vol. 15, pp. 853–970, 1995.

[8] G. Zhang and M. Y. Hu, "Neural network forecasting of the British pound/US dollar exchange rate," *Omega*, vol. 26, no. 4, pp. 495–506, 1998.

[9] W.-C. Chiang, T. L. Urban, and G. W. Baldridge, "A neural network approach to mutual fund net asset value forecasting," *Omega*, vol. 24, no. 2, pp. 205–215, 1996.

[10] F. E. H. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, no. 4, pp. 309–317, 2001.

[11] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 2nd edition, 2000.

[12] K.-R. Muller, A. J. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, and V. N. Vapnik, "Predicting time series with support vector machines," in *Proceedings of the International Conference on Artificial Neural Networks*, pp. 999–1004, Lausanne, Switzerland, 1997.

[13] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear prediction of chaotic time series using support vector machines," in *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing (NNSP '97)*, pp. 511–520, Amelia Island, Fla, USA, September 1997.

[14] V. N. Vapnik, S. E. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," *Advances in Neural Information Processing Systems*, vol. 9, pp. 281–287, 1996.

[15] C.-L. Huang, M.-C. Chen, and C.-J. Wang, "Credit scoring with a data mining approach based on support vector machines," *Expert Systems with Applications*, vol. 33, no. 4, pp. 847–856, 2007.

[16] K.-J. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[17] C.-J. Lu, T.-S. Lee, and C.-C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression," *Decision Support Systems*, vol. 47, no. 2, pp. 115–125, 2009.

[18] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Networks*, vol. 17, no. 1, pp. 113–126, 2004.

[19] G. S. Vijay, H. S. Kumar, P. P. Srinivasa, N. S. Sriram, and R. B. K. N. Rao, "Evaluation of effectiveness of wavelet based denoising schemes using ANN and SVM for bearing condition classification," *Computational Intelligence and Neuroscience*, vol. 2012, Article ID 582453, 12 pages, 2012.

[20] L. Zhou, K. K. Lai, and L. Yu, "Least squares support vector machines ensemble models for credit scoring," *Expert Systems with Applications*, vol. 37, no. 1, pp. 127–133, 2010.

[21] Y. Bao, X. Zhang, L. Yu, K. K. Lai, and S. Wang, "An integrated model using wavelet decomposition and least squares support vector machines for monthly crude oil prices forecasting," *New Mathematics and Natural Computation*, vol. 7, no. 2, pp. 299–311, 2011.

[22] C.-J. Lu and Y. E. Shao, "Forecasting computer products sales by integrating ensemble empirical mode decomposition and extreme learning machine," *Mathematical Problems in Engineering*, vol. 2012, Article ID 831201, 15 pages, 2012.

[23] N. E. Huang, Z. Shen, S. R. Long et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *The Royal Society of London. Proceedings A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[24] N. E. Huang, Z. Shen, and S. R. Long, "A new view of nonlinear water waves: the Hilbert spectrum," *Annual Review of Fluid Mechanics*, vol. 31, pp. 417–457, 1999.

[25] L. Yu, S. Wang, and K. K. Lai, "An EMD-based neural network ensemble learning model for world crude oil spot price forecasting," in *Soft Computing Applications in Business*, B. Prasad, Ed., vol. 230 of *Studies in Fuzziness and Soft Computing*, pp. 261–271, Springer, 2008.

[26] L. Yu, K. K. Lai, S. Wang, and K. He, "Oil price forecasting with an EMD-based multiscale neural network learning paradigm," in *International Conference on Computational Science*, pp. 925–932, 2007.

[27] S. Zhou and K. K. Lai, "An improved EMD online learning-based model for gold market forecasting," in *Proceedings of the 3rd International Conference on Intelligent Decision Technologies*, pp. 75–84, 2011.

[28] K. He, C. Xie, and K. K. Lai, "Estimating real estate value-at-risk using wavelet denoising and time series model," in *Computational Science—ICCS 2008*, vol. 5102 of *Lecture Notes in Computer Science*, pp. 494–503, Springer, Berlin, Germany, 2008.

[29] S. Zhou, K. K. Lai, and J. Yen, "A dynamic meta-learning rate-based model for gold market forecasting," *Expert Systems with Applications*, vol. 39, no. 6, pp. 6168–6173, 2012.

[30] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, *A Practical Guide to Support Vector Classification*, Department of Computer Science and Information Engineering, University of National Taiwan, Taipei, Taiwan, 2003.

[31] J. Lakonishok, A. Shleifer, and R. W. Vishny, "Contrarian investment, extrapolation, and risk," *Journal of Finance*, vol. 49, pp. 1541–1578, 1994.

[32] M. T. Leung, H. Daouk, and A.-S. Chen, "Forecasting stock indices: a comparison of classification and level estimation models," *International Journal of Forecasting*, vol. 16, no. 2, pp. 173–190, 2000.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 801465, 8 pages
http://dx.doi.org/10.1155/2015/801465

Research Article

Accelerated Lifetime Data Analysis with a Nonconstant Shape Parameter

Guodong Wang, Zhanwen Niu, and Zhen He

College of Management and Economics, Tianjin University, Tianjin 300072, China

Correspondence should be addressed to Zhen He; zhhe@tju.edu.cn

Received 9 May 2014; Revised 29 September 2014; Accepted 13 October 2014

Academic Editor: Shifei Ding

Accelerated life test is commonly used for the estimation of high-reliability product. In this paper, we present a simple and efficient approach to estimate the coefficients of acceleration models. Assuming that both scale and shape parameters of Weibull lifetime distribution vary with stress factors, we estimate the parameters of Weibull distribution using maximum likelihood method and reduce the bias of shape parameter estimator. Considering the heteroscedasticity, we compute the estimates of the coefficients of acceleration models through weighted least square method. Additionally, we obtain the confidence interval of low percentile via bootstrapping. We compare the proposed method with other methods using a real lifetime example. Finally, we study the performance of the proposed method by simulation. The simulation results show that our proposed method is effective.

1. Introduction

In reliability experiments, one is often interested in obtaining inference on low percentile at normal conditions. In today's competitive market, for high-reliability products, the experiments are often censored within a limited period of time. Thus, it is very difficult or impossible to get enough lifetime data to estimate reliability. Accelerated life test (ALT) is commonly used for forcing products to fail more quickly. In ALT, products are tested under higher than usual conditions to reduce the test time and to collect more failure time information. Detailed description on ALT can be found in the studies by Lawless [1], Nelson [2], and Meeker and Escobar [3]. Some software packages have been developed and refined for reliability data analysis, such as Minitab, R, and SAS. Rigdon et al. [4] provided a tutorial on lifetime data analysis using the statistical software packages.

Often, the methods for lifetime data analysis assumed that the scale parameter varies with stress factors but the shape parameter remains constant when lifetime data followed a Weibull distribution. In many applications, however, this assumption may not be appropriate, such as metal fatigue, electronics reliability, and reliability physics. Meeter and Meeker [5] developed the statistical models and optimum ALT with a nonconstant shape parameter, and more examples with a nonconstant shape parameter can be found in their references. Seo et al. [6] designed accelerated life test sampling plans with a nonconstant shape parameter of Weibull distribution.

There are only a few attempts trying to analyze lifetime data based on a Weibull distribution with a nonconstant shape parameter. Maximum likelihood method is generally recommended for estimating the coefficients of accelerated failure time models. The estimates can be obtained by solving likelihood equations. However, the closed form does not exist. Instead, they are approximated by numerical methods. The Newton-Raphson algorithm is very sensitive to the starting points, and it may fail when the number of the starting points is big. To ensure the estimates converge, Balakrishnan and Ling [7] found maximum likelihood estimates through expectation maximization (EM) algorithm. Wang and Kececioglu [8] presented a generalized linear model (GLM) approach to estimate the coefficients of acceleration models. However, the EM or GLM approach does not seem to have enjoyed extensive application probably because it is too "statistical" for some practitioners to intuitively grasp.

In this paper, we present an approach for estimating the coefficients of accelerated failure time models using weighted

least square method. Firstly, we obtain the maximum likelihood estimates of Weibull shape and scale parameters. Secondly, we compute asymptotic covariance matrix by inverting the observed Fisher information matrix. Finally, we get the corresponding coefficients for acceleration models by weighted least square method. Furthermore, we introduce a procedure for performing confidence intervals of low percentiles via bootstrapping. It is well known that the maximum likelihood estimators are biased. We reduce the bias of shape parameter estimator by unbiasing factor method or modified maximum likelihood method.

The paper is organized as follows: in the following section, we introduce model description and assumption of ALT. Then, we provide a method including coefficients estimation and confidence interval estimation of low percentiles based on bootstrapping. Next we compare our proposed method with EM and GLM methods via a real lifetime example. And then we study the performance of the proposed method by simulation. Finally, we present some conclusions and future work.

2. Model Description and Assumption

The Weibull distribution is a very popular distribution for modeling the lifetime data because of its flexibility in being able to model multiple types of failure mechanisms. For T, a random variable denoting the lifetime data, the commonly used parameterization of the two-parameter Weibull probability density function (PDF) is

$$f(t;\eta,\beta) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1}\exp\left(-\left(\frac{t}{\eta}\right)^{\beta}\right), \tag{1}$$

and the cumulative distribution function (CDF) is

$$F(t;\eta,\beta) = 1-\exp\left(-\left(\frac{t}{\eta}\right)^{\beta}\right), \tag{2}$$

where $t > 0$ is the time to failure and $\eta > 0$ and $\beta > 0$ are the scale and shape parameters, respectively.

We consider an ALT with right censored data involving m stress levels, and the number of the samples of ith stress level is n_i. We assume that the failure times of the ith stress level follow a Weibull distribution with scale parameter η_i and shape parameter β_i. η_i and β_i depend on the (possibly transformed) stress levels through the acceleration models

$$\begin{aligned}\log\eta_i &= \mathbf{a}'\mathbf{X}_i,\\ \log\beta_i &= \mathbf{b}'\mathbf{X}_i,\end{aligned} \tag{3}$$

where $\mathbf{a}$ and $\mathbf{b}$ are the vectors of regression parameters and $\mathbf{X}_i$ is the ith vector of designed stress level.

3. Proposed Method

This section contains a presentation of proposed method for estimating the parameters of acceleration models and bootstrapping for confidence intervals of low percentiles. We estimate low percentiles by maximum likelihood method and weighted least square method and then predict intervals by bootstrapping.

3.1. Estimating Parameters and Reducing Bias. Let t_{ij} be the lifetime for the jth item within the ith stress level, and it follows a Weibull distribution. The likelihood function with right censoring present is

$$\mathscr{L}\left(t_{ij};\eta_i,\beta_i\right) = C\prod_{j=1}^{n_i}\left(f\left(t_{ij}\right)\right)^{\delta_{ij}}\left(1-F\left(t_{ij}\right)\right)^{1-\delta_{ij}}, \tag{4}$$

where $\delta_{ij} = 1$ if the item fails and $\delta_{ij} = 0$ if the item is right censored. C is a constant dependent on censoring type.

The log likelihood function for the ith stress level with right censoring present is

$$\begin{aligned}\ell\left(t_{ij};\eta_i,\beta_i\right) &= \log C + \log\beta_i\sum_{j=1}^{n_i}\delta_{ij} - \beta_i\sum_{j=1}^{n_i}\delta_{ij}\log\eta_i\\ &\quad + \left(\beta_i-1\right)\sum_{j=1}^{n_i}\delta_{ij}\log t_{ij} - \sum_{j=1}^{n_i}\left(\frac{t_{ij}}{\eta_i}\right)^{\beta_i}.\end{aligned} \tag{5}$$

The shape parameter and scale parameter can be estimated by solving likelihood equations or maximizing formula (5).

The lifetime data may be censored and the variances of error terms may not be constant for different stress levels, so there is a problem of heteroscedasticity. Thus, we use weighted least square method for estimating parameters. The parameter estimates, $\widehat{\mathbf{a}}$ and $\widehat{\mathbf{b}}$, can be computed by

$$\begin{aligned}\widehat{\mathbf{a}} &= \left(\mathbf{X}'\mathbf{W}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}^{-1}\log\widehat{\boldsymbol{\eta}},\\ \widehat{\mathbf{b}} &= \left(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{V}^{-1}\log\widehat{\boldsymbol{\beta}},\end{aligned} \tag{6}$$

where $\mathbf{X}$ is a matrix of designed stress levels and

$$\begin{aligned}\mathbf{log}\widehat{\boldsymbol{\eta}} &= \left(\log\widehat{\eta}_1,\ldots,\log\widehat{\eta}_m\right)',\\ \mathbf{W} &= \left\langle\widehat{\mathrm{Var}}\left(\log\widehat{\eta}_i\right)\right\rangle = \begin{bmatrix}\widehat{\mathrm{Var}}\left(\log\widehat{\eta}_1\right) & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \widehat{\mathrm{Var}}\left(\log\widehat{\eta}_m\right)\end{bmatrix};\\ \mathbf{log}\widehat{\boldsymbol{\beta}} &= \left(\log\widehat{\beta}_1,\ldots,\log\widehat{\beta}_m\right)',\\ \mathbf{V} &= \left\langle\widehat{\mathrm{Var}}\left(\log\widehat{\beta}_i\right)\right\rangle = \begin{bmatrix}\widehat{\mathrm{Var}}\left(\log\widehat{\beta}_1\right) & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \widehat{\mathrm{Var}}\left(\log\widehat{\beta}_m\right)\end{bmatrix}.\end{aligned} \tag{7}$$

The estimated variances of $\log\widehat{\eta}_i$ and $\log\widehat{\beta}_i$ can be obtained by inverting the observed Fisher information matrix. Now, we introduce the procedure in brief.

From formula (5), it can be shown that

$$-\frac{\partial^2\ell}{\partial\beta_i^2} = \frac{1}{\beta_i^2}\sum_{j=1}^{n}\delta_{ij} + \sum_{j=1}^{n}\left(\frac{t_{ij}}{\eta_i}\right)^{\beta_i}\log^2\left(\frac{t_{ij}}{\eta_i}\right),$$

$$-\frac{\partial^2\ell}{\partial\eta_i^2} = \frac{\beta_i(\beta_i+1)}{\eta_i^2}\sum_{j=1}^{n}\left(\frac{t_{ij}}{\eta_i}\right)^{\beta_i} - \frac{\beta_i}{\eta_i^2}\sum_{j=1}^{n}\delta_{ij},$$

$$-\frac{\partial^2\ell}{\partial\beta_i\partial\eta_i} = -\frac{\partial^2\ell}{\partial\eta_i\partial\beta_i} = \frac{1}{\eta_i}\sum_{j=1}^{n}\delta_{ij} - \frac{\beta_i}{\eta_i}\sum_{j=1}^{n}\left(\frac{t_{ij}}{\eta_i}\right)^{\beta_i}\log\left(\frac{t_{ij}}{\eta_i}\right) - \frac{1}{\eta_i}\sum_{j=1}^{n}\left(\frac{t_{ij}}{\eta_i}\right)^{\beta_i}. \tag{8}$$

Rinne [9] pointed out that if there does not exist censoring within the ith stress level, then the estimated asymptotic variance-covariance matrix of $\widehat{\boldsymbol{\theta}}_i = (\widehat{\eta}_i, \widehat{\beta}_i)'$ is

$$\widehat{\mathrm{Var}}\left(\widehat{\boldsymbol{\theta}}_i\right) = \left[\begin{array}{cc} -E\left(\dfrac{\partial^2\ell}{\partial\eta_i^2}\right) & -E\left(\dfrac{\partial^2\ell}{\partial\eta_i\partial\beta_i}\right) \\ -E\left(\dfrac{\partial^2\ell}{\partial\beta_i\partial\eta_i}\right) & -E\left(\dfrac{\partial^2\ell}{\partial\beta_i^2}\right) \end{array}\right]^{-1}_{|\boldsymbol{\theta}_i=\widehat{\boldsymbol{\theta}}_i} = \frac{1}{n_i}\left[\begin{array}{cc} 1.1087\dfrac{\widehat{\eta}_i^2}{\widehat{\beta}_i^2} & 0.2570\widehat{\eta}_i \\ 0.2570\widehat{\eta}_i & 0.6079\widehat{\beta}_i^2 \end{array}\right]. \tag{9}$$

If there exists censoring within the ith stress level, the estimated asymptotic variance-covariance matrix is then

$$\widehat{\mathrm{Var}}\left(\widehat{\boldsymbol{\theta}}_i\right) = \left[\begin{array}{cc} -\dfrac{\partial^2\ell}{\partial\eta_i^2} & -\dfrac{\partial^2\ell}{\partial\eta_i\partial\beta_i} \\ -\dfrac{\partial^2\ell}{\partial\beta_i\partial\eta_i} & -\dfrac{\partial^2\ell}{\partial\beta_i^2} \end{array}\right]^{-1}_{|\boldsymbol{\theta}_i=\widehat{\boldsymbol{\theta}}_i}. \tag{10}$$

We calculate the estimates of variances by $\widehat{\mathrm{Var}}(\log\widehat{\eta}_i) = \widehat{\mathrm{Var}}(\widehat{\eta}_i)/\widehat{\eta}_i^2$ and $\widehat{\mathrm{Var}}(\log\widehat{\beta}_i) = \widehat{\mathrm{Var}}(\widehat{\beta}_i)/\widehat{\beta}_i^2$.

It is well known that the maximum likelihood estimate of the Weibull shape parameter is biased. The bias may be very serious in the case of small sample or heavy censoring. Yang and Xie [10] proposed an estimator for the shape parameter of the Weibull distribution under the notion of parameter orthogonalization. The estimator proposed by Yang and Xie [10] can reduce the bias for complete and censored data. It can be easily proved that the ratio between the estimator obtained from the study by Yang and Xie [10] and the shape parameter is a pivotal quantity for complete and type II censoring. We can reduce the bias further via unbiasing factor method.

The theoretical justification for using unbiasing factor method is based on the existence of pivotal quantity. Thoman et al. [11] proved that there are two pivotal quantities with shape parameter and scale parameter of Weibull distribution for complete data, and then Monte Carlo method is used to obtain the unbiasing factor of the shape parameter estimator. Ross [12] introduced a formula for calculating the unbiasing factor of the shape parameter estimator in the case of complete data. Ross [13] extended this work to the case of type II censored data with 50% or less censoring.

In this paper, we reduce the bias of maximum likelihood estimator of shape parameter using modified maximum likelihood method which is proposed by Yang and Xie [10] for the case of the failure time is type I censored. When the failure time is complete or type II censored, we reduce the bias using unbiasing factor method.

3.2. Interval Prediction. In reliability data analysis, interest is focused on the low quantities. These low percentiles provide engineers with an evaluation of the product's early failures along with providing information for specification limits, warranty, and cost analysis, but we cannot compute confidence intervals around predicted values because the Fisher information matrix does not exist.

Bootstrapping is a straightforward way to derive estimates of standard errors and confidence intervals for complex estimators of complex parameters of the distribution. DiCiccio and Efron [14] presented several types of bootstrap confidence intervals including standard, percentile, and bootstrap-t. Edwards et al. [15] compared two wood plastic composite extruders using bootstrapping confidence interval of percentiles.

We now describe a procedure for constructing the 100 $(1-\alpha)$% percentile bootstrapping confidence interval which involves the following steps.

(1) Obtain $\widehat{\eta}_i$ and $\widehat{\beta}_i$ from the original samples and reduce the bias of $\widehat{\beta}_i$.

(2) Simulate bootstrapping samples from Weibull distributions with the same data type.

(3) Obtain estimates of $\widehat{\eta}_i$ and $\widehat{\beta}_i$, $\widehat{\eta}_i^*$ and $\widehat{\beta}_i^*$, based on the bootstrapping samples and reduce the bias of $\widehat{\beta}_i^*$.

(4) Compute asymptotic variance-covariance matrix and obtain the estimates of coefficients $\widehat{\mathbf{a}}^*$ and $\widehat{\mathbf{b}}^*$.

(5) Compute $\widehat{t}_p^* = \exp(\widehat{\mathbf{a}}^{*\prime}\mathbf{x})(-\log(1-p))^{\wedge}(1/\exp(\widehat{\mathbf{b}}^{*\prime}\mathbf{x}))$, where $\mathbf{x}$ is the vector of stress level.

(6) Repeat steps (2)–(5) K times to obtain a sample of bootstrapping estimate, $\widehat{t}_p^k$, for $k = 1, 2, \ldots, K$, and then arrange the bootstrapping in ascending order.

The 100 $(1-\alpha)$% percentile bootstrapping confidence interval for t_p is given by

$$\left(\widehat{t}_p^{(\alpha/2)(K+1)}, \widehat{t}_p^{(1-(\alpha/2))(K+1)}\right). \tag{11}$$

4. Illustrative Example

In this section, we demonstrate the practical application of our proposed method and compare our proposed method

Table 1: Failure data for the seven operating policies.

Voltage X_1	Operation type X_2	Load X_3	Time to failure (hours)
3	1	0.20	71.79 85.24 96.01 100.68 104.21 112.65 114.84 115.03 128.66
3	1	0.48	51.99 74.19 74.69 91.05 91.05 100.11 112.99
6	1	0.30	9.03 10.62 12.38 13.92 15.48 18.82
6	1	0.65	4.27 5.32 5.76 6.68 7.62 8.66 8.78 9.19 10.33 10.78
3	0	0.48	84.57 115.79 151.20 154.85 164.55 186.24
6	0	0.65	6.57 8.25 8.50 8.87 8.96 9.74 9.98 11.58 12.09 13.89
3	0	0.08	152.74 176.32 195.35 213.32 214.07 228.26 242.18 264.56

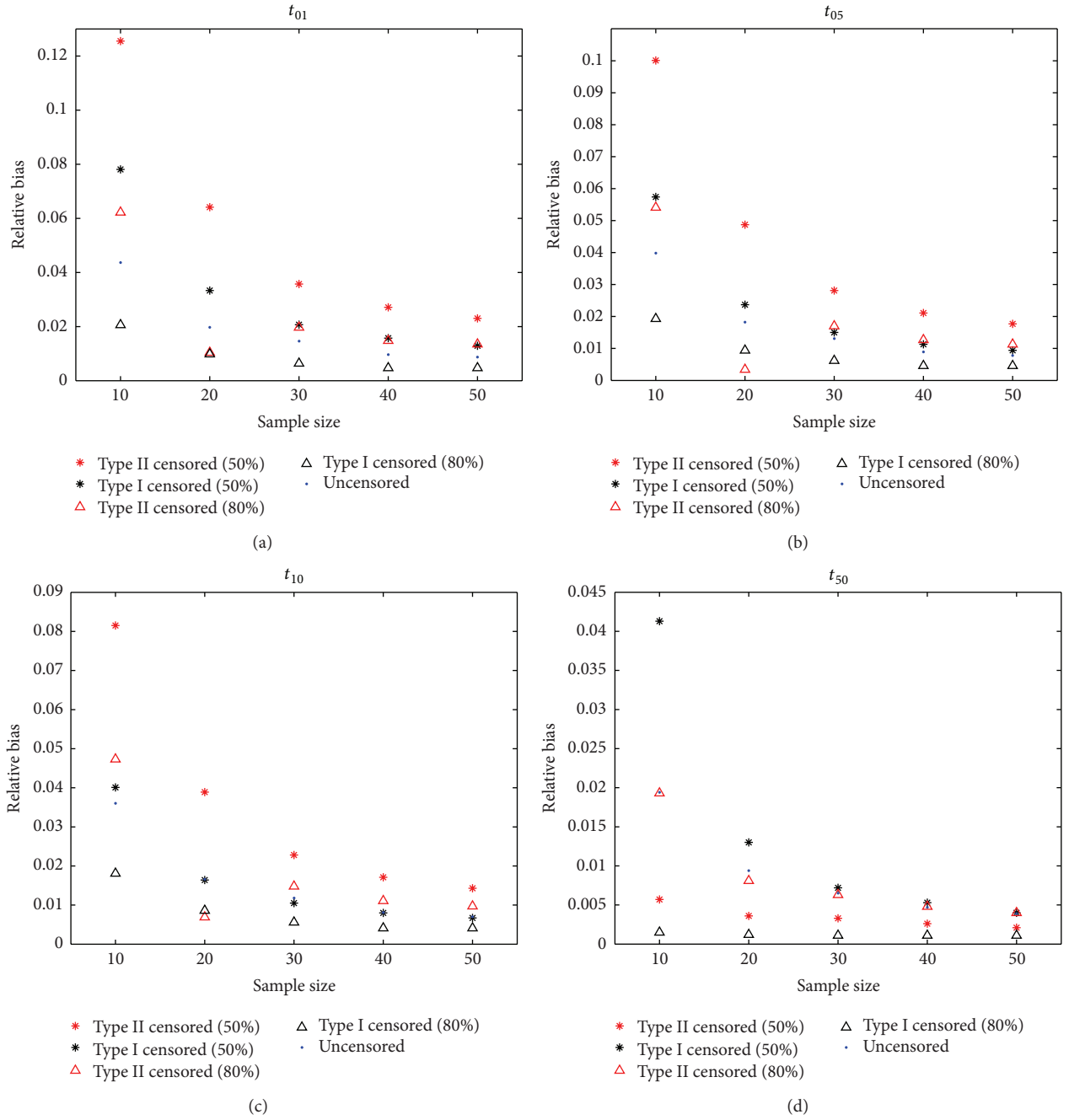


Figure 1: The relative biases of percentiles for the case of $m = 7$.

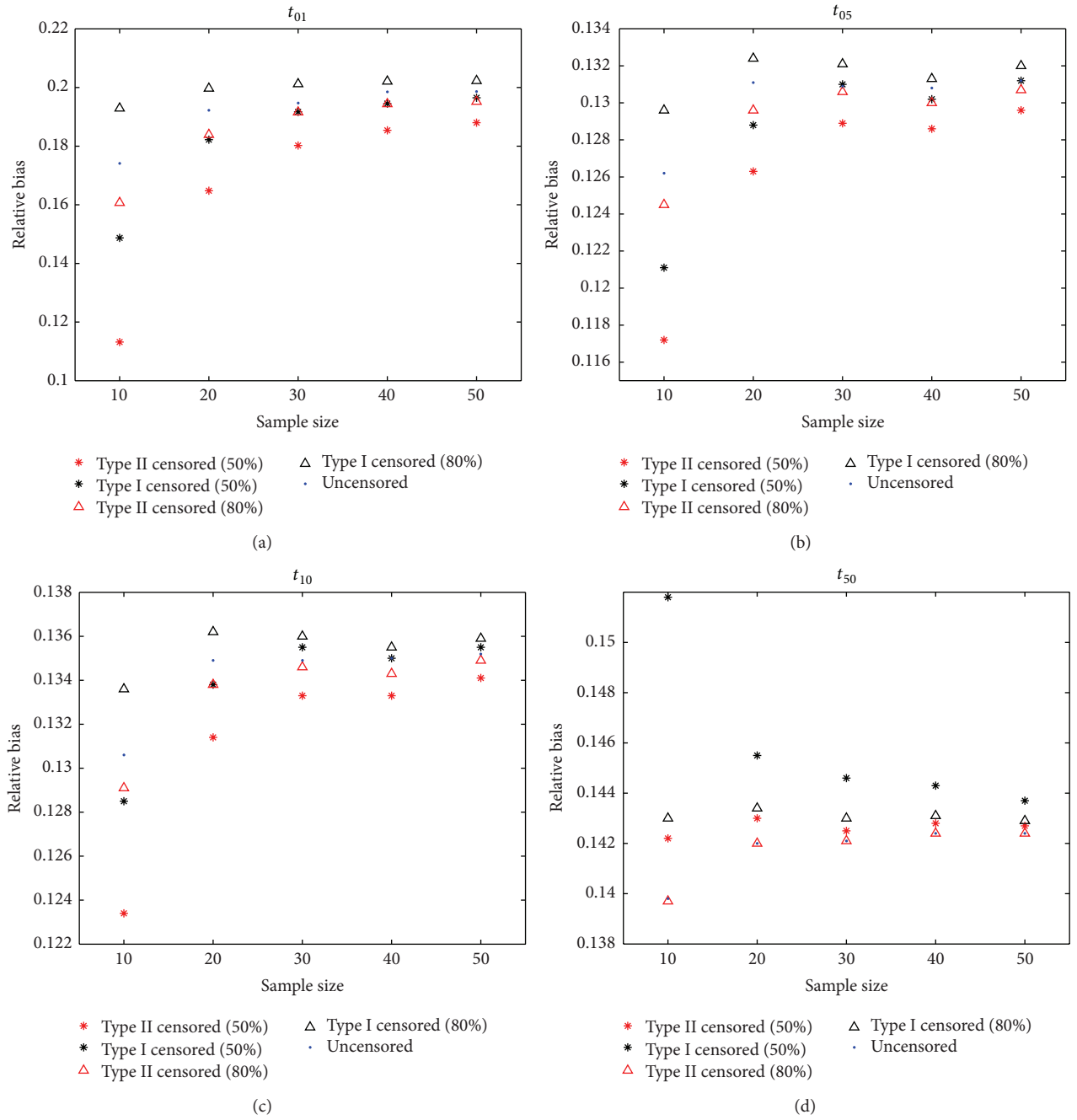


Figure 2: The relative biases of percentiles for the case of $m = 4$.

with EM and GLM methods. Teng and Kolarik [16] designed and performed a life test experiment to study the on/off cycling characteristics of small DC motors. The voltage (X_1), operation type (X_2), and load (X_3) were considered to be stress factors. The lifetime data are summarized in Table 1.

We consider the main factors and 2-interaction effects in this example. Because the lifetime data is complete, we calculate the estimates of Weibull parameters by unbiasing factor method. We repeat steps (2)–(5) 10,000 times and get the values of unbiasing factors for $\widehat{\beta}$ that are shown in Table 2.

Table 2: Unbiasing factor values of different number of samples.

Number	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
Unbiased factor value	1.3302	1.2638	1.2201	1.1922	1.1645

The results in Table 2 indicate that $\widehat{\beta}$ is greater than true value of β. The bias of $\widehat{\beta}$ gradually becomes smaller when n increases from 5 up to 10. But the relative bias is very serious even when $n = 10$. Table 3 shows the values of estimates

Table 3: The estimates of shape and scale parameters.

Stress level	1	2	3	4	5	6	7
$\widehat{\beta}$	6.4018	4.2943	3.4102	3.7409	3.9950	4.3877	5.9420
$\widehat{\eta}$	109.0104	90.9359	14.2697	8.4077	152.7174	10.5409	222.5437

Table 4: The estimates of **a** and **b**.

Predictor	Proposed method		GLM method		EM method	
	$\widehat{\mathbf{a}}$	$\widehat{\mathbf{b}}$	$\widehat{\mathbf{a}}$	$\widehat{\mathbf{b}}$	$\widehat{\mathbf{a}}$	$\widehat{\mathbf{b}}$
Intercept	7.4322	2.6977	7.410	2.667	7.4044	2.7464
X_1	−0.6506	−0.2788	−0.640	−0.208	−0.6380	−0.2338
X_2	−0.9019	0.4384	−0.910	0.338	−0.9033	0.3339
X_3	−0.0774	−2.6829	0.006	−1.879	0.0019	−2.0063
$X_1 X_2$	0.0808	−0.0527	0.081	−0.035	0.0793	−0.0258
$X_1 X_3$	0.2939	−0.4335	0.301	−0.441	0.3077	−0.5125
$X_2 X_3$	−0.2880	0.5635	−0.309	0.367	−0.3097	0.4082

obtained through maximum likelihood method. Bias of the estimator, $\widehat{\beta}$, shown in Table 3 has been reduced via unbiasing factor method.

We estimate the coefficients of acceleration models using different methods. Table 4 shows the results of estimates of **a** and **b** obtained by our proposed method, GLM method, and EM method. We see in Table 4 that the estimates of **a** and **b** obtained by different methods have similar values.

Practitioners may be more interested in the lifetime of p-percentile. We calculate the percentiles in the case of $p = 0.01$, 0.05, 0.10, 0.50. In order to compare our proposed method with GLM and EM methods, we evaluate the estimates of low percentiles by the relative bias (RB), where $\mathrm{RB}(\widehat{t}_{ip}) = |\widehat{t}_{ip} - t_{ip}|/t_{ip}$. The procedure is described as follows.

(1) Calculate $t_{ip} = \exp(\log \eta_i + (\Phi^{-1}_{\mathrm{SEV}}(p)/\beta_i))$, where $\Phi^{-1}_{\mathrm{SEV}}(p)$ is the inverse cumulative distribution function for the smallest extreme value distribution and β_i and η_i are obtained from Table 3.

(2) Estimate $\widehat{\mathbf{a}}$ and $\widehat{\mathbf{b}}$ using different methods and then calculate $\widehat{\beta}_i$ and $\widehat{\eta}_i$, respectively.

(3) Calculate $\widehat{t}_{ip} = \exp(\log \widehat{\eta}_i + (\Phi^{-1}_{\mathrm{SEV}}(p)/\widehat{\beta}_i))$ using $\widehat{\beta}_i$ and $\widehat{\eta}_i$ obtained from step (2).

(4) Compare our proposed method with GLM and EM methods according to the values of RB.

The results are shown in Table 5. From Table 5 we can find that the values of RB_2 are close to the values of RB_3. RB_1 is the smallest, meaning that our proposed method is the best.

Additionally, our proposed method can obtain confidence interval using bootstrapping. The number of resampling is 10,000 times. We calculate the percentiles and their inferences in the case of $p = 0.01, 0.05, 0.10, 0.50$. The results are shown in Table 6.

5. Simulation Study

This section studies the effect of several factors on the performance of the proposed approach. In this paper, the particular factors used are the number of stress levels m, the sample size at the ith stress level n_i, censoring time t_c, and the proportion of noncensoring data q.

Teng and Kolarik [16] designed a life experiment which is cited in Section 4. We study the cases of $m = 4, 7$, $n_i = 10, 20, 30, 40, 50$, and $q = 0.5, 0.8, 1$. For the case of $m = 4$, we choose the 1st, 3rd, 5th, and 7th stress levels as new stress levels and only consider main factors. For type I censoring, the life experiment of the ith stress level is terminated at t_c which is subject to $F(t_c) = q$. The results are based on 10,000 simulations.

Figure 1 shows the changes of relative biases for different percentiles in the case of $m = 7$. The relative biases of percentiles are the averages of all stress levels. We see in Figure 1 that the relative biases for all the types of censoring and the proportion of noncensoring data become smaller when n_i increases from 10 up to 50. The relative biases are less than 0.1 in most cases.

Figure 2 shows the changes of relative biases for different percentiles in the case of $m = 4$. From Table 2 we find that the relative biases are less than 0.2 in most cases. For $p = 0.05$, 0.10, and 0.50, the relative biases are less than 0.15.

6. Conclusions and Future Work

In this paper, we propose a new approach to estimate the coefficients of acceleration models. Our method is much simpler and can be complemented easily. The confidence interval of low percentile is difficult to calculate analytically, so we obtain the numerical solution via bootstrapping.

Many reliability life tests contain blocking or subsampling. This is often because treatments are directly applied to test stands rather than individual specimen. Freeman and Vining [17] presented two-stage method to deal with subsampling. Wang et al. [18] extended Freeman and Vining's method such that it can compute confidence interval of low percentile, but their research assumes that the shape parameter is constant. The clear future work is to analyze reliability data with subsampling under shape and scale parameters that are varied.

Conflict of Interests

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

Table 5: Relative bias of $\widehat{t}_p$.

Stress level	$t_{0.01}$			$t_{0.05}$			$t_{0.10}$			$t_{0.50}$		
	RB_1	RB_2	RB_3	RB_1	RB_2	RB_3	RB_1	RB_2	RB_3	RB_1	RB_2	RB_3
1	0.02	13.12	14.07	0.01	8.58	9.25	0.01	6.63	7.18	0.01	1.69	1.96
2	0.18	26.92	26.07	0.01	17.27	16.86	0.01	13.25	13.01	0.01	3.36	3.52
3	0.48	43.06	41.99	0.03	27.08	26.57	0.03	20.60	20.31	0.01	5.17	5.34
4	0.41	20.16	21.30	0.03	13.04	13.81	0.03	10.03	10.65	0.01	2.53	2.79
5	0.17	35.59	35.27	0.01	22.56	22.26	0.01	17.21	16.92	0.01	4.29	4.03
6	0.32	17.32	17.71	0.03	11.40	11.67	0.02	8.88	9.11	0.02	2.55	2.67
7	0.14	16.27	16.38	0.01	10.66	10.73	0.01	8.27	8.32	0.00	2.26	2.26

RB_1, RB_2, and RB_3 are relative biases of the proposed method, GLM method, and EM method (in %), respectively.

Table 6: The percentile estimates and bootstrapping confidence intervals.

Stress level	$t_{0.01}$	$t_{0.05}$	$t_{0.10}$	$t_{0.50}$
1	335.63	658.67	887.12	1933.73
	[20.5, 706.1]	[139.0, 1200.3]	[289.6, 1674.9]	[849.8, 5851.5]
2	115.83	266.04	384.08	1004.16
	[1.5, 401.6]	[15.2, 597.2]	[42.4, 726.0]	[376.6, 1482.2]
3	412.76	659.81	811.68	1395.83
	[109.0, 701.8]	[282.0, 1067.0]	[411.9, 1302.3]	[886.0, 2552.7]
4	92.54	207.08	295.54	749.69
	[7.1, 187.9]	[39.7, 323.6]	[81.5, 414.4]	[376.3, 965.4]
5	425.63	589.89	681.34	993.51
	[154.4, 589.5]	[307.5, 729.7]	[412.9, 804.0]	[777.8, 1058.3]
6	73.77	160.95	227.14	559.60
	[6.5, 149.5]	[33.6, 249.0]	[67.4, 315.4]	[294.4, 723.8]
7	388.28	487.25	538.65	700.28
	[143.8, 509.9]	[243.0, 637.1]	[282.2, 712.1]	[354.5, 1039.6]
8	58.67	124.91	174.39	417.63
	[1.2, 183.9]	[10.0, 264.9]	[24.8, 317.4]	[178.5, 620.8]

Acknowledgment

This research was supported in part by Grants 71401123, 70931004, 71071107, and 71225006 from the National Natural Science Foundation of China.

References

[1] J. F. Lawless, *Statistical Models and Methods for Lifetime Data*, John Wiley & Sons, New York, NY, USA, 1982.

[2] W. B. Nelson, *Accelerated Testing: Statistical Models, Test Plans and Data Analysis*, John Wiley & Sons, Hoboken, NJ, USA, 1990.

[3] W. Q. Meeker and L. A. Escobar, *Statistical Methods for Reliability Data*, John Wiley & Sons, New York, NY, USA, 1998.

[4] S. E. Rigdon, B. R. Englert, I. A. Lawson, C. M. Borror, D. C. Montgomery, and R. Pan, "Experiments for reliability achievement," *Quality Engineering*, vol. 25, no. 1, pp. 54–72, 2012.

[5] C. A. Meeter and W. Q. Meeker, "Optimum accelerated life tests with a nonconstant scale parameter," *Technometrics*, vol. 36, no. 1, pp. 71–83, 1994.

[6] J. H. Seo, M. Jung, and C. M. Kim, "Design of accelerated life test sampling plans with a nonconstant shape parameter," *European Journal of Operational Research*, vol. 197, no. 2, pp. 659–666, 2009.

[7] N. Balakrishnan and M. H. Ling, "Expectation maximization algorithm for one shot device accelerated life testing with weibull lifetimes, and variable parameters over stress," *IEEE Transactions on Reliability*, vol. 62, no. 2, pp. 537–551, 2013.

[8] W. Wang and D. B. Kececioglu, "Fitting the Weibull log-linear model to accelerated life-test data," *IEEE Transactions on Reliability*, vol. 49, no. 2, pp. 217–223, 2000.

[9] H. Rinne, *The Weibull Distribution: A Handbook*, CRC Press, Boca Raton, Fla, USA, 2008.

[10] Z. Yang and M. Xie, "Efficient estimation of the Weibull shape parameter based on a modified profile likelihood," *Journal of Statistical Computation and Simulation*, vol. 73, no. 2, pp. 115–123, 2003.

[11] D. R. Thoman, L. J. Bain, and C. E. Antle, "Inferences on the parameters of the Weibull distribution," *Technometrics*, vol. 11, pp. 445–460, 1969.

[12] R. Ross, "Formulas to describe the bias and standard deviation of the ML-estimated Weibull shape parameter," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 1, no. 2, pp. 247–253, 1994.

[13] R. Ross, "Bias and standard deviation due to Weibull parameter estimation for small data sets," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 3, no. 1, pp. 28–42, 1996.

[14] T. J. DiCiccio and B. Efron, "Bootstrap confidence intervals," *Statistical Science*, vol. 11, no. 3, pp. 189–228, 1996.

[15] D. J. Edwards, R. V. León, T. M. Young, F. M. Guess, and K. A. Crookston, "Comparison of two wood plastic composite extruders using bootstrap confidence intervals on measurements of sample failure data," *Quality Engineering*, vol. 25, no. 1, pp. 23–33, 2012.

[16] N. H. Teng and W. J. Kolarik, "On/off cycling under multiple stresses," *IEEE Transactions on Reliability*, vol. 38, no. 4, pp. 494–498, 1989.

[17] L. J. Freeman and G. G. Vining, "Reliability data analysis for life test experiments with subsampling," *Journal of Quality Technology*, vol. 42, no. 3, pp. 233–241, 2010.

[18] G. D. Wang, Z. W. Niu, L. Qu, Z. He, and Z. J. Li, "Bootstrapping analysis of lifetime data with subsampling," in *Proceedings of the Industrial and Systems Engineering Research Conference*, 2014.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 671872, 11 pages
http://dx.doi.org/10.1155/2014/671872

Research Article

Fitness Estimation Based Particle Swarm Optimization Algorithm for Layout Design of Truss Structures

Ayang Xiao,[1] Benli Wang,[1] Chaoli Sun,[2] Shijie Zhang,[1] and Zhenguo Yang[1]

[1] *Research Centre of Satellite Technology, Harbin Institute of Technology, Harbin 150001, China*
[2] *Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology, Taiyuan 030024, China*

Correspondence should be addressed to Ayang Xiao; ayang.xiao@gmail.com

Received 2 August 2014; Revised 8 September 2014; Accepted 9 September 2014; Published 29 September 2014

Academic Editor: Shifei Ding

Due to the fact that vastly different variables and constraints are simultaneously considered, truss layout optimization is a typical difficult constrained mixed-integer nonlinear program. Moreover, the computational cost of truss analysis is often quite expensive. In this paper, a novel fitness estimation based particle swarm optimization algorithm with an adaptive penalty function approach (FEPSO-AP) is proposed to handle this problem. FEPSO-AP adopts a special fitness estimate strategy to evaluate the similar particles in the current population, with the purpose to reduce the computational cost. Further more, a laconic adaptive penalty function is employed by FEPSO-AP, which can handle multiple constraints effectively by making good use of historical iteration information. Four benchmark examples with fixed topologies and up to 44 design dimensions were studied to verify the generality and efficiency of the proposed algorithm. Numerical results of the present work compared with results of other state-of-the-art hybrid algorithms shown in the literature demonstrate that the convergence rate and the solution quality of FEPSO-AP are essentially competitive.

1. Introduction

As a typical real world project, truss structural analysis is considered to be computationally expensive [1]. Moreover, truss layout optimization, which solves truss sizing variables (e.g., cross-sectional areas of elements) and shape variables (e.g., coordinates of nodes) simultaneously, is known as a typical multimodal and highly nonlinear problem [2]. The various differences (e.g., physical nature, magnitude, continuity, etc.) between the two types of variables make the truss layout optimization problem a difficult task [3]. Hence, in the past decades, researchers mainly focused on solving truss layout optimization via multilevel methods [4].

Vanderplaats and Moses [5] proposed an alternating gradient method to decompose truss layout optimization into a number of subproblems, each of which optimized a subset of the design variables. The subproblems are solved iteratively until a converged optimal solution is found. Zhou [6] used a similar two-level approximation concept to optimize the cross-sectional areas of the members and the coordinates of the joints. Gil and Andreu [7] used fully stressed design method and conjugate gradient method to optimize the sizing and shape parameters of bridges, respectively.

Due to the strong coupling between the variables, the search efficiency of multilevel methods is often limited [8]. Aiming for addressing this issue and also benefiting from recent rapid advances in computational power, single-level methods which optimize all design variables simultaneously are becoming popular and competitive. Wang et al. [9] presented an optimality criteria (OC) algorithm for spatial truss layout optimization. Fourie and Groenwold [10] used new operators, namely, the elite velocity and the elite particle, in the standard particle swarm optimization (PSO) algorithm to optimize the truss layout, with the purpose of increasing the probability of migration to regions with high fitness. In fact, various metaheuristic algorithms including simulated annealing (SA) [11], genetic algorithm (GA) [12], charged system search (CSS) [13], and artificial bee colony algorithm (ABC) [14] have been introduced to address truss layout optimization problems.

More recently, researchers turn to hybridizing different techniques to further enhance the searching efficiency of metaheuristic algorithms. Lingyun et al. [18] proposed a niche hybrid genetic algorithm to solve the truss shape and sizing optimization in a simple and effective manner. Kaveh and Zolghadr [23] developed a hybridized CSS-BBBC algorithm with trap recognition capability for truss layout optimization. Zuo et al. [24] proposed a hybrid OC-GA approach for fast and global truss layout optimization; Kaveh and Javadi [25] used harmony search and ray optimizer to enhance the PSO algorithm to optimize truss layout under multiple frequency constraints; Liu and Ye [26] designed a genetic simulated annealing algorithm for domes layout optimization. Gholizadeh [17] proposed a hybridized cellular automata and PSO algorithm for truss layout optimization.

However, the common weakness of metaheuristic algorithms based structural optimization is that a huge number of structural analyses are required, which is quite time-consuming. In this paper, we propose a new hybridized algorithm, termed FEPSO-AP for truss layout optimization that aims to enhance the optimal efficiency by using the fitness estimations to partly substitute the computationally expensive fitness calculations. The finite element method (FEM) is adopted to evaluate the structural performance. Empirical results demonstrate that the proposed method is highly promising for truss layout optimization.

2. Statement of Truss Layout Optimization Problem

The main aim of truss layout optimization can be formulated as follows:

$$\begin{aligned} &\text{Minimize:} \quad w(\mathbf{X}) = \sum_{i=1}^{ne} \rho_i \cdot A_i \cdot l_i, \\ &\text{Subject to:} \quad g(\mathbf{X}) \le 0, \end{aligned} \tag{1}$$

where w and g denote the weight of truss structure and the maximum violated structural constraint; $\mathbf{X}$ is the vector of all design variables; ρ_i, l_i, and A_i represent the material density, length, and cross-sectional area of the ith truss element, respectively.

Different types of constraints might be considered simultaneously depending on the problem to be solved. Four typical design constraints involved in this work can be stated by

$$\begin{aligned} g_i^{\sigma}(\mathbf{X}) &= \frac{\sigma_i(\mathbf{X})}{\sigma_{i,\text{all}}} - 1 \le 0, \quad i = 1, 2, \ldots, ne, \\ g_j^{d}(\mathbf{X}) &= \frac{d_j(\mathbf{X})}{d_{j,\text{all}}} - 1 \le 0, \quad j = 1, 2, \ldots, nj, \\ g_i^{b}(\mathbf{X}) &= \frac{|\sigma_i(\mathbf{X})|}{\sigma_{i,er}} - 1 \le 0, \quad \text{if } \sigma_i(\mathbf{X}) < 0, \\ g_k^{f}(\mathbf{X}) &= \frac{f_k(\mathbf{X})}{f_{k,\text{all}}} - 1 \le 0, \quad k = 1, 2, \ldots, nk, \end{aligned} \tag{2}$$

in which g_i^{σ} is the truss element stress constraint, while σ_i and $\sigma_{i,\text{all}}$ stand for the actual largest stress of the ith truss element and its stress limit, respectively; g_j^{d} is the nodal displacement constraint, while d_j and $d_{j,\text{all}}$ stand for the actual largest nodal displacement of the jth node and its displacement limit, respectively; g_i^{b} is the local buckling constraint, while $\sigma_{i,er}$ stands for the Euler critical stress of the ith element; g_k^{f} is the structural natural frequency constraint, while f_k and $f_{k,\text{all}}$ stand for the kth structural natural frequency and its frequency limit; ne, nj, and nk represent the number of structural elements, nodes, and constrained natural frequencies, respectively.

3. FEPSO-AP Algorithm

This section describes the fitness estimation based PSO algorithm with an adaptive penalty function approach (FEPSO-AP) developed in this research. As FEPSO-AP integrates PSO, FE, and AP, this section recalls the basic concept of PSO algorithm, fitness estimation strategy, and adaptive constraint handling approach. Finally, a framework of FEPSO-AP algorithm is presented.

3.1. The PSO Algorithm. As one of the most popular metaheuristic algorithms, PSO has found a wide application in real world projects for its structural concision and searching efficiency [17]. In a standard PSO [27], it is assumed that each of the particles has a position and a certain velocity. The position of particle represents a candidate solution to the optimization problem, and the velocity of particle determines the particle's movement. Hence, the flying of particles can be considered as the swarm searching of design domain.

If the particle flies from its current position to the next position, its velocity and position are updated by

$$\begin{aligned} \vec{v}_i(t+1) &= \omega \vec{v}_i(t) + c_1 \mathbf{r}_1 \left(\vec{p}_i(t) - \vec{x}_i(t)\right) \\ &\quad + c_2 \mathbf{r}_2 \left(\vec{p}_g(t) - \vec{x}_i(t)\right), \\ \vec{x}_i(t+1) &= \vec{x}_i(t) + \vec{v}_i(t+1), \end{aligned} \tag{3}$$

in which $\vec{v}_i(t)$ and $\vec{x}_i(t)$ indicate the velocity and position of the ith particle at iteration t; $\vec{p}_i(t)$ and $\vec{p}_g(t)$ represent the historical best position of the ith particle and the global best position of all particles till iteration t, respectively. ω stands for the inertia weight; c_1 and c_2 are cognitive and social parameters, respectively. $\mathbf{r}_1$ and $\mathbf{r}_2$ are diagonal matrixes, of which the diagonal elements are uniformly distributed random numbers among the range of $(0, 1)$.

3.2. The Fitness Estimation Strategy. Sun et al. [28] proposed a novel fitness estimation strategy for PSO to solve computationally expensive problems.

According to (3), the positions of any two arbitrary particles selected from the swarm can be formulated by

$$\begin{aligned} \vec{x}_i(t+1) &= \left(1 + \omega - \varphi_1 - \varphi_2\right) \vec{x}_i(t) + \varphi_1 \vec{p}_i(t) \\ &\quad + \varphi_2 \vec{p}_g(t) - \omega \vec{x}_i(t-1), \end{aligned}$$

$$\vec{x}_j(t+1) = \left(1+\omega-\varphi_1'-\varphi_2'\right)\vec{x}_j(t) + \varphi_1'\vec{p}_j(t) + \varphi_2'\vec{p}_g(t) - \omega\vec{x}_j(t-1), \tag{4}$$

in which $\varphi_1 = c_1\mathbf{r}_1$, $\varphi_2 = c_2\mathbf{r}_2$, $\varphi_1' = c_1\mathbf{r}_1'$, and $\varphi_2' = c_2\mathbf{r}_2'$. It can be found that the above two equations are both related to the global best position $\vec{p}_g(t)$. Hence, (4) can be abbreviated to

$$\begin{aligned}&\varphi_2\vec{x}_j(t+1) + \varphi_2'\left(1+\omega-\varphi_1-\varphi_2\right)\vec{x}_i(t) + \varphi_2'\varphi_1\vec{p}_i(t) + \varphi_2\omega\vec{x}_j(t-1)\\ &= \varphi_2'\vec{x}_i(t+1) + \varphi_2\left(1+\omega-\varphi_1'-\varphi_2'\right)\vec{x}_j(t) + \varphi_2\varphi_1'\vec{p}_j(t) + \varphi_2'\omega\vec{x}_i(t-1).\end{aligned} \tag{5}$$

Use a virtual position $\vec{x}_v(t+1)$ to present the value of (5). Supposing the influences of all coefficients are similar, Figure 1 describes the geometrical relationship among the virtual position $\vec{x}_v(t+1)$ and all subitems contained by (5). As shown in Figure 1, the fitness of virtual position $\vec{x}_v(t+1)$ can be estimated by either the fitness of $\vec{x}_j(t+1)$, $\vec{x}_i(t)$, $\vec{p}_i(t)$, and $\vec{x}_j(t-1)$ or the fitness of $\vec{x}_i(t+1)$, $\vec{x}_j(t)$, $\vec{p}_j(t)$, and $\vec{x}_i(t-1)$. Hence, if the current fitness value and the historical best fitness of every particle at iteration t and $t-1$ are obtained already, the explicit relationship between the fitness of particle i and j at iteration $t+1$ can be expressed in the following formulations:

$$\begin{aligned}&f^e\left(\vec{x}_j(t+1)\right)\\ &= d_v^j(t+1)\\ &\quad\cdot\left\{\alpha\cdot\left[\frac{f\left(\vec{x}_i(t+1)\right)}{d_v^i(t+1)} + \frac{f\left(\vec{x}_i(t-1)\right)}{d_v^i(t-1)} + \frac{f\left(\vec{x}_j(t)\right)}{d_v^j(t)} + \frac{f\left(\vec{p}_j(t)\right)}{d_v^{pj}(t)}\right]\right.\\ &\quad\left. - \frac{f\left(\vec{x}_j(t-1)\right)}{d_v^j(t-1)} + \frac{f\left(\vec{x}_i(t)\right)}{d_v^i(t)} + \frac{f\left(\vec{p}_i(t)\right)}{d_v^{pi}(t)}\right\},\end{aligned} \tag{6}$$

$$\alpha = \frac{\left(1/d_v^j(t+1)\right) + \left(1/d_v^j(t-1)\right) + \left(1/d_v^i(t)\right) + \left(1/d_v^{pi}(t)\right)}{\left(1/d_v^i(t+1)\right) + \left(1/d_v^i(t-1)\right) + \left(1/d_v^j(t)\right) + \left(1/d_v^{pj}(t)\right)}, \tag{7}$$

in which $d_v^i(t+1)$, $d_v^i(t)$, $d_v^i(t-1)$, $d_v^j(t+1)$, $d_v^j(t)$, $d_v^j(t-1)$, $d_v^{pi}(t)$, and $d_v^{pj}(t)$ represent the distance between the virtual position $\vec{x}_v(t+1)$ and the real positions $\vec{x}_i(t+1)$, $\vec{x}_i(t)$, $\vec{x}_i(t-1)$, $\vec{x}_j(t+1)$, $\vec{x}_j(t)$, $\vec{x}_j(t-1)$, $\vec{p}_i(t)$, and $\vec{p}_j(t)$, respectively.

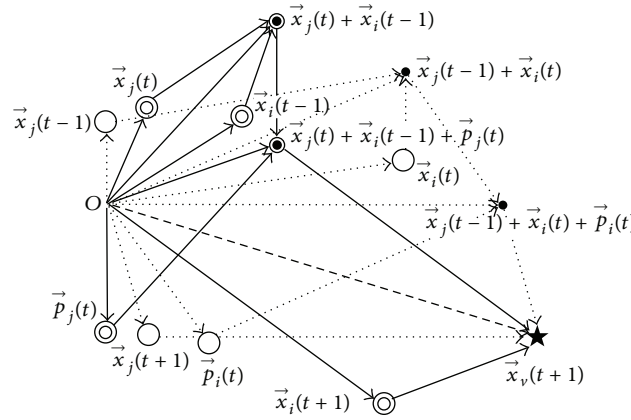


Figure 1: Illustration of the virtual position.

3.3. Adaptive Constraint Handling Approach. As mentioned in Section 2, truss layout optimization problem is often taken as a multiple constrained optimization problem. Coello Coello [29] pointed out that the constraint handling procedure plays an important role in the exploration and exploitation of constrained optimization problems. Similar to other metaheuristic algorithms, FEPSO is designed for unconstrained problems. Hence, it is necessary to incorporate constraint handling approach into FEPSO to solve truss layout optimization problems.

In this work, we use penalty function methods (PEM) to transfer a constrained problem into an unconstrained problem by adding the influence of violated constraints to the initial fitness function. A pseudoobjective function is stated as follows:

$$\Phi(\mathbf{X}) = f(\mathbf{X}) + W \times G(\mathbf{X}), \tag{8}$$

where W and $G(\mathbf{X})$ are positive penalty factor and penalty function. The common difficulty existing in standard PEM is to set the constant penalty factor W properly. Runarsson and Yao [30] pointed out that if the penalty factor W turns out to be too large, the searching landscape would be quite rough, and thus qualified exploitation and exploration of the design domain are hard to be achieved; on the contrary, if the penalty factor W is too small, it would be quite possible to lose feasible solutions.

As summarized in [29], the common target of adaptive penalty factors is to make a good balance between the objective function and the constraints violation. In this work, we developed a laconic adaptive penalty factor as follows:

$$W = 2^{1-\rho}, \tag{9}$$

in which ρ is the ratio of feasible solutions in current population. It can be concluded that if W is too small, the obtained unfeasible solutions will increase the value of ρ, and thus, the algorithm would be pushed to explore more feasible solutions. On the other hand, if the feasible solutions congregate in the population, the value of W will approach zero, and thus a detailed exploitation would be performed.

3.4. The Framework of FEPSO-AP Algorithm. The penalty function $G(\mathbf{X})$ of truss layout optimization problem is determined by

$$G(\mathbf{X}) = w(\mathbf{X}) \times V(\mathbf{X}), \tag{10}$$

$$V(\mathbf{X}) = \max\{0, g(\mathbf{X})\}, \tag{11}$$

in which $\mathbf{X}$ is the structural design vector, $V(\mathbf{X})$ is the violated constraints, and $w(\mathbf{X})$ and $g(\mathbf{X})$ are defined by (1).

According to (8)–(10), the pseudoobjective function can be abbreviated to

$$\Phi(\mathbf{X}) = w(\mathbf{X}) \times \left(1 + 2^{1-\rho} \times V(\mathbf{X})\right). \tag{12}$$

To transfer the formulated truss layout optimization problem into an unconstrained optimization problem, the adaptive penalty function, the fitness estimation strategy, and the PSO algorithm are assembled as follows.

Step 1. Set the termination condition of optimization and the initial values of ω, c_1, and c_2. Set the iteration times iter = 1.

Step 2. Generate the initial positions $\mathbf{X} = \{\vec{x}_i(t)\}$ and the initial velocities $\mathbf{V} = \{\vec{v}_i(t)\}$ of particles population randomly, $i = 1, 2, 3, \ldots,$ popsize.

Step 3. Calculate the fitness values $f(\vec{x}_i(t))$ of all particles and update every particle's historical best position $\vec{p}_i(t)$ and the global best position $\vec{p}_g(t)$ by

$$f(\vec{x}_i(t)) = \Phi(\vec{x}_i(t)), \tag{13}$$

$$\begin{aligned} \vec{p}_i(t) &= \min\left\{f\left(\vec{x}_i\left(t'\right)\right)\right\}, \quad t' = 1, 2, 3, \ldots, t, \\ \vec{p}_g(t) &= \min\left\{f(\vec{x}_i(t))\right\}, \quad i = 1, 2, 3, \ldots, \text{popsize}. \end{aligned} \tag{14}$$

Step 4. Update the velocity $\vec{v}_i(t+1)$ and the position $\vec{x}_i(t+1)$ of every particle based on (3). Set the iteration times iter = iter + 1.

Step 5. Calculate the distances between every pair of particles.

(5.1) Select particle i randomly and calculate its fitness value $f(\vec{x}_i(t+1))$ based on (13).

(5.2) Choose a particle j nearby particle i and calculate the virtual position $\vec{x}_v(t+1)$ by using the historical information of particle j based on (5).

(5.3) If $\vec{x}_v(t+1)$ does not coincide with $\vec{x}_i(t+1)$, $\vec{x}_j(t)$, $\vec{p}_j(t)$, $\vec{x}_i(t-1)$, $\vec{x}_j(t+1)$, $\vec{x}_i(t)$, $\vec{p}_i(t)$, or $\vec{x}_j(t-1)$, estimate the fitness value $f^e(\vec{x}_j(t+1))$ of particle j by (6); otherwise, calculate the fitness value of particle j based on (13).

(5.4) Repeat (5.1)–(5.4) till the fitness values of all particles are evaluated.

Step 6. Update every particle's historical best position $\vec{p}_i(t+1)$ and the global best position $\vec{p}_g(t+1)$. If $\vec{p}_g(t+1)$ is obtained by estimation, recalculate the corresponding particle's fitness value by (13), and rechoose the global best position $\vec{p}_g(t+1)$. Repeat this step till a nonestimated value of $\vec{p}_g(t+1)$ is obtained.

Step 7. Repeat Steps 4–6 till the termination condition is reached. Output the global best position $\vec{p}_g(t+1)$ and its objective value.

4. Benchmark Examples

The following four benchmark examples have been used to demonstrate the generality and efficiency of the FEPSO-AP algorithm:

(i) a planar 15-bar truss subjected to a single load condition and stress constraints,

(ii) a spatial 25-bar truss subjected to a single load condition under stress and displacement constraints,

(iii) a planar 37-bar truss subjected to multiple frequency constraints,

(iv) a planar 47-bar truss subjected to three load conditions under stress and local buckling constraints.

Programs of FEPSO-AP algorithm and structural finite element method (FEM) algorithm are developed by using MATLAB R2013a. A personal computer with a Pentium E5700 processor and 2 GB memory under the Microsoft Windows 7 operating system has been used to run the optimization software.

For all benchmarks examined in this study, the FEPSO-AP algorithm parameters are set as the usual constants of standard PSO which are obtained by [31]: $\omega = 0.7298$, $c_1 = 2.05$, and $c_2 = 2.05$. According to the design dimensions of four benchmarks, the population sizes are set as 46, 26, 38, and 88, while the maximum numbers of FEM analyses are set as 4000, 4500, 8000, and 20000, respectively.

Twenty-five independent runs are performed with the best one being selected for each problem.

4.1. Planar 15-Bar Truss. The original geometry of planar 15-bar truss is shown in Figure 2(a). Two nodes (ID: 1 and 5) are totally fixed and the x-coordinates of other two nodes (ID: 4 and 8) are fixed as well. The single loading condition is listed in Table 1.

Table 1: Loading condition acting on the planar 15-bar truss.

Case	Node	F_x (kips)	F_y (kips)
1	8	0	−10.0

All design variables are classified into 23 groups:

sizing variables: A_i, $i = 1, 2, \ldots, 15$;

shape variables: $x_2 = x_6$; $x_3 = x_7$; y_2; y_3; y_4; y_6; y_7; y_8.

Material parameters and design constraints are listed in Table 2.

Figure 2(b) shows the optimum design found by this work, of which two nodes (ID: 4 and 8) are highly overlapped.

Table 2: Material parameters, design constraints, and search range of the planar 15-bar truss optimization problem.

Category	Values
Material Parameters	
Density	0.1 lb/in^3
Modulus of elasticity	1×10^4 ksi
Constraints	
Stress	The allowable elements stress interval: [−25 ksi, 25 ksi]
Search range	
Shape variables	100 in. $\le x_2 \le$ 140 in.; 220 in. $\le x_3 \le$ 260 in.; 100 in. $\le y_2 \le$ 140 in.; 100 in. $\le y_3 \le$ 140 in.; 50 in. $\le y_4 \le$ 90 in.; −20 in. $\le y_6 \le$ 20 in.; −20 in. $\le y_7 \le$ 20 in.; 20 in. $\le y_8 \le$ 60 in.; $x_{1,5} = 0$ in.; $y_1 = 120$ in.; $x_{4,8} = 360$ in.
Sizing variables	S = {0.111, 0.141, 0.174, 0.220, 0.270, 0.287, 0.347, 0.440, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.800, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.300, 10.850, 13.330, 14.290, 17.170, 19.180} in.^{2} $A_i \in S, \ i = 1, 2, \ldots, 15$

Table 3: Comparison of optimized designs found for the planar 15-bar truss.

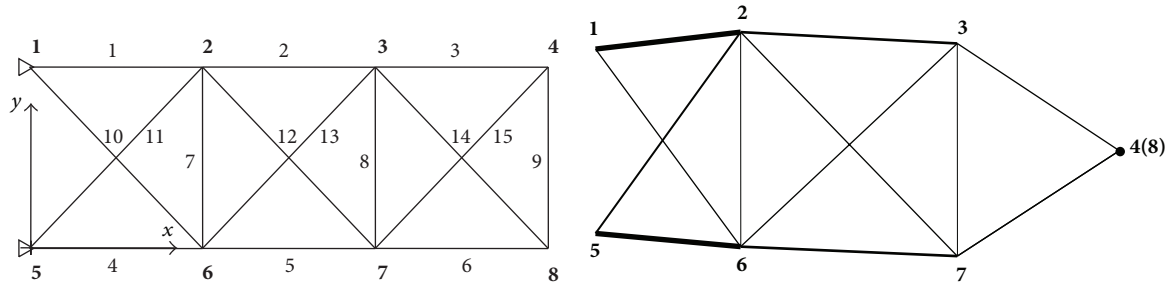
No.	Variable	FA [15]	FM-GA [16]	PSO [17]	CPSO [17]	SCPSO [17]	FEPSO-AP
1	A_1	0.954	1.081	0.954	1.174	0.954	1.081
2	A_2	0.539	0.539	1.081	0.539	0.539	0.539
3	A_3	0.220	0.287	0.270	0.347	0.270	0.270
4	A_4	0.954	0.954	1.081	0.954	0.954	0.954
5	A_5	0.539	0.539	0.539	0.954	0.539	0.539
6	A_6	0.220	0.141	0.287	0.141	0.174	0.111
7	A_7	0.111	0.111	0.141	0.141	0.111	0.111
8	A_8	0.111	0.111	0.111	0.111	0.111	0.111
9	A_9	0.287	0.539	0.347	1.174	0.287	0.347
10	A_{10}	0.440	0.440	0.440	0.141	0.347	0.347
11	A_{11}	0.440	0.539	0.270	0.440	0.347	0.440
12	A_{12}	0.220	0.270	0.111	0.440	0.220	0.287
13	A_{13}	0.220	0.220	0.347	0.141	0.220	0.287
14	A_{14}	0.270	0.141	0.440	0.141	0.174	0.111
15	A_{15}	0.220	0.287	0.220	0.347	0.270	0.270
16	x_2	114.9670	101.5775	106.052	102.287	137.222	100.009
17	x_3	247.0400	227.9112	239.025	240.505	259.909	248.078
18	y_2	125.9190	134.7986	130.356	112.584	123.501	131.524
19	y_3	111.0670	128.2206	114.273	108.043	110.002	123.211
20	y_4	58.2980	54.8630	51.987	57.795	59.936	54.077
21	y_6	−17.5640	−16.4484	1.814	−6.430	−5.180	−9.039
22	y_7	−5.8210	−13.3007	9.183	−1.801	4.219	−14.905
23	y_8	31.4650	54.8572	46.909	57.799	57.883	54.084
Best weight (lb)		75.5473	76.6854	82.2344	77.6153	72.6153	74.1673
Maximum displacement (in.)		24.9993	24.9992	24.9999	24.9909	24.9912	24.9999
Number of structural analyses		8,000	8,000	4,500	4,500	4,500	4,000

Table 3 compares the best design found by this work with those reported in the literature. It can be seen that the results achieved by the proposed algorithm are quite close to the best results reported in the literature.

4.2. Spatial 25-Bar Truss. The original geometry of spatial 25-bar truss is shown in Figure 3(a). Two nodes (ID: 1 and 2) are totally fixed and the z-coordinates of four nodes (ID: 7, 8, 9. and 10) are fixed as well. The loading condition is listed in Table 4.

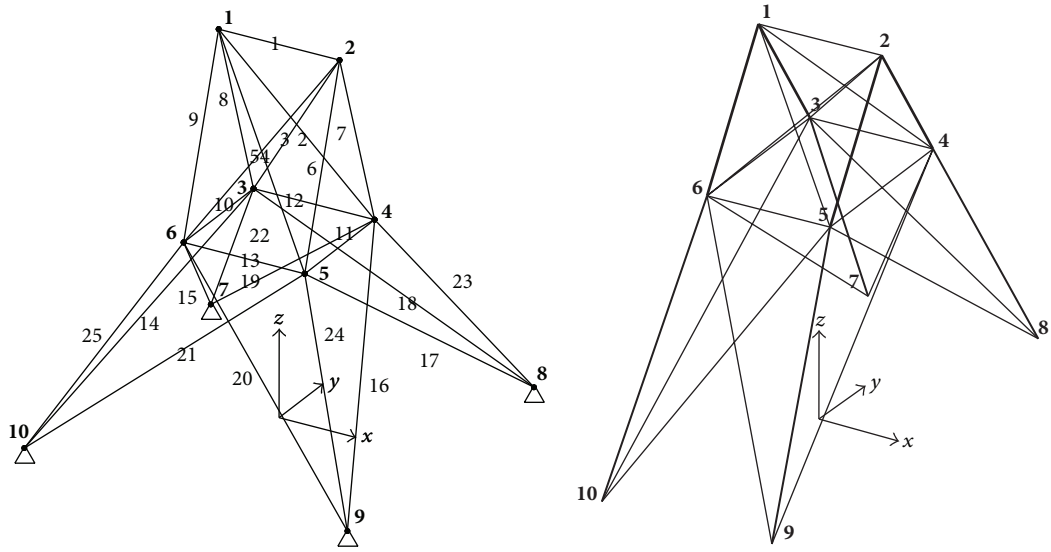
To ensure the structural symmetries, all design variables are classified into 13 groups:

sizing variables: A_1; $A_2 = A_3 = A_4 = A_5$; $A_6 = A_7 = A_8 = A_9$; $A_{10} = A_{11}$; $A_{12} = A_{13}$; $A_{14} = A_{15} = A_{16} = A_{17}$; $A_{18} = A_{19} = A_{20} = A_{21}$; $A_{22} = A_{23} = A_{24} = A_{25}$;



(a) Geometry and element definitions of the planar 15-bar truss (b) Best solution of the planar 15-bar truss

Figure 2: Layout optimization of the planar 15-bar truss.



(a) Geometry and element definitions of the spatial 25-bar truss (b) Best solution of the spatial 25-bar truss

Figure 3: Layout optimization of the spatial 25-bar truss.

Table 4: Loading condition acting on the spatial 25-bar truss.

Case	Node	F_x (kips)	F_y (kips)	F_z (kips)
1	1	1.0	−10.0	−10.0
	2	0.0	−10.0	−10.0
	3	0.5	0.0	0.0
	6	0.6	0.0	0.0

shape variables: $x_4 = x_5 = -x_3 = -x_6$; $x_8 = x_9 = -x_7 = -x_{10}$; $y_3 = y_4 = -y_5 = -y_6$; $y_7 = y_8 = -y_9 = -y_{10}$; $z_3 = z_4 = z_5 = z_6$.

Material parameters and design constraints are listed in Table 5.

Figure 3(b) shows the optimum design found by this work. Table 6 compares the best design found by this work with those reported in the literature. It can be concluded that by using the proposed algorithm it is possible to achieve the best feasible results at a low computational cost.

4.3. Planar 37-Bar Truss. The original geometry of planar 37-bar truss is shown in Figure 4(a). Eleven nodes (ID: 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20) are totally fixed and the x-coordinates of all other nine nodes (ID: 3, 5, 7, 9, 11, 13, 15, 17, and 19) are fixed as well. A nonstructural mass of 10 kg is attached to all free nodes on the lower chord.

To ensure the structural symmetric about the y-axis, all design variables are classified into 19 groups:

sizing variables: $A_1 = A_{27}$; $A_2 = A_{26}$; $A_3 = A_{24}$; $A_4 = A_{25}$; $A_5 = A_{23}$; $A_6 = A_{21}$; $A_7 = A_{22}$; $A_8 = A_{20}$; $A_9 = A_{18}$; $A_{10} = A_{19}$; $A_{11} = A_{17}$; $A_{12} = A_{15}$; $A_{13} = A_{16}$; A_{14};

shape variables: x_3; x_5; x_7; x_9; $y_3 = y_{19}$; $y_5 = y_{17}$; $y_7 = y_{15}$; $y_9 = y_{13}$; y_{11}.

Material parameters and design constraints are listed in Table 7.

Figure 4(b) shows the optimum design explored by this work. Table 8 compares the best design presented by this work with those reported in the literature. It can be seen that

Table 5: Material parameters, design constraints, and search range of the spatial 25-bar truss optimization problem.

Category	Values
Material parameters	
Density	0.1 lb/in^3
Modulus of elasticity	1×10^4 ksi
Constraints	
Stress	The allowable elements stress interval: [−40 ksi, 40 ksi]
Displacement	The allowable nodal displacement interval: [−0.35 in., 0.35 in.]
Search range	
Shape variables	20 in. $\le x_4 \le$ 60 in.; 40 in. $\le x_8 \le$ 80 in.; 40 in. $\le y_4 \le$ 80 in.; 100 in. $\le y_8 \le$ 140 in.; 90 in. $\le z_4 \le$ 130 in.; $-x_1 = x_2 = 37.5$ in.; $y_{1,2} = 0$ in.; $z_{1,2} = 200$ in.
Sizing variables	$S = \{0.1, 0.2, 0.3, \ldots, 3.2, 3.3, 3.4\}$ in.^{2} $A_i \in S$, $i = 1, 2, \ldots, 25$

Table 6: Comparison of optimized designs found for the spatial 25-bar truss.

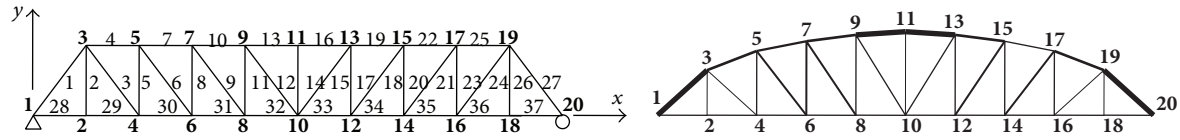
No.	Variable	FA [15]	FM-GA [16]	PSO [17]	CPSO [17]	SCPSO [17]	FEPSO-AP
1	A_1	0.1	0.1	0.1	0.3	0.1	0.1
2	A_2	0.1	0.1	0.1	0.1	0.1	0.1
3	A_6	0.9	1.1	1.1	1.0	1.0	1.0
4	A_{10}	0.1	0.1	0.1	0.1	0.1	0.1
5	A_{12}	0.1	0.1	0.4	0.1	0.1	0.1
6	A_{14}	0.1	0.1	0.1	0.1	0.1	0.1
7	A_{18}	0.1	0.2	0.4	0.2	0.1	0.1
8	A_{22}	1.0	0.8	0.7	0.9	0.9	0.9
9	x_4	37.3200	33.0487	27.6169	33.4976	36.9520	36.8958
10	y_4	55.7400	53.5663	51.6196	62.3735	54.5786	54.1337
11	z_4	126.6200	129.9092	129.9071	114.5945	129.9758	130.0000
12	x_8	50.1400	43.7826	42.5526	40.0531	51.7317	51.9924
13	y_8	136.4000	136.8381	132.7241	133.6695	139.5316	140.0000
	Best weight (lb)	118.83	120.1149	129.2076	123.5403	117.2271	117.3022
	Maximum displacement (in.)	0.3500	0.3500	0.3503	0.3505	0.3518	0.3500
	Maximum stress (ksi)	18.8302	17.1574	16.4391	15.5913	19.9702	20.0182
	Minimum stress (ksi)	−9.4017	−6.4822	−10.9931	−6.4102	−9.4049	−9.4024
	Number of structural analyses	6,000	10,000	4,500	4,500	4,500	4,500

Table 7: Material parameters, design constraints, and search range of the planar 37-bar truss optimization problem.

Category	Values
Material parameters	
Density	7800 kg/m^3
Modulus of elasticity	2.1×10^{11} N/m^2
Constraints	
Natural frequencies	$f_1 \ge 20$ Hz; $f_2 \ge 40$ Hz; $f_3 \ge 60$ Hz;
Search range	
Shape variables	$x_1 = 0$ m; $x_{2,3} = 1$ m; $x_{4,5} = 2$ m; $x_{6,7} = 3$ m; $x_{8,9} = 4$ m; $x_{10,11} = 5$ m; $x_{12,13} = 6$ m; $x_{14,15} = 7$ m; $x_{16,17} = 8$ m; $x_{18,19} = 9$ m; $x_{20} = 10$ m; $0 \text{ m} \le y_{3,5,7,9,11} \le 3$ m;
Sizing variables	1×10^{-4} m$^2 \le A_i \le 10 \times 10^{-4}$ m^2, $i = 1, 2, \ldots, 14$ $A_j = 4 \times 10^{-3}$ m^2, $i = 28, 29, \ldots, 30$

Table 8: Comparison of optimized designs found for the planar 37-bar truss.

No.	Variable	OC [9]	GA [18]	PSO [19]	RO [20]	FEPSO-AP
1	A_1	3.2508	2.8932	2.6797	3.0124	3.4197
2	A_2	1.2364	1.1201	1.1568	1.0623	0.9766
3	A_3	1.0000	1.0000	2.3476	1.0005	0.8313
4	A_4	2.5386	1.8655	1.7182	2.2647	2.8073
5	A_5	1.3714	1.5962	1.2751	1.6339	1.2997
6	A_6	1.3681	1.2642	1.4819	1.6717	1.6483
7	A_7	2.4290	1.8254	4.6850	2.0591	2.4972
8	A_8	1.6522	2.0009	1.1246	1.6607	1.5379
9	A_9	1.8257	1.9526	2.1214	1.4941	1.7590
10	A_{10}	2.3022	1.9705	3.8600	2.4737	2.7069
11	A_{11}	1.3103	1.8294	2.9817	1.5260	1.3046
12	A_{12}	1.4067	1.2358	1.2021	1.4823	1.4004
13	A_{13}	2.1896	1.4049	1.2563	2.4148	3.0476
14	A_{14}	1.0000	1.0000	3.3276	1.0034	0.5947
15	y_3	1.2086	1.1998	0.9637	1.0010	0.8756
16	y_5	1.5788	1.6553	1.3978	1.3909	1.2546
17	y_7	1.6719	1.9652	1.5929	1.5893	1.4446
18	y_9	1.7703	2.0737	1.8812	1.7507	1.5889
19	y_{11}	1.8502	2.3050	2.0856	1.8336	1.6480
Natural frequencies (Hz)	f_1	20.0850	20.0013	20.0001	20.056	20.020
	f_2	42.0743	40.0305	40.0003	40.035	40.022
	f_3	62.9383	60.0000	60.0001	60.030	60.233
	f_4	74.4539	73.0444	73.0440	74.387	72.137
	f_5	90.0576	89.8244	89.8240	85.929	84.065
Best weight (kg)		366.50	368.84	377.20	364.04	362.8812
Number of structural analyses		—	—	20,000	32,000	8,000



(a) Geometry and element definitions of the planar 37-bar truss

(b) Best solution of the planar 37-bar truss

Figure 4: Layout optimization of the planar 37-bar truss.

Table 9: Loading conditions acting on the planar 47-bar truss.

Case	Node	F_x (kips)	F_y (kips)
1	17 and 22	6.0	−14.0
2	17	6.0	−14.0
3	22	6.0	−14.0

the results achieved by the proposed algorithm are even better than the best results reported in the literature.

4.4. Planar 47-Bar Truss. The original geometry of planar 47-bar truss is shown in Figure 5(a). Four nodes (ID: 15, 16, 17, and 22) are totally fixed and the y-coordinates of other two nodes (ID: 1 and 2) are fixed as well. The loading conditions are listed in Table 9.

To ensure the structural symmetric about the y-axis, all design variables are classified into 44 groups:

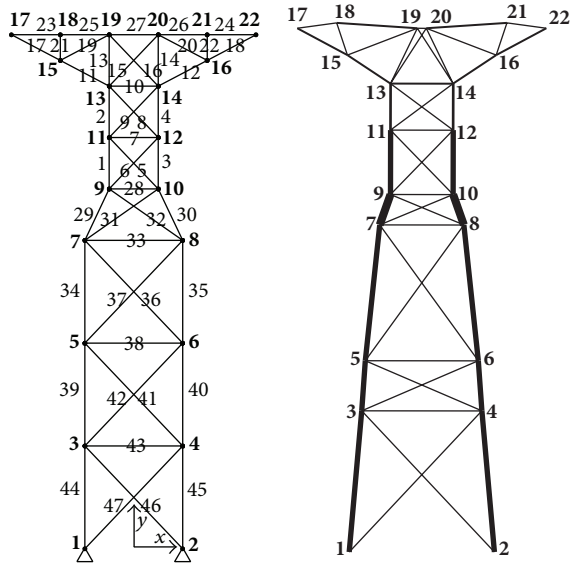
sizing variables: $A_1 = A_3$; $A_2 = A_4$; $A_5 = A_6$; A_7; $A_8 = A_9$; A_{10}; $A_{11} = A_{12}$; $A_{13} = A_{14}$; $A_{15} = A_{16}$; $A_{17} = A_{18}$; $A_{19} = A_{20}$; $A_{21} = A_{22}$; $A_{23} = A_{24}$; $A_{25} = A_{26}$; A_{27}; A_{28}; $A_{29} = A_{30}$; $A_{31} = A_{32}$; A_{33}; $A_{34} = A_{35}$; $A_{36} = A_{37}$; A_{38}; $A_{39} = A_{40}$; $A_{41} = A_{42}$; A_{43}; $A_{44} = A_{45}$; $A_{46} = A_{47}$;

shape variables: $x_1 = -x_2$; $x_3 = -x_4$; $y_3 = y_4$; $x_5 = -x_6$; $y_5 = y_6$; $x_7 = -x_8$; $y_7 = y_8$; $x_9 = -x_{10}$; $y_9 = y_{10}$; $x_{11} = -x_{12}$; $y_{11} = y_{12}$; $x_{13} = -x_{14}$; $y_{13} = y_{14}$; $x_{19} = -x_{20}$; $y_{19} = y_{20}$; $x_{18} = -x_{21}$; $y_{18} = y_{21}$.

Material parameters and design constraints are listed in Table 10.

Table 10: Material parameters, design constraints, and search range of the planar 47-bar truss optimization problem.

Category	Values
Material parameters	
Density	$0.3\,\text{lb/in}^3$
Modulus of elasticity	3×10^4 ksi
Constraints	
Stress	The allowable elements stress interval: [−15 ksi, 20 ksi]
Local buckling	$\left\|(\sigma_c)_i\right\| \le \beta E A_i / l_i^2, \quad i = 1, 2, \ldots, 47, \quad \beta = 3.96$
Search range	
Shape variables	$0\text{ in.} \le x_i \le 120\text{ in.}, (i = 2, 4, 6, 8); \ -30\text{ in.} \le x_j \le 90\text{ in.} \ (j = 10, 12, 14, 20); 30\text{ in.} \le x_{21} \le 150\text{ in.}$; $60\text{ in.} \le y_4 \le 180\text{ in.}; 180\text{ in.} \le y_6 \le 300\text{ in.}; 300\text{ in.} \le y_8 \le 420\text{ in.}; 360\text{ in.} \le y_{10} \le 480\text{ in.}; 420\text{ in.} \le y_{12} \le 540\text{ in.}; 480\text{ in.} \le y_{14} \le 600\text{ in.}; 540\text{ in.} \le y_{20} \le 660\text{ in.}; 540\text{ in.} \le y_{21} \le 660\text{ in.}$
Sizing variables	$S = \{0.1, 0.2, 0.3, \ldots, 4.8, 4.9, 5.0\}\,\text{in.}^2 \quad A_i \in S, \quad i = 1, 2, \ldots, 47$



(a) Geometry and element definitions of the planar 47-bar truss

(b) Best solution of the planar 47-bar truss

Figure 5: Layout optimization of the planar 47-bar truss.

Figure 5(b) shows the optimum design identified by this work. Table 11 compares the best design of this work with those reported in the literature. It can be seen that by using the proposed algorithm it is possible to achieve better results at a lower computational cost.

5. Conclusion

In this work, a new hybrid PSO algorithm is proposed to solve a quite challenging task in truss optimization area: truss layout optimization with multiple constraints.

Two computational techniques are adopted to further enhance the performance of PSO algorithm. In the first fitness estimation strategy, the evaluation of particles is partly substituted by the estimation of similar particles, with the purpose to reduce the computational cost of real world optimization problem. In the second adaptive penalty function approach, the iteration information is merged into the penalty function to find a good balance between the exploration and exploitation of the constrained design domain. The resulted algorithm is termed as FEPSO-AP.

Four benchmark truss layout optimization problems, subject to nodal displacement constraints, element stress constraints, natural frequency constraints, and local buckling constraints, are used to verify the performance of FEPSO-AP. Numerical results demonstrate that three out of four benchmarks, to which the FEPSO-AP based optimization is applied, delivered the best feasible designs to the author's knowledge. Moreover, the convergence rate of the FEPSO-AP algorithm is quite competitive comparing to other state-of-the-art hybrid algorithms published in the former literatures.

Conflict of Interests

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

Table 11: Comparison of optimized designs found for the planar 47-bar truss.

No.	Variable	GA [21]	FSD-ES [22]	PSO [17]	CPSO [17]	SCPSO [17]	FEPSO-AP
1	A_3	2.50	2.70	2.80	2.60	2.50	3.20
2	A_4	2.20	2.50	2.70	2.50	2.50	2.80
3	A_5	0.70	0.70	0.80	0.70	0.80	0.70
4	A_7	0.10	0.10	1.10	0.30	0.10	0.10
5	A_8	1.30	0.90	0.80	1.20	0.70	0.60
6	A_{10}	1.30	1.10	1.30	1.10	1.40	1.60
7	A_{12}	1.80	1.80	1.80	1.60	1.70	1.90
8	A_{14}	0.50	0.70	0.90	0.80	0.80	0.90
9	A_{15}	0.80	0.90	1.20	1.10	0.90	1.00
10	A_{18}	1.20	1.30	1.40	1.30	1.30	2.00
11	A_{20}	0.40	0.30	0.30	0.30	0.30	0.10
12	A_{22}	1.20	1.10	1.40	0.80	0.90	0.40
13	A_{24}	0.90	1.00	1.10	1.00	1.00	1.40
14	A_{26}	1.00	0.90	1.20	1.00	1.10	1.50
15	A_{27}	3.60	0.80	1.60	0.90	5.00	1.20
16	A_{28}	0.10	0.10	1.00	0.10	0.10	0.70
17	A_{30}	2.40	2.70	2.80	2.70	2.50	5.00
18	A_{31}	1.10	0.80	0.80	0.90	1.00	1.00
19	A_{33}	0.10	0.10	0.10	0.10	0.10	0.10
20	A_{35}	2.70	3.00	3.00	3.00	2.80	3.00
21	A_{36}	0.80	0.90	0.90	1.00	0.90	0.50
22	A_{38}	0.10	0.00	0.10	0.20	0.10	0.10
23	A_{40}	2.80	3.20	3.30	3.30	3.00	3.30
24	A_{41}	1.30	1.00	0.90	0.90	1.00	0.30
25	A_{43}	0.20	0.10	0.10	0.10	0.10	0.10
26	A_{45}	3.00	3.30	3.30	3.30	3.20	3.50
27	A_{46}	1.20	1.10	1.20	1.10	1.20	0.40
28	x_2	114.0000	100.9724	98.8628	99.3630	101.3393	87.7275
29	x_4	97.0000	80.4772	78.6595	83.4439	85.9111	72.4352
30	y_4	125.0000	136.8699	146.7331	126.3845	135.9645	162.6451
31	x_6	76.0000	64.3908	66.5231	69.5148	74.7969	67.2113
32	y_6	261.0000	247.0491	239.0901	218.2013	237.7447	218.2041
33	x_8	69.0000	55.2589	55.6936	58.0004	64.3115	50.6507
34	y_8	316.0000	338.4534	327.7882	322.2272	321.3416	375.4549
35	x_{10}	56.0000	48.7333	48.8641	51.4015	53.3345	36.6525
36	y_{10}	414.0000	409.7380	398.6775	401.5626	414.3025	408.7230
37	x_{12}	50.0000	43.4742	43.1400	46.8605	46.0277	36.9960
38	y_{12}	463.0000	472.1479	464.7831	458.3021	489.9216	483.4295
39	x_{14}	54.0000	44.8349	37.8993	46.8885	41.8353	37.9558
40	y_{14}	524.0000	512.1901	511.0450	527.8575	522.4161	535.7644
41	x_{20}	1.0000	3.8414	18.2341	16.2354	1.0005	4.6875
42	y_{20}	587.0000	591.1449	594.0710	610.8496	598.3905	599.7416
43	x_{21}	99.0000	84.5040	90.9369	98.3239	97.8696	101.4535
44	y_{21}	631.0000	630.3472	621.3943	624.9580	624.0552	605.4302
	Best weight (lb)	1925.7897	1842.6609	1975.8393	1908.8301	1864.0985	1799.7037
	Maximum stress (ksi)	19.9528	20.0000	19.0636	19.3351	19.4735	19.9808
	Minimum stress (ksi)	−14.9973	−15.0000	−14.9999	−14.9986	−15.0000	−14.9986
	Number of buckling elements	0	0	0	0	0	0
	Number of structural analyses	100,000	55,802	25,000	25,000	25,000	20,000

References

[1] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and Multidisciplinary Optimization*, vol. 41, no. 2, pp. 219–241, 2010.

[2] A. Kaveh and A. Zolghadr, "Democratic PSO for truss layout and size optimization with frequency constraints," *Computers and Structures*, vol. 130, pp. 10–21, 2014.

[3] A. Xiao, B. Wang, and Y. Jin, "Evolutionary truss layout optimization using the vectorized structure approach," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, pp. 2879–2886, June 2013.

[4] M. W. Dobbs and L. P. Felton, "Optimization of truss geometry," *Journal of the Structural Division*, vol. 95, pp. 2105–2118, 1969.

[5] G. N. Vanderplaats and F. Moses, "Structural optimization by methods of feasible directions," *Computers and Structures*, vol. 3, no. 4, pp. 739–755, 1973.

[6] M. Zhou, "Geometrical optimization of trusses by a two-level approximation concept," *Structural Optimization*, vol. 1, no. 4, pp. 235–240, 1989.

[7] L. Gil and A. Andreu, "Shape and cross-section optimization of a truss structure," *Computers & Structures*, vol. 79, no. 7, pp. 681–689, 2001.

[8] K. Deb and S. Gulati, "Design of truss-structures for minimum weight using genetic algorithms," *Finite Elements in Analysis and Design*, vol. 37, no. 5, pp. 447–465, 2001.

[9] D. Wang, W. H. Zhang, and J. S. Jiang, "Truss optimization on shape and sizing with frequency constraints," *AIAA Journal*, vol. 42, no. 3, pp. 622–630, 2004.

[10] P. C. Fourie and A. A. Groenwold, "The particle swarm optimization algorithm in size and shape optimization," *Structural and Multidisciplinary Optimization*, vol. 23, no. 4, pp. 259–267, 2002.

[11] O. Hasançebi and F. Erbatur, "Layout optimisation of trusses using simulated annealing," *Advances in Engineering Software*, vol. 33, no. 7-10, pp. 681–696, 2002.

[12] H. Kawamura, H. Ohmori, and N. Kito, "Truss topology optimization by a modified genetic algorithm," *Structural and Multidisciplinary Optimization*, vol. 23, no. 6, pp. 467–472, 2002.

[13] A. Kaveh and S. Talatahari, "Optimal design of skeletal structures via the charged system search algorithm," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 893–911, 2010.

[14] M. Sonmez, "Artificial Bee Colony algorithm for optimization of truss structures," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2406–2418, 2011.

[15] L. F. F. Miguel, R. H. Lopez, and L. F. F. Miguel, "Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm," *Advances in Engineering Software*, vol. 56, pp. 23–37, 2013.

[16] H. Rahami, A. Kaveh, and Y. Gholipour, "Sizing, geometry and topology optimization of trusses via force method and genetic algorithm," *Engineering Structures*, vol. 30, no. 9, pp. 2360–2369, 2008.

[17] S. Gholizadeh, "Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization," *Computers and Structures*, vol. 125, pp. 86–99, 2013.

[18] W. Lingyun, Z. Mei, W. Guangming, and M. Guang, "Truss optimization on shape and sizing with frequency constraints based on genetic algorithm," *Computational Mechanics*, vol. 35, no. 5, pp. 361–368, 2005.

[19] H. M. Gomes, "Truss optimization with dynamic constraints using a particle swarm algorithm," *Expert Systems with Applications*, vol. 38, no. 1, pp. 957–968, 2011.

[20] A. Kaveh and M. Khayatazad, "Ray optimization for size and shape optimization of truss structures," *Computers and Structures*, vol. 117, pp. 82–94, 2013.

[21] O. Hasançebi and F. Erbatur, "Layout optimization of trusses using improved GA methodologies," *Acta Mechanica*, vol. 146, no. 1-2, pp. 87–107, 2001.

[22] A. Ahrari, A. A. Atai, and K. Deb, "Simultaneous topology, shape and size optimization of truss structures by fully stressed design based on evolution strategy," *Engineering Optimization*, 2013.

[23] A. Kaveh and A. Zolghadr, "Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability," *Computers & Structures*, vol. 102-103, pp. 14–27, 2012.

[24] W. Zuo, J. Bai, and B. Li, "A hybrid OC-GA approach for fast and global truss optimization with frequency constraints," *Applied Soft Computing Journal*, vol. 14, pp. 528–535, 2014.

[25] A. Kaveh and S. M. Javadi, "Shape and size optimization of trusses with multiple frequency constraints using harmony search and ray optimizer for enhancing the particle swarm optimization algorithm," *Acta Mechanica*, vol. 225, no. 6, pp. 1595–1605, 2014.

[26] W. Liu and J. Ye, "Collapse optimization for domes under earthquake using a genetic simulated annealing algorithm," *Journal of Constructional Steel Research*, vol. 97, pp. 59–68, 2014.

[27] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, vol. 1, pp. 39–43, October 1995.

[28] C. Sun, J. Zeng, J. Pan, S. Xue, and Y. Jin, "A new fitness estimation strategy for particle swarm optimization," *Information Sciences*, vol. 221, pp. 355–370, 2013.

[29] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.

[30] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.

[31] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 534162, 8 pages
http://dx.doi.org/10.1155/2014/534162

Research Article

A Guaranteed Global Convergence Social Cognitive Optimizer

Jia-ze Sun,[1,2] Shu-yan Wang,[1] and Hao Chen[1]

[1] *School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*
[2] *School of Information Science and Technology, Northwest University, Xi'an 710127, China*

Correspondence should be addressed to Jia-ze Sun; sunjiaze@126.com

Received 11 July 2014; Accepted 10 September 2014; Published 29 September 2014

Academic Editor: Shifei Ding

From the analysis of the traditional social cognitive optimization (SCO) in theory, we see that traditional SCO is not guaranteed to converge to the global optimization solution with probability one. So an improved social cognitive optimizer is proposed, which is guaranteed to converge to the global optimization solution. The global convergence of the improved SCO algorithm is guaranteed by the strategy of periodic restart in use under the conditions of participating in comparison, which helps to avoid the premature convergence. Then we give the convergence proof for the improved SCO based on Solis and Wets' research results. Finally, simulation results on a set of benchmark problems show that the proposed algorithm has higher optimization efficiency, better global performance, and better stable optimization outcomes than the traditional SCO for nonlinear programming problems (NLPs).

1. Introduction

Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, natural or artificial in artificial intelligence field. Swarms often are large groups of small insects in which each member performs a simple role, but the action produces complex behavior as a whole. The emergence of such complex behavior extends beyond swarms. Similar complex social structures also occur in higher-order animals and insects that do not swarm: colonies of ants, flocks of birds, packs of wolves, or colonies of bees, and so on.

Human has higher adaptability and social intelligence than insect swarm, and human intelligence derives from the interactions of individuals, including interacting with the environment, in a social world from the study of social cognitive theory. Xie et al. [1] present social cognitive optimization (SCO) algorithm which is a novel heuristic swarm intelligence optimization algorithms. SCO algorithm has a probabilistic iterative procedure of lots of learning agents. A learning agent obtains vicarious capability by tournament selection and shares information by the knowledge library with symbolizing capability. Because SCO algorithm fully makes use of the interactions and share of the entire social swarm, it greatly improves the convergence speed and accuracy of the swarm intelligence algorithm and makes it better than many other well-used intelligent optimization methods, such as PSO and ACO, in many applications. Such applications include nonlinear programming problems (NLPs), nonlinear complementarity's problem (NCP) [2], fractional programs [3], nonlinear system of equation [4], engineering design problems [5], and Web service composition selection [6].

Many researchers improved the traditional SCO. Wang et al. [7] improved SCO algorithm through joining self-organizing migrating algorithm (SOMA) migration in the process of SCO and adding two parameters in SCO algorithm to solve the SAT problem, which showed the improved SCO may obtain the quick convergence rate in the optimization early and only small effect on the last result. Ma et al. [8] brought in the chaos and Kent mapping function to modify and optimize the conditions of neighborhood search and got more reasonable knowledge points which were distributed more uniformly for solving the nonlinear constraint problems. Sun et al. [9] presented a hybrid social cognitive optimization algorithm based on elitist strategy and chaotic optimization is proposed to solve constrained nonlinear programming problems, which partitions learning agents into three groups in proportion: elite learning agents,

chaotic learning agents, and common learning agents. Zhizhong et al. [10] improve the social cognitive optimization, put the improved SCO algorithm into the framework of culture algorithm, constructed a novel algorithm, culture social cognitive optimization (C-SCO), and used C-SCO to solve the QoS-aware cloud service composition problem. Sun et al. [11] present a social cognitive optimization algorithm (SCO) to generate optimal evidence weight values for the Dempster-Shafer (D-S) evidence model based on historical training data.

These improved algorithms, called hybrid optimization algorithms, are mainly based on empirical analysis of the experiment while the global convergence analysis of hybrid algorithm has not been studied in theory. Because the SCO is originated from simulation social cognitive progress and involves sophisticated stochastic behavior, it is hard to perform theoretical analysis, resulting in the lack of solid theoretical foundation. Particularly, the performance of SCO as optimization techniques requires theoretical support. The lack of theoretical foundation injures the further development of SCO and blocks the application of SCO in problems where serious algorithms are required.

In this paper, from the analysis of the traditional social cognitive optimization (SCO) in theory, we see that traditional SCO is not guaranteed to converge to the global optimization solution with probability one. So a novel social cognitive optimizer is called stochastic SCO that is guaranteed to converge to the global optimization solution. The global convergence of the improved SCO algorithm is guaranteed by the strategy of periodic restart in use under the conditions of participating in comparison, which helps to avoid the premature convergence. Then we give the convergence proof for the stochastic SCO based on Solis and Wets' research results [12]. Finally, simulation results on a set of benchmark problems show that the proposed algorithm has higher optimization efficiency, better global performance, and better stable optimization outcomes than the traditional SCO for NLPs.

The remainder of this paper is organized as follows. In Section 2, we survey the traditional SCO and analyse global convergence of traditional SCO algorithm. In Section 3, our improvement to traditional SCO is described concretely and the proof of the global convergence of the proposed algorithms is presented. In Section 4 the typical experiments are employed to evaluate the performance of the improved SCO and the conclusions are showed in Section 5, and finally the last section presents the acknowledgment and the appendix.

2. Convergence Analysis of Traditional SCO Algorithm

2.1. Social Cognitive Optimization Algorithm (SCO). Social cognitive theory (SCT) agrees that people learn by observing others, with the environment, behavior, and cognition all as the chief reciprocal factors in influencing development. Human learning possesses the abilities to symbolize, learn from others, plan alternative strategies, regulate one's own behavior, and engage in self-reflection. So human has higher adaptability and social intelligence than insect swarm. By introducing human social intelligence based on SCT to artificial system, Xie et al. [1] proposed social cognitive optimization (SCO) algorithm in 2002. In SCO optimization procedure, a knowledge library with symbolizing capability consists of a number of knowledge points which are denoted by the location x in search space S and its fitness values; learning agents, on behalf of human individuals, in possession of a knowledge point in the knowledge library, act observational learning via the neighborhood local searching by observing the selected model from tournament selection. The neighborhood local searching for x_2 referring to x_1 is finding a new point x', which is for d dimension

$$x'_d = U\left(x_{1,d}, x_{m,d}\right), \tag{1}$$

where $U(a,b)$ is a uniform distribution which is usually generated by linear congruential method within $[a,b]$, $x_{m,d} = 2 * x_{2,d} - x_{1,d}$. SCO algorithm basic steps are clearly described in [1].

2.2. Basic Conception and Theory for Global Convergence Theorem. The general global optimization problem (P) used here is defined as

$$\min_{x \in S} f(x), \tag{2}$$

where x is a vector of n decision variables, S is an n-dimensional feasible region and is assumed to be nonempty, a subset of R^n, and $f(x)$ is a real-valued function defined over S from R^n to R. The goal is to find a value for x contained in S that minimizes f. Notice that the feasible region may include both continuous and discrete variables. Denote the global optimal solution to (P) by (x^{**}, y^{**}), where

$$\begin{aligned} x^{**} &= \arg\min_{x \in S} f(x), \\ y^{**} &= f\left(x^{**}\right) = \min_{x \in S} f(x). \end{aligned} \tag{3}$$

Solis and Wets [12] provide a convergence proof, in probability, to the global minimum for general step size algorithms with conditions on the method of generating the step length and direction.

Conceptual algorithm [12] is as follows.

Step 0. Find x^0 in S and set $k = 0$.

Step 1. Generate ξ^k from the sample space (R^n, B, μ_k).

Step 2. Set $x^{k+1} = D(x^k, \xi^k)$, choose μ_{k+1}, set $k = k + 1$, and return to Step 1.

(R^n, B, μ_k) is probability space on iteration k. The μ_k are probability measures corresponding to distribution functions defined on R^n as conditional probability measures. The B is Borel subsets of R^n. B is the σ-algebra of subset of R^n. The map D with domain $S \times R^n$ and S satisfies the following condition.

Hypothesis 1 (H1). Consider $f(D(x,\xi)) \le f(x)$; if $\xi \in S$, then $f(D(x,\xi)) \le f(\xi)$.

Clearly, the monotone sequence $\{f(x^k)\}_{k=1}^{\infty}$ converges to the infimum of f on S. In order to avoid excluding some pathological situations, we replace the search for the infimum by that for α, the essential infimum of f on S, defined as follows:

$$\alpha = \inf\{x \mid v(z \in S \mid f(z) < x) > 0\}, \quad (4)$$

where v is a nonnegative measure defined on the (Borel) subsets B of R^n with $v(S) > 0$. Typically $v(A)$ is simply the n-dimensional volume of the set A; more generally v is the Lebesgue measure.

Hypothesis 2 (H2). For any subset A of S with $v(A) > 0$ we have that

$$\prod_{k=0}^{\infty}(1 - \mu_k(A)) = 0. \quad (5)$$

It means that given any subset A of S with positive Lebesgue measure the probability of repeatedly missing the set A, when generating the random samples ξ^k, must be zero.

This requires that the sampling strategy determined by the choice of the ξ^k cannot rely exclusively on distribution functions concentrated on proper subsets of S of lower dimension (such as discrete distributions), or that the sampling strategy consistently ignore a part of S with positive "volume" (with respect to v).

Theorem 1 (convergence theorem (global search)). *Suppose that f is a measurable function. S is a measurable subset of R^n and (Hypotheses 1 and 2) is satisfied. Let $\{x^k\}_{k=1}^{+\infty}$ be a sequence generated by the algorithm. Then*

$$\lim_{k\to\infty} P\left[x^k \in R_\varepsilon\right] = 1, \quad (6)$$

where $P[x^k \in R_\varepsilon]$ is the probability that, at step k, the point x^k generated by the algorithm is in R_ε. R_ε is the optimality region.

2.3. Convergence Analysis of Traditional SCO Algorithm. Although traditional SCO may outperform other evolutionary algorithms in the early iterations, its performance may not be competitive as the number of generations is increased. Traditional SCO algorithm is not guaranteed to converge to global optimal solution. If the optimization algorithm satisfies the Hypotheses 1 and 2, general convergence proofs are given.

The traditional SCO algorithm saves the best solution in the knowledge, so it obviously satisfies Hypothesis 1. But, global optimization point GB_P of the agents in SCO algorithm will not be set at a random solution in the search space in the end of every generation. It is obvious that the algorithm does not satisfy Hypothesis 2 and is not an optimization algorithm with global search convergence properties according to Theorem 1. So the traditional SCO algorithm is not guaranteed to converge to global optimal solution with probability one.

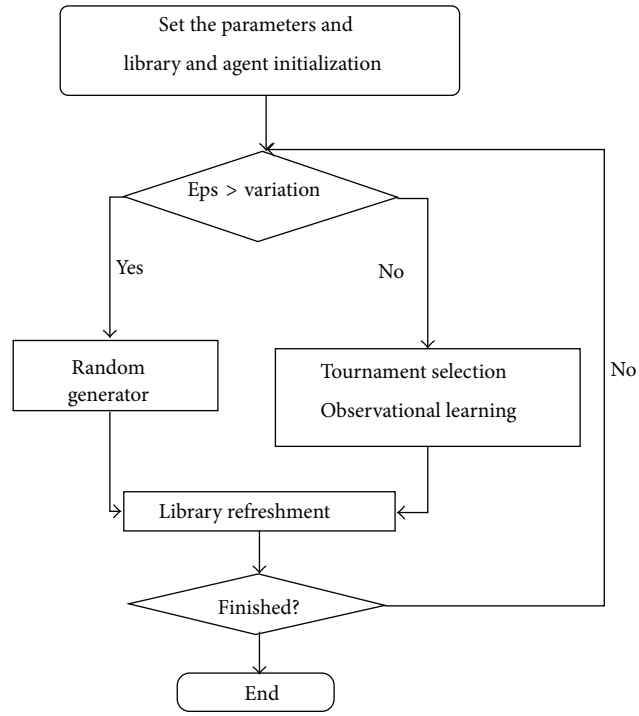


Figure 1: Flowchart of SSCO.

3. Method

3.1. Overview. Swarm intelligence optimization algorithms are also called metaheuristic algorithms because they provide a high-level framework which can be adapted to solve optimization problems. So, when swarm intelligence is used to solving a specific problem, it must be modified to fit the problem.

The SCO's iterative procedure is rooted in human intelligence with the social cognitive theory. When the learning agent falls into the local optimal, learning agents learn without observation, but with full randomicity by stochastic search, which helps to increase the global ergodicity of the knowledge library and to avoid premature convergence.

In this paper, a novel stochastic social cognitive optimization (SSCO) algorithm based on periodic partial reinitialization is proposed to solve NLPs to improve the global convergence speed of social cognitive optimization (SCO) algorithm. Simulation results show that the performance of SSCO is evidently better than traditional SCO for NLPs.

3.2. Stochastic Social Cognitive Optimization (SSCO) Algorithm. In our study, we incorporate the periodic partial reinitialization of the population into the SCO to enhance the overall performance of the algorithm. Figure 1 shows the flowchart of SSCO. The SSCO is described as follows.

Step 1 (initialization). (a) Set the parameters of SSCO: N_p, N_a, T, τ_W, τ_B, T_R, ε_R, where N_p denotes the number of knowledge points in knowledge library; N_a denotes the number of learning agents. T denotes the times of maximum learning cycle; τ_B denotes the tournament width; and T_R

Table 1: Summary of eight test cases.

Func.	n	Type of func.	p	LI	NE	NI	a
G_1	13	Quadratic	0.0111%	9	0	0	6
G_2	20	Nonlinear	99.8474%	0	0	2	1
G_3	5	Quadratic	52.1230%	0	0	6	2
G_4	2	Cubic	0.0066%	0	0	2	2
G_5	10	Quadratic	0.0003%	3	0	5	6
G_6	2	Nonlinear	0.8560%	0	0	2	0
G_7	7	Polynomial	0.5121%	0	0	4	2
G_8	8	Linear	0.0010%	3	0	3	6

denotes the maximum iterate number of reinitialization; ε_R denotes the maximum variation of fitness in T_R generation.

(b) Randomly create N_p knowledge points in knowledge library (KL), and then evaluate their fitness values basis on objective function, and save the global optimization point GB_P: the best point with the best fitness.

(c) Assign a knowledge point in KL to a learning agent randomly, but not repeatedly.

Step 2. For each learning agent, SCA learning cycle is as follows.

(a) If the variation of fitness of the global optimization point SCA in the previous T_R generation is less than ε_R, a new point TS_O is randomly generated; if the new point TS_O is better than GB_P, TS_O is assigned to GB_P.

(b) Otherwise, we have the following.

(1) Tournament selection: select a best knowledge point TS_P from arbitrary τ_B knowledge points in KL not repeatedly with SCA itself.

(2) Observational learning: compare the fitness of TS_P with that of SCA. The neighborhood local searching for the better referring to the worse is finding a new point TS_O according to (1); if the new point TS_O is better than GB_P, TS_O is assigned to GB_P.

(c) Library refreshment: remove the worst knowledge point TS_w in KL, and add the new point into KL.

Step 3. Repeat Step 2 until a stop condition (e.g., maximum number of iterations or a satisfactory fitness value). The total evaluation times are $T_e = N_p + N_a * T$.

3.3. Convergence Analysis of SSCO Algorithm. Traditional SCO algorithm is not guaranteed to converge to global optimal solution with probability one. According to Theorem 1, the proof presented here casts the SSCO into the framework of a global stochastic search algorithm, thus allowing the use of Theorem 1 to prove convergence. It remains to show that the SSCO satisfies both (H1) and (H2).

Let $\{\mathrm{GB}_{P,t}\}$ be a sequence generated by the SSCO algorithm, where $\mathrm{GB}_{P,t}$ is the current best position of the swarm at time t.

Define function D:

$$D\left(\mathrm{GB}_{P,t}, x_{i,t}\right) = \begin{cases} \mathrm{GB}_{P,t}, & f\left(g\left(x_{i,t}\right)\right) \geq f\left(\mathrm{GB}_{P,t}\right) \\ g\left(x_{i,t}\right), & f\left(g\left(x_{i,t}\right)\right) \geq f\left(\mathrm{GB}_{P,t}\right), \end{cases}$$

$$f\left(D\left(x,\xi\right)\right) \leq f\left(x\right), \quad \text{if } \xi \in S, \text{ then } f\left(D\left(x,\xi\right)\right) \leq f\left(\xi\right). \tag{7}$$

The definition of D above clearly complies with hypothesis H1, since the sequence $x_{i,t}$ is monotonic by definition because of always saving the best solution in the knowledge.

If the SSCO algorithm satisfies hypothesis H2, the union of the sample spaces of the agents must cover S, so that $S \subseteq \bigcup_{i=1}^{s} M_{i,k}$ at time step t, where $M_{i,k}$ denotes the support of the sample space of agent i. Because every learning agent has a chance to get a random solution in the search space in every generation when the iteration is static in a certain precision, $M_{i,k} = S$, $\bigcup_{i=1}^{s} M_{i,k} = S$. Define the Borel subset A of S, and $A = M_{i,k}$; then $\nu(A) > 0$, $\mu_k(A) = \sum_{i=1}^{s} \mu_{k,t}(A) = 1$. Thus by hypothesis H2 satisfied by Theorem 1, SSCO can be convergent to global best solution with probability one.

4. Numerical Experiment

4.1. Experimental Settings. Nonlinear programming problems (NLPs) always are nonconvex, highly nonlinear, nondifferentiable, and discontinuous, which is a constrained global optimization problem; the traditional deterministic algorithms for solving the NLPs are very difficult. Furthermore, the constrained global optimization is NP-hard [13], which does not admit efficient deterministic solutions in practice.

In our experiments, the test cases of eight benchmark NLPs from literature [1] and literature [13] in Table 1, except four problems because of lack of result in the referenced literatures, will be applied to show the way in which the proposed algorithm works. The eight benchmark problems almost include all the kinds of the constraints (linear inequalities LI, nonlinear inequalities NI, and nonlinear equalities NE). With respect to constraint handling of the NLPs in the proposed algorithms, there are two effective methods [14], basic constraint handing (BCH) rule for common inequalities constraint and adaptive constraints relaxing (ACR) rule for equalities constraint. The rules are described in detail in [15].

Table 1 lists the parameters of each test case: number of variables, type of the function, relative size of the feasible region in the search space given by the ratio p, the number of

Table 2: Summary of results of the SSCO on eight test cases.

NLP.	Type	Opt.	GA			PSO	SCO	SSCO		
			Worst	Best	Avg.	Avg.	Avg.	Worst	Best	Avg.
G_1	min	−15	−14.6154	−14.7864	−14.7082	−14.7951	−14.8891	−14.8767	−15.0000	−14.9953
G_2	max	0.803553	0.79119	0.79953	0.79671	0.79592	0.75475	0.79124	0.80109	0.79638
G_3	min	−30665.5	−30645.9	−30645.5	−30645.3	−30655.429	−30665.539	−30665.539	−30665.539	−30665.539
G_4	min	−6961.814	−5473.9	−6952.1	−6342.6	−6781.913	−6961.812	−6961.805	−6961.814	−6961.813
G_5	min	24.306	25.069	24.620	24.826	24.713	24.742	24.681	24.306	24.359
G_6	max	0.095825	0.0291438	0.0958250	0.0891568	0.091573	0.095158	0.095231	0.095825	0.095811
G_7	min	680.63	683.18	680.91	681.16	690.642	680.699	680.811	680.631	680.671
G_8	min	7049.33	9659.3	7147.9	8163.6	7531.9	7407.7	7296.35	7049.41	7124.72

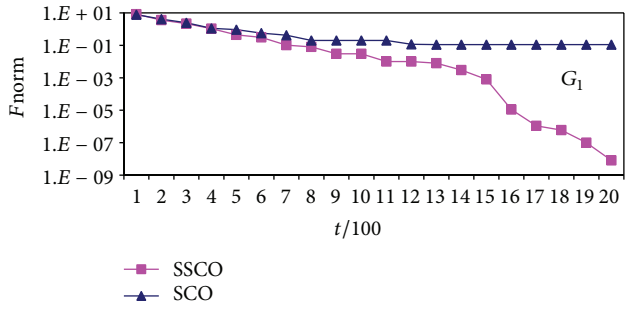


Figure 2: Fnorm versus $t/100$ for G_1 by SSCO and SCO.

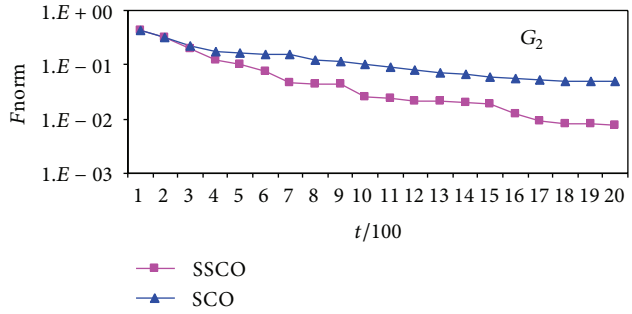


Figure 3: Fnorm versus $t/100$ for G_2 by SSCO and SCO.

constraints of each category (LI, NE, and NI), and the number a of active constraints at the optimum.

4.2. Result and Discussion. To evaluate the performance of the SSCO, we compare SSCO with standard gray-coded GA and traditional social cognitive optimization. The experiments setting for SSCO is the same as that of SCO from [1]. Let $T_R = 5$, $N_p = 98$, $N_a = 14$, $T = 2000$, (for G_6, $T = 200$). Each problem is executed 50 times. We calculate the best solution, worst solution, and mean solution by means of having statistical computation for each running of the SSCO and other algorithms. The experimental results obtained by other algorithms are provided in [1].

Table 2 shows the comparison of the test results between the SSCO and the known three algorithms. Opt. is the optimum value of each NLP. The results indicate that SSCO is superior to the known two algorithms from the viewpoints of the best solution, worst solution, and mean solution. Meanwhile, we can see that best solutions obtained by SSCO are better than other three algorithms since those solutions are much close to the true optimal solutions.

Figures 2, 3, 4, 5, 6, 7, 8, and 9 show the relative fitness value $F\text{norm} = |F\text{best} - F\text{opt}|$, which are performed by SCO and SSCO, versus $t/100$ for different benchmark NLPs, respectively, where t is current generation number. The SSCO, which has periodic partial reinitialization, shows higher convergence velocity and higher sustainable evolutionary capability at the process of evolution than traditional SCO.

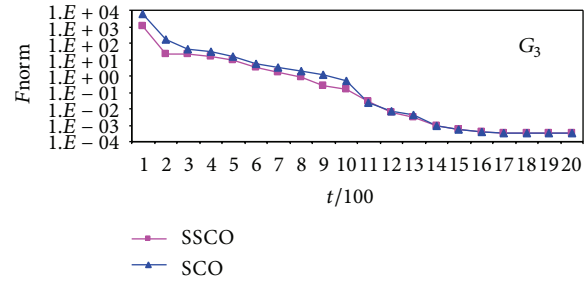


Figure 4: Fnorm versus $t/100$ for G_3 by SSCO and SCO.

Furthermore, the iterative process of the SSCO has no more computing than traditional SCO.

For SSCO and SCO, the only difference is that the SSCO agents include the strategy of periodic restart, which is guaranteed to converge to the global optimization solution. It makes no effect on the total computing, however, because the SSCO considers static individuals are substituted with new randomly generated individuals to avoid the premature convergence in the learning iteration; the SSCO shows higher performance than SCO, PSO, and GA in all the eight test cases.

From the above analysis, the SSCO has higher efficiency in solving NLPs for reaching the near-optimal solutions. Consequently, the experimental results indicate that the SSCO has better robustness, effectiveness, and stableness than SCO and GA reported in the literature.

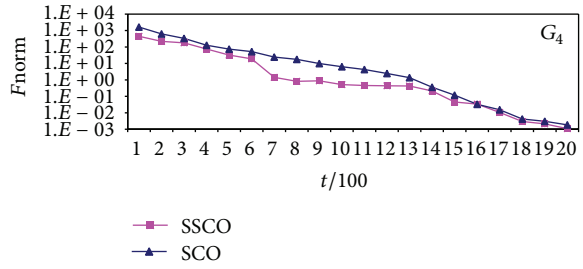


Figure 5: Fnorm versus $t/100$ for G_4 by SSCO and SCO.

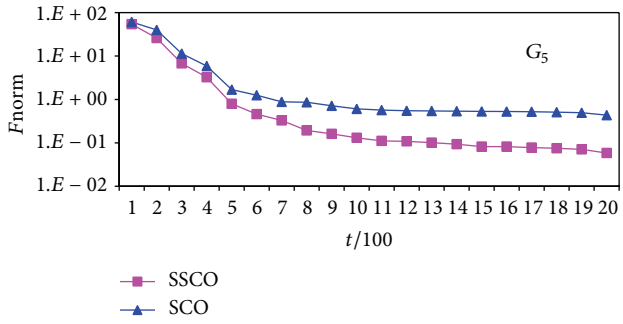


Figure 6: Fnorm versus $t/100$ for G_5 by SSCO and SCO.

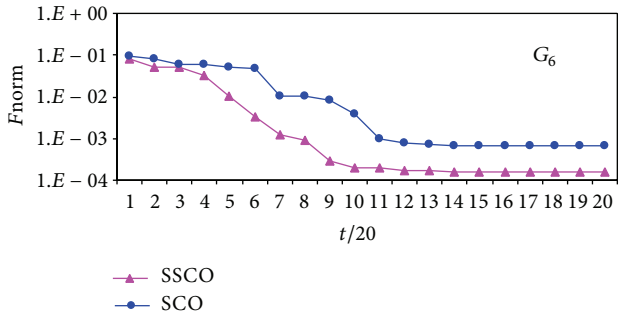


Figure 7: Fnorm versus $t/100$ for G_6 by SSCO and SCO.

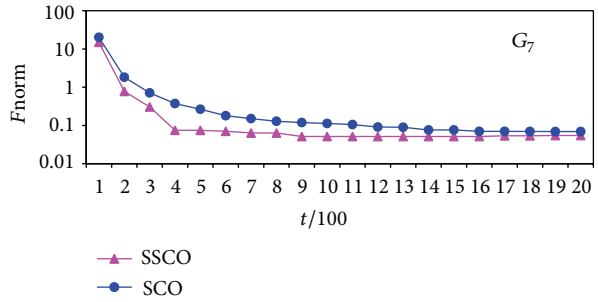


Figure 8: Fnorm versus $t/100$ for G_7 by SSCO and SCO.

5. Conclusions

A stochastic social cognitive optimization (SSCO) algorithm with the strategy of periodic restart is proposed in this paper for solving NLPs. The periodic restart of SCO for static individuals can avoid the premature convergence, improve the global searching performance, and ensure the algorithm to obtain stable optimal solution. The convergence proof for the stochastic SCO is given based on Solis and Wets' research results. The final experiment results indicate that the new algorithm has good capability to find optimal solution.

Appendix

The appendix provides the description of eight test functions.

G_1:

$$\text{Minimize} \quad G_1(\vec{x}) = 5x_1 + 5x_2 + 5x_3 + 5x_4 - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$$

$$\begin{aligned} \text{Subject to} \quad & 2x_1 + 2x_2 + x_{10} + x_{11} \le 10, \\ & 2x_1 + 2x_3 + x_{10} + x_{12} \le 10, \\ & 2x_2 + 2x_3 + x_{11} + x_{12} \le 10, \\ & -8x_1 + x_{10} \le 0, \quad -8x_2 + x_{11} \le 0, \\ & -8x_3 + x_{12} \le 0, \quad -2x_4 - x_5 + x_{10} \le 0, \\ & -2x_6 - x_7 + x_{11} \le 0, \quad -2x_8 - x_9 + x_{12} \le 0, \end{aligned} \tag{A.1}$$

where

$$\begin{aligned} & 0 \le x_i \le 1, \quad i = 1, \ldots, 9, \\ & 0 \le x_i \le 100, \quad i = 10, 11, 12, \\ & 0 \le x_{13} \le 1 \end{aligned} \tag{A.2}$$

$$G_1(\vec{x}^*) = -15.$$

G_2:

$$\text{Maximize} \quad G_2(\vec{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$$

$$\text{Subject to} \quad \prod_{i=1}^{n} x_i \ge 0.75, \quad \sum_{i=1}^{n} x_i \ge 7.5n, \tag{A.3}$$

where

$$0 \le x_i \le 10 \quad \text{for } 1 \le i \le n. \tag{A.4}$$

$$G_2(\vec{x}^*) = 0.803553.$$

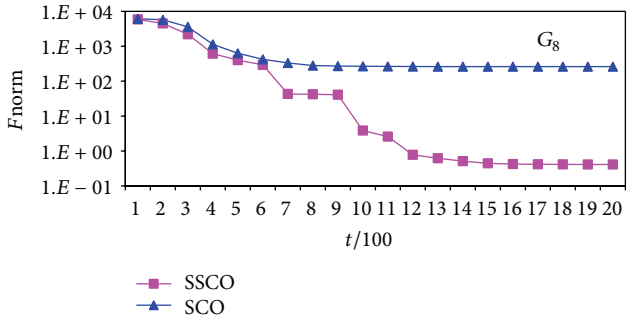


FIGURE 9: Fnorm versus $t/100$ for G_8 by SSCO and SCO.

G_3:

$$\begin{aligned}
\text{Minimize}\quad & G_3(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 \\
& + 37.293239x_1 - 40792.141 \\
\text{Subject to}\quad & 0 \le 85.334407 + 0.0056858x_2x_5 \\
& + 0.0006262x_1x_4 - 0.0022053x_3x_5 \le 92 \\
& 90 \le 80.51249 + 0.0071317x_2x_5 \\
& + 0.0029955x_1x_2 + 0.0021813x_3^2 \le 110 \\
& 20 \le 9.300961 + 0.0047026x_2x_5 \\
& + 0.0012547x_1x_3 + 0.0019085x_3x_4 \le 25,
\end{aligned} \tag{A.5}$$

where

$$\begin{aligned}
& 78 \le x_1 \le 102, \quad 33 \le x_2 \le 45, \\
& 27 \le x_i \le 45 \quad \text{for } i = 3, 4, 5 \\
& G_3(\vec{x}^*) = -30665.5.
\end{aligned} \tag{A.6}$$

G_4:

$$\begin{aligned}
\text{Minimize}\quad & G_4(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3 \\
\text{Subject to}\quad & (x_1 - 5)^2 + (x_2 - 5)^2 - 100 \ge 0 \\
& -(x_1 - 6)^2 - (x_2 - 5)^2 + 82.81 \ge 0,
\end{aligned} \tag{A.7}$$

where

$$\begin{aligned}
& 13 \le x_1 \le 100, \quad 0 \le x_2 \le 100, \\
& G_4(\vec{x}^*) = -6961.81381.
\end{aligned} \tag{A.8}$$

G_5:

$$\begin{aligned}
\text{Minimize}\quad & G_5(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 \\
& + (x_3 - 10)^2 + 4(x_4 - 5)^2 \\
& + (x_5 - 3)^2 + 2(x_6 - 1)^2 \\
& + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 \\
& + (x_{10} - 7)^2 + 45 \\
\text{Subject to}\quad & 105 - 4x_1 - 5x_2 + 3x_7 - 9x_8 \ge 0, \\
& -3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2x_3^2 \\
& + 7x_4 + 120 \ge 0, \\
& -10x_1 + 8x_2 + 17x_7 - 2x_8 \ge 0, \\
& -x_1^2 - 2(x_2 - 2)^2 + 2x_1x_2 - 14x_5 + 6x_6 \ge 0, \\
& 8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 \ge 0, \\
& -5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 \ge 0, \\
& 3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \ge 0, \\
& -0.5(x_1 - 8)^2 - 2(x_2 - 4)^2 - 3x_5^2 + x_6 + 30 \ge 0,
\end{aligned} \tag{A.9}$$

where

$$\begin{aligned}
& -10.0 \le x_i \le 10.0, \quad i = 1, \ldots, 10, \\
& G_5(\vec{x}^*) = 24.3062091.
\end{aligned} \tag{A.10}$$

G_6:

$$\begin{aligned}
\text{Maximize}\quad & G_6(\vec{x}) = \frac{\sin^3(2\pi x_1) \cdot \sin(2\pi x_2)}{x_1^3 \cdot (x_1 + x_2)} \\
\text{Subject to}\quad & x_1^2 - x_2 + 1 \le 0 \\
& 1 - x_1 + (x_2 - 4)^2 \le 0,
\end{aligned} \tag{A.11}$$

where

$$\begin{aligned}
& 0 \le x_1 \le 10, \quad 0 \le x_2 \le 10, \\
& G_6(\vec{x}^*) = 0.095825.
\end{aligned} \tag{A.12}$$

G_7:

$$\begin{aligned}\text{Minimize}\quad & G_7(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 \\ & + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 \\ & - 4x_6x_7 - 10x_6 - 8x_7 \\ \text{Subject to}\quad & 127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geq 0, \\ & 282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \geq 0 \\ & 196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \geq 0, \\ & -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7 \geq 0,\end{aligned} \tag{A.13}$$

where

$$\begin{aligned}& -10.0 \leq x_i \leq 10.0, \quad i = 1, \ldots, 7, \\ & G_7(\vec{x}^*) = 680.6300573.\end{aligned} \tag{A.14}$$

G_8:

$$\begin{aligned}\text{Minimize}\quad & G_8(\vec{x}) = x_1 + x_2 + x_3 \\ \text{Subject to}\quad & 1 - 0.0025(x_4 + x_6) \geq 0, \\ & 1 - 0.0025(x_5 + x_7 - x_4) \geq 0 \\ & 1 - 0.01(x_8 - x_5) \geq 0, \\ & x_1x_6 - 833.33252x_4 - 100x_1 + 83333.333 \geq 0 \\ & x_2x_7 - 1250x_5 - x_2x_4 + 1250x_4 \geq 0, \\ & x_3x_8 - 1250000 - x_3x_5 + 2500x_5 \geq 0,\end{aligned} \tag{A.15}$$

where

$$\begin{aligned}& 100 \leq x_1 \leq 10000, \quad 1000 \leq x_i \leq 10000, \\ & i = 2, 3, \quad 100 \leq x_i \leq 10000, \quad i = 4, \ldots, 8, \\ & G_8(\vec{x}^*) = 7049.330923.\end{aligned} \tag{A.16}$$

Conflict of Interests

The authors (Jia-ze Sun, Shu-yan Wang, and Hao Chen) declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (611721701, 61203311, 61105064, and 61050003), by the Scientific Research Program of Shaanxi Provincial Education Department (12JK0732 and 13JK1183), by the Natural Science Foundation of XUPT (ZL2013-26), and by special funding for Key Discipline Construction of General Institutions of Higher Learning from Shaanxi Province and special funding for course development for Xi'an University of Posts and Telecommunications.

References

[1] X.-F. Xie, W.-J. Zhang, and Z.-L. Yang, "Social cognitive optimization for nonlinear programming problems," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 779–783, Beijing, China, November 2002.

[2] J. Z. Sun, S. Y. Wang, J. K. Zhang, and X.-P. Cao, "SCO algorithm based on entropy function for NCP," *Computer Engineering and Applications*, vol. 46, no. 21, pp. 40–42, 2010, (Chinese).

[3] K. J. Zhang, Z. J. Sun, and X. L. Kou, "Social cognitive optimization for fractional programs," *Computer Engineering and Design*, vol. 29, no. 21, pp. 5543–5545, 2008 (Chinese).

[4] J. Sun Z and J. Zhang, "Solving nonlinear systems of equations based on social cognitive optimization," *Computer Engineering and Applications*, vol. 44, no. 28, pp. 42–44, 2008 (Chinese).

[5] X. F. Xie and W. J. Zhang, "Solving engineering design problems by social cognitive optimization," in *Genetic and Evolutionary Computation—GECCO 2004*, vol. 3102 of *Lecture Notes in Computer Science*, pp. 261–262, Springer, Berlin, Germany, 2004.

[6] Y.-P. Chen, J.-K. Zhang, J.-Z. Sun, Q.-H. Zheng, and Z.-Z. Li, "A service selection model using mixed intelligent optimization," *Chinese Journal of Computers*, vol. 33, no. 11, pp. 2116–2125, 2010 (Chinese).

[7] L.-J. Wang, C.-J. Wei, and C.-X. Li, "Social cognitive optimization modified with self-organizing migrating algorithm," *Computer Engineering*, vol. 22, pp. 201–203, 2008.

[8] L. Ma, R.-X. Wang, and Y. P. Chen, "Solving the nonlinear problem based on the modified social cognitive optimization algorithm," *Computer Engineering*, vol. 10, pp. 170–172, 2011.

[9] J. Z. Sun, G. H. Geng, S. Y. Wang, and M. Q. Zhou, "Hybrid social cognitive optimization algorithm for constrained nonlinear programming," *Journal of China Universities of Posts and Telecommunications*, vol. 19, no. 3, pp. 91–99, 2012.

[10] L. Zhi-zhong, W. Zhi-Jian, and X. Xiao, "Research on cloud service composition based on culture social cognitive optimization algorithm," *Computer Science*, vol. 40, no. 5, pp. 103–106, 2013.

[11] J.-Z. Sun, G.-H. Geng, and X.-Y. Pan, "D-S algorithm based on SCO for matching fragments," *Applied Mechanics and Materials*, vol. 411–414, pp. 1876–1879, 2013.

[12] F. J. Solis and R. J.-B. Wets, "Minimization by random search techniques," *Mathematics of Operations Research*, vol. 6, no. 1, pp. 9–30, 1981.

[13] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," *Evolutionary Computation*, vol. 7, no. 1, pp. 19–44, 1999.

[14] X.-F. Xie, W.-J. Zhang, and D.-C. Bi, "Handling equality constraints by adaptive relaxing rule for swarm algorithms," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, pp. 2012–2016, Portland, Ore, USA, June 2004.

[15] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 481791, 9 pages
http://dx.doi.org/10.1155/2014/481791

Research Article

Flower Pollination Algorithm with Dimension by Dimension Improvement

Rui Wang and Yongquan Zhou

College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

Correspondence should be addressed to Yongquan Zhou; yongquanzhou@126.com

Received 8 July 2014; Accepted 16 August 2014; Published 1 September 2014

Academic Editor: Shifei Ding

Flower pollination algorithm (FPA) is a new nature-inspired intelligent algorithm which uses the whole update and evaluation strategy on solutions. For solving multidimension function optimization problems, this strategy may deteriorate the convergence speed and the quality of solution of algorithm due to interference phenomena among dimensions. To overcome this shortage, in this paper a dimension by dimension improvement based flower pollination algorithm is proposed. In the progress of iteration of improved algorithm, a dimension by dimension based update and evaluation strategy on solutions is used. And, in order to enhance the local searching ability, local neighborhood search strategy is also applied in this improved algorithm. The simulation experiments show that the proposed strategies can improve the convergence speed and the quality of solutions effectively.

1. Introduction

In recent years, more and more bioinspired algorithms are proposed, such as genetic algorithm (GA) [1], simulated annealing (SA) [2], particle swarm optimization (PSO) [3], firefly algorithm (FA) [4], glowworm swarm optimization (GSO) [5], monkey search (MS) [6], bacterial foraging optimization algorithm (BFOA) [7], invasive weed optimization (IWO) [8], cultural algorithms (CA) [9], and harmony search (HS) [10]. Because of their advantages of global and parallel efficiency, robustness, and universality, swarm intelligence algorithms have been widely used in engineering optimization, scientific computing, automatic control, and other fields.

Flower pollination algorithm (proposed by Yang in 2012) [11] is a new population-based intelligent optimization algorithm by simulating flower pollination behavior. And FPA has been extensively researched to solve Integer Programming Problems [12], Sudoku Puzzles [13], and Wireless Sensor Network Lifetime Global Optimization [14] in the last two years by scholars. It is estimated that there are over 250,000 types of flowering plants in nature. And researchers of biology considered that almost four-fifths of all plant species are flowering species. Flower pollination behavior stems from the purpose of reproduction. From the biological evolution point of view, the objective of flower pollination is the survival of the fittest and the optimal reproduction of species. All these factors and processes of flower pollination interact so as to achieve optimal reproduction of the flowering plants. In nature, pollination can be divided into two parts: abiotic and biotic. Almost 90% pollen grains are transferred by insects and animals; we call this biotic pollination. The other 10% pollen grains are transferred by wind [15, 16]. They do not need pollinators. And we call this form abiotic pollination. Pollinators can be very diverse; researches show almost 200,000 kinds of pollinators.

Self-pollination and cross-pollination are two different ways of pollination [17]. Cross-pollination means pollination can occur from pollen of a flower of a different plant, and self-pollination is just the opposite. Biotic, cross-pollination can occur at long distance; the pollinators such as bees, bats, birds can fly a long distance; thus they can be considered as the global pollinators. And these pollinators can fly as Lévy flight behavior [18], with fly distance steps obeying a Lévy distribution. Thus, this can inspire to design new optimization algorithm. Flower pollination algorithm is an optimization algorithm which simulates the flower pollination behavior

mentioned above; flower pollination algorithm can also be divided into global pollination process and local pollination process.

2. FPA with Dimension by Dimension Improvement

In order to enhance the global searching and local searching abilities, we applied three optimization strategies to basic flower pollination algorithm (FPA); those were local neighborhood searching strategy (LNSS) [19], dimension by dimension evaluation and improvement strategy (DDEIS), and dynamic switching probability strategy (DSPS).

2.1. Local Neighborhood Search Strategy (LNSS). FPA (developed by Yang and Deb) uses differential evolution (DE) algorithm [20] to do local search. And experiment results show that the local search ability of DE is limited. Thus, we add LNSS to local search process to enhance its exploitation ability.

Firstly, we should explain a model (local neighborhood model). In this model, each vector uses the best vector of only a small neighborhood rather than the entire population to do the mutation. We suppose that there exists a differential evolutionary population $P_G = [X_{1,G}, X_{2,G}, \ldots, X_{i+1,G}]$, and each $X_{i,G}$ $(i = 1, 2, 3, \ldots, NP)$ is a parameter vector, and its dimension is D. Each vector subscript index is randomly divided to ensure the diversity of each neighborhood. For each vector $X_{i,G}$, we can define a neighborhood, and the radius is k $(2k + 1 < NP)$. The neighborhood consists of vector $X_{i-k,G}, \ldots, X_{i,G}, \ldots, X_{i+k,G}$. Assume that the vectors accord the subscript indices in a ring topology structure. We can take $X_{NP,G}$ and $X_{2,G}$ as two direct neighbors of $X_{1,G}$. The concept of local neighborhood model is shown in Figures 1 and 2. The neighborhood topology here is static and determined by the collection of vector subscript indices. And the local neighborhood model can be expressed in the following formula:

$$X_{i,G+1} = X_{i,G} + \alpha\left(X_{n\_\text{best}_i,G} - X_{i,G}\right) + \beta\left(X_{p,G} - X_{q,G}\right), \tag{1}$$

where $X_{n\_\text{best}_i,G}$ is the best vector of $X_{i,G}$ neighborhood, $p, q \in [i-k, i+k]$ $(p \neq q \neq i)$, and α, β are two scale factors.

2.2. Dimension by Dimension Evaluation and Improvement Strategy (DDEIS). Flower pollination algorithm uses the whole update and evaluation strategy on solutions. For solving multidimensional function optimization problems, this strategy may deteriorate the convergence speed and the quality of solution due to interference phenomena among dimensions. To overcome this shortage, we add this strategy to FPA in local search process.

In FPA, Lévy flight can improve the diversity of population and strengthen the global search ability of the algorithm. But for multidimensional objective function, overall update evaluation strategy will affect the convergence rate and quality of solutions. DDEIS updates dimension by dimension.

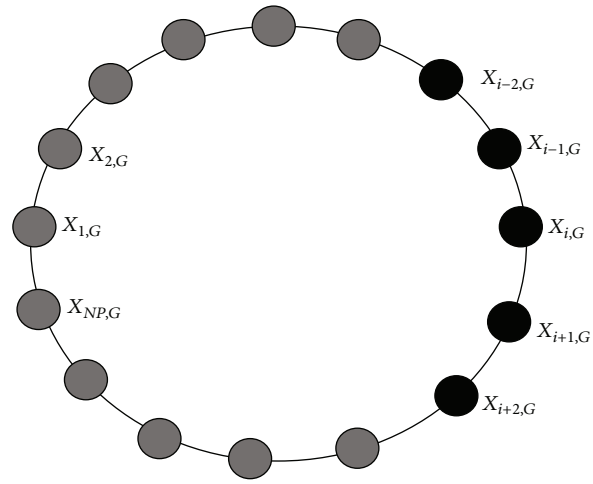


Figure 1: Neighborhood ring topology.

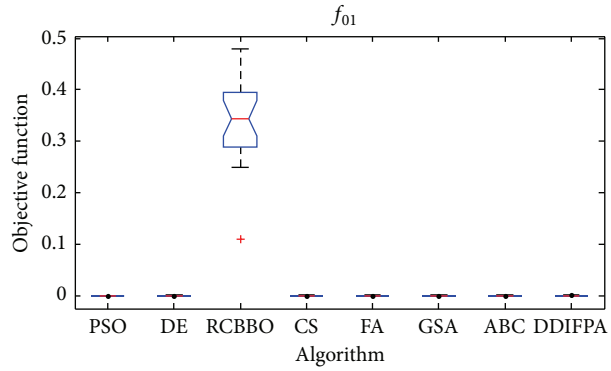


Figure 2: ANOVA tests for f_{01}.

Assume that objective function is $f(X) = X_1^2 + X_2^2 + X_3^2$ and $X_{g,i} = (0.5, 0.5, 0.5)$ is a solution to $f(X)$.

The objective value $f(X_{g,i}) = 0.75$. We use formula (1) to update $X_{g,i}$ and get $X_{g+1,i} = (0, 1, -1)$. For example, when first dimension value of $X_{g,i}$ updates from 0.5 to 0, combined with the value of other dimensions, we can get a new $X_{g+1,i} = (0, 0.5, 0.5)$. The objective value $f(X_{g+1,i}) = 0.5 < f(X_{g,i})$; it can improve current solution. Thus, we accept this update and update operation into the next dimension. If first dimension value of $X_{g,i}$ updates from 0.5 to 1, we can get a new $X_{g+1,i} = (1, 0.5, 0.5)$. The objective value $f(X_{g+1,i}) = 1.5 > f(X_{g,i})$, it fails to improve $X_{g,i}$, and we should abandon the current dimension updated value and update operation into the next dimension. The strategy is described in Algorithm 1.

2.3. Dynamic Switching Probability Strategy (DSPS). In FPA, local search and global search are controlled by a switching probability $p \in [0, 1]$, and it is a constant value. We suppose that a reasonable algorithm should do more global search at the beginning of searching process and global search should be less in the end. Thus, we applied the dynamic switching probability strategy (DSPS) to adjust the proportion of two

```
temp2 = X_{g,i};
temp = X_{g+1,i};
for m = 1 : d
    temp3 = temp2;
    temp2(m) = temp(m);
    if fitness(temp2 ) > fitness(temp3),
        temp2(m) = X_{g,i}(m);
    endif
endfor
```

Algorithm 1: Dimension by dimension evaluation and improvement strategy.

kinds of searching process. Switching probability p can alter according to the following formula:

$$p = 0.6 - 0.1 \times \frac{(\text{Max\_iter} - t)}{\text{Max\_iter}}, \quad (2)$$

where Max_iter is the maximum iterations of the DDIFPA and t is current iteration. Specific implementation steps of FPA with dimension by dimension improvement (DDIFPA) can be summarized in the pseudocode shown in Algorithm 2.

3. Numerical Simulation Experiments

In this section, we applied 12 standard test functions [21] to evaluate the optimal performance of FPA with dimension by dimension improvement (DDIFPA). The mean and standard deviation results of 20 independent runs for each algorithm have been summarized in Table 2. The 12 standard benchmark functions have been widely used in the literature. The dimensions, scopes, optimal values, and iterations of 12 functions are in Table 1. We also do some high-dimensional tests, and the results are showed in Table 3.

3.1. Experimental Setup. All of the algorithm was programmed in MATLAB R2012a; numerical experiment was set up on AMD Athlont (tm) II∗4640 processor and 2 GB memory.

3.2. Comparison of Each Algorithm Performance. The proposed DDIFPA algorithm is compared with mainstream swarm intelligence algorithms FPA [11], PSO [22], DE [23], RCBBO [24], GSA [25], FA [26], CS [27], and ABC [28], respectively, using the mean and standard deviations to compare their optimal performances. The setting values of algorithm control parameters of the mentioned algorithms are given as follows.

PSO parameters setting: weight factor $\omega = 0.6$, $c_1 = c_2 = 2$. The population size is 100 [22].

DE parameters setting: $F = 0.5$ and CR = 0.9 in accordance with the suggestions given in [23]; the population size is 100.

ABC parameters setting: *limit* = $5D$ has been used as recommended in [22]; the population size is 50 because this algorithm has two phases.

RCBBC parameters setting: maximum immigration rate: $I = 1$, maximum emigration rate: $E = 1$, and *mutation probability*: $m_{\max} = 0.005$ have been used as recommended in [24]; the population size is 100.

CS parameters setting: $\beta = 1.5$ and $\rho_0 = 1.5$ have been used as recommended in [27]; the population size is 50 because this algorithm has two phases.

GSA parameters setting: $G_0 = 100, \alpha = 20$ and K_0 which is set to *NP* and is decreased linearly to 1 have been used as recommended in [25]; the population size is 100.

FA parameters setting: $\alpha_0 = 0.5$, $\beta_0 = 0.2$, and $\gamma = 1$ have been used as recommended in [26]; the population size is 100.

FPA parameters setting: the population size is 50 because this algorithm has two phases [11].

DDIFPA parameters setting: the population size is 50 because this algorithm has two phases.

From the rank of each function in Table 2, we can conclude that DDIFPA provides many of the best results are better than FPA and other algorithms, especially for functions f_{01}, f_{03}, and f_{05}. For f_{01} the mean and standard deviation of DDIFPA are much higher than FPA. For f_{03}, the mean and standard deviation of DDIFPA are 117 orders of magnitude higher than GSA and 127 orders of magnitude higher than FPA. For f_{04}, DDIFPA and FPA fail to give the best optimal solution. For f_{05}, the mean and standard deviation of DDIFPA are 2 orders of magnitude higher than FPA.

Figures 2 and 3 show the graphical analysis results of ANOVA test. As can be seen in Figure 2, when solving function f_{01}, most of the algorithms can obtain the stable optimal value after 20 independent runs except RCBBO algorithm, and, in Figure 3, when solving the function f_{05}, DDIFPA is more stable than other algorithms.

Figures 4 and 5 show the fitness function curve evolution of each algorithm for f_{01} and f_{05}. From the two figures, we can conclude that DDIFPA has a faster convergence rate and a higher optimizing precision.

For multimodal functions f_{06} to f_{10} with many local minima, the final results are more important because these functions can reflect the ability of algorithm to escape from poor local optima and obtain the global optimum.

As can be seen in Table 2, for f_{06} and f_{07}, DDIFPA are in first place; ABC achieve the optimal value when solving f_{07}. For f_{08} the mean and standard deviation of DDIFPA are 15 orders of magnitude higher than FPA. For f_{09}, ABC and DDIFPA all achieve the optimal value and the standard

Objective min or max $f(x), x = (x_1, x_2, \ldots, x_d)$
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution g_* in the initial population
while ($t <$ *MaxGeneration*)
  for $i = 1 : n$ (all n flowers in the population)
    Get p according to formula (2);
    if rand $< p$
      Draw a (d-dimensional) step vector L which obeys a Lévy distribution
      Global pollination via $x_i^{t+1} = x_i^t + \gamma L(\lambda)(g_* - x_i^t)$;
    else
      Draw ε from a uniform distribution in [0, 1];
      Local pollination via $X_{i,G+1} = X_{i,G} + \alpha(X_{n\_best_i,G} - X_{i,G}) + \beta(X_{p,G} - X_{q,G})$;
      where $\alpha = \beta = \varepsilon$;
    end if
      Evaluate new solutions via DDEIS
      If new solutions are better, update them in the population
  end for
      find the current best solution g_*
end while

Algorithm 2: FPA with dimension by dimension improvement (DDIFPA).

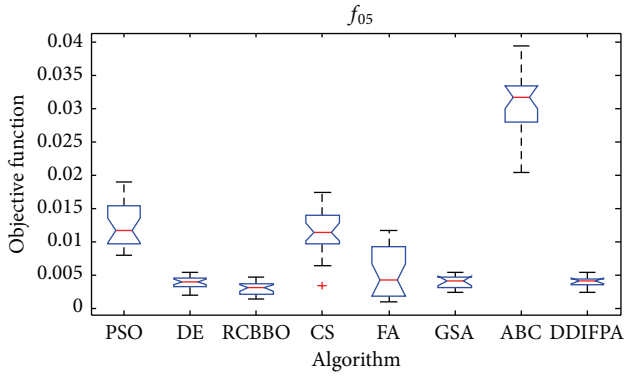


Figure 3: ANOVA tests for f_{05}.

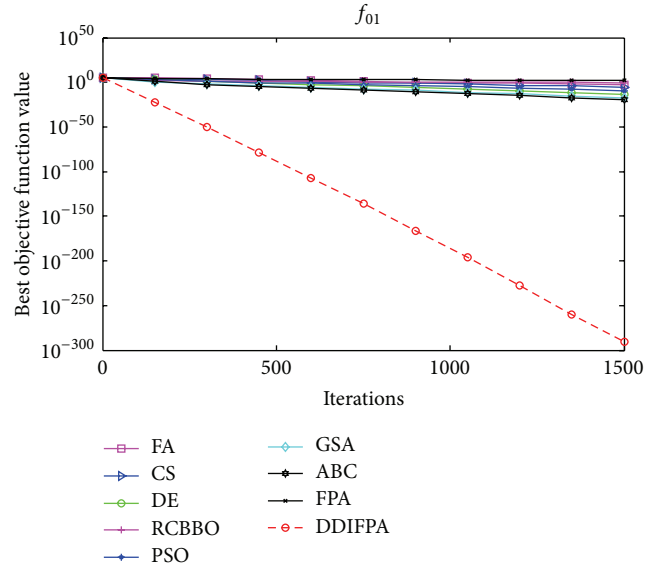


Figure 4: Fitness function curve evolution for f_{01}.

deviations are all 0. For f_{10}, the mean of DDIFPA is 11 orders of magnitude higher than ABC, and the standard deviation of DDIFPA is 27 orders of magnitude higher than ABC.

Figures 6 and 7 show the graphical analysis results of the ANOVA tests. Figure 6 shows that RCBBO, ABC, and DDIFPA can obtain the relatively stable optimal values. Figure 7 shows that when solving function f_{10}, most of the algorithms can obtain the stable optimal value after 20 independent runs.

Figures 8 and 9 show the fitness function curve evolution. From Figure 9, we can conclude that both ABC and DDIFPA converge to the optimal solution. From Figure 9, we can conclude that DDIFPA converges to a more precise point than other algorithms, and its convergence speed is faster.

From Table 2, f_{11} and f_{12} are multimodal low-dimensional functions. For f_{11}, the solutions of most of the algorithms are accurate in 3 to 4 decimal places, and the rank of DDIFPA is second. For f_{12}, the rank is second too, and the experiment results show that DDIFPA can do a good job in solving multimodal low-dimensional problems.

3.3. Experimental Analysis. We have carried out benchmark validations for unimodal and multimodal test functions using the proposed algorithm (DDIFPA) with three improvement strategies (local neighborhood search strategy, dimension by dimension evaluation and improvement strategy, and dynamic switching probability strategy). An optimization

Table 1: Benchmark test functions.

Benchmark test functions	Dimension	Range	Optimum	Iterations
$f_{01} = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]$	0	1500
$f_{02} = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]$	0	2000
$f_{03} = \max_i \{\lvert x_i \rvert, 1 \le i \le D\}$	30	$[-100, 100]$	0	5000
$f_{04} = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	30	$[-30, 30]$	0	5000
$f_{05} = \sum_{i=1}^{D} i x_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]$	0	3000
$f_{06} = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	$[-500, 500]$	$-418.9829^* n$	3000
$f_{07} = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	30	$[-5.12, 5.12]$	0	3000
$f_{08} = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right) + 20 + e$	30	$[-32, 32]$	0	1500
$f_{09}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{2}}\right) + 1$	30	$[-600, 600]$	0	2000
$f_{10}(x) = \frac{\pi}{D}\left\{\begin{array}{c} 10\sin^2(\pi y_1) \\ +\sum_{i=1}^{D-1}(y-1)^2\left[1 + 10\sin^2(\pi y_1)\right] + (y_D - 1)^2 \end{array}\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$, $y_i = 1 + \frac{x_i + 1}{4}$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a \le x_i \le a \\ k(-x_i - z)^m, x_i < a \end{cases}$	30	$[-50, 50]$	0	1500
$f_{11}(x) = -\sum_{i=1}^{4} c_i \exp\left[\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right]$	3	$[0, 1]$	-3.8628	100
$f_{12}(x) = -\sum_{i=1}^{10}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	-10.5364	100

process can be divided into two key components (local search and global search); we use a dynamic switching probability $p \in [0, 1]$ to control the whole searching process. LNSS and DDEIS are applied to the local search process and enhance its exploitation ability. Among 12 test functions listed above, f_{01} to f_{05} are unimodal, and the remarkable achievements confirm that DDIFPA have stronger exploitation ability than FPA and other algorithms. And DSPS, which could improve the ability of escape from poor local optima, was applied to enhance the exploration ability. That also balanced exploitation and exploration dynamically. For multimodal benchmark functions (f_{06} to f_{12}), we can conclude that DDIFPA converges to a more precise point than other algorithms, and its convergence speed is faster. Our simulation results for finding the global optima of various test functions suggest that DDIFPA can outperform the FPA and other mentioned algorithms in terms of both precision and convergence speed.

3.4. High-Dimensional Functions Test. In previous sections, 12 standard test functions are applied to evaluate the optimal performances of the FPA with dimension by dimension improvement (DDIFPA) in the case of low dimension. In order to evaluate the performances of DDIFPA comprehensively, we also do some high-dimensional tests in f_1, f_2, f_4, f_7, f_{10}. The test results are shown in Table 3. As can be seen in Table 3, DDIFPA can also solve high-dimensional problems efficiently and stably.

4. Conclusions

In this paper, three optimization strategies (local neighborhood search strategy, dimension by dimension evaluation and improvement strategy, and dynamic switching probability strategy) have been applied to FPA to improve its deficiencies. By 12 typical standard benchmark functions simulation,

Table 2: Experiment results of bench mark functions for different algorithms.

Functions		PSO	DE	RCBBO	CS	FA	GSA	ABC	FPA	**DDIFPA**
f_{01}	Mean	$3.33E-10$	$5.60E-14$	0.3737	$5.66E-06$	$1.70E-03$	$3.37E-18$	$2.99E-20$	123.5791661	$4.62E-289$
	Std.	$7.04E-10$	$4.41E-14$	0.1181	$2.86E-06$	$4.06E-04$	$8.09E-19$	$2.15E-20$	52.5878636	0
	Rank	5	4	8	6	7	3	2	9	1
f_{02}	Mean	$6.66E-11$	$4.73E-10$	0.1656	$2.00E-03$	$4.53E-02$	$8.92E-09$	$1.42E-15$	8.27840887	$3.61E-196$
	Std.	$9.26E-11$	$1.78E-10$	0.0342	$8.10E-04$	$3.38E-02$	$1.33E-09$	$5.53E-16$	2.19245633	0
	Rank	3	4	8	6	7	5	2	9	1
f_{03}	Mean	7.9997	0.2216	7.9738	3.2388	0.0554	$9.93E-10$	18.5227	3.759984116	$4.52E-127$
	Std.	2.535	0.243	2.6633	0.6644	0.0101	$1.19E-10$	4.2477	1.301640714	$5.83E-127$
	Rank	7	4	8	5	3	2	9	6	1
f_{04}	Mean	46.9202	0.2657	64.6907	8.0092	38.1248	20.0819	0.0441	32.25150881	0.065392617
	Std.	38.0312	1.0293	36.2782	1.9188	30.3962	0.1722	0.0707	12.70746052	0.094325113
	Rank	8	3	9	4.0000	7	5	1	6	2
f_{05}	Mean	0.0135	0.0042	0.003	0.0096	0.0082	0.0039	0.0324	0.029755135	0.003729368
	Std.	0.0041	0.0014	0.0012	0.0028	0.0093	0.0013	0.0059	0.013355902	0.000996113
	Rank	7	4	1	6	5	3	9	8	2
f_{06}	Mean	$-8.83E+03$	$-1.13E+04$	$-1.26E+04$	$-9.15E+03$	$-6.22E+03$	$-3.05E+03$	$-1.25E+04$	−8448.868832	−12569.48662
	Std.	611.159	$1.81E+03$	0.5758	$2.53E+02$	$7.72E+02$	$3.39E+02$	61.1186	292.6519355	$1.92E-12$
	Rank	6	4	2	5	8	9	3	7	1
f_{07}	Mean	18.2675	134.6789	0.0385	51.2202	23.5213	7.2831	0	75.97229041	0
	Std.	4.7965	28.8598	0.0154	8.1069	8.3683	1.8991	0	12.38031696	0
	Rank	5	9	3	7	6	3	1	8	1
f_{08}	Mean	$3.87E-06$	$7.47E-08$	0.1947	2.375	0.0094	$1.47E-09$	$1.19E-09$	3.706074906	$4.44E-15$
	Std.	$2.86E-06$	$3.11E-08$	0.0461	1.1238	0.0014	$1.44E-10$	$5.01E-10$	0.519323445	0
	Rank	5	4	7	8	6	2	3	9	1
f_{09}	Mean	0.0168	0	0.2765	$4.49E-05$	0.0025	0.01265	0	4.559765038	0
	Std.	0.0205	0	0.0796	$8.96E-05$	$4.69E-04$	0.0216	0	1.769274137	0
	Rank	7	1	8	4	5	6	1	9	1
f_{10}	Mean	0.0083	$4.71E-15$	0.002	0.5071	$8.87E-06$	$2.04E-20$	$1.19E-21$	4.3551	$1.57E-32$
	Std.	0.0287	$3.26E-15$	0.0023	0.2662	$2.80E-06$	$4.53E-21$	$1.08E-21$	1.1793	$2.89E-48$
	Rank	7	4	6	8	5	3	2	9	1
f_{11}	Mean	−3.8628	−3.8628	−3.8627	−3.8628	−3.8613	−3.8625	−3.8628	−3.861091803	−3.862782148
	Std.	$3.13E-12$	$2.30E-15$	$1.41E-04$	$1.40E-05$	0.0037	$3.88E-04$	$1.37E-10$	0.00149557	$1.43E-13$
	Rank	3	1	6	5	8	7	4	9	2
f_{12}	Mean	−8.9611	−10.5364	−9.3514	−9.7534	−10.2297	−8.2651	−10.5339	−5.006727392	−10.53636839
	Std.	2.8381	$3.97E-06$	2.6288	0.4913	1.5332	2.8868	0.0054	1.235737304	$7.4406E-05$
	Rank	7	1	6	5	4	8	3	9	2

Table 3: High-dimensional functions test results.

Functions	Dimensions	Means	Std.	Best	Worst
f_{01}	1000	$6.47933690759837E-284$	0	$9.49725772050939E-288$	$3.19181958962329E-283$
f_{02}	500	$2.24079551697238E-193$	0	$5.54622672996093E-195$	$5.86867756931104E-193$
f_{04}	500	$3.4272355502299E-17$	$2.42530519040223E-17$	$6.91784434902547E-18$	$6.99285915411276E-17$
f_{07}	500	0	0	0	0
f_{10}	500	$1.57054477178664E-32$	0	$1.57054477178664E-32$	$1.57054477178664E-32$

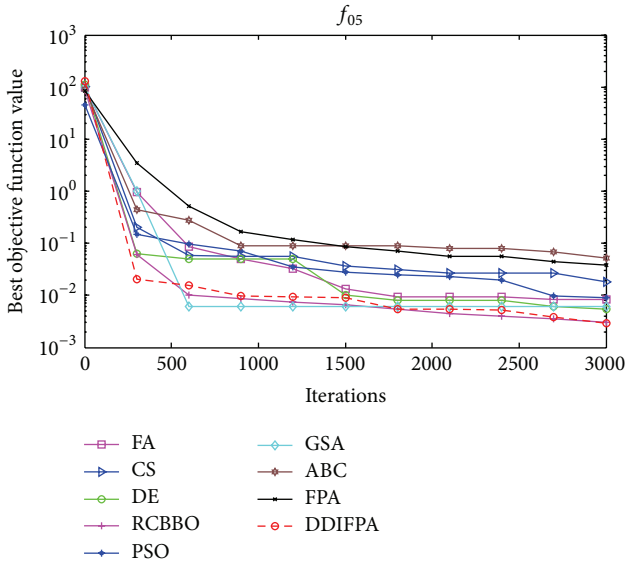


Figure 5: Fitness function curve evolution for f_{05}.

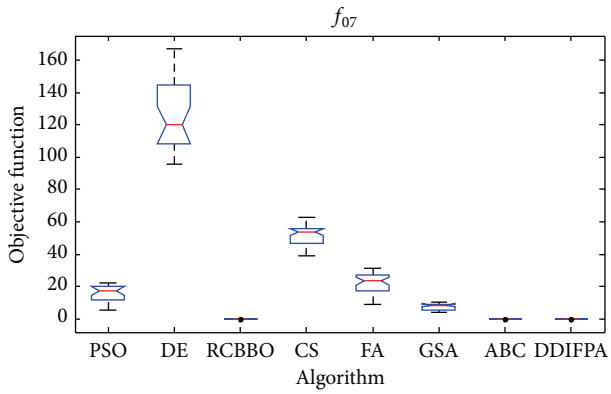


Figure 6: ANOVA tests for f_{07}.

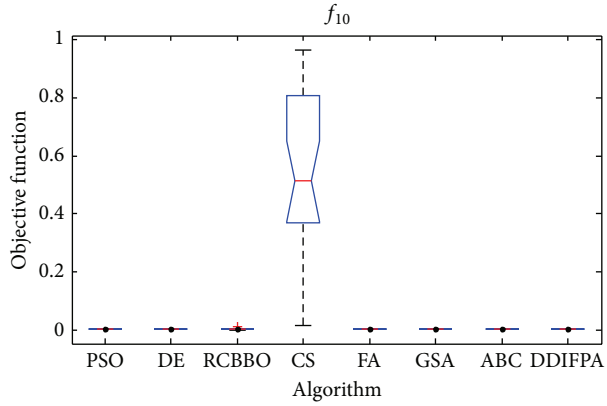


Figure 7: ANOVA tests for f_{10}.

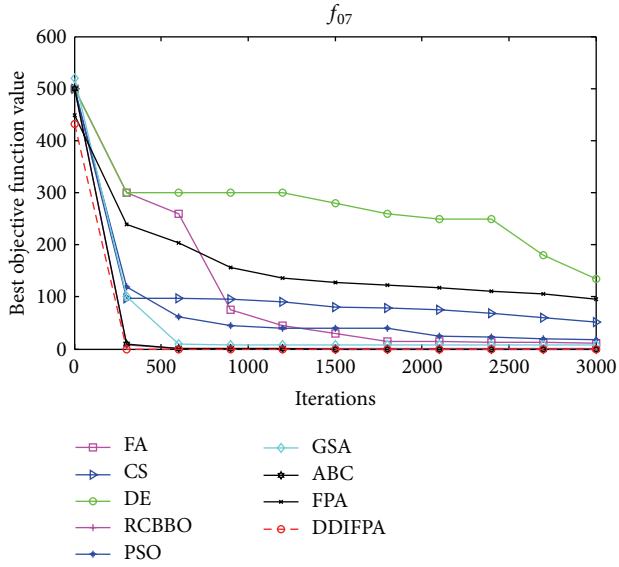


Figure 8: Fitness function curve evolution for f_{07}.

the results show that DDIFPA algorithm generally has strong global searching ability and local optimization ability, and effectively avoid the defects of other algorithms fall into local optimization. DDIFPA has improved the convergence speed and convergence precision of FPA. The experiment results show that it is an effective algorithm to solve complex functions optimization problems.

In this paper, we only consider the global optimization. The algorithm can be extended to solve other problems such as constrained optimization problems and multiobjective optimization problem. In addition, many engineering design problems are typically difficult to solve. The application of the proposed FPA with dimension by dimension improvement in engineering design optimization may prove fruitful.

Conflict of Interests

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

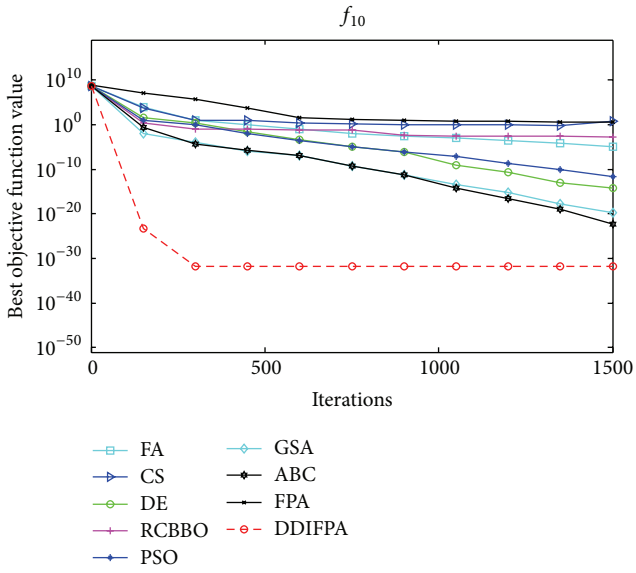


Figure 9: Fitness function curve evolution for f_{10}.

Acknowledgments

This work is supported by the National Science Foundation of China under Grant nos. 61165015 and 61463007, the Key Project of Guangxi Science Foundation under Grant no. 2012GXNSFDA053028, and the Key Project of Guangxi High School Science Foundation under Grant nos. 20121ZD008 and 201203YB072.

References

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, Mass, USA, 1992.

[2] S. Kirkpatrick Jr., C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 11, pp. 650–671, 1983.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, 1995.

[4] X.-S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers*, vol. 29, no. 2, pp. 175–184, 2013.

[5] Z. Yongquan, L. Jiakun, and Z. Guangwei, "Leader glowworm swarm optimization algorithm for solving nonlinear equations systems," *Przeglad Elektrotechniczny*, vol. 88, no. 1, pp. 101–106, 2012.

[6] A. Mucherino and O. Seref, "Monkey search: a novel metaheuristic search for global optimization," in *Proceedings of the Conference on Data Mining, Systems Analysis, and Optimization in Biomedicine*, pp. 162–173, Gainesville, Fla, USA, March 2007.

[7] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.

[8] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, no. 4, pp. 355–366, 2006.

[9] R. G. Reynolds, "Cultural algorithms: theory and applications," in *New Ideas in Optimization*, D. W. Corne, M. Dorigo, and F. Glover, Eds., pp. 367–378, McGraw-Hill, Maidenhead, UK, 1999.

[10] B. Alatas, "Chaotic harmony search algorithms," *Applied Mathematics and Computation*, vol. 216, no. 9, pp. 2687–2699, 2010.

[11] X. S. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation*, vol. 7445 of *Lecture Notes in Computer Science*, pp. 240–249, 2012.

[12] I. El-Henawy and M. Ismail, "An improved chaotic flower pollination algorithm for solving large integer programming problems," *International Journal of Digital Content Technology and Its Applications*, vol. 8, no. 3, pp. 72–81, 2014.

[13] O. Abdel Raouf, I. El-henawy, and M. Abdel-Baset, "A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles," *International Journal of Modern Education and Computer Science*, vol. 3, pp. 38–44, 2014.

[14] M. Sharawi, E. Emary, I. A. Saroit, and H. El-Mahdy, "Flower pollination optimization algorithm for wireless sensor network lifetime global optimization," *International Journal of Soft Computing and Engineering*, vol. 4, no. 3, pp. 54–59, 2014.

[15] Wikipedia article on pollination, http://en.wikipedia.org/wiki/Pollination.

[16] B. J. Glover, *Understanding Flowers and Flowering: An Integrated Approach*, Oxford University Press, 2007.

[17] X. S. Yang and S. Deb, "Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, J. R. Gonzalez, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds., vol. 284 of *Studies in Computational Intelligence*, pp. 101–111, Springer, Berlin, Germany, 2010.

[18] I. Pavlyukevich, "Lévy flights, non-local search and simulated annealing," *Journal of Computational Physics*, vol. 226, no. 2, pp. 1830–1844, 2007.

[19] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.

[20] X.-S. Yang and S. Deb, "Two-stage eagle strategy with differential evolution," *International Journal of Bio-Inspired Computation*, vol. 4, no. 1, pp. 1–5, 2012.

[21] A. Hedar, "Test function," http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedarfiles/TestGOfiles/Page364.htm.

[22] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, December 1995.

[23] 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.

[24] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[25] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[26] X. Yang S, "Firefly algorithm, Lévy flights and global optimization," in *Research and Development in Intelligent Systems XXVI*,

M. Bramer, R. Ellis, and M. Petridis, Eds., pp. 209–218, Springer, London, UK, 2010.

[27] X. S. Yang and S. Deb, "Cuckoo search via Levy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing*, pp. 210–214, IEEE Publication, New York, NY, USA, 2009.

[28] 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.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 867149, 9 pages
http://dx.doi.org/10.1155/2014/867149

Research Article

A Genetic Algorithm Based Multilevel Association Rules Mining for Big Datasets

Yang Xu,[1] Mingming Zeng,[1] Quanhui Liu,[1] and Xiaofeng Wang[2]

[1] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, China*
[2] *Institute of Computing Technology, Chinese Academy of Sciences, China*

Correspondence should be addressed to Yang Xu; xuyang@uestc.edu.cn

Received 1 July 2014; Revised 5 August 2014; Accepted 14 August 2014; Published 26 August 2014

Academic Editor: Shifei Ding

Multilevel association rules mining is an important domain to discover interesting relations between data elements with multiple levels abstractions. Most of the existing algorithms toward this issue are based on exhausting search methods such as Apriori, and FP-growth. However, when they are applied in the big data applications, those methods will suffer for extreme computational cost in searching association rules. To expedite multilevel association rules searching and avoid the excessive computation, in this paper, we proposed a novel genetic-based method with three key innovations. First, we use the category tree to describe the multilevel application data sets as the domain knowledge. Then, we put forward a special tree encoding schema based on the category tree to build the heuristic multilevel association mining algorithm. As the last part of our design, we proposed the genetic algorithm based on the tree encoding schema that will greatly reduce the association rule search space. The method is especially useful in mining multilevel association rules in big data related applications. We test the proposed method with some big datasets, and the experimental results demonstrate the effectiveness and efficiency of the proposed method in processing big data. Moreover, our results also manifest that the algorithm is fast convergent with a limited termination threshold.

1. Introduction

Exploring knowledge in *Big data* is appealing in the state of the art of data mining research [1]. Due to its high volume and complexity, resourceful domain knowledge or hidden patterns are potentially useful for human decision support [2]. It is especially for the case of multilevel association rule mining approaches to discover interesting relations among data elements with multiple levels of abstractions. Successful applications include spatial data analysis [3], emergency event analysis [4], sensor network data mining [5], and gene ontology mining. However, most existing multilevel association rules mining algorithms rely on exhaustive scans of the database to find frequent patterns across different abstraction levels, such as the most renowned Apriori algorithm [6] and Frequent Pattern tree algorithm (FP-tree) [7]. When the dataset scales up, those algorithms will suffer for the excessive computation cost and the system will retard due to its heavy scan of the large database. When the algorithms are used in big data applications, the bottleneck becomes more prominent. For example, in gene ontology domain, the annotations have rapidly grown to more than 80 million by 2012. If considering the complicated relationships between gene items in various hierarchical levels, the complexity of mining association rules in multilevels has been classified as NEXP-COMPLETE [8]. Therefore a fast multilevel association rules mining algorithm for big datasets that is scalable and can be performed in parallel computation environment becomes imperative.

In this paper, we made our initial effort toward this issue by building a genetic algorithm (GA) based heuristic method for effective multilevel association rules mining in big datasets. By taking advantage of the genetic algorithm, which can efficiently find multiple solutions concurrently in a large multidimensional problem without performing exhaustive searches, our proposed method can improve the mining performance while keeping a desired accuracy but avoiding the exhausting enumeration on association rule candidates. In summary, there are three major contributions.

First, to make our GA-based approach possible, we design a new tree like encoding schema to model the genetic candidates of multilevel association rules so that a feasible implementation of genetic operators can be built. This representation model is based on the application's domain knowledge, where each attributes (items) to be mined can be briefly illustrated as a catalog tree. Then, each valid multilevel association rule can be modeled as a subtree of the catalog tree, but each leaf node should be assigned with a binary number to indicate whether it is an antecedent or a consequent in the rule.

Next, based on the encoding schema, we build its unique GA-based operators to make the multilevel association rules mining possible: (1) individual initiation function to build the subtree with valid multilevel association rule representations; (2) crossover function that allows subtree to cross over at one of their common nodes to produce new generations; (3) selection function which is based on our designed fitness function to select stronger association rules.

As the third contribution, based on our analysis of our fitness function design, we have found that our GA-based method is adaptive and robust as its termination threshold can be reached fast with a fixed time table. We have built our experiments with different big datasets and the results manifest the performance of our design as well as the fast convergence with a limited termination threshold.

The remainder of the paper is organized as follows: Section 2 is the related work; the problem of multilevel association rules mining is formally described in Section 3; in Section 4, the genetic algorithm based multilevel association rules mining is presented; in Section 5, the performance of proposed method is evaluated on several big datasets; the conclusions are drawn in Section 6.

2. State of the Art

Many researchers have focused on multilevel association rules mining. The first branch are Apriori [6] based methods. To mine multilevel association rules, these methods are either adding all the ancestors of frequent items in the corresponding transaction database, for example, cumulate [9], or exhaustively finding all frequent items in every concept level, for example, ML-T2L1 algorithm [10] and the Level-Crossing algorithm [11]. Another branch is FP-growth [7] based methods, such as methods proposed in [12, 13]. Additionally, Cao et al. [12] and Tang et al. [13] expanded the FP-tree with ancestors of items. Wan built an approach through grouping and merging the single level association rules generated by FP-growth. Comparing with Apriori based methods, the FP-tree based methods inherit the merit of FP-tree algorithm which takes less times to scan the dataset and finds the multilevel frequent items. However, when they are utilized to analyze big data, the computational and memory cost will increase exponentially which leads to a prominent bottleneck in big data analysis.

In addition, there are some other approaches to improve the efficiency of multilevel association rules mining. Vejdani et al. proposed a method that extracted multilevel membership functions by Ant Colony Systems algorithm without specifying the actual minimum support [14]. To enhance the efficiency of computing, Mahmoudi et al. optimized Vejdani's method by fixing the functions for each item followed by computing minimum supports [15]. Wang et al. took advantages of the OLAP and data mining technology in multilevel association rules mining which brought efficiency and flexibility [16]. Besides, mining association rules with genetic algorithm (GA) [17] based methods have also been explored. The GA-based methods are able to quickly scan association rule candidate set with large amount of candidates. According to the previous research work [18], the GA-based methods can discover high-level prediction rules. This is because the GA-based methods perform a global search on association rules and can handle the data with attribute interactions better, comparing with the greedy rule induction algorithms. Previous researches have thoroughly explored single-level association rules mining with GA, such as mining single-objective rules [19] and mining multiobjective rules [20]. However, in the big data analysis context, strong association rules are always in multilevel forms, and mining multilevel association rules in big data needs more efficient methods. The GA-based multilevel association rules mining method proposed in this paper is one attempt to efficiently find multilevel association rules in big data.

3. Problem Description

The multilevel association rules mining problem can be described as follows: there are a set of items $I = \{i_1, i_2, \ldots, i_n\}$ and Γ is a catalog tree that briefly defines the multilevel categorizing relationships between items as the domain knowledge. i_1 is a parent of i_2 and i_2 is a child of i_1 if there is an edge in Γ from i_1 to i_2. We denote $\hat{\imath}$ as an ancestor of i and i as a descendant of $\hat{\imath}$ if there is a path from $\hat{\imath}$ to i in Γ. Only leaf nodes are presented in the database. An illustration of a catalog tree in a supermarket domain is shown as in Figure 1.

D is a database of transactions where each transaction T in D is a set of items such that $T \subseteq I$. Each transaction is associated with an identifier TID. Items in T are expected to be leaves in Γ. Note that a transaction T *supports* an item $x \in I$ if x is in T or x is an ancestor of some items in T. In addition, a transaction T *supports* $X \subseteq I$ if T supports every item in X.

A multilevel association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I, Y \subseteq I$, and $X \cap Y = \varnothing$. No item in Y is an ancestor of any item in X; that is, $Y \cap \text{ancestors}(x) = \varnothing$. This is because a rule of the form "$x \Rightarrow \text{ancestor}(x)$" is trivially true with 100% confidence, which is redundant. Both X and Y can contain items from any level of Γ.

The rule $X \Rightarrow Y$ holds in transaction set D with *support* s, where s is the percentage of transactions in D that support $X \cup Y$. $X \cup Y$ indicates the probability $P(X \cup Y)$. The rule $X \Rightarrow Y$ has *confidence* c in transaction set D, where c is the percentage of the transactions which support X in D that meanwhile support Y. This can be represented as the conditional probability $P(Y \mid X)$. Then,

$$\begin{aligned} \text{Support}\,(X \Longrightarrow Y) &= P\,(X \cup Y)\,, \\ \text{Confidence}\,(X \Longrightarrow Y) &= P\,(Y \mid X)\,. \end{aligned} \tag{1}$$

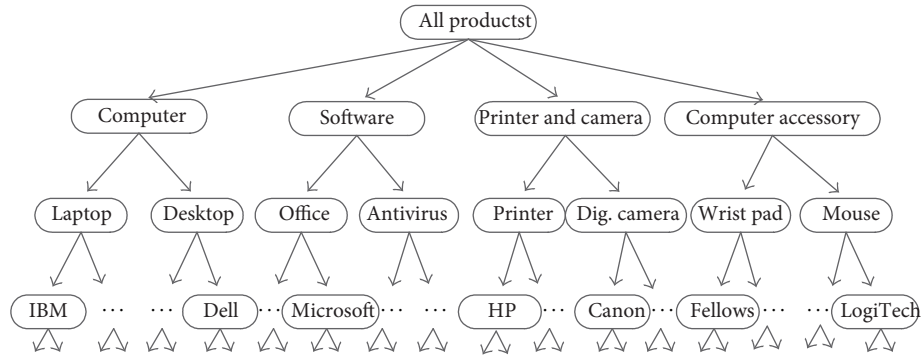


Figure 1: A catalog tree in supermarket domain.

Table 1: Association rule threshold.

Rules	Minsupp	Minconf
Computer ⇒ *Office*	5%	50%
HPprinter ⇒ *Canon Camera*	1%	50%

Table 2: Valid association rules.

Rules	Support	Confidence
Computer ⇒ *Office*	10%	66%
HPprinter ⇒ *Canon Camera*	0.8%	66%

Example. Let Γ be the catalog tree shown in Figure 1. The *Minsupp* and *Minconf* are shown in Table 1, and two of the rules on item sets are shown in Table 2. Note that the rule "Computer ⇒ Office" satisfies the minimum support (5%) and the minimum confidence (50%), but the rule "HP printer ⇒ Canon Camera" does not satisfy the minimum support (1%). Therefore, rule "Computer ⇒ Office" is considered as a valid multilevel association rule.

To evaluate the rules discovered from the multilevel abstraction, we prefer the following.

(i) Support Confidence. The rules with larger support and higher confidence are preferred, where larger support reflects that the rules are more general, and higher confidence reflects the certainty of discovered rules in the domain statistics.

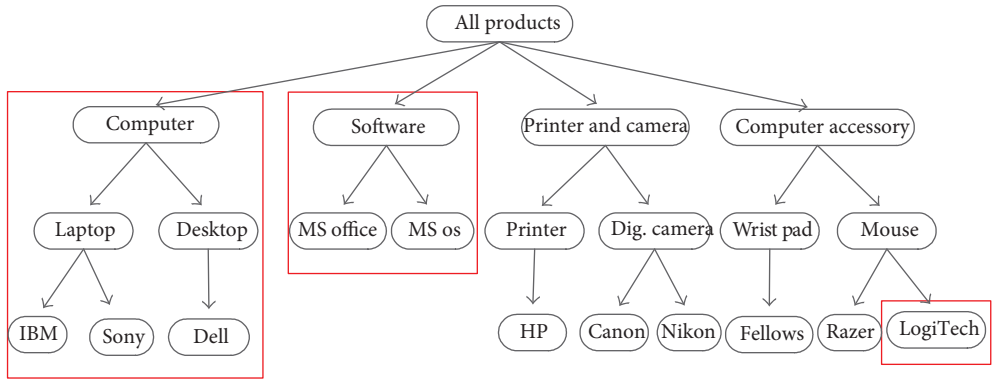
(ii) Interest. The rule in a proper level of catalog tree is preferred. Mining association rules at low levels may lead to uninteresting rules that are too trivial; that is, "IBM_ThinkPaD_R40/P4M ⇒ Symantec_Norton_Antivirus_2003." However, mining association rules at high levels usually leads to common sense, for example, "Computer ⇒ Software."

Mining of multilevel associations involves items at different levels of abstractions and its exhaustive computation complexity has been classified as NEXP-COMPLETE. A dataset that contains k items in the primitive level can potentially generate up to $2^k - 1$ primitive and nonempty frequent item sets. Particularly, in the *Bigdata* context, with the number of items in the catalogue and transactions increasing rapidly, the computational and memory consumptions of the traditional methods will be expanded exponentially. It is worthy of noting that the FP-tree algorithm enhances the efficiency of mining association rules, but it can hardly mine the multilevel association rules, especially the cross-level association rules. Thus, in big data analysis context, a novel heuristic method is imperative to mine multilevel association rules.

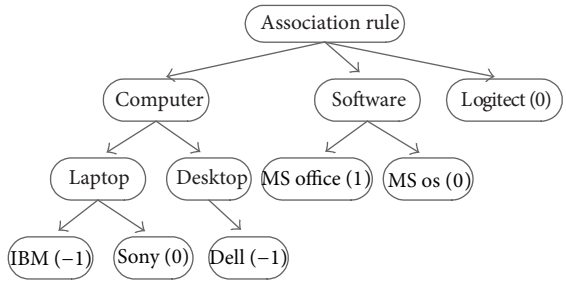
4. GA-Based Approach

Genetic algorithm is a heuristic search approach that mimics the process of natural evolution and generates solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. Its essence is an efficient, parallel, and global search method, which can automatically obtain and accumulate knowledge about the search space, and control the search space in order to achieve the optimal solution adaptively in the search process. In the traditional multilevel association rules mining algorithms, we have to generate almost all candidate items and test them against the entire database. However most of the mining process is in vain and leads to heavy computational cost. The genetic algorithm offers a novel way to solve these problems. By efficiently testing the most likely candidate items preferentially, GA-based method can control the search space and achieve the optimal solution adaptively during the association rules searching. Therefore, by taking this advantage, the association rules search space will be greatly reduced and the performance of mining method can be dramatically improved.

4.1. Encoding Scheme. When GA is applied to mine multilevel association rules, a key is to encode and automatically generate candidates of the association rules in a GA-based form. Because the classic GA-based encoding schema is not feasible to mine multilevel association rules, we propose a new category tree based encoding scheme to represent the association rule candidates. Each valid multilevel association rule can be modeled as a subtree of the catalog tree, but each leaf node should be assigned with a binary number to indicate whether it is an antecedent or a consequent in the rule. The goal of the algorithm is to find valid candidates



(a) Random pruning the catalogue tree



(b) An association rule

Figure 2: Illustration of the encoding schema.

by the evolution of subtrees. The structure of the GA-based encoding subtree can defined as follows:

```
Class TreeNode {int  Id;
                string attributeName;
                int  assignedValue;
                TreeNode parentNode;
                List ⟨TreeNode⟩ childList; }.
```
(2)

In this representation, every leaf represents a commodity and is assigned to a value of 0, 1, or −1 as shown in Figure 2. The antecedents of the association rule can be expressed by the commodities assigned to 0 and the consequents can be expressed by the ones assigned to 1. The commodities assigned to −1 do not join the association rules. With randomly pruning the catalogue tree and assigning the values of the leaves, we can initiate the subtrees with the first generation of the association rules. For example, the items in red rectangle of Figure 2(a) are regarded as the children of the association rules tree. The multilevel association rules can be generated and polished by the processes of roll-up, mutation, crossover, and selection operator.

Example. As shown in Figure 2(b), the leaves assigned to 0 represent the antecedents of the association rule and the leaves assigned to 1 represent the consequents, so the association rule can be represented as (Sony ∩ MS os ∩ Logitech) ⇒ (MS office). The rule implies that people who buy Sony laptops and Logitech mouses, MS os, will be likely to buy MS office software.

4.2. Genetic Operators. Initially, according to the given catalog tree, we will randomly prune the catalog tree to get the subtrees as the children of the association rules trees. Then we randomly assign the leaves of each association rule tree to −1, 0, or 1 and make sure that the association rule tree maintains 0 and 1 at the same time. In the same way, we can get the appropriate number of the initial population.

Selection operator defines how to choose the individuals that will create the offsprings for the next generation. The selection operator is based on the fitness function that the offsprings with high fitness will have higher probabilities to be selected. In this paper, we use "roulette wheel" [21] selection, and the higher the fitness of an individual is, the more likely it is to be selected to reproduce.

After the high fitness individuals are selected, the crossover operator can be applied. This function allows a pair of selected subtrees to cross over at one of their randomly chosen common nodes to produce new generations so as to avoid generating invalid rules. In particular, only attribute values will be exchanged if only leaf nodes are crossed over. The crossover of the root node is prohibited because no new rules will be produced. An example of crossover process in a real domain is illustrated as in Figure 3.

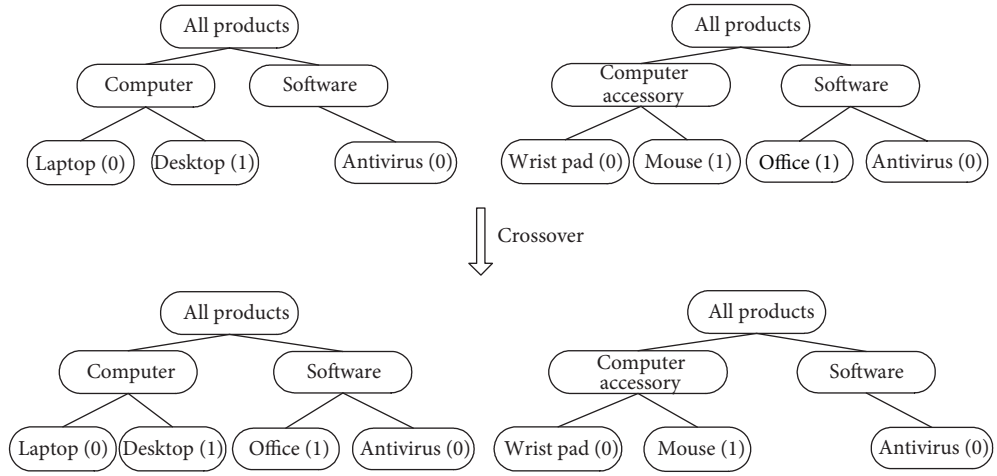
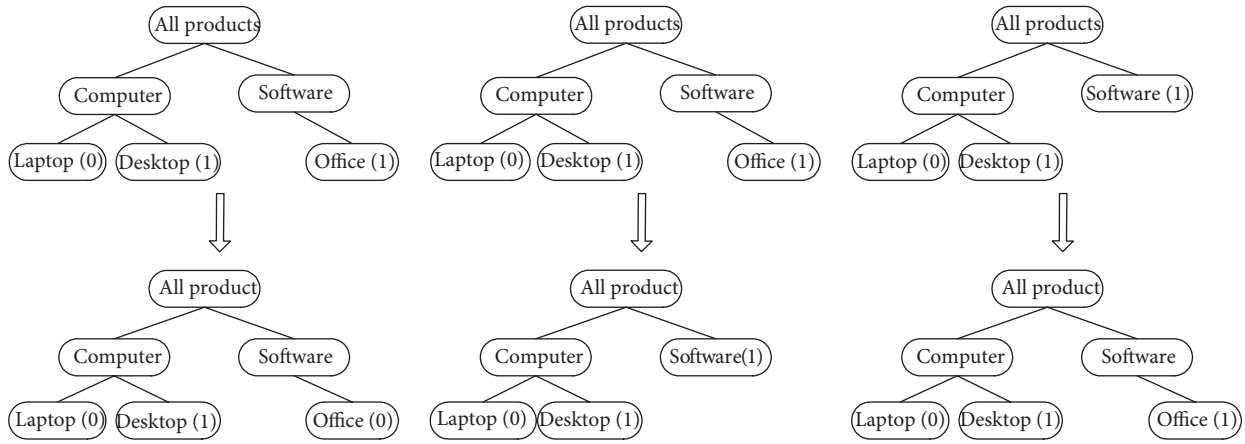


Figure 3: Illustration of the crossover.



(a) Alter attribute value (b) Prune subtree (c) Add subtree

Figure 4: Illustration of the mutation.

Mutation operator plays an important role in maintaining the diversity of the population during the mutations. In our schema, we define three types of mutation operators:

(1) randomly choose a leaf node and assign with an alternative attribute value;

(2) randomly choose a nonroot node and prune its subtree;

(3) randomly choose a nonroot node and add a subtree to the node.

An illustration of the mutation operators is shown as in Figure 4, where the node with thick border indicates that it is the one to be chosen and mutated.

4.3. Fitness Function. The fitness function plays an important role in our GA schema. It is used either to evaluate the offsprings that will be selected into the next generation or to act as the terminate condition, when there have been enough association rules with higher fitness values that are more than the predefined threshold. To build the fitness function, we have to combine the *support* and *confidence* attributes, which are necessary to described an association rule in domain, in our fitness definition. Therefore, the fitness of an association rule $A \Rightarrow B$ is defined as

$$F(A \Longrightarrow B) = \alpha \times \text{support}(A \Longrightarrow B) + \beta \times \text{confidence}(A \Longrightarrow B). \quad (3)$$

Parameters α and β are the important factors to balance the weight of the *support* and *confidence* in the fitness function, and $\alpha + \beta = 1$. To mine valid association rules from the big data base with our GA approach, the threshold of the fitness function has to be predefined. As the threshold is relevant to the *support* and *confidence* attributes, we should set the thresholds of minimum support *min_sup* and the minimum confidence *min_conf* for the algorithm. In our approach, other than uniformly using the same thresholds for all levels, we use different *min_sup* for different levels of association rules. The deeper the level is, the smaller the corresponding thresholds will be. Furthermore, the more leaf

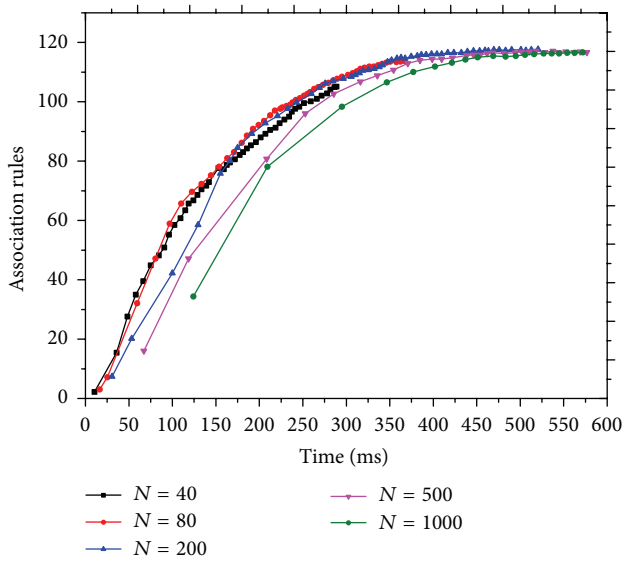


Figure 5: With a limited population and a limited time, most valid association rules could be mined in dataset 1.

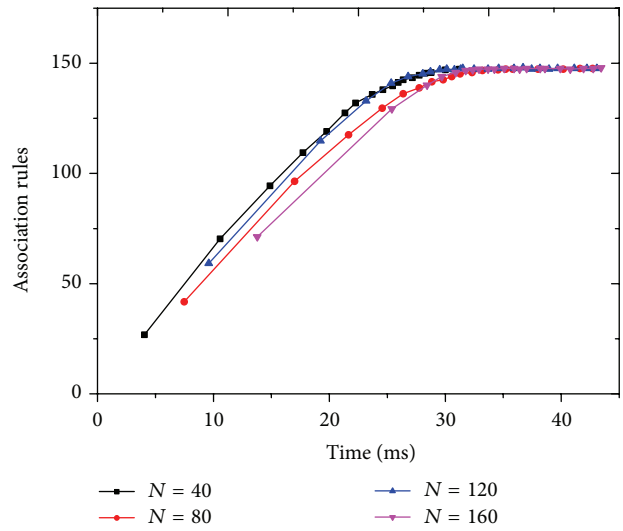


Figure 6: With a limited population and a limited time, most valid association rules could be mined in dataset 2.

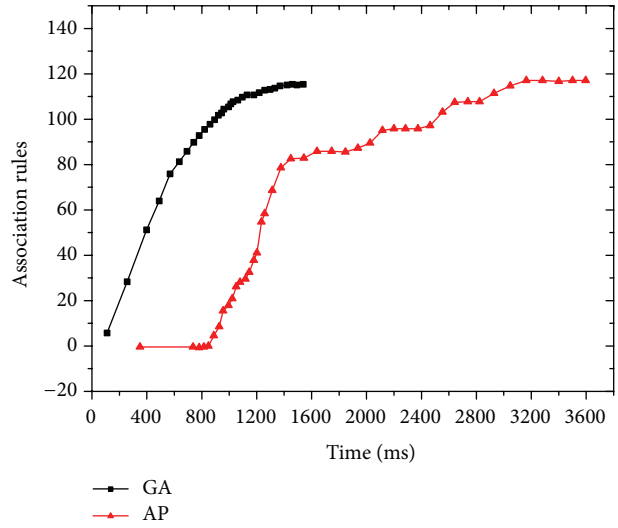


Figure 7: Number of association rules mined from 4000 transactions in dataset 1.

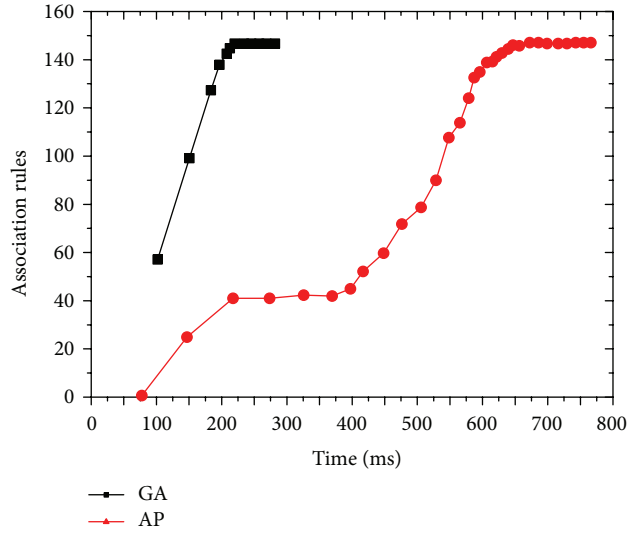


Figure 8: Number of association rules mined from 4000 transactions in dataset 2.

nodes the ancestor has, the higher *min_sup* of the ancestor will be [13].

5. Experimental Results

In this section, we build various experiments to analyse the performances of our design. We briefly use two different transaction databases to mine the multilevel association rules: "Dataset 1" from University of Regina (http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/datasets.php/) and "Dataset 2" from California State University Los Angeles (http://www.calstatela.edu/centers/hipic/contents/researchy/cloudComputing/files/market/).

We also use the classic Apriori algorithm as the benchmark to compare with our GA-based algorithm. All the experiments run on a PC with Core i5 CPU and 4 GB RAM. For the default settings, we set the size of the population in our GA-based algorithm for dataset 1 as 200 while for dataset 2 as 200. The initial *min_sup* is 0.01 and *min_conf* is 0.5.

In the first experiment, we test whether valid association rules can be mined with a fixed number of initial generations and in a limited of period. In dataset 1, the initial generation size varies from 40 to 1000, and, in dataset 2, it varies from 50 to 160. The results are shown in Figures 5 and 6. We can conclude that if the population is too small, the performance of the GA-based algorithm will be similar to the random algorithm. But if the population is too large, although we can obtain enough association rules quickly, the

Table 3: Number of association rules mined from 1000 and 2000 transactions in two datasets.

Results from dataset 1								
Number of transactions	Type	Time (s)						
		62.5	125	250	400	1000	2000	2500
1000	GA	32.8	67.7	94.4	97	97	97	97
	AP	0	0	0	1	56	98	98
2000	GA	7	30.4	66.9	92.3	116.6	116.6	116.6
	AP	0	0	0	0	35	97	118
Results from dataset 2								
Number of transactions	Type	Time (s)						
		15	20	40	100	200	400	600
500	GA	57.6	97.2	150.9	151	151	151	151
	AP	0	0	0	46.4	130	151	151
1000	GA	0	57.3	130.9	147	147	147	147
	AP	0	0	0	39	74	141	147

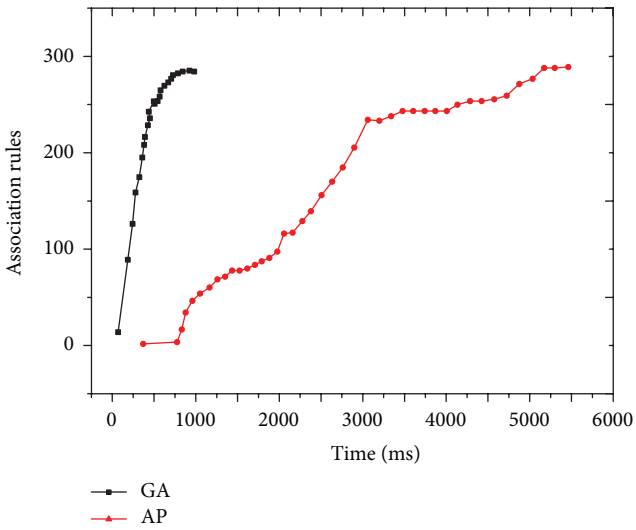


Figure 9: Mining efficiencies of GA-based and Apriori algorithm with different *min_sup* and *min_conf* in dataset 1.

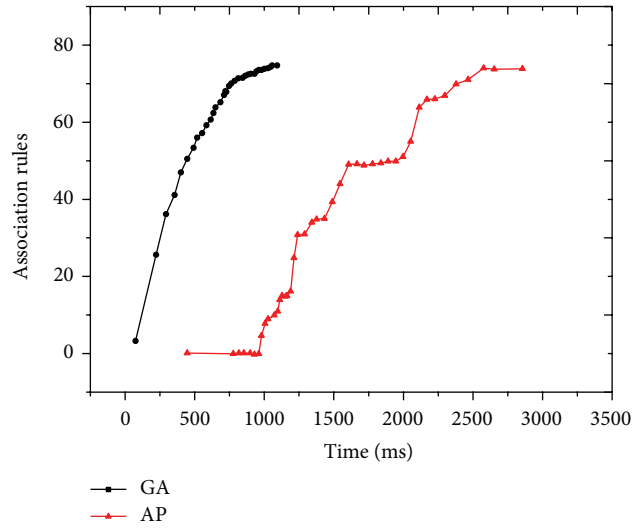


Figure 10: Mining efficiencies of GA-based and Apriori algorithm with different *min_sup* and *min_conf* in dataset 2.

computational complexity grows fast. However, as we can see, in both datasets, there is a good balance that, with a limited population and a limited time period, most valid association rules have been mined. Therefore we select 200 and 120 as the default population for dataset 1 and dataset 2, respectively, which performs well in our algorithm.

In the rest of this section, we compare the efficiency of our GA-based algorithm with the Apriori algorithm in the two different datasets. In Figures 7 and 8, we fixed 4000 transactions in each dataset. As it has shown, with the progress of the time step, GA-based approach can find valid association rules much quicker in both datasets. Although as an exhaustive approach, Apriori algorithm may be able to find a few more valid rules than ours if there is no time limit. However, in most big data applications, the system response time is a critical criterion of its performance and our approach is more valuable to obtain most association rules in a short period of time. In Table 3, we presented more detailed results when we picked 1000 and 2000 transaction records from each dataset. As the time goes by, we reach a consistent conclusion that our GA-based approach is more capable of finding valid association rules than the exhaustive approach.

When we changed the thresholds of the minimal support and confidence for the valid association rules, the output for both GA-based algorithm and the Apriori algorithm is changed. As shown in Figure 9, in dataset 1, we reduced the *min_sup* to 0.005 and the *min_conf* to 0.25. In Figure 10, we increased the *min_sup* to 0.02 and the *min_conf* to 0.75 for dataset 2. In both experiments, the GA-based algorithm finds more association rules in a short of period than the Apriori algorithm, no matter what the multilevel association thresholds are.

In the next group of experiments, we tested how valuable the mined association rules are from either the GA-based algorithm or the Apriori. To measure its value, we use the same formula of fitness function and set $\alpha = 0.9$, $\beta = 0.1$.

Table 4: Average fitness of association rules mined from 1000 and 2000 transactions in two datasets.

Results from dataset 1									
Number of transactions	Type	Time (s)							
		20	62.5	125	250	400	1000	2000	2500
1000	GA	0.14	0.17	0.17	0.16	0.16	20	62.5	125
	AP	0	0	0	0	0.13	0.15	0.16	0.16
2000	GA	0	0.21	0.24	0.23	0.22	0.20	0.20	0.20
	AP	0	0	0	0	0	0.16	0.20	0.20
Results from dataset 2									
Number of transactions	Type	Time (s)							
		15	20	40	100	200	400	600	1000
500	GA	0.35	0.36	0.35	0.35	0.35	0.35	0.35	0.35
	AP	0	0	0	0.29	0.36	0.35	0.35	0.35
1000	GA	0	0.36	0.37	0.35	0.35	0.35	0.35	0.35
	AP	0	0	0	0.26	0.28	0.36	0.35	0.35

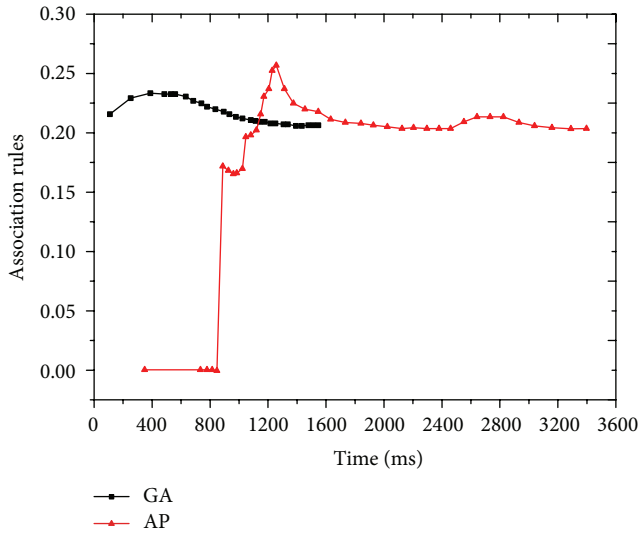


Figure 11: Average fitness of association rules mined from 4000 transactions in dataset 1.

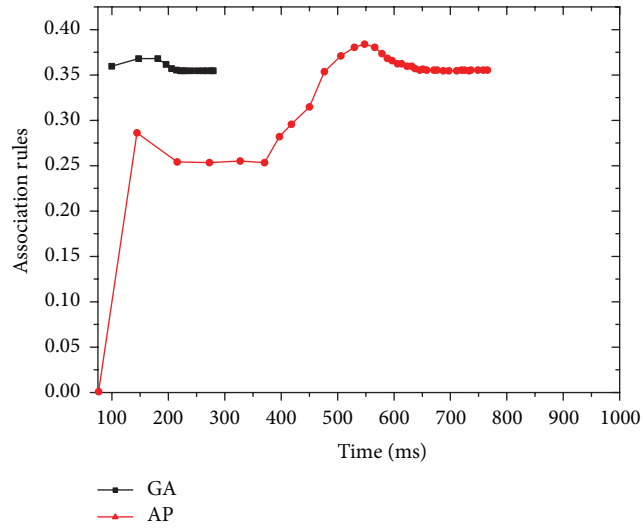


Figure 12: Average fitness of association rules mined from 4000 transactions in dataset 2.

The experiment results in different datasets with the 4000 transactions are shown in Figures 11 and 12, and the results in datasets with different number of transactions are shown in Table 4.

Consistent with the conclusions above, with the progress of the time horizon, GA-based approach can find high valuable association rules much quicker than Apriori in both datasets.

6. Conclusion and Future Work

In this paper, we have presented a novel genetic based algorithm to mine multilevel association rules in big date sets. By utilizing the application domain knowledge that could be briefly explained as a catalog tree, we introduce a special subtree based encoding schema to make the GA-based algorithm possible. In addition, we personalized the initiation, crossover and mutation functions for the tree-based genetic operators. Based on our simulations and the experiment results, we can see that by building the dynamic fitness function from the multilevel support and confidence threshold, this algorithm is adaptive and convergent. Moreover, we test its performance in different databases and the algorithm performs better than the classic Apriori algorithm with faster and more accurate mining the high quality multilevel association rules.

Although we have proved that our GA-based approach is capable of dealing with some key challenges in the multilevel association rule mining in big databases, we leave many of the others in the future. Firstly, our approach is only good in the domains that the items in the association rules can be organized as a catalog tree. But when it is applied in some other domain with unstructured item sets, our basic design does not match. Second, our GA-based approach should be encoded in a distributed and parallel computation

environment so as to optimize its performance. Moreover, deployment in real domain is the key to evaluate our approach and polish the algorithm for better performance.

Conflict of Interests

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

Acknowledgments

This research has been sponsored in part by National Natural Science Foundation of China nos. 61370151 and 61202211, National Science and Technology Support Program of China 2012BAI22B05, and Huawei Research Foundation YB2013120141.

References

[1] E. Dumbill, "Making sense of big data," *Big Data*, vol. 1, no. 1, pp. 1–2, 2013.

[2] K. Cukier and V. Mayer-Schoenberger, "The rise of big data: how it's changing the way we think about the world," *Foreign Affairs*, vol. 92, no. 3, 2013.

[3] B. Petelin, I. Kononenko, V. Malačič, and M. Kukar, "Multi-level association rules and directed graphs for spatial data analysis," *Expert Systems with Applications*, vol. 40, no. 12, pp. 4957–4970, 2013.

[4] B. Fan and J. Luo, "Spatially enabled emergency event analysis using a multi-level association rule mining method," *Natural Hazards*, vol. 67, no. 2, pp. 239–260, 2013.

[5] D. Han, Y. Shi, W. Wang et al., "Research on multi-level association rules based on geosciences data," *Journal of Software*, vol. 8, no. 12, pp. 3269–3276, 2013.

[6] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference: Very Large Data Bases*, vol. 1215, pp. 487–499, 1994.

[7] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *SIGMOD Record*, vol. 29, no. 2, pp. 1–12, 2000.

[8] G. Alterovitz, M. Xiang, M. Mohan, and M. F. Ramoni, "GO PaD: the gene ontology partition database," *Nucleic Acids Research*, vol. 35, supplement 1, pp. D322–D327, 2007.

[9] R. Srikant and R. Agrawal, "Mining Generalized Association Rules," in *VLDB*, vol. 95, pp. 407–419, 1995.

[10] J. Han and Y. Fu, "Discovery of multiple-level association rules from large databases," in *Proceedings of the 21st International Conference: Very Large Data Bases*, vol. 95, pp. 420–431, 1995.

[11] R. S. Thakur, R. C. Jain, and K. R. Pardasani, "Mining level-crossing association rules from large databases," *Journal of Computer Science*, vol. 2, no. 1, 2006.

[12] H. Cao, Z. Jiang, and Z. Sun, "Fast mining algorithm for multi-level association rules based on FP-tree," *Computer Engineering*, vol. 19, no. 25, 2007.

[13] H. Tang, M. Wu, and Y. He, "Improved multilevel association rule mining algorithm," *Computer Engineering*, vol. 16, no. 16, 2011.

[14] E. Vejdani, F. Saadatmand, M. Niazi, and M. H. Yaghmaee, "Extracting membership functions by ACS algorithm without specifying actual minimum support," in *Proceeding of the International Conference on Multimedia Computing and Information Technology (MCIT '10)*, pp. 13–16, Sharjah, United Arab Emirates, March 2010.

[15] E. V. Mahmoudi, E. Sabetnia, M. N. Torshiz, M. Jalali, and G. T. Tabrizi, "Multi-level fuzzy association rules mining via determining minimum supports and membership functions," in *Proceedings of the 2nd International Conference on Intelligent Systems, Modelling and Simulation (ISMS '11)*, pp. 55–61, Kuala Lumpur, Malaysia, January 2011.

[16] Y. Wang, L. Yu, Q. Wen et al., "Improved multi-level association rule in mining algorithm based on a multidimensional data cube," in *Proceedings of the Consumer Electronics, Communications and Networks (CECNet '13)*, pp. 355–358, 2013.

[17] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2, pp. 95–99, 1988.

[18] A. A. Freitas, *A Survey of Evolutionary Algorithm s for Data Mining and Knowledge Discovery*, Advances in Evolutionary Computing Natural Computing Series, 2003.

[19] M. R. Kumar and D. K. Iyakutti, "Application of genetic algorithms for the prioritization of association rules," *IJCA Special Issue on Artificial Intelligence Techniques-Novel Approaches and Practical Applications*, pp. 1–3, 2011.

[20] M. K. Gupta and G. Sikka, "Association rules extraction using multi-objective feature of genetic algorithm," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 2, San Francisco, Calif, USA, October 2013.

[21] C. R. Houck, J. Joines, and M. G. Kay, "A genetic algorithm for function optimization: a matlab implementation," *NCSU-IE TR*, vol. 95, no. 9, 1995.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 395263, 8 pages
http://dx.doi.org/10.1155/2014/395263

Research Article

An Optimized Classification Algorithm by Neural Network Ensemble Based on PLS and OLS

Weikuan Jia,[1] Dean Zhao,[1] Yuyang Tang,[1] Chanli Hu,[1] and Yuyan Zhao[2]

[1] *School of Electrical and Information Engineering, Jiangsu University, Zhenjiang 212013, China*
[2] *Changzhou College of Information Technology, Changzhou 213164, China*

Correspondence should be addressed to Dean Zhao; dazhao@ujs.edu.cn

Received 19 May 2014; Accepted 30 June 2014; Published 20 July 2014

Academic Editor: Shifei Ding

Using the neural network to classify the data which has higher dimension and fewer samples means overmuch feature inputs influence the structure design of neural network and fewer samples will generate incomplete or overfitting phenomenon during the neural network training. All of the above will restrict the recognition precision obviously. It is even better to use neural network to classify and, therefore, propose a neural network ensemble optimized classification algorithm based on PLS and OLS in this paper. The new algorithm takes some advantages of partial least squares (PLS) algorithm to reduce the feature dimension of small sample data, which obtains the low-dimensional and stronger illustrative data; using ordinary least squares (OLS) theory determines the weights of each neural network in ensemble learning system. Feature dimension reduction is applied to simplify the neural network's structure and improve the operation efficiency; ensemble learning can compensate for the information loss caused by the dimension reduction; on the other hand, it improves the recognition precision of classification system. Finally, through the case analysis, the experiment results suggest that the operating efficiency and recognition precision of new algorithm are greatly improved, which is worthy of further promotion.

1. Introduction

Neural network (NN) is based on the intelligent computing, which makes use of the computer network to imitate the biological neural network. It shows powerful function in dealing with nonlinear process and large-scale computing. The essence of classification system is regarded as an input/output system. The transformational relations are mainly about three aspects which includes numerical fitting, logical reasoning, and fuzzy transforming; all of these can be well expressed by neural network. The research of neural network classification has obtained widespread application in many fields [1].

However, while dealing with the small sample, it is a big challenge to use neural network to classify. Overmuch feature inputs will influence the structure design of the neural network and restrict the operating efficiency. However, fewer numbers of samples easily lead to incomplete training or overfitting, which restricts the final classification precision. Thus, in this paper, we introduce the feature dimension reduction and ensemble learning into neural network algorithm.

The principle of feature dimension reduction theory [2, 3] is to extract most useful feature from many complex feature variables, which eliminates the influence of repetition or correlation factors. It means that reducing the feature dimension as much as possible on the premise of solving the problems normally. Because feature dimension reduction will cause the loss of information, but it can't influence the problem solving. To reduce the inputs of the neural network, we will design and simplify structure conveniently. Optimized neural network algorithm based on feature dimension reduction has obtained gratifying achievement in many fields [4–6]. The traditional feature extraction algorithm which depends on the measure of variance (such as principal component analysis, factor analysis, etc.) is hard to get ideal low dimensional data for small sample, which is with higher dimension and fewer samples. In this paper, we adopt partial least square (PLS) to feature dimension reduction algorithm [7, 8], which can take

the level of correlation feature variables and the dependent variables into consideration and obtain the relatively ideal low-dimensional data with strong interpretation. It has a unique advantage for dealing with small sample data. Using PLS to optimize the classification algorithm and neural network, it has got some progress [9–11].

Hansen and Salamon first proposed neural network ensemble learning [12]. They proved the performance of a series of integrated neural network is better than the best single neural network by the experiment; the generalization ability and recognition precision of multiple classifier integration system have been improved obviously. In order to improve the classification precision, the method of multiple classifiers integration has been considered. Meanwhile, the ensemble learning reaches a climax, a series of concepts and algorithms have been proposed [13–15], and applied in many fields. The history of neural network has more than 70 years; hundreds of models are proposed and different models have their own advantage in dealing with different problems. However, it is not perfect for the nature of neural network. In this paper, choose BP, RBF, Elman three neural networks as subclassifier to study. The integrator design is the key point of ensemble optimization algorithm, which is to determine the final recognition precision of integrated classification system. The key of ensemble learning is how to determine the weights of each subclassifier; it attracts many scholars' research interests [16–18]. This study is using ordinary least squares (OLS) principle to ensemble learning for classification system, through establishing the regression equation to determine the weights of three subclassifiers.

Brief and to the point, in view of the nature of small sample and neural network, in this paper, an ensemble optimized classification algorithm by neural network based on PLS and OLS is proposed. The new algorithm by PLS dimension reduction is to improve the operating efficiency and the recognition precision through OLS ensemble learning. The new algorithm aims at providing a high efficiency and precision classification system. Finally, through the test of two data sets, one from the actual production, the other one form UCI standard data sets, the experimental results suggest that the new algorithm is valid and worthy of further popularization and application.

2. PLS Dimension Reduction Algorithm

Partial least squares is the characteristic development of ordinary least squares (OLS), its basic idea is that the feature variable matrix X is reduced; at the same time, it gives consideration to the correlation of the dependent variable matrix Y. Suppose there are n feature variables, $x_1, x_2, \ldots, x_n$, p dependent variables, $y_1, y_2, \ldots, y_p$, after preprocessing, matrix X is reduced into

$$X = TP^T + E. \tag{1}$$

Here, T is the score matrix, P is the load matrix, and E is the residual error matrix. Matrix multiplication of TP^T can be expressed as the sum products of score vector t_i (the ith column of matrix T) and load vector p_i (the ith column of matrix P); then the above formula can be written as

$$X = \sum_{i=1}^{n} t_i p_i^T + E, \quad i = 1, 2, \ldots, n. \tag{2}$$

Similarly, matrix Y is decomposed into

$$Y = UQ^T + F. \tag{3}$$

Here, U is the score matrix, Q is the load matrix, and F is the residual error matrix. Matrix multiplication of UQ^T can be expressed as the sum products of score vector u_j (the jth column of matrix U) and load vector q_j (the jth column of matrix Q); then the above formula can be written as

$$Y = \sum_{j=1}^{p} u_j q_j^T + F, \quad j = 1, 2, \ldots, r. \tag{4}$$

PLS analysis separately extracted the scores t and u form corresponding X and Y; they are the linear combination of feature variables and dependent variables. And both scores satisfied the maximum load of variation information of feature variables and the dependent variables; the covariance between them is the largest. Establishment of the regression equation is

$$u_j = b_k t_i. \tag{5}$$

Here, b_k is regression coefficient; the formula can be expressed in matrix form as

$$Y = BX. \tag{6}$$

Here, B is coefficients matrix;

$$B = W\left(P^T W\right)^{-1} Q^T. \tag{7}$$

Here, W is the weights matrix.

PLS aims to each dimension iterative calculation by using each other's information; each iteration continuously according to residual information of X, Y to adjust t_i, u_j for the second round extracted, until the residual matrix element of absolute value approximate to zero. The precision satisfied the requirements; then the algorithm stops. In the iteration process, t_i, u_j can maximize the expression of variance of X and Y simultaneously.

PLS regression does not need to use all the components to establish the regression equation, which just need to select the front l components ($0 \le l \le n$) and then get better regression equation. Generally, K-fold cross-validation method is used to calculate prediction residual sum of squares and determine the number of components extracted, reaching the purpose of dimension reduction.

3. Neural Network Ensemble Optimization

3.1. Subclassifier (Individual Neural Network). Select BP, RBF, and Elman regression three different types of neural networks

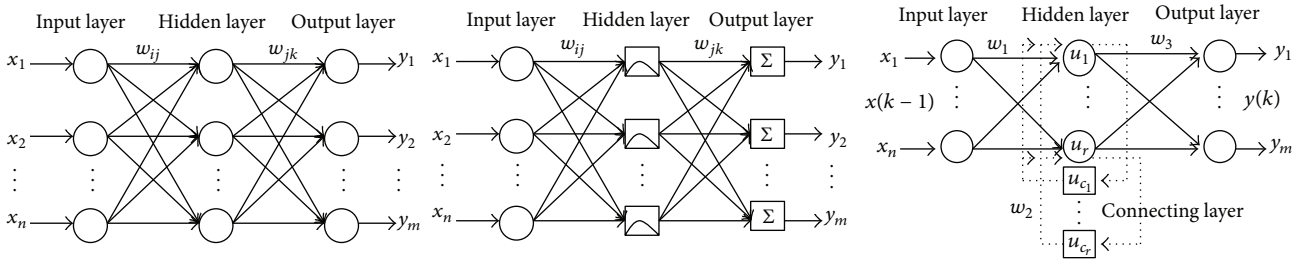


Figure 1: The topology structures of BP, RBF, and Elman neural network.

as subclassifier in this paper. These three subclassifiers, to a certain extent, play a complementary role. The topology structure of three neural networks is as shown in Figure 1.

BP neural network is denoted by subclassifier I; it adopts the error back propagation learning. A three-layer BP neural network can imitate any nonlinear function with arbitrary precision. It has good adaptive robustness and generalization ability. With the expanding scope of application, the defects of BP neural network emerge gradually, which includes fixed learning rate that caused long training time, easily trapped in local minimum, and the overfitting illusion, hidden layer, and its uncertain neurons numbers. All these above become the inherent defects of BP network.

RBF neural network is denoted by subclassifier II; it is a kind of feed forward model, which can adjust the hidden layer adaptively in the training process according to the specific problems. The distribution of hidden layer units is determined by the training sample's capacity, category, and distribution. It can dynamically determine the hidden layer, even its center and width; meanwhile, the convergence speed is fast. The biggest advantage is to use linear learning algorithm to complete the work done by nonlinear learning algorithm and maintain high precision of nonlinear algorithm; it has the characteristics of the best approximation and global optimal. It can use the sum of local approximation to attain the global approximation of training data, so using the sum of low-order local approximation can be finished the training data fitting. During the training process, it is easily appear over-fitting phenomenon, low learning ability, and so on. All of these insufficient will lead the bigger prediction error, further influence the recognition precision of RBF neural network.

Elman regression neural network is denoted by subclassifier III; it is a kind of feedback model that adds a context layer based on hidden layer of BP neural network. The context layer is regarded as a delay operator; it can delay and store the output of hidden layer and achieve the memory. That means the system has the ability of adapting the time-varying dynamic characteristics and strong global stability. Thus, Elman neural network optimization is always based on BP neural network, naturally; it inevitably inherits the inherent defects of BP neural network which will lead to the unsatisfactory operating efficiency.

3.2. Construct of Neural Network Ensemble Algorithm. Consider the complex pattern classification problems; single classifier is usually difficult to achieve the ideal recognition precision and has some of its own deficiencies. The generalization ability and recognition precision of multiple classifier integrated system will be more outstanding obviously. Each classifier is regarded as subclassifier of integrated system; the main idea of ensemble learning is mainly about using many subclassifiers to solve the same problem and integrating the outputs of each subclassifier, finally obtaining the results of integrated classification system. The purpose is to improve the ability of generalization and recognition of the learning algorithm.

Neural network ensemble is a pattern classification system, which integrates a series of single neural networks. The performance of the integrated system is better than any single neural network. The main purpose of neural network ensemble is to improve the recognition precision of classification system. Obviously, determining the weights of each subclassifier is the key of ensemble algorithm. The main task of neural network ensemble algorithm is seeking the weights of each subclassifier based on the characteristics of subclassifier and reducing the recognition error of integrated classification system.

There are three subclassifiers for ensemble learning, in order to deal with the small sample, adopt dimension reduction before ensemble learning and optimizing original data, and then we further attempt to establish neural network ensemble learning models. Figure 2 presents the established ensemble learning model. Its operating principle consists of dimension reduction for small sample by PLS algorithm, getting the low-dimensional data as the input of each subclassifier and regarding the output of each subclassifier as the input of integrator. The outputs of sub-classifiers weighted learning by integrator, finally the system gets relatively optimal classification results.

Figure 2 presents the basic flow chart of neural network ensemble algorithm; this study applies three subclassifiers, for example. Respectively, we use three subclassifiers to recognize the pending sample data; the cognition results of subclassifiers I are denoted by A, the cognition results of subclassifiers II are denoted by B, and the cognition results of subclassifiers III are denoted by C. Three subclassifiers are independent of each other, so are the results.

Respectively, the weights of three subclassifiers are denoted by w_1, w_2, and w_3, the output of integrated system is denoted by Y, and the value of Y is obtained by sum

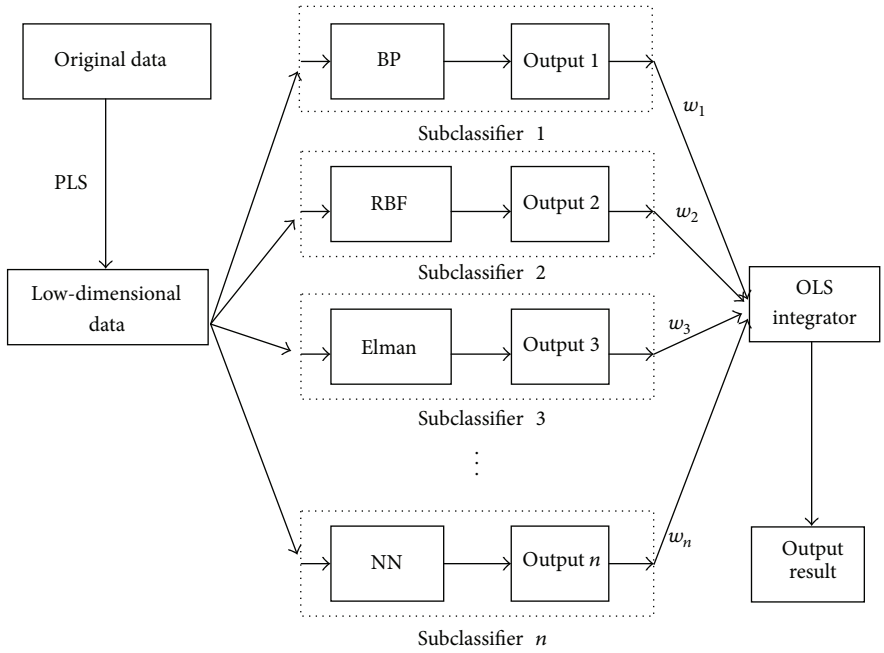


Figure 2: Optimized classification algorithm assembled with neural network.

of weighted A, B, C, and the ensemble learning model is established as

$$Y = Aw_1 + Bw_2 + Cw_3. \tag{8}$$

Model (8) reflects the follow discussion obviously. While $w_1 = 1$, $w_2 = 0$, $w_3 = 0$, it means that only subclassifier I works, the system output results are determined by the subclassifier I. While $w_1 = 0$, $w_2 = 1$, $w_3 = 0$, it means that only subclassifier II works; the system output results are determined by the subclassifier II. While $w_1 = 0$, $w_2 = 0$, $w_3 = 1$, it means that only subclassifier III works, the system output results are determined by subclassifier III. The optimized integrated system, that is, find out the optimal weights of each sub-classifier, and then makes the recognition results of model (8) to achieve optimal state.

3.3. Optimized Weights. The study of neural network ensemble learning, that is, the optimized the model (8), is to determine the value of three individual neural networks' weights w_1, w_2, w_3. We use the thought of multiple regressions to establish ternary regression equation for the output of three subclassifiers and determine the optimal estimated value of weights w_1, w_2, w_3 by OLS algorithm.

Suppose the recognized results of three subclassifiers are denoted by a_i, b_i, c_i, respectively, for the ith training sample; the output of integrated system is denoted by y_i and the actual value of the sample is y_i^o, where $i = 1, 2, \ldots, n$. By the actual value Y^o of training sample, the ternary regression equation is established as follows:

$$Y = \theta + \alpha A + \beta B + \gamma C + \varepsilon, \quad \varepsilon \sim N\left(0, \sigma^2\right). \tag{9}$$

Here, α, β, γ present the partial regression coefficient; using the method of maximum likelihood estimation to estimate unknown parameters α, β, γ, θ, it needs the minimum residual error value Q:

$$Q = \sum_{i=1}^{n} \left(y_i^o - y_i\right)^2. \tag{10}$$

Satisfy, namely,

$$\min Q = \sum_{i=1}^{n} \left(y_i^0 - \theta - \alpha a_i - \beta b_i - \gamma c_i\right)^2. \tag{11}$$

If we make Q minimum, α, β, γ, θ should satisfy the following equation sets:

$$\begin{aligned}
\frac{\partial Q}{\partial \alpha} &= \sum_{i=1}^{n} \left(y_i^0 - \theta - \alpha a_i - \beta b_i - \gamma c_i\right) a_i = 0, \\
\frac{\partial Q}{\partial \beta} &= \sum_{i=1}^{n} \left(y_i^0 - \theta - \alpha a_i - \beta b_i - \gamma c_i\right) b_i = 0, \\
\frac{\partial Q}{\partial \gamma} &= \sum_{i=1}^{n} \left(y_i^0 - \theta - a_i\alpha - \beta b_i - \gamma c_i\right) c_i = 0, \\
\frac{\partial Q}{\partial \theta} &= \sum_{i=1}^{n} \left(y_i^0 - \theta - a_i\alpha - \beta b_i - \gamma c_i\right) = 0.
\end{aligned} \tag{12}$$

Set

$$X = \begin{bmatrix} 1 & a_1 & b_1 & c_1 \\ 1 & a_2 & b_2 & c_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & a_n & b_n & c_n \end{bmatrix}_{n\times 4}, \qquad \eta = \begin{bmatrix} \theta \\ \alpha \\ \beta \\ \gamma \end{bmatrix}_{4\times 1}. \tag{13}$$

The equation set (12) translates into matrix form

$$X^T X\eta = X^T Y^o. \tag{14}$$

Obtain the least squares estimate of coefficient matrix η:

$$\eta = \left(X^T X\right)^{-1} X^T Y^o. \tag{15}$$

By the coefficient matrix η, get the values of regression coefficients α, β, γ, and the values of weights w_1, w_2, and w_3 can be further obtained:

$$w_1 = \frac{\alpha}{\alpha+\beta+\gamma}, \qquad w_2 = \frac{\beta}{\alpha+\beta+\gamma}, \qquad w_3 = \frac{\gamma}{\alpha+\beta+\gamma}. \tag{16}$$

When values of w_1, w_2, w_3 as formula (16), then substitute them into formula (8), will makes the output value Y is closer to the actual value Y^o, so the residual error Q reach the minimum.

Consider

$$\min Q = \sum_{i=1}^{n} \left(y_i^o - y_i\right)^2 = \sum_{i=1}^{n} \left(a_i w_1 + b_i w_2 + c_i w_3 - y_i^o\right). \tag{17}$$

Formula (16) is the optimal weights of the three subclassifiers by training; model (8) achieves the most optimal and further establish ensemble optimization algorithm.

3.4. Elementary Parameters of Ensemble Optimized Algorithm. To sum up the above, using PLS for small sample dimension reduction realized the preliminary optimization of the classification system and then optimized the weights of each subclassifier by OLS. For the high dimensional and small sample problems, we may as well try to establish ensemble optimized classification algorithm by neural network based on PLS and OLS.

Some parameters setting of the new algorithm are as follows.

(1) Data Preprocessing. Data preprocessing eliminates the incommensurability by different data index distribution and numerical differences, which ensures the quality of the application of data from the source. Using standardized transformation obtains the data which is in accordance with the distribution of $N(0, 1)$. The standardized transformation formula is

$$x'_{ij} = \frac{\left(x_{ij} - \overline{x_j}\right)}{\sqrt{(1/n)\sum_{i=1}^{n}\left(x_{ij} - \overline{x_j}\right)^2}}. \tag{18}$$

(2) Parameters of PLS. PLS feature dimension reduction is using K-fold cross validation (K-CV) method to calculate prediction residual sum of squares. This method can effectively avoid overlearning or under-fitting and get more persuasive result.

(3) Parameters of BP Neural Network. The number of neurons using Gao's empirical formula [19] to determine

$$s = \left(0.43nm + 0.12m^2 + 2.54n + 0.77m + 0.35\right)^{1/2} + 0.51. \tag{19}$$

In the formula, s, n, m, respectively, are on behalf of the number of neurons of hidden layer, input layer, and output layer. The neural network training uses trainlm (LM) algorithm, which is the combination of gradient descent and quasi-newton algorithm. Its advantage is give full play to the gradient descent algorithm can rapid convergence at the beginning training, and the quasi-newton algorithm can quickly produce an ideal search direction near the extremum. Connection weights and threshold learning use the Learngdm algorithm.

(4) Set the Center of RBF Neural Network. Set up the center of the RBF network and choose the center of the basis function empirically; as long as the distribution of the training samples can represent the given problem, the s centers of the uniform distribution can be chosen according to experience; the distance is d; choose the width of Gaussian basis function

$$\sigma = \frac{d}{\sqrt{2s}}. \tag{20}$$

Select the basis function by K-clustering method; exploit the center of clustered class as the center of the basis function.

(5) Parameters of Elman Neural Network. Elman neural network, which is optimized by BP neural network, the number of context layer is the same as the hidden layer. Parameter set is equal to BP neural network.

3.5. Steps of Ensemble Optimized Algorithm. The basic steps of neural network ensemble algorithm based on PLS and OLS are as follows.

Step 1. Normalize the original data according to formula (18); get the characteristic variables matrix X and dependent variable matrix Y.

Step 2. Respectively, extract the first pair component t_1, u_1 from X, Y and make up to the maximum correlation, respectively; establish the regression equation of X on t_1 and Y on u_1.

Step 3. Using residual error matrix E and F instead of X and Y, repetition Step 2, until the absolute value of the residual matrix elements is close to zero.

Step 4. With K-CV method, by the principles of cross-validation and residual sum of squares to determine the number of components extracted.

Step 5. From the perspective of information feature compression, get the compression matrix X and Y, as new samples.

Table 1: Comparison of every classifier's performance for Test 1.

Network model	Weights	Training steps	SSE	Accuracy, %
Subclassifier I (BP NN)	0.3207	1132	2.3167	73.33
Subclassifier II (EBF NN)	0.2457	45	4.0261	66.67
Subclassifier III (Elman NN)	0.4336	881	1.0595	80.00
PLS-Elman	—	475	0.8676	80.00
Assembling system	—	—	0.3284	100

Step 6. Divide the new samples into two parts as training samples and simulation samples according to the need of the problem.

Step 7. Set up three subclassifiers, respectively; classification training, the output of the three subclassifiers are A, B, C;

Step 8. By the output of three subclassifiers, establish the ternary regression model (9).

Step 9. Transform (9) into equations set (12) by maximum likelihood estimation.

Step 10. Get the formula (13) by solving equations set (12) and using OLS theory.

Step 11. By regression coefficient and formula (16), the weight of three subclassifiers can be obtained.

Step 12. Get the optimal solution of integrated model (8) and terminate the algorithm.

4. Case Analysis

Respectively, use three subclassifiers, PLS-Elman neural network and neural network ensemble algorithm to test the data set and contrast the test results.

From the follow three aspects to evaluate the performance of each algorithm (model), which includes convergence steps, sum of squared errors and recognition accuracy rate. Convergence steps, we test 10 times, the experiment tests 10 times, we record the best once, and list in table. The sum of squared errors, the sum of squares of the difference of predicted value and actual value, is usually used to estimate the degree of closeness between recognition value and actual value. In the circumstances of the same accuracy simulation, the smaller the error sum of squares is, the higher the precision of the algorithm is. Accuracy rate, the ratio of correct recognized samples and simulation samples, which reflects the recognize accuracy of each algorithm.

In order to illustrate the validity of new algorithm better, we use two data sets for testing. One data set is agricultural pests forecasting data, which is from the actual production [20], and the other data set is the ionosphere data subset of radar, which is from the UCI machine learning standard data sets [21].

4.1. Test 1. This data comes from agricultural production, using the meteorological factor to predict the occurrence degree of wheat midge. The data set includes 60 samples from 1941 to 2000, which regards 14 feature variables (meteorological factors) as the input of neural network. The single output presents the occurrence degree of wheat midge.

4.1.1. Algorithm Performance. Select the last 30 samples to test among 15 training samples and 15 simulation samples, in accordance with the characteristics of small sample. By PLS dimension reduction, the data extracts 6 features, which means the dimensions of data from reduce 14 to 6. The simulation test results are listed in Table 1.

4.1.2. Recognize Precision of Algorithm. In order to better illustrate the classification ability of the new algorithm, we continue to do the following test; divide the selected samples into 25 training samples and 5 simulation samples. Compared with the simulation results and the actual value, the test results are listed in Table 2.

4.2. Test 2. We using another UCI data set to test the new algorithm, the radar data includes 351 samples that have 34 characteristics and each sample is used to predict the quality of the radar, which means 34 inputs and 1 output.

4.2.1. Performance of Each Algorithm. We selected the front 40 samples, among 20 training samples and 20 simulation samples. It is in accordance with the characteristics of small sample. By PLS dimension reduction, this data extracts 19 features; that is, the dimensions of data reduce from 34 to 19. The simulation test results are listed in Table 3.

4.2.2. Every Algorithm's Recognize Precision. In order to better illustrate the classification ability of the new algorithm, we continue to do the following test and divide the selected samples into 35 training samples and 5 simulation samples. Comparing the simulation results with the actual value, the test results are listed in Table 4.

4.3. Results. According to the results of Tables 1 to 4, compared among the three subclassifiers, RBF neural network training is fastest, but the recognition accuracy is the worst; Elman neural network's recognition accuracy is the best; BP

Table 2: Comparison of every classifier's accuracy for Test 1.

Network model	Weights	Training steps	1996	1997	1998	1999	2000	SSE
Actual value	—	—	1	2	2	2	1	—
Subclassifier I (BP NN)	0.3179	1036	0.5031	1.8796	2.2593	1.6927	1.1268	0.4392
Subclassifier II (EBF NN)	0.2311	45	1.8109	2.3207	1.3358	2.2819	0.6651	1.3932
Subclassifier III (Elman NN)	0.4510	925	1.5218	1.9076	1.8978	2.1038	0.8255	0.3325
PLS-Elman	—	447	1.4129	2.2106	1.9073	1.8775	0.8691	0.2556
Assembling system	—	—	1.2919	1.9942	1.8828	2.0145	0.9451	0.1022

Table 3: Comparison of every classifier's performance for Test 2.

Network model	Weights	Training steps	SSE	Accuracy, %
Subclassifier I (BP NN)	0.3301	976	2.0533	70.00
Subclassifier II (EBF NN)	0.2117	45	3.1048	60.00
Subclassifier III (Elman NN)	0.4582	907	1.8368	75.00
PLS-Elman	—	459	1.4739	80.00
Assembling system	—	—	0.4469	100.00

Table 4: Comparison of every classifier's accuracy for Test 2.

Network model	Weights	Training steps	1	2	3	4	5	SSE
Actual value	—	—	0	1	0	1	0	—
Subclassifier I (BP NN)	0.3127	1205	−0.2163	1.5213	0.2027	0.6994	−0.1050	0.4610
Subclassifier II (EBF NN)	0.2208	45	0.3255	1.4126	−0.1685	1.5012	0.2167	0.6027
Subclassifier III (Elman NN)	0.4665	877	−0.1294	0.8927	−0.5101	1.2093	−0.0972	0.3417
PLS-Elman	—	408	0.1273	0.5792	−0.2216	1.1749	0.1132	0.2858
Assembling system	—	—	−0.0561	1.2041	−0.2118	1.1143	−0.0303	0.1036

neural network's training is the slowest; on the training speed and recognition accuracy, Elman is slightly better than BP neural network.

Comparing PLS-Elman algorithm with traditional Elman neural network, it has better training speed and is slightly better in recognition precision. It shows that the recognition precision of classifier is not influenced; on the contrary, the operating efficiency has been improved after PLS optimization.

The classification ability of ensemble optimization is the best. The data of Tables 1 and 3 prove that Comparing with other traditional neural network, the ensemble learning algorithm with the highest recognition accuracy rate and minimum error, it can suggest that the new algorithm with the highest recognition precision.

The experimental results reflect that the recognition precision of ensemble algorithm is obviously higher than any subclassifier, to a certain degree, compensating for the information loss caused by data dimension reduction. Recognition results meet the ideal requirements, which means the new algorithm is effective.

5. Conclusion and Discussion

From the two groups of experimental results above, The optimized classification algorithm by PLS, algorithm, improves the training speed of subclassifier, and and the recognition accuracy is not reduced. It shows that the new algorithm reduces the inputs of neural network by PLS feature dimension reduction, which is convenient for designing the network structure and improving the operating efficiency. The recognition accuracy rate and test error of integrated system have been greatly improved, which shows that the classification precision of the ensemble algorithm has been greatly improved and higher than any subclassifier. The purpose of neural network ensemble is to improve the recognition precision of pattern classification and the point of ensemble learning is to determine the weights of each

sub,classifier effectively. In this paper, using OLS principle to design integrator, establish multiple regression model and further determine the weight of each subclassifier.

In view of the small sample classification problem, this paper proposes an ensemble optimized neural network classification algorithm based on PLS and OLS. PLS has unique advantage to reduce dimension for small sample data, which obtains ideal low-dimensional data. By OLS algorithm, do neural network ensemble learning, to determine the weights of each subclassifier. The new algorithm has higher operating efficiency and classification precision, which shows the worthness of further popularization and application.

Conflict of Interests

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

Acknowledgments

This work is supported by the National Natural Science Foundation of China (nos. 60875052, 61203014, and 61379101); Priority Academic Program Development of Jiangsu Higher Education Institutions; Major Projects in the National Science & Technology Pillar Program during the Twelfth Five-year Plan Period (no. 2011BAD20B06); Major Projects in Jiangsu Province Science & Technology Pillar Program Plan (Industrial) (no. BE2012136); The Specialized Research Fund for the Doctoral Program of Higher Education of China (no. 20133227110024), Ordinary University Graduate Student Research Innovation Projects of Jiangsu Province (no. KYLX14_1062).

References

[1] S. F. Ding, W. K. Jia, C. Y. Su, L. W. Zhang, and Z. Z. Shi, "Neural network research progress and applications in forecast," in *Advances in Neural Networks—ISNN 2008*, vol. 5264 of *Lecture Notes in Computer Science*, pp. 783–793, Springer, New York, NY, USA, 2008.

[2] S. F. Ding, H. Zhu, W. K. Jia, and C. Y. Su, "A survey on feature extraction for pattern recognition," *Artificial Intelligence Review*, vol. 37, no. 3, pp. 169–180, 2012.

[3] F. X. Song, X. M. Gao, S. H. Liu, and J. Y. Yang, "Dimensionality reduction in statistical pattern recognition and low loss dimensionality reduction," *Chinese Journal of Computers*, vol. 28, no. 11, pp. 1915–1922, 2005.

[4] B. Garlik and M. Křivan, "Identification of type daily diagrams of electric consumption based on cluster analysis of multi-dimensional data by neural network," *Neural Network World*, vol. 23, no. 3, pp. 271–283, 2013.

[5] K. R. Janes, S. X. Yang, and R. R. Hacker, "Pork farm odour modelling using multiple-component multiple-factor analysis and neural networks," *Applied Soft Computing Journal*, vol. 6, no. 1, pp. 53–61, 2005.

[6] J. Zhou, A. H. Guo, B. Celler, and S. Su, "Fault detection and identification spanning multiple processes by integrating PCA with neural network," *Applied Soft Computing A*, vol. 14, pp. 4–11, 2014.

[7] H. W. Wang, *Partial Least Squares Regression Method and Application*, National Defense Industry Press, Beijing, China, 2000.

[8] G. Z. Li, R. W. Zhao, H. N. Qu, and M. Y. You, "Model selection for partial least squares based dimension reduction," *Pattern Recognition Letters*, vol. 33, no. 5, pp. 524–529, 2012.

[9] J. Marques and D. Erik, "Texture analysis by a PLS based method for combined feature extraction and selection," in *Machine Learning in Medical Imaging*, vol. 7009 of *Lecture Notes in Computer Science*, pp. 109–116, 2011.

[10] S. F. Ding, W. K. Jia, X. Z. Xu, and C. Su, "Elman neural network algorithm based on PLS," *Acta Electronica Sinica*, vol. 38, no. 2, pp. 71–75, 2010.

[11] X. G. Tuo, M. Z. Liu, and L. Wang, "A PLS-based weighted artificial neural network approach for alpha radioactivity prediction inside contaminated pipes," *Mathematical Problems in Engineering*, vol. 2014, Article ID 517605, 5 pages, 2014.

[12] 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.

[13] D. Opitz and R. Maclin, "Popular ensemble methods: an empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[14] Z. H. Zhou and S. F. Chen, "Neural network ensemble," *Chinese Journal of Computers*, vol. 25, no. 1, pp. 1–8, 2002.

[15] N. Kourentzes, D. K. Barrow, and S. F. Crone, "Neural network ensemble operators for time series forecasting," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4235–4244, 2014.

[16] C. X. Zhang and J. S. Zhang, "A survey of selective ensemble learning algorithms," *Chinese Journal of Computers*, vol. 34, no. 8, pp. 1399–1410, 2011.

[17] Z. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial Intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.

[18] M. Alhamdoosh and D. Wang, "Fast decorrelated neural network ensembles with random weights," *Information Sciences*, vol. 264, pp. 104–117, 2014.

[19] D. Q. Gao, "On structures of supervised linear basis function feed forward three-layered neural networks," *Chinese Journal of Computers*, vol. 21, no. 1, pp. 80–86, 1998.

[20] Y. M. Zhang, *The Application of Artificial Neural Network in the Forecasting of Wheat Midge*, Northwest A&F University, Shaanxi Province, China, 2003.

[21] http://mlearn.ics.uci.edu/databases/.