Zhihua Cai
Chengyu Hu
Zhuo Kang
Yong Liu (Eds.)

LNCS 6382

Advances in Computation and Intelligence

5th International Symposium, ISICA 2010
Wuhan, China, October 2010
Proceedings

Springer

Lecture Notes in Computer Science 6382

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Zhihua Cai   Chengyu Hu   Zhuo Kang
Yong Liu (Eds.)

Advances in Computation and Intelligence

5th International Symposium, ISICA 2010
Wuhan, China, October 22-24, 2010
Proceedings

Springer

Volume Editors

Zhihua Cai
China University of Geosciences, School of Computer Science
Wuhan, Hubei, 430074, P.R. China
E-mail: zhcai@cug.edu.cn

Chengyu Hu
China University of Geosciences, School of Computer Science
Wuhan, Hubei, 430074, P.R. China
E-mail: huchengyu@cug.edu.cn

Zhuo Kang
Wuhan University, Computation Center
Wuhan, Hubei 430072, P.R. China
E-mail: kang_whu@yahoo.com

Yong Liu
The University of Aizu, School of Computer Science
and Engineering
Tsuruga, Ikki-machi, Aizu-Wakamatsu City
Fukushima, 965-8580, Japan
E-mail: yliu@u-aizu.ac.jp

Library of Congress Control Number: Applied for

CR Subject Classification (1998): I.2, F.1, J.3, I.5, I.2.11, I.2.6

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN	0302-9743
ISBN-10	3-642-16492-7 Springer Berlin Heidelberg New York
ISBN-13	978-3-642-16492-7 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

LNCS 6382 is the first volume of the proceedings of the Fifth International Symposium on Intelligence Computation and Applications (ISICA 2010) held in Wuhan, China, October 22–24, 2010. Fifty-three papers among 267 submissions were selected and included in *LNCS* 6382.

The symposium featured the most up-to-date research in ant colony and particle swarm optimization, differential evolution, distributed computing, genetic algorithms, multi-agent systems, multi-objective and dynamic optimization, robot intelligence, statistical learning, and system design.

LNCS 6382 is dedicated to the memory of Lishan Kang. ISICA conferences were one of the first series of international conferences on computational intelligence that combine elements of learning, adaptation, evolution and fuzzy logic to create programs as alternative solutions to artificial intelligence. The idea for ISICA came about after Lishan Kang organized on international symposium on evolutionary computation at Wuhan University in 2000. After he was invited to be the Director of the School of Computer Science, China University of Geosciences, he wondered whether he could establish such discussion forums on computational intelligence at China University of Geosciences. With support from his university, the School of Computer Science organized the first ISICA in 2005, in which some of the leading figures from the scientific computing world were invited, including H.-P. Schwefel, Germany, M. Schoenauer, France, D.J. Evans, UK, T. Higuchi, Japan, Z. Michalewicz, Australia, and X. Yao, UK.

The Second ISICA was jointly held in 2007 with the 7th International Conference on Evolvable Systems: From Biology to Hardware (ICES 2007). Since then, ISICA has become an annual event. Sadly, the founder of ISICA, Lishan Kang, passed away last year. However, his spirit will live with us and inspire us to hold ISICA continually and successfully. Kang firmly believed that evolutionary computation is the foundation of computational intelligence, and computational intelligence is the future of computational science. We truly hope that ISICA will establish a bridge for young researchers to reach this beautiful future one day.

Solutions have been evolving in computational intelligence. So has Kang's research. Kang started his research on the Schwarz algorithm in 1957 under the guidance of Russian computational scientist, I.P. Mysovskich. Although Schwarz algorithm was proposed as early as 1869, it had not attracted enough attention at that time. As a young researcher, Kang became fascinated with Schwarz algorithm and submitted his first paper on the multi-domain Schwarz alternating method in 1959. However, his paper was rejected with the only comment being that it was of no practical use. Twenty years later, with the development of parallel computers, researchers' attention was brought back to Kang's results on Schwarz algorithm. It was like a rebirth of Kang's research. Led by the

Kang, a group at Wuhan University developed the first distributed computing system called WuPP-80 in China in 1982. Kang had solved a number of difficult mathematical physics problems using the multi-domain Schwarz alternating method on WuPP-80. Because of Kang's great achievement on asynchronous parallel computing and his theoretical research on the convergence of the Schwarz algorithm with the size of overlapping domains, he was honored with the fourth-class prize of the National Natural Science Award of China in 1993, which was the highest prize awarded to computer science that year.

After domain decomposition methods reached a level of maturity in the late 1980s, Kang shifted his attention to the new research field of evolutionary computation. There are always great challenges in a new field, but that also means great chances for research. In those years, Kang's students were excited about evolutionary computation. Meanwhile, there were no foreign research journals and books on evolutionary computation at the Wuhan University Library. Kang sent a student to get a few references from the Beijing Library. In such hard conditions, Kang and his student, Yong Liu, wrote the first research book on evolutionary computation in China, *Non-Numerical Algorithms: (II) Genetic Algorithms*, published by China Science Press in 1995.

Being a generation greatly influenced by Chairman Mao, Kang had answered Mao's calling of "Yang Wei Zhong Yong" (to make the foreign things serve China). He had given hundreds of public talks on both the Schwarz alternating method and evolutionary computations at many universities in China starting in 1980s. Late in his life, Kang still insisted on giving lectures at summer teacher workshops in Guiyang for a number of years till he was diagnosed with stomach cancer. Nowadays thousands of students and researchers in China are following in his footsteps.

With the popularity of "Yang Wei Zhong Yong" in the computational field in China, we would also like to call attention to "Gu Wei Jin Yong" (to make the past serve the present) among researchers. Nengchao Wang, for example, has demonstrated in a keynote speech at ICES 2007 how the evolution of Yin and Yang could be used in designing modern computer architectures and algorithms. Hard as it is to believe, even many foreign experts became interested in Wang's speech, given in Chinese. Kang had solved the 100-year-old convergence problem existing in the Schwarz algorithm, while Wang uncovered a thousand-year-old mathematical mystery, and rediscovered how the ancient mathematician Hui Liu calculated the ratio of a circle's area to the square of its radius π to 3.1416 more than a thousand years ago. The methodology used in calculating π actually shares a similar idea with computational intelligence concerning the evolution of solutions.

Finally, on behalf of the Organizing Committee, we would like to warmly thank the sponsor, China University of Geosciences for helping us in sundry ways to achieve our goals for the conference. We wish to express our appreciation to Springer for publishing the proceedings of ISICA 2010. We also wish to acknowledge the dedication and commitment of the *LNCS* and *CCIS* editorial staff. We would like to thank the authors for submitting their work, as well as the Program Committee members and reviewers for their enthusiasm, time and expertise. The invaluable help of active members from the Organizing Committee, including Hengjian Tong, Chengyu Hu, Wei Qiang, Hongwei Zhang and Hao Zhang, in setting up and maintaining the online submission systems, assigning the papers to the reviewers, and preparing the camera-ready version of the proceedings, is highly appreciated. We would like to thank them personally for their help making ISICA 2010 a success.

October 2010

Zhihua Cai
Yong Liu
Zhenhua Li

Organization

ISICA 2010 was organized by the School of Computer Science, China University of Geosciences, sponsored by China University of Geosciences, and supported by Springer.

General Chair

Huiming Tang	China University of Geosciences, China

Program Chairs

Zhihua Cai	China University of Geosciences, China
Yong Liu	University of Aizu, Japan
Zhijian Wu	Wuhan University, China

Publication Chairs

Hengjian Tong	China University of Geosciences, China
Chengyu Hu	China University of Geosciences, China

Local Arrangement Chair

Hui Li	China University of Geosciences, China

Program Committee

Tughrul Arslan	The University of Edinburgh, UK
Javad Sohafi Bonab	Islamic Azad University, Iran
Tan Kay Chen	National University of Singapore, Singapore
Carlos A. Coello	LANIA, Mexico
Guangming Dai	China University of Geosciences, China
Kalyanmoy Deb	Indian Institute of Technology, India
Yaochu Jin	Honda Research Institute Europe, Germany
Pavel Kromer	Technical University of Ostrava, Czech Republic
Yuanxiang Li	Wuhan University, China
Zhenhua Li	China University of Geosciences, China

Steffen Limmer	Friedrich Alexander University Erlangen Nürnberg, Germany
Shiow-Jyu Lin	National Taiwan Normal University, Taiwan
Charles X. Ling	The University of Western Ontario, Canada
Bob McKay	Seoul National University, Korea
Ryszard Tadeusiewicz	AGH University of Science and Technology, Krakow, Poland
Hamid R. Tizhoosh	The University of Waterloo, Canada
Dong-Min Woo	Myongji University, Korea
Zhijian Wu	Wuhan University, China
Shengxiang Yang	University of Leicester, UK
Xin Yao	University of Birmingham, UK
Gary G. Yen	Oklahoma State University, USA
Sanyou Zeng	China University of Geosciences, China
Huajie Zhang	University of New Brunswick, Canada
Qingfu Zhang	University of Essex, UK
Xiufen Zou	Wuhan University, China

Local Co-chair

Zhenhua Li	China University of Geosciences, China
Guangming Dai	China University of Geosciences, China
Sifa Zhang	China University of Geosciences, China
Yi Zeng	China University of Geosciences, China

Local Committee

Shuanghai Hu	China University of Geosciences, China
Wei Qiang	China University of Geosciences, China
Siqing Xue	China University of Geosciences, China
Siwei Jiang	China University of Geosciences, China
Ming Yang	China University of Geosciences, China

Secretaries

Hongwei Zhang	China University of Geosciences, China
Hao Zhang	China University of Geosciences, China
Jing Zai	China University of Geosciences, China

Sponsoring Institutions

China University of Geosciences, Wuhan, China

Table of Contents

Section I: Ant Colony and Particle Swarm Optimization

A Discrete PSO-Based Fault-Tolerant Topology Control Scheme in Wireless Sensor Networks 1
Bingyu You, Guolong Chen, and Wenzhong Guo

Retracted: Feature Selection Using Ant Colony Optimization for Text-Independent Speaker Verification System 13
Javad Sohafi-Bonab and Mehdi Hosseinzadeh Aghdam

GPS Height Fitting Using Gene Expression Programming 25
Xuezhi Yue, Zhijian Wu, Dazhi Jiang, and Kangshun Li

Parameter Evolution for a Particle Swarm Optimization Algorithm 33
Aimin Zhou, Guixu Zhang, and Andreas Konstantinidis

The Ant Colony Optimization Algorithm for Multiobjective Optimization Non-compensation Model Problem Staff Selection 44
Ryszard Tadeusiewicz and Arkadiusz Lewicki

Section II: Differential Evolution

A New Self-adaption Differential Evolution Algorithm Based Component Model 54
Shen Li and Yuanxiang Li

A Novel Interpolation Method Based on Differential Evolution-Simplex Algorithm Optimized Parameters for Support Vector Regression 64
Dongmei Zhang, Wei Liu, Xue Xu, and Qiao Deng

A Variant of Differential Evolution Based on Permutation Regulation Mechanism 76
Dazhi Jiang, Hui Wang, and Zhijian Wu

Differential Evolution Based Band Selection in Hyperspectral Data Classification 86
Xiaobo Liu, Chao Yu, and Zhihua Cai

Diversity Analysis of Opposition-Based Differential Evolution—An Experimental Study 95
Hui Wang, Zhijian Wu, Shahryar Rahnamayan, and Jing Wang

Hybrid Differential Evolution Algorithm with Chaos and Generalized Opposition-Based Learning 103
Jing Wang, Zhijian Wu, and Hui Wang

Self-adapting Differential Evolution Algorithm with Chaos Random for Global Numerical Optimization 112
Ming Yang, Jing Guan, Zhihua Cai, and Lu Wang

Section III: Distributed Computing

A Concurrent-Hybrid Evolutionary Algorithms with Multi-child Differential Evolution and Guotao Algorithm Based on Cultural Algorithm Framework 123
Xia Li, Kunqi Liu, Lixiao Ma, and Huanzhe Li

A Hierarchical Distributed Evolutionary Algorithm to TSP 134
Chengjun Li, Guangfu Sun, Dongmei Zhang, and Songhu Liu

A New Distributed Node Localization Scheme Using a Mobile Beacon 140
Sheng Xiao, Changfeng Xing, and Zhangsong Shi

A Novel Multi-Population Genetic Algorithm for Multiple-Choice Multidimensional Knapsack Problems 148
Qian Zhou and Wenjian Luo

Comparative Analysis for k-Means Algorithms in Network Community Detection 158
Jian Liu

Framework for Distributed Evolutionary Algorithms in Computational Grids 170
Steffen Limmer and Dietmar Fey

Section IV: Genetic Algorithms

A Review of Tournament Selection in Genetic Programming 181
Yongsheng Fang and Jun Li

Genetic Algorithm for Mixed Chinese Postman Problem 193
Hua Jiang, Lishan Kang, Shuqi Zhang, and Fei Zhu

Hybrid Evolutionary Algorithms Design Based on Their Advantages 200
Guangming Lin, Sundong Liu, Fei Tang, and Huijie Wang

Hybridized Optimization Genetic Algorithm for QOS-Based Multicast Routing Problem 211
Yunliang Chen, Jianzhong Huang, and Changsheng Xie

Retroviral Iterative Genetic Algorithm for Real Parameter Function Optimization Problems 220
Renato Simões Moreira, Glauber Duarte Monteiro, Otávio Noura Teixeira, Átila Siqueira Soares, and Roberto Célio Limão de Oliveira

The RM-MEDA Based on Elitist Strategy 229
Li Mo, Guangming Dai, and Jiankai Zhu

Section V: Multi-agent Systems

A Proactive Perception Model in Agent-Based Software Integration and Evolution 240
Qingshan Li, Chengguang Zhao, Haishun Yun, and Lili Guo

Algorithmic Trading Strategy Optimization Based on Mutual Information Entropy Based Clustering 252
Feng Wang, Keren Dong, and Xiaotie Deng

An Improved Similarity Algorithm Based on Hesitation Degree for User-Based Collaborative Filtering 261
Xiangwei Mu, Yan Chen, Jian Yang, and Jingjing Jiang

Intrusion Detection System Based on Support Vector Machine Active Learning and Data Fusion 272
Man Zhao, Jing Zhai, and Zhouqian He

Long-Distant Pipeline Emergency Command Decision-Making System Based on Agent 280
Huagang He, Man Yang, and Ling Qian

Sales Forecasting Based on ERP System through BP Neural Networks 289
Min Zhang, Haifeng Zhang, and Yujuan Huang

Section VI: Multi-objective and Dynamic Optimization

Capacity Allocation Policy of Third Party Warehousing with Dynamic Optimization 297
Chang Lin

Dynamical Multi-objective Optimization Using Evolutionary Algorithm for Engineering 304
Lingling Wang and Yuanxiang Li

Faster Convergence and Higher Hypervolume for Multi-objective Evolutionary Algorithms by Orthogonal and Uniform Design 312
Siwei Jiang and Zhihua Cai

Multi-objective Fuzzy Clustering Method for Image Segmentation Based on Variable-Length Intelligent Optimization Algorithm 329
Yuankang Fang, Ziyang Zhen, Zhiqiu Huang, and Chao Zhang

Perspectives in Dynamic Optimization Evolutionary Algorithm 338
Zhiqiong Bu and Bojin Zheng

Section VII: Robot Intelligence

A Real Time Vision-Based Hand Gestures Recognition System......... 349
Lei Shi, Yangsheng Wang, and Jituo Li

A Virtual Robot Arm Model with Force Feedback and Contact States Identification 359
Chengjun Chen and Niu Li

Improving Reading Comprehension Using Knowledge Model 370
Yue Chen

Modular Robot Path Planning Using Genetic Algorithm Based on Gene Pool 380
Huaming Zhong, Zhenhua Li, Hao Zhang, Chao Yu, and Ni Li

Pathfinder Based on Simulated Annealing for Solving Placement and Routing Problem 390
Zhangyi Yu, Sanyou Zeng, Yan Guo, Nannan Hu, and Liguo Song

Section VIII: Statistical Learning

Application of Data Mining in Multi-Geological-Factor Analysis 402
Jing Chen, Zhenhua Li, and Bian Bian

Applying an Artificial Neural Network to Building Reconstruction 412
Dong-Min Woo, Dong-Chul Park, and Hai-Nguyen Ho

Fast Dimension Reduction Based on NMF 424
Pavel Krömer, Jan Platoš, and Václav Snášel

Frequent Words' Grammar Information in Chinese Chunking 434
Quan Qi, Li Liu, and Yue Chen

Multilabel Classification Using Error Correction Codes 444
Abbas Z. Kouzani

Test-Cost Sensitive Classification Using Greedy Algorithm on Training Data 455
Chang Wan

Section IX: System Design

3-D Magnetotelluric Adaptive Finite-Element Modeling 465
Changsheng Liu, Yan Yu, Zhengyong Ren, and Qi Wu

A Study on Modeling Evolutionary Antenna Based on ST-5 Antenna and NEC2 474
Yuanyuan Fan, Qingzhong Liang, and Sanyou Zeng

A Synthesis of Four-Branch Microwave Antenna by Evolution Algorithm and Orthogonal Experiment 487
Jincui Guo, Jinxin Zou, Yincheng Wang, Xiaojuan Zhao, and Liangjiang Yu

Band Structures of Multilayer Films with Randomness 496
Ping Li, Zhuo Li, and Yong Liu

Fast Principal Component Analysis Based on Hardware Architecture of Generalized Hebbian Algorithm 505
Shiow-Jyu Lin, Yi-Tsan Hung, and Wen-Jyi Hwang

The Data-Based Mathematical Modeling and Parameter Identification in JAK-STAT Signaling Pathway by Using a Hybrid Evolutionary Algorithm 516
Wei Zhang and Xiufen Zou

The Research on the Intelligent Interpreter for ISO 14649 Programs 523
Yu Zhang, Yongxian Liu, and Xiaolan Bai

Erratum

Feature Selection Using Ant Colony Optimization for Text-Independent Speaker Verification System E1
Javad Sohafi-Bonab and Mehdi Hosseinzadeh Aghdam

Author Index 535

A Discrete PSO-Based Fault-Tolerant Topology Control Scheme in Wireless Sensor Networks

Bingyu You, Guolong Chen*, and Wenzhong Guo

College of Mathematics and Computer Science, Fuzhou University,
Fuzhou, 350002, China
youby3536@sina.com

Abstract. Fault-tolerant topology control in wireless sensor networks (WSN) is a NP-hard problem, drawing significant research interests in the past several years. However, most of the previous studies were only based on heuristic approaches to obtain approximate solutions, and so the final network topologies generated are not reasonable sometimes. Aiming at this problem and taking both the issues of power efficient and node failure into consideration, we propose, in this paper, a discrete PSO-based topology control scheme (FTPSO) for minimizing the transmission power of each sensor node, which can improve the network reliability effectively. The final network topologies derived by this algorithm can also preserve at least k=2 vertex disjoint paths between any pair nodes. The results of this algorithm are compared to other approaches to demonstrate the effectiveness of the proposed methodology.

Keywords: WSN; PSO; topology control; 2-connectivity; fault-tolerant.

1 Introduction

In the past few years, wireless sensor networks have been increasingly deployed for civil and military applications. However, due to some unavoidable limitations compared with traditional networks, we need to adjust the transmitting power of each sensor node in order to use the limited energy effectively, which is called *topology control* [1, 2]. Recently, the researches on topology control can be divided into two main categories: centralized and distributed. Compared to distributed schemes, centralized algorithms need to collect the overall network information and require higher maintenance costs. However, the topologies derived by centralized algorithms are usually better than those generated by distributed schemes [3]. Of course, there are several works in the general area of topology control and network design previously.

Usually, wireless sensor networks are randomly distributed in hash environments. Each sensor node may be added to the network system, may change their positions or event run out of their limited energy in most cases. Accordingly, they are likely to fail. However, most previously proposed topology control algorithms don't take the issue of node failure into consideration, which often leads to the reduction of network reliability.

* Corresponding Author.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 1–12, 2010.

Generally, the studies on the evaluation of network reliability can be divided into two main categories: one is to study the probability of network failure due to node failure or link failure, and another is to build low-power and high-connected network topology structures using the knowledge of graph theory [4]. In this paper, we will continue the second direction for further studies. Particularly, we will concentrate on 2-vertex connectivity, and use 2-connectivity to refer to 2-vertex connectivity for simplicity. A 2-connected network can tolerate at least one node failure. Of course, if a graph G is 2-connectivity, there are at least 2 vertex disjoint paths between any pair nodes according to *Menger* Theorem [5].

Notice that, most currently MAC protocols and routing protocols only take bidirectional edges into consideration, and it is also necessary that a message sent over a link can be acknowledged by sending a corresponding message over the same link [6, 7], so in this paper we only take bi-directional edges into consideration and ignore the existence of unidirectional edge.

Nowadays, there are several other results on the topology control and network design for achieving power efficiency and maintaining network 2-connectivity in WSN, and different approaches have been proposed previously [3, 6-12]. However, to our best knowledge, most of them are only based on heuristic approaches to obtain approximate solutions, the final topologies generated by which are unreasonable sometimes.

Sun et al. in [8] have studied the relationship among the number of sensor nodes, the radiation power of each sensor and the network connectivity through simulations, and obtained experiment formulas of 1-vertex and 2-vertex connectivity probability. However, the corresponding topology control scheme hasn't been proposed in the literature. Ram et al. in [9] have taken the issues of connectivity and 2-connectivity into account and presented two topology control schemes based on a centralized spanning tree, called CONNECT and BICONN respectively. Hajiahayi et al. in [10] presented an approximation algorithm called 2-UPVCS to find a minimum power 2-connected network structure. However, their works are only based on some heuristics and then there is no guarantee of 2-connectivity in all cases. Li. et al in [7] has given a counterexample to show that 2-UPVCS algorithm cannot preserve 2-connectivity in some cases and presented their own algorithm FGSS and FLSS.

As constructing a 2-connected network topology while minimizing the overall energy consumption is NP-hard [9], an exact solution is infeasible. Fortunately, swarm intelligence algorithms can be used to solve NP-hard problems effectively. In literature [11] and [12] the authors have designed a genetic algorithm and a quantum genetic algorithm to deal with the problem respectively, but they haven't taken the issue of node failure into consideration, the finally topology structure generated by which will lower the network reliability. Aiming at this problem, in this paper we will proposed a centralized topology control scheme called FTPSO based on a discrete PSO algorithm to construct a power-efficient network structure while maintaining the network property of 2-connecitvity.

The rest of this paper is organized as follows: In Section 2, the system model is described in detail, and then we present our proposed methodology in Section 3. In Section 4, we compare our algorithm with other approaches and evaluate the performance of them. Finally, concluding remarks are made in Section 5.

2 Network Model and Problem Statement

In this section, we focus on the problem of how to obtain a power-efficient network while maintaining the property of 2-connectivity in WSN. We assume that a wireless sensor network composed of considerable numbers of sensor nodes are randomly distributed in a two-dimensional plane. Given the position and the transmission power of each node, a network can be mapped into a graph $G(V,E)$, where V is a finite set of sensor nodes, and E is defined as the wireless connection between nodes.

We use $d(i,j)$ to represent the Euclidean distance between node i and node j, and $r(i)$ is the radius of node i, where $0<=r(i)<= rmax(i)$. The power requirement for supporting the communication between node i and j is given by $p(i,j)=q \times d(i,j)^r$, where q is the receiver's power threshold for signal detection and r is the path loss exponent [1]. Generally, r is equal to 2 or 4. In this paper, r is set as 2.

In order to facilitate the description of our algorithm, we firstly give the following definitions of several terms which will be used throughout this paper.

Definition 1. *Adjacency Matrix: We assume a matrix C is the adjacency matrix of a graph G. As we only take the bidirectional edges into account in this paper, C(i,j)=1 if and only if d(i,j)<=r(i) and d(i,j)<= r(j), otherwise C(i,j)=0.*

Definition 2. *Cut Point: A node is a cut point of a connected graph G such that its removal will cause the resulting topology graph to be disconnected. We use $NC_{(G)}$ to represent the number of cut points in G. The algorithm used in this paper to determine the number of cut points of the network topology is a very sophisticated algorithm [5].*

Definition 3. *Reachable Neighbors: node j is a reachable neighbor of node i, if and only if d(i,j)<=rmax(i).*

Definition 4. *Average Node Degree: The node degree of node i, dc(i) , is defined as the number of node i's reachable neighbors. The smaller the degree of node i, the less number of nodes node i may affect. Obviously, the average node degree is a good indication of the level of communication interference [1, 7].*

Definition 5. *Average Energy Saving [7]:*

$$ES = \frac{E_{ave}}{E_{max}} \times 100\% \tag{1}$$

where E_{ave} is the average transmission power over all the nodes in the network, and E_{max} is the maximum transmission power.

In order to determine whether the network structure is connected, the method used in literature [11] and [12] is adopted in this paper. This concrete process is as follows: Firstly, we define a matrix operation symbols $\otimes$. Assume that $C=A \otimes B$ (where A is a $m \times k$ matrix, B is a $k \times n$ matrix), then

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \ldots \vee (a_{ik} \wedge b_{kj}) \tag{2}$$

$$L = C \otimes C \otimes \ldots\ldots \otimes C \tag{3}$$

$$\gamma(G) = \frac{\left(\sum_{1 \le i \le n} \sum_{1 \le j \le n} L(i, j) \right)}{n \times n} \tag{4}$$

where n is the number of sensor nodes and γ is the connectivity rate. The number of C in (4) is n.

The network is said to be connectivity if any node can communicate with any other node. If the network is connectivity, $\gamma=1$; otherwise $0<\gamma<1$. Topology control algorithms should keep the connectivity of the network. To this end, we include $\gamma=1$ as a constraint. Of course, the connectivity of a graph can be also tested using Depth First Search (DFS) algorithm [5].

In particular, a network is said to be 2-connectivity and hence has the ability to tolerate one node failure, if and only if the removal of any sensor node will not render the topology graph disconnected. In order to construct a 2-connected network structure, we transform the model into the problem of finding a topology graph without cut points. According to the above analysis, we designed the following mathematical model:

$$\min \quad \sum p(i) \tag{5}$$

$$\min \quad NC_{(G)} \tag{6}$$

$$s.t. \quad \gamma(G) = 1 \tag{7}$$

$$1 \le i \le n \tag{8}$$

$$0 \le p\left(i\right) \le p_{max} \tag{9}$$

where $p(i)$ is the radiation power of node i, and $NC_{(G)}$ is the number of cut points in G.

Obviously, the above model is a multi-objective optimization problem. To our knowledge, there are lots of methods to deal with Multi-objective problems such as the *weighted-sum* method, the *utility-function* method, the *compromise approach* and the *Pareto* method and so no [13]. In this paper, we adopt the following approach to transform the multi-objective optimization problem into a single objective optimization problem. If the initial network topology is connected, the objective function of the model can be formulated mathematically as follows:

$$\min \quad f = \ (NC_{(G)} + 1) \times \sum p(i) \tag{10}$$

The purpose of ($NC_{(G)}$ + 1) is to avoid the situation that f equals to 0.

In order to solve this problem, in this paper we design a discrete PSO-based algorithm to deal with it. Simulation results also indicate that our algorithm can converge quickly.

3 Algorithm

3.1 Basic Particle Swarm Optimizaiton

Particle swarm optimization (PSO) is a population-based evolutionary technique developed by Dr. Eberhart and Dr. Kennedy in 1995[14], where each particle represents a candidate solution and has a fitness value which is evaluated by the objective functions. Obviously, each particle can be represented as $X_i=\{X_{i1}, X_{i2}, X_i3,......, X_{in}\}$. After initialization, the system searches for the best particle by evolutions. In every generation, each particle adjusts its own position according to two "*best*" values: one is its best previous position p_{besti}, and another is the best particle in the population g_{best}. After finding the two best values, each particle updates its velocity according to the following equations:

$$v_{\mathrm{i}} = w\times v_{\mathrm{i}} + c_1\times r_1\times(p_{besti} - x_{\mathrm{i}}) + c_2\times r_2\times(g_{best} - x_{\mathrm{i}}) \tag{11}$$

$$x_{\mathrm{i}} = x_{\mathrm{i}} + v_{\mathrm{i}} \tag{12}$$

As shown in the above formulas, the first part of (11) represents the previous velocity. The second part represents the personal thinking of each particle, which is known as the "cognitive" component. The cognitive component encourages each particle to move toward its own best position found so far. The third part, known as "social" component, represents the collaborative effect of the particles, in finding the global optimal solution. [15]

3.2 Discrete Particle Swarm Optimization

3.2.1 Representation of Particles

For a wireless sensor network composed of *n* sensor nodes, we encode each particle as follows:

We use one-dimensional array $X(1{:}n)$ to represent each particle, and the value of $X(i)$ equals to the radius of node *i*. Obviously, this encoding method can effectively reflect the energy consumption of the overall network. Of course, the connectivity rate and the number of cut points can be computed easily according to the value of each particle.

3.2.2 Fitness Value Function

In order to evaluate each particle's adaptability, we take all the optimization objectives and constrains into consideration. As the definition of cut point only makes sense in the case of that the network is connected, in this paper we compare the merits of any two particles as follows:

Firstly, we assume that the network topology is connected if each node communicates with each other using its own maximum radiation power. For any pair particles X_1 and X_2, we use G_1 to represent the corresponding network topology of X_1, and G_2 represents the corresponding network topology of X_2. Assume that

$$P_1 = \sum_{i=1}^{n} X_1(i) \tag{13}$$

$$P_2 = \sum_{i=1}^{n} X_2(i) \tag{14}$$

$$f_1 = (NC_{(G_1)} + 1) \times P_1 \tag{15}$$

$$f_2 = (NC_{(G_2)} + 1) \times P_2 \tag{16}$$

$$\{X_1 \succ X_2\} = \begin{cases} \gamma(G_1) > \gamma(G_2) & ; \\ \gamma(G_1) = \gamma(G_2) = 1 \; and \; f_1 < f_2 & . \end{cases} \tag{17}$$

3.2.3 The Parameter Settings

The values of the relative parameters will affect the convergence of PSO algorithm effectively. In this paper, the inertial weight w is assumed to decrease linearly during the course of the simulation from w_{max} to w_{min}. The parameters c_1 and c_2 are also significantly influence each particle's "flying". In order to improve the efficiency of our algorithm, we adopt the following adjustment strategies [16]:

$$w = \left(w_{max} - w_{min}\right) \times \left(gen_{max} - gen_{cur}\right) / \, gen_{max} + w_{min} \tag{18}$$

$$c_1 = 1.0 - 1.0 \times gen_{cur} \, / \, gen_{max} \tag{19}$$

$$c_2 = 1.0 - c_1 \tag{20}$$

For the purpose of maintaining the population diversity in the evolutionary process, we don't give up a bad new particle directly, but accept the new particle according to the probability *upd*. In this paper, the value of *upd* is updated according to the following equation:

$$upd = (upd_{max} - upd_{min}) \times (gen_{max} - gen_{cur}) + upd_{min} \tag{21}$$

where upd_{max} and upd_{min} are the initial max value and min value of *upd* respectively.

3.2.4 Discrete Procedure of PSO

The notion of mutation operator in GA is incorporated into the first part of (11).

$$A_i^t = F_1(X_i^{t-1}, w) = \begin{cases} M(X_i^{t-1}), r_1 < w \\ X_i^{t-1}, \text{otherwise} \end{cases} \tag{22}$$

where F_1 indicates the mutation operator with the probability of w.

The mutation operator adopted in this paper is as follows: We select a random position ($1 \le i \le n$) of the current particle, and then change the corresponding value into a random value between 0 and *r*(*i*). Of course, we can also change the corresponding value into the Euclidean distance *d*(*i,j*) between node *i* and node *j*, where *j* ∈ *Nbr*(*i*).

The second part and third part of (11) both adopt the notion of crossover operator in *GA*.

$$B_i^t = F_2(A_i^t, c_1) = \begin{cases} C_p(A_i^t), r_2 < c_1 \\ A_i^t, \text{otherwise} \end{cases} \tag{23}$$

$$X_i^t = F_3(B_i^t, c_2) = \begin{cases} C_g(B_i^t), r_3 < c_2 \\ B_i^t, \text{otherwise} \end{cases} \tag{24}$$

where F_2, F_3 indicate the mutation operator with the probability of c_1, c_2 respectively.

There are different crossover operators namely *simple* (*single-point*), *arithmetical*, *blend*, *boundary*, *directional* crossovers et al [13]. The crossover operator adopted in this paper is as follows, which is called arithmetical crossover operator:

Assume that $k_1+k_2=1$, $k_1>0$, $k_2>0$, and then the new particle can be obtained by the following formula:

$$Xnew(i) = k_1 \times X_1(i) + k_2 \times X_2(i) \tag{25}$$

In this paper, we assume $k_1=0.5$ and $k_2 = 0.5$.

Then we can get the following formulas:

$$X_i^t = F_3(F_2(F_1(X_i^{t-1}, w), c_1), c_2) \tag{26}$$

$$X_i^t = \begin{cases} X_i^t \quad , \quad X_i^t < X_i^{t-1} \ or \quad rand < upd \\ X_i^{t-1} \quad , \quad otherwise \end{cases} \tag{27}$$

3.2.5 FTPSO Algorithm Overview

First of all, in order to ensure that the corresponding network topology structure of each particle *X*(1:*n*) is connected, we make *X*(*i*) equals to *rmax*(*i*) ($1 \le i \le n$) in the population initialization. This algorithm FTPSO can be divided into two phases, which we described as follows:

The First Phase:

The sink node collects the initial network information and computes the optimal network topology according to the following steps:

Step 1: Initial the population and the relative parameters
Step 2: Update w, c_1, c_2 et al. according to (18~21)
Step 3: Update each particle's position in the population according to (22~27)
Step 4: Calculate the fitness value $f(x)$

Step 5: Update p_{besti} and g_{best} for each particle
Step 6: If the termination condition is not satisfied, go to Step 2
Step 7: Obtain g_{best} as the approximate optimal solution, and then the sink node adjusts the transmission power of each node according to g_{best}.

The Second Phase:
In order to get a more power-efficient network topology, we can adjust the radiation power of each sensor node again according to the final structure G_f generated by the first phase. Each node requires a power lever that can reach the farthest one-hop neighbor in G_f.

Of course, we can make our algorithm completely local like *FLSS*, if we are willing to lower the approximation guarantee. At this same time, we can remove uni-directional edges in order to get a final topology consisting of only bi-directional edges.

4 Simulation Results and Evaluations

In this section, the performance of the improved discrete PSO method applied to network optimization, were observed in comparison with some previous heuristic approaches. We compared the performance of FTPSO against 2-UPVCS, BICONN and FGSS$_2$ algorithms with respect to several metrics via simulations. In this paper, all the simulations of our improved PSO algorithm were carried out with a population size of 30, and the maximum number of iterations was set to 300. The inertial weight *w* is decreased linearly from 0.9 to 0.4. All the topology control schemes are implemented in MATLAB.

4.1 The First Simulations

Without lost of generality, in the first simulations, we simulate MaxPower (using maximum transmission power), MST (using Prim algorithm), BICONN, FGSS$_2$, 2-UPVCS and FTPSO using the same traffic pattern and random networks for performance measure. We generated 50 sensor nodes randomly distributed in a 100m × 100m region. The maximum transmission range of each sensor node is set as 26m. The final topologies derived by the different algorithms above are shown in Figure.1 respectively.

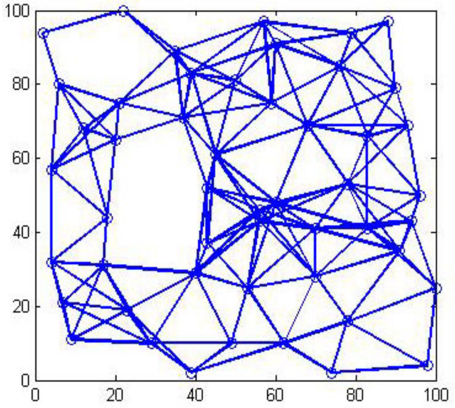


Fig. 1. MaxPower

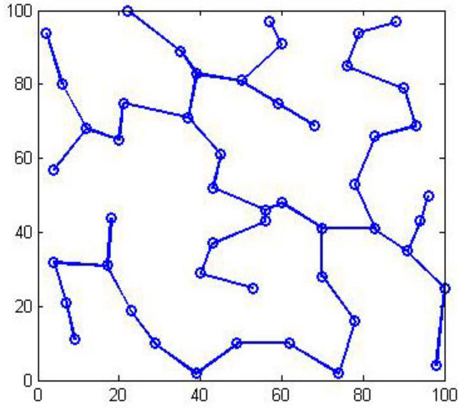


Fig. 2. MST

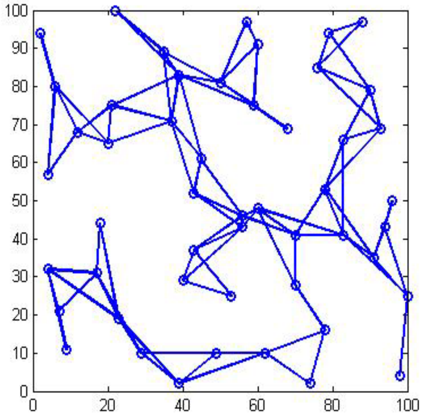


Fig. 3. 2-UPVCS

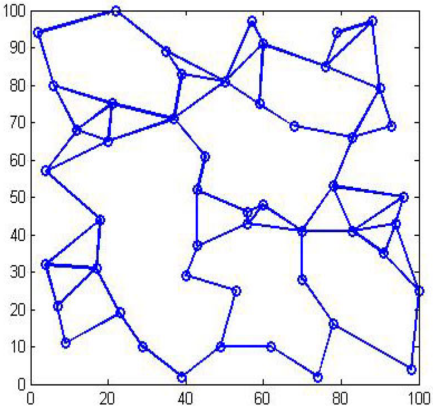


Fig. 4. BICONN

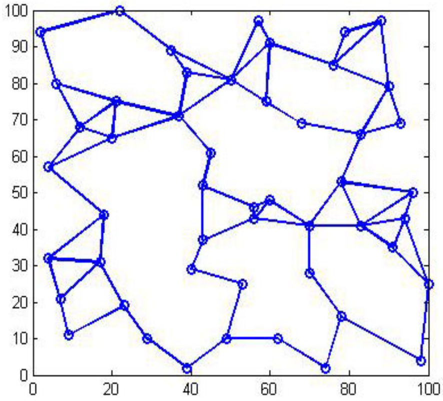


Fig. 5. FGSS$_2$

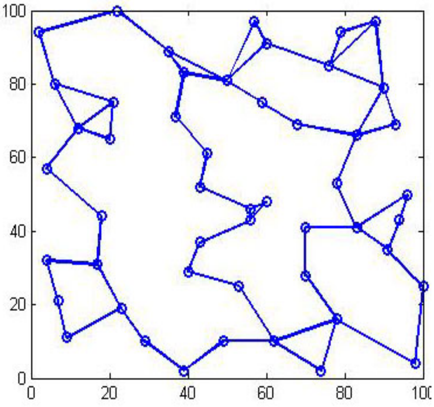


Fig. 6. FTPSO

Obviously, MST can only keep the property of 1-connectivity, and there are several cut points in the final network topology derived under 2-UPVCS. The fault-tolerant abilities of MST and 2-UPVCS are poor. By using smaller transmission power, the number of wireless links can be decreased in the networks. As shown in Fig.1 to Fig.6, FTPSO outperforms 2-UPVCS, BICONN and FGSS2 in the sense that fewer edges are formed in the final network topologies, so it can lower radio interference and provide a better spatial reuse.

4.2 The Second Simulations

Similar to literatures [6, 10], we test the performance of our algorithm using lots of simulations. In the experiments, variable numbers of sensor nodes from 30 to 80 are randomly deployed in the area, and the maximum transmission range of each node is set as 26m, too. The each data point is the average of 50 simulation samples.

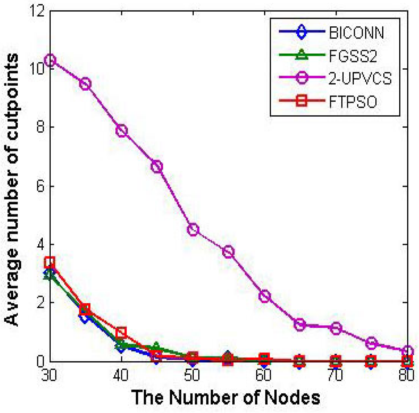


Fig. 7. Comparision of diffenent algorithms with respect to the average numbers of cut points

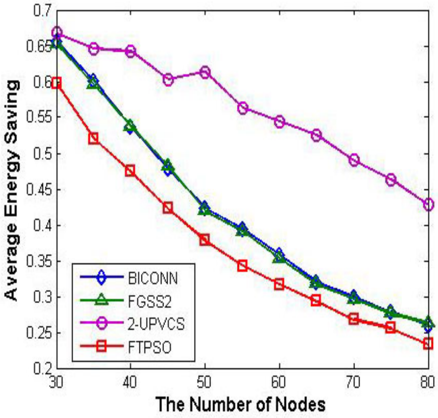


Fig. 8. Comparision of diffenent algorithms with respect to the average energy saving

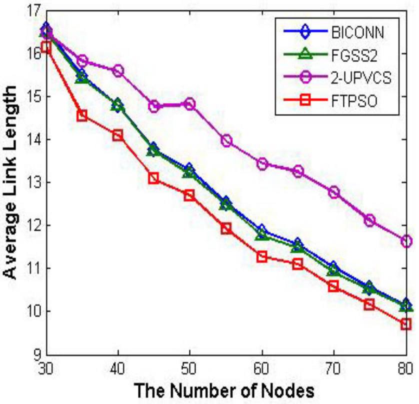


Fig. 9. Comparision of diffenent algorithms with respect to the average link length

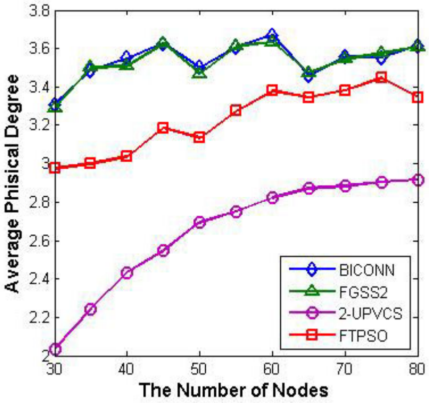


Fig. 10. Comparision of diffenent algorithms with respect to the average physical node degree

Firstly, we are interested in the ability of fault tolerance under different approaches. As shown in Fig.7, the average number of cut points derived under BICONN, FGSS$_2$, and FTPSO are similar with each other as the node density increase, and they are all smaller than 2-UPVCS. Obviously, the fault-tolerant ability derived under 2-UPVCS is so bad. The simulation results also imply that FTPSO can preserve the network property of 2-connectivity.

We also compare other network metrics for the network topologies derived under different algorithms. As shown in Fig.8 and Fig.9, they are the analysis of the average energy saving and the average link length respectively, from which we can conclude that our algorithm can generated a more power-efficient network topology than other heuristic approaches. Fig.10 shows that the average physical node degree under FTPSO is smaller than BICONN and $FGSS_2$, but larger than 2-UPVCS. Although the radio inference derived under 2-UPVCS is better than FTPSO, the core metrics of energy consumption and fault-tolerant ability under 2-UPVCS are worse than FTPSO.

All the above simulation results show that our algorithm FTPSO has made a trade off between power efficiency and fault tolerance, and it can also reduce radio interference effectively. Obviously, the performance of the final network topologies derived under FTPSO has been greatly improved.

5 Conclusion Remarks

Using 2-connecitivity and power-efficiency as our objectives, in this paper we have presented an application of DPSO for the fault tolerant topology control in WSN. As mentioned before, compared to previous heuristic approaches, FTPSO can provide better network topologies. Simulation results also showed that our algorithm can not only reduce the entire network energy consumption effectively, but also lower the number of cut points as possible. As part of our future researches, we will pursue the following problems: (1) how to analyze the reliability in WSN effectively; (2) how to construct a high-reliability network topology?

Acknowledgments. This work supported by the National Natural Science Foundation of China under Grant No.10871221, the Key Project of Fujian Provincial Natural Science Foundation of China under Grant No.A0820002, the Technology Innovation Platform Project of Fujian Province under Grant No.2009J1007, Fujian Provincial Natural Science Foundation of China under Grant No.2009J01284 and 2008F3063, the project development foundation of Education Committee of Fujian province under Grand No.JA08011 and JA09002.

References

1. Sun, L.M.: Wireless sensor networks. Tsinghua University Press, Beijing (2005)
2. Zhang, X., Lu, S.L., Chen, G.H., Chen, D.X., Xie, L.: Topology Control for Wireless Sensor Networks. Journal of Software 18(4), 943–954 (2007)
3. Sun, L.S., Zhang, R.H., Wu, W.B.: On constructing 2-connected 2-dominating set using two centralized algorithms. Computer Engineering and Applications 45(15), 107–110 (2009)
4. Yue, W., Cai, W.D., Duan, Q.: Computing Vulnerability of Network Based on Genetic Algorithm. Journal of System Simulation 21(6), 1628–1632 (2009)
5. Gao, S.X.: Graph theory and network flow theory. Higher Education Press, Beijing (2009)
6. Shen, Z., Chang, Y.L., Cui, C., Zhang, X.: A Topology Maintenance Algorithm Based on Shortest Path Tree for Wireless Ad hoc Networks. Journal of Electronics & Information Technology 29(2), 323–327 (2007)
7. Li, N., Hou, J.C.: FLSS: A fault-tolerant topology control algorithm for wireless networks. In: Proceedings of the Annual International Conference on Mobile Computing and Networking, pp. 275–286 (2004)

8. Sun, Y.J., Sun, Y.G., Chen, B.J., Fang, Z.H.: Research on the One Vertex and Two Verteices Connectivity Reliability in the Wireless Sensor Networks. Chinese Journal of Sensors and Actuators 17(3), 0379–0385 (2004)
9. Ram, R., Regina, R.H.: Topology control of multihop wireless networks using transmit power adjustment. In: Proc. IEEE INFOCOM, vol. 2, pp. 404–413 (2000)
10. Hajiaghayi, M., Nicole, N., Mirrokni, V.S.: Power Optimization in Fault-Tolerant Topology Control Algorithms for Wireless Multi-hop Networks. IEEE/ACM Transactions on Networking 15(6), 1345–1358 (2007)
11. Zhang, J., Yang, H., Zhu, A.: Topology Controlment Based on Genetic Algorithm in Static Ad-hoc Network. Wuhan University Journal (Natural Science Edition) 51(3), 327–332 (2005)
12. Sun, L.J., Guo, J., Lu, K., Wang, R.C.: Topology control based on quantum genetic algorithm in sensors network. Journal on Communications 27(12), 1–5 (2006)
13. Mitsuo, G., Runwei, C.: Genetic Algorithms and Engineering Optimization. Tsinghua University Press, Beijing (2004)
14. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. of the IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Service Center, Piscataway (1995)
15. Ratnaweera, A., Halgamuge, S.K.: Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients. IEEE Transactions on Evolutionary Computation 8, 240–255 (2004)
16. You, B.Y., Chen, G.L., Guo, W.Z.: Topology control in wireless sensor networks based on discrete particle swarm optimization. In: Proc. of IEEE International Conference on Intelligent Computing and Intelligent Systems, vol. 1, pp. 269–273 (2009)

Retracted: Feature Selection Using Ant Colony Optimization for Text-Independent Speaker Verification System

Javad Sohafi-Bonab[1] and Mehdi Hosseinzadeh Aghdam[2]

[1] Department of Computer Engineering, Islamic Azad University,
Bonab Branch, Bonab, Iran
[2] Department of Computer Engineering & IT, Payame Noor University,
Bonab, Iran
sohafi.j.b@gmail.com, hosseinzadeh@comp.ui.ac.ir

Abstract. With the growing trend toward remote security verification procedures for telephone banking, biometric security measures and similar applications, automatic speaker verification (ASV) has received a lot of attention in recent years. The complexity of ASV system and its verification time depends on the number of feature vectors, their dimensionality, the complexity of the speaker models and the number of speakers. In this paper, we concentrate on optimizing dimensionality of feature space by selecting relevant features. It presents another method that is based on ant colony optimization (ACO). The performance of the proposed algorithm is compared to the performance of genetic algorithm on the task of feature selection in TIMIT corpora. The results of experiments indicate that with the optimized feature set, the performance of the ASV system is improved.

Keywords: Ant colony optimization, Feature selection, Genetic algorithm, Speaker verification, Gaussian mixture model universal background model.

1 Introduction

Automatic speaker recognition (ASR) systems are generally divided into two categories, namely: automatic speaker identification (ASI) systems which are designed to answer the question “who is the speaker?” or automatic speaker verification systems that aim to answer the question “is the speaker who they claim to be?” Automatic speaker verification refers to the task of verifying speaker's identity using speaker-specific information contained in speech signal. Speaker verification methods are totally divided into text-dependent and text-independent applications. When the same text is used for both training and testing, the system is called to be text-dependent while for text-independent operation; the text used to train and test of the ASV system is completely unconstrained. Text independent speaker verification requires no restriction on the type of input speech. In contrast, text independent speaker verification usually gives less performance than text dependent speaker verification, which requires test input to be the same sentence as training data [1].

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 13–24, 2010.

Applications of speaker verification can be found in biometric person authentication such as an additional identity check during credit card payments over the Internet while, the potential applications of speaker identification can be found in multi-user systems. For instance, in speaker tracking the task is to locate the segments of given speaker(s) in an audio stream [2]. It has also potential applications in automatic segmentation of teleconferences and helping in the transcription of courtroom discussions.

Among too many methods which are proposed for FS, population-based optimization algorithms such as genetic algorithm (GA)-based method and ant colony optimization (ACO)-based method have attracted a lot of attention. These methods attempt to achieve better solutions by application of knowledge from previous iterations. Genetic algorithms are optimization techniques based on the mechanism of natural selection. They used operations found in natural genetics to guide itself through the paths in the search space [3]. Because of their advantages, recently, GAs have been widely used as a tool for feature selection in data mining [4].

ACO algorithm was firstly used for solving traveling salesman problem (TSP) [5] and then has been successfully applied to a large number of difficult problems like the quadratic assignment problem (QAP) [6], routing in telecommunication networks, graph coloring problems, scheduling, etc. This method is particularly attractive for feature selection as there seems to be no heuristic that can guide search to the optimal minimal subset every time [7].

This paper proposes an ACO algorithm for feature selection in ASV systems based on GMM-UBM and apply it to larger feature vectors containing Mel-frequency cepstral coefficients (MFCCs) and their delta coefficients, two energies and linear prediction cepstral coefficients (LPCCs) and their delta coefficients. Then, feature vectors are applied to a Gaussian mixture model universal background model (GMM-UBM) which is a text-independent speaker verification Model. Finally the verification quality and the length of selected feature vector are considered for performance evaluation.

2 An Overview of ASV Systems

The typical process in most proposed ASV systems involves: some form of preprocessing of the data (silence removal) and feature extraction, followed by some form of speaker modeling to estimate class dependent feature distributions. A comprehensive overview can be found in [8]. Adopting this strategy the ASV problem can be further divided into the two problem domains of:

- Preprocessing, feature extraction and selection
- Speaker modeling and matching.

These two steps are described in following sections in more details.

The original signal, the speech waveform, contains all information about the speaker, and each step in the extraction process can only reduce the mutual information or leave it unchanged. The objective of the feature extraction is to reduce the dimension of the input signal and thereby reduce the complexity of the system. The main task for the feature extraction process is to pack as much speaker-discriminating information as possible into as few features as possible.

The choice of features in any proposed ASV system is of primary concern, because if the feature set does not yield sufficient information then trying to estimate class dependent feature distributions is futile [9]. Most feature extraction techniques in speaker verification were originally used in speech recognition. However, the focus in using these techniques was shifted to extract features with high variability among people. Most commonly used features extraction techniques, such as Mel-frequency cepstral coefficients (MFCCs) and linear prediction cepstral coefficients (LPCCs) have been particularly popular for ASV systems in recent years. This transforms give a highly compact representation of the spectral envelope of a sound [10].

The speaker modeling stage of the process varies more in the literature. The purpose of speaker modeling is characterizing an individual which is enrolled into an ASV system with the aim of defining a model (usually feature distribution values). The three most popular methods in previous works are Gaussian mixture models (GMM) [10], Gaussian mixture models universal background model (GMM-UBM) [11] and vector quantization [12]. Other techniques such as decision trees [13], support vector machine (SVM) [14] and artificial neural network (ANN) [15] have also been applied. In this paper GMM-UBM is used for speaker modeling.

2.1 GMM-UBM Approach

GMM-UBM is the predominant approach used in speaker recognition systems, particularly for text-independent task [11]. Given a segment of speech *Y* and a speaker *S*, the speaker verification task consists in determining if *Y* was spoken by *S* or not. This task is often stated as basic hypothesis test between two hypotheses: *Y* is from the hypothesized speaker $S(H_0)$, and *Y* is not from the hypothesized speaker $S(H_1)$. A likelihood ratio (*LR*) between these two hypotheses is estimated and compared to a decision threshold *Φ*. The LR test is given by:

$$LR(Y,H_0,H_1)=\frac{p(Y\mid H_0)}{p(Y\mid H_1)} \tag{1}$$

Where *Y* is the observed speech segment, $p(Y|H_0)$ is the likelihood function for the hypothesis H_0 evaluated for *Y*, $p(Y|H_1)$ is the likelihood function for H_1 and Φ is the decision threshold for accepting or rejecting H_0. If $LR(Y,H_0,H_1)>\Phi$, H_0 is accepted else H_0 is rejected. A model denoted¸ λ_{hyp} represents H_0, it is learned using an extract of speaker *S* voice. The model λ_{UBM} represents the alternative hypothesis, H_1, and is usually learned using data gathered from a large set of speakers.

The likelihood ratio statistic becomes $\frac{p(Y\mid\lambda_{hyp})}{p(Y\mid\lambda_{UBM})}$. Often, the logarithm of this statistic is used giving the Log *LR* (*LLR*):

$$LLR(Y)=\log p(Y\mid\lambda_{hyp})-\log p(Y\mid\lambda_{UBM}) \tag{2}$$

In the presented approach, the models are Gaussian Mixture Models which estimate a probability density function by:

$$p(x \mid \lambda) = \sum_{i=1}^{M} w_i N(x_t, \mu_i, \Sigma_i) \tag{3}$$

Where $w_i, i = 1, \cdots .c$ are the mixture weights that satisfy $0 \le w_i \le 1, \sum_{i=1}^{c} w_i = 1$ and $N(x_t, \mu_i, \Sigma_i)$ are the d-variate Gaussian component densities with mean vectors μ_i and covariance matrices Σ_i

$$N(x_t, \mu_i, \Sigma_i) = \frac{\exp\left\{-\frac{1}{2}(x_t - \mu_i)'\Sigma_i^{-1}(x_t - \mu_i)\right\}}{(2\pi)^{d/2}\left|\Sigma_i\right|^{1/2}} \tag{4}$$

Usually a large number of components in the mixture and diagonal covariance matrices are used.

2.2 Universal Background Model

The UBM has been introduced and successfully applied to speaker verification. It aims at representing the inverse hypothesis in the Bayesian test, i.e. it is designed to compute the data probability not to belong to the targeted speaker, i.e. λ_{hyp}. A UBM is learned with multiple audio files from different speakers, usually several hundreds. For speaker verification, some approaches consist in having specific UBM models, such as a UBM model per gender or per channel.

The UBM is trained with the EM algorithm on its training data. For the speaker verification process, it fulfills two main roles:

- It is the a priori model for all target speakers when applying Bayesian adaptation to derive speaker models.
- It helps to compute logarithm likelihood ratio much faster by selecting the best Gaussian for each frame on which likelihood is relevant [11].

3 Ant Colony Optimization

In the early 1990s, ant colony optimization was introduced by M. Dorigo and colleagues as a novel nature-inspired meta-heuristic for the solution of hard combinatorial optimization problems. An ant colony optimization algorithm is essentially a system based on agents which simulate the natural behavior of ants, including mechanisms of cooperation and adaptation. The inspiring source of ACO is the foraging behavior of real ants [16].

The ACO algorithm is based on a computational paradigm inspired by real ant colonies and the way they function. The underlying idea was to use several constructive computational agents (simulating real ants). A dynamic memory structure incorporating information on the effectiveness of previous choices based on the obtained results, guides the construction process of each agent.

The paradigm is based on the observation made by ethologists about the medium used by ants to communicate information regarding shortest paths to food by means of pheromone trails. A moving ant lays some pheromone on the ground, thus making a path by a trail of this substance. While an isolated ant moves practically at random, exploration, an ant encountering a previously laid trail can detect it and decide with high probability to follow it, exploitation, and consequently reinforces the trail with its own pheromone. What emerges is a form of autocatalytic process through which the more the ants follow a trail, the more attractive that trail becomes to be followed. The process is thus characterized by a positive feedback loop, during which the probability of choosing a path increases with the number of ants that previously chose the same path. The mechanism above is the inspiration for the algorithms of the ACO family [17].

4 ACO Algorithm for Feature Selection

As mentioned earlier given a feature set of size *n*, the FS problem is to find a minimal feature subset of size $s(s<n)$ while retaining a suitably high accuracy in representing the original features. Therefore, there is no concept of path. A partial solution does not define any ordering among the components of the solution, and the next component to be selected is not necessarily influenced by the last component added to the partial solution [18]. Furthermore, solutions to an FS problem are not necessarily of the same size. To apply an ACO algorithm to solve a feature selection problem, these aspects need to be addressed. The first problem is addressed by redefining the way that the representation graph is used.

The feature selection problem may be reformulated into an ACO-suitable problem. The main idea of ACO is to model a problem as the search for a minimum cost path in a graph. Here nodes represent features, with the edges between them denoting the choice of the next feature. The search for the optimal feature subset is then an ant traversal through the graph where a minimum number of nodes are visited that satisfies the traversal stopping criterion.

The basic ingredient of any ACO algorithm is a constructive heuristic for probabilistically constructing solutions. A constructive heuristic assembles solutions as sequences of elements from the finite set of solution components. A solution construction starts with an empty partial solution. Then, at each construction step the current partial solution is extended by adding a feasible solution component from the set of solution components [16]. A suitable heuristic desirability of traversing between features could be any subset evaluation function for example, an entropy-based measure or rough set dependency measure [19]. In proposed algorithm verification quality is mentioned as heuristic information for feature selection. The heuristic desirability of traversal and node pheromone levels are combined to form the so-called ***probabilistic transition rule***, denoting the probability that ant *k* will include feature *i* in its solution at time step *t*:

$$P_i^k(t) = \begin{cases} \dfrac{[\tau_i(t)]^{\alpha}.[\eta_i]^{\beta}}{\sum\limits_{u \in J^k}[\tau_u(t)]^{\alpha}.[\eta_u]^{\beta}} & if\ i \in J^k \\ 0 & otherwise \end{cases} \qquad (5)$$

where, J^k is the set of feasible features that can be added to the partial solution; τ_i and η_i are respectively the pheromone value and heuristic desirability associated with feature *i*. α and β are two parameters that determine the relative importance of the pheromone value and heuristic information.

After all ants have completed their solutions, pheromone evaporation on all nodes is triggered, and then according to equation (6) each ant *k* deposits a quantity of pheromone, $\Delta\tau_i^k(t)$, on each node that it has used.

$$\Delta\tau_i^k(t)=\begin{cases}\omega.\gamma(S^k(t))+\varphi.(n/|S^k(t)|) & if\ i\in S^k(t)\\ 0 & otherwise\end{cases} \qquad (6)$$

Where $S^k(t)$ is the feature subset found by ant *k* at iteration *t*, and $|S^k(t)|$ is its length. The pheromone is updated according to both the measure of the verification quality, $\gamma(S^k(t))$, and feature subset length. ω and φ are two parameters that control the relative weight of verification quality and feature subset length, $\omega\in$ [0,1] and $\varphi=1-\omega$. This formula means that the verification quality and feature subset length have different significance for feature selection task. In our experiment we assume that verification quality is more important than subset length, so they were set as ω =0.7, φ=0.3.

In practice, the addition of new pheromone by ants and pheromone evaporation are implemented by the following rule applied to all the nodes:

$$\tau_i(t+1)=(1-\rho)\tau_i(t)+\Delta\tau_i^g(t)+\sum\nolimits_{k=1}^{m}\Delta\tau_i^k(t) \qquad (7)$$

Where, *m* is the number of ants at each iteration and $\rho\in$ (0,1) is the pheromone trail decay coefficient. The main role of pheromone evaporation is to avoid stagnation, that is, the situation in which all ants constructing the same solution. *g* indicates the best ant at each iteration. All ants can update the pheromone according to equation (7) and the best ant deposits additional pheromone on nodes of the best solution. This leads to the exploration of ants around the optimal solution in next iterations.

The overall process of ACO feature selection can be seen in Figure 1. The process begins by generating a number of ants which are then placed randomly on the graph i.e. each ant starts with one random feature. Alternatively, the number of ants to place on the graph may be set equal to the number of features within the data; each ant starts path construction at a different feature. From these initial positions, they traverse nodes probabilistically until a traversal stopping criterion is satisfied. The resulting subsets are gathered and then evaluated. If an optimal subset has been found or the algorithm has executed a certain number of times, then the process halts and outputs the best feature subset encountered. If none of these conditions hold, then the pheromone is updated, a new set of ants are created and the process iterates once more.

Typically an ASV system consists of several essential parts including feature extraction and feature selection. After preprocessing of speech signals, feature extraction is used to transform the input signals into a feature set (feature vector). Feature selection is applied to the feature set to reduce the dimensionality of it. ACO is used

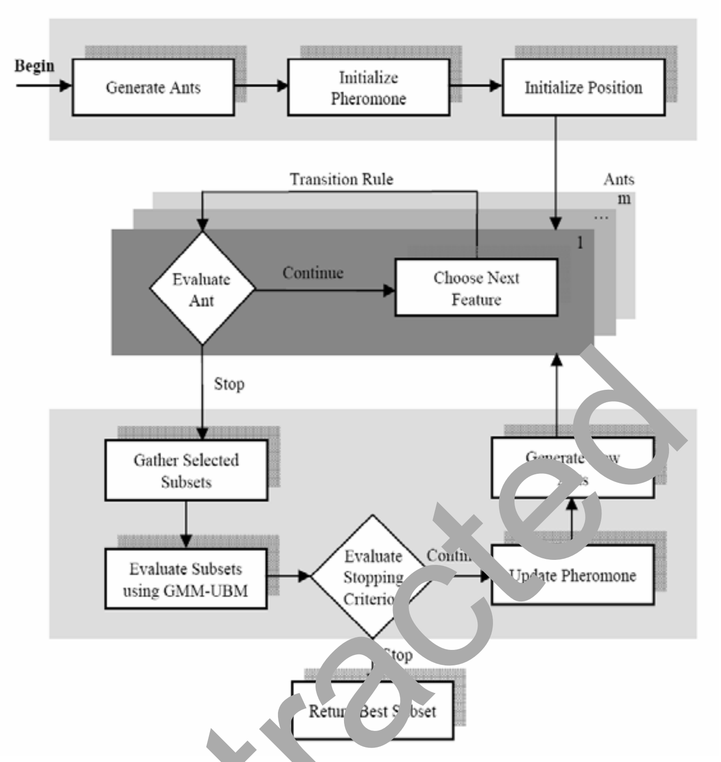


Fig. 1. Overall process of ACO for feature selection in ASV

to explore the space of all subsets of given feature set. The performance of selected feature subsets is measured by invoking an evaluation function with the corresponding reduced feature space and measuring the specified verification result. The best feature subset found is then output as the recommended set of features to be used in the actual design of the ASV system.

5 Experimental Results

The TIMIT corpus [20] is used in this paper. This corpus contains 630 speakers (438 male and 192 female) representing 8 major dialect regions of the United States, each speaking ten sentences. There are two sentences that are spoken by all speakers and the remaining eight are selected randomly from a large database.

The speech signal is recorded through a high quality microphone with a sampling frequency of 16 kHz in a quiet environment, with no session interval between recordings. Eight sentences (SX, SI) were used to develop each speaker model, and the remaining 2 SA sentences were used to test each speaker. The 40 speakers included in both the test and train directories were used during the TIMIT(40) trials.

The evaluation of the speaker verification system is based on detection error tradeoff (DET) curves, which show the tradeoff between false alarm (*FA*) and false rejection (*FR*) errors. Typically equal error rate (EER), which is the point on the curve where *FA* = *FR*, is chosen as evaluation measure. We also used detection cost function (DCF) defined as [21]:

$$DCF = C_{miss}.FRR.P_{t\arg et} + C_{FA}.FAR.(1 - P_{t\arg et}) \tag{8}$$

Where P_{target} is the priori probability of target tests with $P_{t\arg et} = 0.01$, *FRR* and *FAR* are false rejection rate and false acceptance rate respectively at an operating point and the specific cost factors $C_{miss} = 10$ and $C_{FA} = 1$. Hence, the point of interest is shifted towards low *FA* rates.

Various values were tested for the parameters of proposed algorithm. The results show that the highest performance is achieved by setting the parameters to values shown in table 1.

Table 1. GA and ACO Parameter Settings

	Population	Iteration	Crossover probability	Mutation probability	Initial pheromone	α	β	ρ
GA	30	50	0.7	0.005	-	-	-	-
ACO	30	50	-	-	1	1	0.1	0.2

Experiments were conducted on a subset of TIMIT corpora consist of 24 male and 16 female speakers of different accent that were selected randomly. Data were processed in 20 ms frames (320 samples) with 50% overlaps. Frames were segmented by Hamming window and pre-emphasized with $\alpha = 0.97$ to compensate the effect of microphone's low pass filter. At first, feature vector was created by extracting MFCCs from silence removed data for each frame. In the next step, delta coefficients were calculated based on the MFCCs and appended to existing feature vector. Furthermore, two energies were applied to input vectors as described earlier. Then we consider the LPCCs and their delta coefficients respectively and append them to the feature vector. The final feature set contains *F*=50 features. Table 2 shows the overall set of features.

Finally, verification process was performed using the GMM-UBM approach. The performance criterion is due to EER and DCF according to an adopted decision threshold strategy.

Table 2. The Overall Feature Set

Feature Name	Order
Mel Frequency Cepstral Coefficient (MFCC)	12
Linear Prediction Cepstral Coefficient (LPCC)	12
First diff of MFCC (Δ - MFCC)	12
First diff of LPCC(Δ - LPCC)	12
Energy	2
Total	50

The feature subset length and verification quality are two criteria which are considered to assess the performance of algorithms. Comparing the first criterion, we noted that both ACO-based and GA-based algorithms reduce the dimensionality of feature space. Furthermore, the ACO-based algorithm selected a smaller subset of features than the GA-based algorithm. Table 3 shows the number of selected features by ACO-based and GA-based approaches. As we can see in table 3, ACO can degrade dimensionality of features over 80%.

Table 3. Selected Features of GA and ACO

Selection Method	Number of Selected Features	Percentages of Selected Features
GA- GMM-UBM(16)	16	32%
GA-GMM-UBM(32)	17	34%
GA-GMM-UBM(64)	16	32%
ACO-GMM-UBM(16)	10	20%
ACO-GMM-UBM(32)	9	18%
ACO-GMM-UBM(64)	10	20%

The second performance criterion is due to EER and DCF according to an adopted decision threshold strategy. The EER and DCF for GMM-UBM, GA-based and ACO-based algorithms with different number of Gaussian (16, 32 and 64) were shown in table 4.

Table 4. EER and DCF for GMM-UBM, GA-based and ACO-based Algorithms with Different Number of Gaussians.

Number of Gaussians	GMM-UBM		GA		ACO	
	EER	DCF	EER	DCF	EER	DCF
16	5.08	0.0563	4.966	0.0554	4.318	0.0401
32	4.56	0.0505	3.679	0.0389	2.634	0.0309
64	6.667	0.0707	4.961	0.0532	4.23	0.0386

Ideally the number of mixtures, *M*, should approximate the number of natural classes in data. If *M* is less than the number of natural classes, closely placed clusters will fuse to form larger clusters whose variance is higher. This results in a larger percentage of frames, both the true speaker and imposter, getting average scores. The performance goes down because these frames cannot be discriminated. This is a case of trying to under-fit training data. When *M* is more than the number of natural classes, larger clusters are broken up into smaller sub-clusters or spurious frames get represented as separate clusters. The new clusters will have lower variance. This would lead to increase in average and low scoring frames and hence would lead to lower performance. This is a case of trying to over-fit training data. As could be seen in table 4, speaker verification performance is found to increase as *M* is increased but begins to drop after a point at which over-fitting starts taking place.

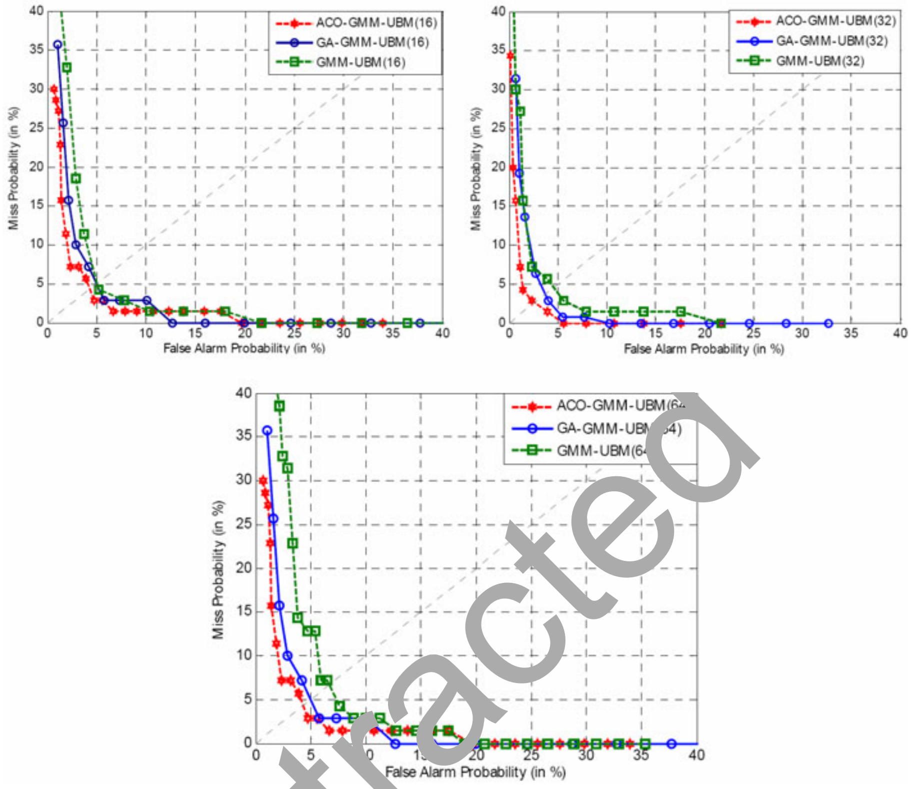


Fig. 2. DET curves for GMM-UBM, GA and ACO with 16, 32 and 64 Gaussians

DET curves for GA-based and ACO-based algorithms with 16, 32 and 64 Gaussians are shown in Figure 2. From the results, it can be seen that ACO-GMM-UBM yields a significant improvement in speed than the baseline GMM-UBM approach. The improvement is due to the selection of optimal feature set by ACO algorithm.

From the results and figures, we can see that, compared with GA, ACO is quicker in locating the optimal solution. In general, it can find the optimal solution within tens of iterations. If exhaustive search is used to find the optimal feature subset in the TIMIT dataset, there will be tens of billions of candidate subsets, which is impossible to execute. But with ACO, at the 50th iteration the optimal solution is found.

Ant colony optimization has powerful exploration ability; it is a gradual searching process that approaches optimal solutions. The running time of ACO is affected more by the problem dimension (feature numbers), and the size of data. For some datasets with more features, after finding a sub-optimal solution, the GA cannot find a better one. However, ACO can search in the feature space until the optimal solution is found. The GA is affected greatly by the number of features.

Ant colony optimization comprises a very simple concept, and the ideas can be implemented in a few lines of computer code. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed. This optimization technique does not suffer, however, from some of the difficulties of GAs; interaction in the colony enhances rather than detracts from progress toward the solution. Further, an ant colony system has memory, which the genetic algorithm does not have. Changes in genetic populations result in the destruction of previous knowledge of the problem. In ant colony optimization, ants that past optima are tugged to return towards them; knowledge of good solutions is retained by all ants.

6 Conclusion

In this paper, we have addressed the problem of optimizing the acoustic feature set by ACO technique for text-independent speaker verification system based on GMM-UBM. Ant colony optimization selected the relevant features among all Mel-cepstrum coefficients in order to increase the performance of our ASV system. We compare its performance with another prominent population-based feature selection method, genetic algorithm. The experimental results on subset of TIMIT database showed that ACO is able to select the more informative features without loosing the performance than GA. The feature vectors size reduced over 80% which led to a less complexity of our ASV system. Moreover, verification process in the test phase speeds up because less complexity is achieved by the proposed system in comparison with current ASV systems.

References

1. Xiang, B., Berger, T.: Efficient Text-Independent Speaker Verification with Structural Gaussian Mixture Models and Neural Network. IEEE Trans. Speech and Audio Processing 11(5) (2003)
2. Lapidot, I., Guterman, H., Cohen, A.: Unsupervised speaker recognition based on competition between self-organizing maps. IEEE Trans. Neural Networks 13(2), 877–887 (2002)
3. Srinivas, M., Patnik, L.M.: Genetic Algorithms: A Survey. IEEE Computer Society Press, Los lamitos (1994)
4. Haydar, A., Demirekler, M., Yurtseven, M.K.: Feature selection using genetic algorithm and its application to speaker verification. Electron. Lett. 34(15), 1457–1459 (1998)
5. Dorigo, M., Maniezzo, V., Colorni, A.: Ant System: Optimization by a colony of cooperating agents. IEEE Trans. Systems, Man, and Cybernetics-Part B 26(1), 29–41 (1996)
6. Maniezzo, V., Colorni, A.: The Ant System Applied to the Quadratic Assignment Problem. IEEE Trans. Knowledge and Data Engineering 11(5), 769–778 (1999)
7. Aghdam, M.H., Ghasem-aghaee, N., Basiri, M.E.: Text Feature Selection Using Ant Colony Optimization. Int. Expert Systems with Applications 36(3), 6843–6853 (2009)
8. Bimbot, F.: A tutorial on text-independent speaker verification. Eurasip Journal on Applied Signal Processing, 430–451 (2004)
9. Basiri, M.E., Ghasem-Aghaee, N., Aghdam, M.H.: Using ant colony optimization-based selected features for predicting post-synaptic activity in proteins. In: Marchiori, E., Moore, J.H. (eds.) EvoBIO 2008. LNCS, vol. 4973, pp. 12–23. Springer, Heidelberg (2008)
10. Cheung-chi, L.: GMM-Based Speaker Recognition for Mobile Embedded Systems. Ph.D. thesis, Univ. of Hong Kong (2004)
11. Neiberg, D.: Text Independent speaker verification using adapted Gaussian mixture models. Ph.D. thesis, Centre for Speech Technology (CTT) Department of Speech, Music and Hearing KTH, Stockholm, Sweden (2001)

12. Linde, Y., Buzo, A., Gray, R.: An Algorithm for Vector Quantizer Design. IEEE Trans. Communications 28(1), 84–95 (1980)
13. Navratil, J., Jin, Q., Andrews, W.D.: Campbell, Phonetic speaker recognition using maximum-likelihood binary-decision tree models. In: Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Hong Kong (2003)
14. Wan, V.: Speaker Verification Using Support Vector Machines. Ph.D. thesis, Univ. Sheffield, U.K (2003)
15. Wouhaybi, R., Al-Alaou, M.A.: Comparison of neural networks for speaker recognition. In: Proc. 6th IEEE Int. Conf. Electronics, Circuits Systems, pp. 125–128 (1999)
16. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. Theoretical Computer Science, 243–278 (2005)
17. Montemanni, R., Gambardella, L.M., Rizzoli, A.E., Donati, A.V.: A new algorithm for a Dynamic Vehicle Routing Problem based on Ant Colony System. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, Technical Report IDSIA-23-02 (2002)
18. Blum, C., Dorigo, M.: The hyper-cube framework for ant colony optimization. IEEE Trans. Systems, Man, and Cybernetics -Part B 34(2), 1161–1172 (2004)
19. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishing, Dordrecht (1991)
20. Garofolo, J.: DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM. National Institute of Standards and Technology (1990)
21. Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M.: The DET Curve in Assessment of Detection Task Performance. In: Proc. Euro. speech, vol. 4, pp. 1895–1898 (1997)

GPS Height Fitting Using Gene Expression Programming

Xuezhi Yue[1,2], Zhijian Wu[1], Dazhi Jiang[3], and Kangshun Li[2]

[1] State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072
[2] School of Science, Jiangxi University of Science and Technology, Ganzhou 341000
[3] School of Science, Shantou University, Shantou 515000
mount_yue@yahoo.com.cn

Abstract. In Global Position System (GPS) height fitting methods, the traditional mathematical model fittings are more stable and general, but the fitting accuracy is usually not intended because of the error of model itself. Gene Expression Programming (GEP) as a kind of newly invented Genotype/phenotype based genetic algorithm can conquer the problem effectively. A GPS height fitting method based on GEP is given in this paper. By experiments and making the analysis and comparison with conicoid function and polyhedral function fitting methods, the results indicate that the GPS height fitting method based on GEP is effective and has better accuracy than traditional mathematical model methods to some extent.

Keywords: Gene Expression Programming; GPS height fitting; conicoid function fitting; polyhedral function fitting.

1 Introduction

Global Positioning System (GPS) has been widely applied in engineering survey especially for the large construction control networks. It can get a high precision when used in the plane position, but with a poor precision in surface position. To improve the accuracy of height, there are a lot of mathematical models in GPS height fitting including polynomial curve and surface function, polyhedral function, spline function and neural network model and so on[1]. All these traditional mathematical model fittings are more stable and general, but the fitting accuracy is usually not intended because of the error of model itself.

Gene Expression Programming (GEP) is a kind of new genetic algorithm based on genotype/phenotype by Candida Ferreira [2]. In recent years, Candida Ferreira has solved many problems effectively by GEP including function parameter optimization, evolutionary modeling, neural network, classification, and TSP problem and so on. In this paper a new GPS height fitting method based on GEP is given to improve the accuracy of GPS height fitting because of the super modeling feature.

The paper is organized as follows. Section 2 describes the principal of GPS height fitting. Section 3 gives the algorithm procedure of modeling using GEP. Section 4 applies the fitting method based on GEP to the examples and make analysis and comparison with the traditional mathematical model and Section 5 presents the conclusions.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 25–32, 2010.

2 GPS Height Fitting

Data collected by GPS is geodetic height data under WGS84 coordinate, however, that in our country is the normal height in quasigeoid. So there is a mapping between the two different kinds of data. In engineering applications, geodetic height is always represented by H, normal height is represented by h, height anomaly is represented by ξ, and the relationship among the three data represented as $\xi = H - h$ [1]. In a specific area, if a number of leveling points (i.e., normal height) are given, GPS survey can be carried on these points to calculate corresponding height anomaly for each point. Then a function model describing the relationship between height anomaly and position of the given points can be built to simulate the quasigeoid height in the area. So the height anomaly of each point in the area can be got by interpolation methods, and the normal height of each point can also be got.

2.1 GPS Height Fitting Methods

In terms of the different mathematical models, GPS height fitting methods mainly include the followings: curve fitting, conicoid function fitting, polyhedral function fitting, spline function fitting and neural network model and so on. Now conicoid function fitting and polyhedral function fitting are introduced.

The principle of conicoid function fitting[1] is that the height anomaly is approximately regarded as the surface function of every position within a certain range when the value of the height anomaly is stable. The height anomaly and the normal height of other points can be calculated based on the function.

$$\xi = a_0 + a_1\Delta x_0 + a_2\Delta y_0 + a_3\Delta x_0{}^2 + a_4\Delta y_0{}^2 + a_5\Delta x_0\Delta y_0 + \varepsilon \tag{1}$$

where a_0 is the height anomaly of reference point; a_1, a_2 is the vertical deviation of x, y direction of reference point; a_3, a_4, a_5 is the gradient of the vertical deviation; $\Delta x_0, \Delta y_0$ is the difference between each point and reference point P_0.

The principle of polyhedral function fitting[1] is that a mathematical surface adjustment problem of data points is solved from the geometric point of view when any smooth mathematical surface is always approximate to the sum of a series of regular mathematical surface.

$$\xi(x, y) = \sum_{i=1}^{n} a_i Q(x, y, x_i, y_i) \tag{2}$$

Where a_i is undetermined coefficient; $Q(x, y, x_i, y_i)$ is kernal function of second degree of X and Y ; the center of kernal function is (x_i, y_i).

The kernal function can be arbitrarily selected, the normal kernal function of second degree:

$$Q(x, y, x_i, y_i) = \left[(x - x_i)^2 + (y - y_i)^2 + d^2\right]^k \tag{3}$$

where d is an arbitrary constant called smooth factor whose value is one usually; $Q(x, y, x_i, y_i)$ is positive hyperbolic function when $k = \frac{1}{2}$▯ $Q(x, y, x_i, y_i)$ is reverse hyperbolic function when $k = -\frac{1}{2}$; (x, y) is the coordinate of unknown point; (x_i, y_i) is the coordinate of given point.

2.2 Index of Fitting Accuracy

There are two kinds of norms respectively called internal accuracy and external accuracy.

The formula about internal accuracy is:

$$\mu = \pm\sqrt{[vv] \big/ n_1 - 1} \tag{4}$$

where v is a vector whose component called residual error $v_i = \xi_i - \xi_i', i = 1,2,\ldots,n_1$, ζ_i is the height anomaly of the *i*th given point, ζ_i' is the fitting height anomaly of the *i*th given point after interpolation, n_1 is the number of given points.

The formula about external accuracy is:

$$m = \pm\sqrt{[vv] \big/ n_2 - 1} \tag{5}$$

where v is a vector whose component called residual error $v_i = \xi_i - \xi_i', i = 1,2,\ldots,n_2$, ζ_i is the height anomaly of the *i*th check point, ζ_i' is the fitting height anomaly of the *i*th check point after interpolation, n_1 is the number of check points.

3 GPS Fitting Algorithm Based on GEP

Gene expression programming (GEP) is, like genetic algorithms (GAs) and genetic programming (GP), a genetic algorithm as it uses populations of individuals, selects them according to fitness, and introduces genetic variation using one or more genetic operators. The implementation techniques of GEP mainly have three parts: encoded mode, genetic operators and fitness function choice.

Step 1: Select the function set $F = \{+, -, \times, /, S, C, L, E, Q\}$ and the terminal set $T = \{a, b\}$, S denotes sine function, C denotes cosine function, L denotes the natural logarithm, E denotes exponential function, Q denotes square root function. *a* denotes X coordinate, *b* denotes Y coordinate; In this case, $n = 2$; and we set the head length of gene $h = 8$, then $t = h + (n-1) = 9$,the length of the gene is $h + t = 8 + 9 = 17$.

Step 2: Initial population and randomly produce 100 chromosomes, each chromosome is made of five genes. The joint function among genes is sum. For the calculation of genes, Gene Read & Compute Machine(GRCM) in reference [3] is used to compute the fitness directly without transforming the genotype into expression tree.

Step 3: Calculate the fitness of each chromosome based on relative error and sort them, the best individual is preserved; In the paper the fitness function based on relative error is used:

$$f_i = \sum_{j=1}^{C_i} (M - \left| \frac{C_{(i,j)} - T_{(j)}}{T_{(j)}} \cdot 100 \right|) \tag{6}$$

where M is an constant that is the maximum of each fitness value f_i, $C(i,j)$ the value returned by the individual chromosome i for fitness case j (out of C_t fitness cases) and $T_{(j)}$ is the target value for fitness case j ; C_t is the number of fitness cases.

Step 4: Carry out the genetic operations. The genetic operations include mutation, insertion sequence elements, root IS elements, gene operation, one-point recombination, two-points recombination, gene recombination. Set $p_m = 0.044$,

$$p_{IS} = p_{RIS} = p_G = 0.1, p_{1r} = p_{2r} = 0.3, p_{gr} = 0.1;$$

where p_m is the probability of mutation, p_{IS} is the probability of insertion sequence elements, p_{RIS} is the probability of root IS elements, p_G is the probability of gene operation, p_{1r} is the probability of one-point recombination, p_{2r} is the probability of two-points recombination, p_{gr} is the probability of gene recombination.

Step 5: Output the result and save them if the maximum generation or the best fitness is achieved , continue step3 or else. In this paper the maximum generation is 1000, the best fitness is $C_t M$.

The difference between GEP modeling and traditional mathematical modeling is that some structure units of model is needed to be determined by the feature of problem in GEP modeling but the structure of model must to be determined in traditional mathematical modeling at first, secondly, The optimum of problem is searched through evolutionary algorithm in GEP modeling [5].

4 Data Experiments

Experiment 1: In a control network, 17 GPS control points are measured in reference[6]. Among these control points, 8 points named A01,A05,A06,A07,A08, A15,A18,A20 are fitting points, other 9 points named A02,A04,A09,A10, A11,A12,A13,A14,A16 are predication points. In table 1, data of 8 fitting points including X coordinate, Y coordinate, geodetic height, normal height and height anormaly are listed. In table 2, data of 9 predication points are also listed in detail.

Table 1. Data of fitting points A01,A05,A06,A07,A08,A15,A18,A20

Number	X coordinate	Y coordinate	geodetic height	Normal height	Height anomaly
A01	3521183.6484	500891.8669	19.655	9.8863	9.7687
A05	3521322.0221	502115.5435	18.511	8.6972	9.8138
A06	3514949.2321	501602.0906	18.927	9.0875	9.8395
A07	3521163.9205	498930.0555	18.647	8.9463	9.7007
A08	3515215.5761	498496.1918	18.700	8.9569	9.7431
A15	3522555.8199	502019.6094	16.494	6.6907	9.8033
A18	3514332.5668	499328.6312	15.755	5.9957	9.7593
A20	3513632.6633	501089.3235	16.394	6.5587	9.8353

Table 2. Data of predication points A02,A04,A09,A10,A11,A12,A13,A14,A16

Number	X coordinate	Y coordinate	geodetic height	Normal height	Height anormaly
A02	3515101.7750	500405.4400	18.766	8.9670	9.7990
A04	3515266.4573	499288.1836	18.801	9.0472	9.7538
A09	3521729.1624	503176.9303	18.921	9.0618	9.8592
A10	3514672.9886	502272.9326	18.426	8.5527	9.8733
A11	3522171.4953	500970.8756	16.578	6.8147	9.7633
A12	3513964.6180	500250.4336	17.272	7.4893	9.7827
A13	3521987.6092	499737.3555	16.331	6.6292	9.7018
A14	3514045.6710	499746.5023	15.201	5.4211	9.7799
A16	3513756.9532	500651.7743	17.042	7.2243	9.8177

The mathematical model is as followed:

$$\xi = F(x,y) = \frac{\cos\sqrt{\sqrt{\sqrt{x}}}}{\cos(\cos(\frac{x}{y})} + \ln\sqrt{e^{(\frac{\cos y}{\sqrt{x}} - \ln x)}} + \ln(x + \sqrt{2x} + \sqrt{x+y}) \tag{7}$$

$$+ \sin(\sin(\mathrm{co\ \ s}(\sqrt{\frac{x}{\sqrt{y}}}))) + \cos\sqrt{\sqrt{x}}$$

where ξ denotes the height anomaly, x denotes parameter a, y denotes parameter b.

Table 3. The fitting data of Height anomaly and residual error of predication points A02,A04, A09,A10,A11,A12,A13,A14,A16

Number	Height anomaly by polyhedral function fitting	Residual error	Height anomaly by GEP fitting	Residual error
A02	9.8001	-0.0011	9.7963	0.0027
A04	9.7537	0.0001	9.7446	0.0092
A09	9.8662	-0.0070	9.8587	0.0005
A10	9.8744	-0.0011	9.8815	-0.0082
A11	9.7630	0.0003	9.7565	0.0068
A12	9.7678	0.0149	9.7993	-0.0166
A13	9.6854	0.0164	9.7017	0.0001
A14	9.7801	-0.0002	9.7764	0.0035
A16	9.8193	-0.0016	9.8191	-0.0014
External accuracy		±0.0083		±0.0079

From above tables, it shows that external accuracy based on GEP is 0.0079,and increased 4% than that based on polyhedral function fitting.

Experiment 2: In a GPS control network of smooth terrain along the river,17 GPS control points are measured in reference[7]. Among these control points 10 points named A08,A23,A10,A05,A07,A11,A22,A26,A04,A28 are fitting points, other 7 points named A02,A03,A06,A09,A21,A01,A20 are predication points. In table 4, data of 10 fit points including X coordinate, Y coordinate, geodetic height, normal height and height anormaly are listed. In table 5, data of 7 predication points are also listed in detail.

Table 4. Data of fitting points A08,A23,A10,A05,A07,A11,A22,A26,A04,A28

Number	X coordinate	Y coordinate	geodetic height	Normal height	Height anormaly
A08	3566375.346	499179.740	26.424	5.3896	21.0344
A23	3566854.849	498567.506	30.741	9.7334	21.0076
A10	3566324.251	498659.474	25.790	4.7728	21.0172
A05	3563826.318	499348.917	30.059	9.0030	21.05
A07	3564312.203	500321.498	31.206	10.1204	21.0856
A11	3564001.762	500035.270	31.338	10.2591	21.0789
A22	3567961.396	498691.018	30.160	9.1622	20.9978
A26	3567524.909	500219.017	31.183	10.1314	21.0516
A04	3565549.066	498813.558	31.110	10.0815	21.028
A28	3568016.260	499235.369	30.332	9.3182	21.0138

Table 5. Data of predication points A02,A03,A06,A09,A21,A01,A20

Number	X coordinate	Y coordinate	geodetic height	Normal height	Height anormaly
A02	3565858.080	499248.000	31.225	10.1818	21.0432
A03	3564029.592	499613.378	30.134	9.0686	21.0654
A06	3566091.401	499632.434	30.709	9.6536	21.0554
A09	3564827.161	500392.773	29.487	8.4044	21.0826
A21	3566814.699	499080.199	30.989	9.9617	21.0273
A01	3564231.786	499937.723	30.442	9.3676	21.0744
A20	3567660.247	499189.334	31.128	10.1110	21.017

The mathematical model is as followed:

$$\xi = f(x,y) = e^{\sqrt{\sqrt{\ln x}}} + e^{\sqrt{\sqrt{\ln(y-\sqrt{x})}}} + e^{\sin(\sqrt{\sqrt{y}}\times\frac{\sqrt{x}}{\sqrt{y}})} + e^{\ln\sqrt{\sqrt{\ln(y-\sqrt{y})}}} + \sqrt{e^{\sqrt{e^{\sqrt{\ln(\ln(\sqrt{y}))}}}}} \tag{8}$$

Table 6. The fitting data of height anomaly and residual error of predication points A02,A03, A06,A09,A21,A01,A20

Number	height anomaly by conicoid function fitting	Residual error	Height anomaly by GEP fitting	Residual error
A02	21.0424	0.0008	21.0395	0.0037
A03	21.0648	0.0006	21.0662	-0.0008
A06	21.0529	0.0025	21.0496	0.0058
A09	21.0850	-0.0024	21.0825	0.0001
A21	21.0270	0.0003	21.0254	0.0019
A01	21.0746	-0.0002	21.0741	0.0003
A20	21.0194	-0.0024	21.0214	-0.0044
External accuracy		±0.0018		±0.0034

From above table, it shows that external accuracy based on GEP is 0.0034, and increased 14% than that based on polyhedral function fitting.

5 Conclusions

GPS height fitting based on GEP conquer the difficulties and shortcomings in the traditional mathematical fitting. The chromosome of GEP is simple, linear, compact and easy to genetic operation. By GEP modeling, the accurate function expression can be achieved as long as sufficient experimental data. Programmers are not required to understand the specific issues and seek the prior provisions of the structure of the objective function. Through the analysis of different GPS height fitting methods, The GEP modeling can automatically identify the implicit relationship in the data and be access to the complex functions that better reflect the actual data. The result of experiments show that GPS Height fitting method based on GEP is feasible and effective.

Acknowledgments. This work was supported by the Jiangxi Province Science & Technology Pillar Program (No.: 2009BHB16400), and the National Natural Science Foundation of China (No.: 61070008).

References

1. Xu, S.: GPS Principle and Application. Wuhan University Press, Wuhan (1998)
2. Ferreira, C.: Gene expression programming: A New Adaptive Algorithm for Solving Problems. Complex System 13(2), 87–129 (2001)
3. Ferreira, C.: Gene Expression Programming in Problem Solving. In: Invited Tutorial of the 6th Online World Conference on Soft Computing in Industrial Applications, pp. 10–24 (2001)
4. Jiang, D., Wu, Z., Kang, L., Cao, B., Li, K.: A New Method Used in Gene Expression Programming: GRCM. Journal of System Simulation 18(6), 1466–1468 (2006)
5. Li, K., Li, Y., Tang, M., Zhou, A., Wu, Z.: Application of Genetic Programming on Statistical Modeling. Journal of System Simulation 17(7), 1597–1600 (2005)
6. Li, J., Yang, Y., Gao, J., Zhou, G., Li, B.: Compare and Analysis of GPA Height Fit by Polyhedral Function and Conicoid Function. Shandong Metallurgy 28(3), 42–43 (2006)
7. Hu,W., Gao,C.: GPS Height Principle and Application.China Communication Press (2004).

Parameter Evolution for a Particle Swarm Optimization Algorithm

Aimin Zhou[1], Guixu Zhang[1], and Andreas Konstantinidis[2]

[1] East China Normal University, Shanghai, China
{amzhou,gxzhang}@cs.ecnu.edu.cn
[2] Frederick University of Cyprus, Cyprus
com.ca@fit.ac.cy

Abstract. Setting appropriate parameters of an evolutionary algorithm (EA) is challenging in real world applications. On one hand, the characteristics of a real world problem are usually unknown. On the other hand, in different running stages of an EA, the best parameters may be different. Thus adaptively tuning algorithm parameters online is preferred. In this paper, we propose to use an estimation of distribution algorithm (EDA) to do this for a particle swarm optimization (PSO) algorithm. The major characteristic of our approach is that there are two evolving processes simultaneously: one for tackling the original problem, and the other for optimizing PSO parameters. For the former evolving process, a set of particles are maintained; while for the later, a probability distribution model of the PSO parameters is maintained throughout the run. In the reproduction procedure, the PSO parameters are firstly sampled from the model, and then new particles are generated by the PSO operator. The feedback from the newly generated particles is used to evaluate the PSO parameters and thus to update the probability model. The new approach is applied to a set of test instances and the preliminary results are promising.

1 Introduction

The parameter tuning plays a key role in applying *evolutionary algorithms* (EAs) to real world applications [1]. The success of an EA depends not only on the algorithm itself but also on the problem to be solved. In algorithm design, we could tune the parameters either by repeated running or by analyzing the properties of benchmark probelems. However in applications, the characteristics of the problems maybe incomplete or the problems may not have closed forms, and the repeated experiments may be expensive. Thus the strategies for tuning parameters in algorithm design are no longer suitable in such cases. Furthermore, to achieve the best performance, the parameters of an algorithm may not be fixed throughout the run. For example, at the beginning stages, to get better diversity, the mutation probability might be high; while at the later stages, to achieve better convergence, the mutation probability needs to be low. To overcome the shortcomings of offline parameter tuning strategies, many research turn to set algorithm parameters adaptively online [2].

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 33–43, 2010.

Most of widely adaptively parameter tuning methods could be classified into the following categories.

- **Randomly selecting parameters:** With this strategy, the users give a set of candidate parameter settings by guessing or using prior knowledge. The algorithm then randomly select a parameter setting from the given set [3]. The initial parameter set is importance to the algorithm success.
- **Adaptively tuning by feedback:** By this strategy, the parameters are adaptively adjusted by heuristic rules with take feedbacks from previous parameter changes [4]. How to define the feedback is a key issue.
- **Encoding parameters into chromosomes:** The parameters are incorporated into the chromosomes and evolve with decision variables [5]. There is not much additional work to implement this strategy. However, encoding the parameters increases the complexity of the EA search space and may slow down the search.
- **Parameter evolving:** Cooperating with the main algorithm, another EA works on the parameters and its optimal solutions, i.e. the best parameters, are used in the main algorithm [6].

In this paper, we follow the idea of parameter evolving strategy. An estimation of distribution algorithm (EDA) [7] [8] is applied to tune the parameters of a particle swarm optimization (PSO) [9] [10]. In our approach, an EDA and a PSO evolve simultaneously. The EDA works on the PSO parameter space, while the PSO works on the original problem search space. In the running process, the EDA maintains a multivariate histogram probabilistic model of PSO parameters; and the PSO maintain a set of candidate solutions (particles). In each generation, the PSO parameters are firstly sampled from the EDA model; secondly, new particles are generated by the PSO operator; thirdly, the probability model is updated according to the performances of the sampled parameters.

The rest of the paper is organized as follows. The next section introduces the background of our work, including the problems, the PSO model which are used in the paper, and a brief introduction of EDA. Section 3 presents the details of the proposed method. Section 4 describes and analyzes the experimental results. The final section concludes the paper and outlines future research work.

2 Background

2.1 Optimization Problem

In this paper, we consider the following global optimization problems.

$$\begin{aligned} &\min f(x) \\ &\text{s.t } x \in \Omega \end{aligned} \tag{1}$$

where $\Omega \subset R^n$ is the decision space and $x = (x_1, \cdots, x_n)^T$ is the decision variable vector. $f : \Omega \rightarrow R$ is a continuous objective function and R is the objective space.

2.2 Particle Swarm Optimization

Among various techniques for global optimization, *particle swarm optimization* (PSO) is a promising one. PSO is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995 [9] [10], inspired by social behavior of bird flocking or fish schooling. Mathematically, the ith particle at generation t, $x_i(t)$, is updated as,

$$\begin{cases} v_i(t+1) = wv_i(t) \\ \qquad\qquad\quad +c_1 r_{1,i}(x^G(t) - x_i(t)) \\ \qquad\qquad\quad +c_2 r_{2,i}(x_i^L(t) - x_i(t)) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases}, \tag{2}$$

where v denotes the volocity, $x^G(t)$ is the global best particle, $x_i^L(t)$ is the local best particle found so far, w is the inertia weight, c_1, c_2 are the acceleration constants, $r_{1,i}, r_{2,i}$ are two dialog matrix of which the dialog elements are uniformly randomly sampled from $[0, 1]$.

Let $\alpha_i(t) = (w_i(t), c_{1,i}(t), c_{2,i}(t))^T$ be the parameter vector for generating the itsh particle at generation t, the above generation procedure could be denoted as

$$(v_i(t+1), x_i(t+1)) := \text{generate}(v_i(t), x_i(t), x_i^L(t), x^G(t), \alpha_i(t)). \tag{3}$$

After the generation process, the global best particle the the local best particle are then updated.

2.3 Estimation of Distribution Algorithm

Estimation of distribution algorithms (EDA) are a new evolutionary computation paradigm [7] [8]. A major difference between EDAs and traditional EAs is in the offspring reproduction procedure. There is no crossover or mutation in EDAs. Instead, they build a probability model of promising solutions by extracting the global population distribution information and sample new solutions from the model thus built.

3 Evolving PSO Parameters by EDA

3.1 Algorithm Framework

To adaptively set the PSO parameters, we use an EDA to evolve these parameters with the main PSO procedure. In each generation t, our approach, named *parameter evolution guided particle swarm optimization* (PEPSO), maintains

- a set of particles: $\{x_i(t), i = 1, \cdots, N\}$,
- a set of velocity vectors: $\{v_i(t), i = 1, \cdots, N\}$,
- a set of local best particles: $\{x_i^L(t), i = 1, \cdots, N\}$,
- a global best particle: $x^G(t)$, and
- a probability distribution model of parameters: $P(\alpha(t))$.

where N is the population size, and $\alpha(t) = (w(t), c_1(t), c_2(t))^T$ denotes the PSO parameter vector.

The main framework of PEPSO is as follows.

Step 0 Initialization: Set $t := 0$. Uniformly randomly generate a set of particle $\{x_i(t), i = 1, \cdots, N\}$, a set of velocity vectors $\{v_i(t), i = 1, \cdots, N\}$ in the search space Ω. Evaluate these particles by (1) and find the global best particle $x^G(t)$. Let $\{x_i^L(t) = x_i(t), i = 1, \cdots, N\}$. Initialize the parameter distribution model $P(\alpha(t))$.

Step 1 Stopping Condition: If stopping condition is met, stop and return $x^G(t)$.

Step 2 Reproduction: For each particle $i = 1, \cdots, N$, sample a parameter vector $\alpha_i(t)$ from $P(\alpha(t))$, generate a new particle $x_i(t + 1)$ by (3), and evaluate this particle.

Step 3 Particle Updating: Update the global best particle $x^G(t + 1)$ and local best particle $x_i^L(t + 1), i = 1, \cdots, N$.

Step 4 Model Updating: Update the probability model $P(\alpha(t + 1))$ by the improvements of the particles.

Step 5 Set $t := t + 1$ and go to **Step 1**.

In the following, we discuss the probability model definition and implementation.

3.2 Probability Distribution Model of Parameters

Since the algorithm parameters may correlate with each other, we use a multivariate histogram probabilistic model to model the distribution of continuous parameters. Let the boundaries of the parameter vector $\alpha = (\alpha_1, \cdots, \alpha_m)^T$ be $[\alpha_1^L, \alpha_1^U] \times \cdots \times [\alpha_m^L, \alpha_m^U]$, and each dimension be divided into D subsets, the parameter search space is thus divided into D^m bins.

The multivariate histogram probabilistic model is defined as

$$P(\alpha(t)) = (p_1(t), \cdots, p_{D^m}(t))^T \tag{4}$$

where $0 \leq p_i(t) \leq 1$ denotes the probability of a parameter vector from the ith bin, and $\sum_{i=1}^{D^m} p_i(t) = 1$.

Fig.1 shows a multivariate histogram probability model in the case of 2-dimensional parameter vector. The search space is divided into $5^2 = 25$ bins, and the hight of each bar denotes the probability of a parameter vector is from that bin.

3.3 Model Sampling and Updating

Let $q_i(t) > 0$ denotes a frequency of the ith bin to be used in generation t, then the probability is approximated as $p_i(t) = \frac{q_i(t)}{\sum_j q_j(t)}$. Instead of maintaining a probability vector, we maintain a frequency vector $q_i(t), i = 1, \cdots, D^m$ in the running process.

In the initialization step, the frequency vector is set to $q_i(0) = 5.0, i = 1, \cdots, D^m$.

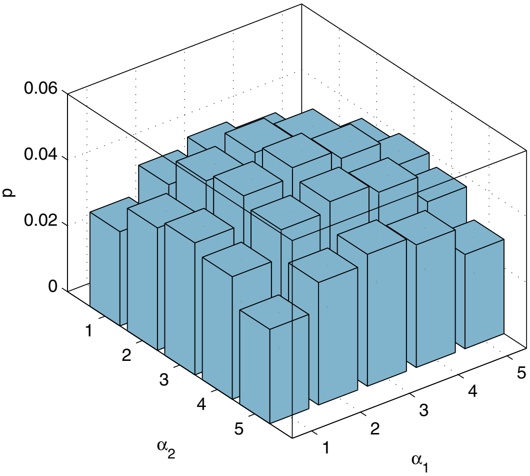


Fig. 1. Illustration of multivariate histogram probabilistic model in the case of 2-dimensional parameter vector

In generating the ith particle, a parameter vector $\alpha_i(t)$ is sampled as follows. Firstly, we randomly select a bin index $I_i(t)$ according to the probability distribution model $P(\alpha(t))$. We then uniformly randomly sample a vector $\alpha_i(t)$ from the $I_i(t)$th bin.

Define the improvement of the ith particle as

$$\Delta_i(t) = \begin{cases} f(x_i(t)) - f(x_i(t+1)) & \text{if } f(x_i(t)) > f(x_i(t+1)) \\ 0 & \text{otherwise} \end{cases} .$$

Let

$$a_{i,j}(t) = \begin{cases} 1 \text{ if } I_i(t) = j \\ 0 \text{ otherwise} \end{cases} .$$

The contribution of parameters from the jth bin is defined as

$$C_j(t) = \sum_{i=1}^{N} a_{i,j}(t) \frac{\Delta_i(t)}{\max_k \Delta_k(t)} .$$

We then update the frequency vector as follows,

$$q_j(t) = \begin{cases} 1.0 & \text{if } (1-\beta)q_j(t) + \frac{C_j(t)}{\max_j C_j(t)} < 1 \\ 10.0 & \text{if } (1-\beta)q_j(t) + \frac{C_j(t)}{\max_j C_j(t)} > 10.0 \\ (1-\beta)q_j(t) + \frac{C_j(t)}{\max_j C_j(t)} & \text{otherwise} \end{cases}$$

where β is an algorithm parameter and it is set to be 0.75 in the experiments. The frequency is fixed in the range of $[1.0, 10.0]$.

4 Experimental Results and Analysis

4.1 Test Instances

We use 13 test instances [11] in the experiments. They are Sphere function, Schwefel 2.22 function, Schwefel 1.2 function, Schwefel 2.21 function, Rosenbrock fuction, Step function, Noisy Quartic function, Schwefel 2.26 function, Rastrigin function, Ackley function, Griewank function, and two Penalized functions. Each of these functions has a global minimum value of 0. The details of these functions are listed in Tab 1.

Table 1. Test instances used in our experiments

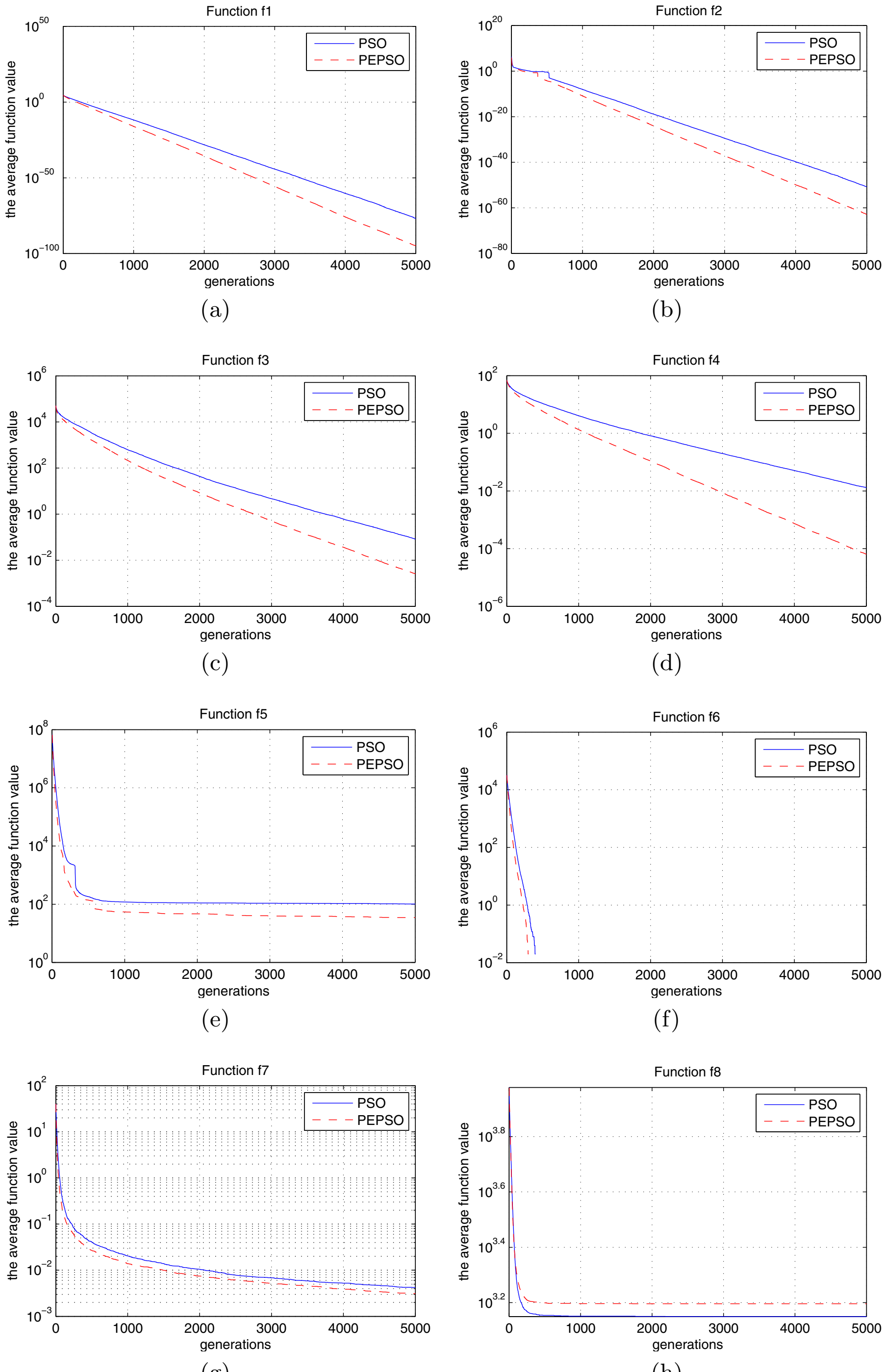
Ω	$f(\cdot)$
$[-100, 100]^n$	$f_1(x) = \sum_{i=1}^{n} x_i^2$
$[-10, 10]^n$	$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \Pi_{i=1}^{n} \lvert x_i \rvert$
$[-100, 100]^n$	$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$
$[-100, 100]^n$	$f_4(x) = \max_i \{\lvert x_i \rvert\}$
$[-30, 30]^n$	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
$[-100, 100]^n$	$f_6(x) = \sum_{i=1}^{n} \lfloor x_i + 0.5 \rfloor^2$
$[-1.28, 1.28]^n$	$f_7(x) = \sum_{i=1}^{n} i x_i^4 + rand[0, 1)$
$[-500, 500]^n$	$f_8(x) = \sum_{i=1}^{n} -i x_i \sin \sqrt{\lvert x_i \rvert} + 418.98288727243369n$
$[-5.12, 5.12]^n$	$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_i) + 10]$
$[-32, 32]^n$	$f_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$
$[-600, 600]^n$	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$
$[-50, 50]^n$	$f_{12}(x) = \frac{\pi}{n} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$
	$+ \sum_{i=1}^{n} u(x_i, 10, 100, 4)$
	$y_i = 1 + \frac{1}{4}(x_i + 1)$
	$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$
$[-50, 50]^n$	$f_{13}(x) = 0.1\{10 \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + 10 \sin^2(3\pi x_{i+1})]$
	$+(x_n - 1)^2 [1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$
	the $u(x_i, a, k, m)$ is the same as in f_{12}

4.2 Experimental Parameter Setting

In the experiments, we compare the proposed PEPSO with a general PSO. The parameters for PSO are $w = 0.5$, and $c_1 = c_2 = 2.05$ as used in most PSO algorithms. The parameter for PEPSO are as follows: the search range of w is

Table 2. The statistical results (*mean* ± *std.*) of PSO and PEPSO on the 13 test instances over 50 runs after 1000, 3000 and 5000 generations

		1000	3000	5000
f_1	PSO	1.548e-12±2.693e-12	5.530e-45±1.873e-44	1.471e-77±6.468e-77
	PEPSO	**9.587e-17**±1.623e-16	**1.756e-56**±3.405e-56	**1.398e-95**±7.907e-95
f_2	PSO	1.086e-08±3.602e-08	3.433e-30±9.232e-30	1.824e-51±6.527e-51
	PEPSO	**1.429e-11**±4.260e-11	**7.476e-38**±1.398e-37	**1.263e-63**±5.074e-63
f_3	PSO	6.208e+02±2.916e+02	4.642e+00±3.908e+00	8.114e-02±9.738e-02
	PEPSO	**2.118e+02**±1.067e+02	**4.947e-01**±4.704e-01	**2.550e-03**±2.881e-03
f_4	PSO	4.073e+00±1.258e+00	1.989e-01±1.090e-01	1.312e-02±1.250e-02
	PEPSO	**1.344e+00**±4.469e-01	**8.813e-03**±6.994e-03	**6.424e-05**±7.231e-05
f_5	PSO	1.194e+02±4.243e+02	1.079e+02±4.249e+02	1.015e+02±4.252e+02
	PEPSO	**5.417e+01**±3.916e+01	**3.941e+01**±3.026e+01	**3.477e+01**±2.912e+01
f_6	PSO	**0.000e+00**±0.000e+00	**0.000e+00**±0.000e+00	**0.000e+00**±0.000e+00
	PEPSO	**0.000e+00**±0.000e+00	**0.000e+00**±0.000e+00	**0.000e+00**±0.000e+00
f_7	PSO	2.043e-02±6.191e-03	6.852e-03±2.398e-03	4.202e-03±1.561e-03
	PEPSO	**1.384e-02**±4.793e-03	**5.218e-03**±1.885e-03	**3.113e-03**±1.018e-03
f_8	PSO	**1.416e+03**±3.486e+02	**1.412e+03**±3.468e+02	**1.412e+03**±3.468e+02
	PEPSO	1.572e+03±3.916e+02	1.5699e+03±3.914e+02	1.570e+03±3.914e+02
f_9	PSO	4.208e+01±1.854e+01	3.475e+01±1.971e+01	3.409e+01±1.931e+01
	PEPSO	**3.500e+01**±2.024e+01	**2.840e+01**±1.622e+01	**2.728e+01**±1.645e+01
f_{10}	PSO	1.803e-07±1.589e-07	8.420e-15±1.705e-15	7.923e-15±5.024e-16
	PEPSO	**5.615e-09**±1.371e-08	**7.852e-15**±7.033e-16	**7.709e-15**±9.736e-16
f_{11}	PSO	**8.368e-03**±1.073e-02	**7.928e-03**±9.858e-03	**7.928e-03**±9.858e-03
	PEPSO	1.429e-02±2.018e-02	1.365e-02±2.010e-02	1.218e-02±1.770e-02
f_{12}	PSO	**2.073e-03**±1.466e-02	**1.571e-32**±5.529e-48	**1.571e-32**±5.529e-48
	PEPSO	4.146e-03±2.932e-02	4.146e-03±2.932e-02	4.146e-03±2.932e-02
f_{13}	PSO	3.302e-09±1.837e-08	**1.350e-32**±1.106e-47	**1.350e-32**±1.106e-47
	PEPSO	**2.101e-13**±8.957e-13	**1.350e-32**±1.106e-47	**1.350e-32**±1.106e-47



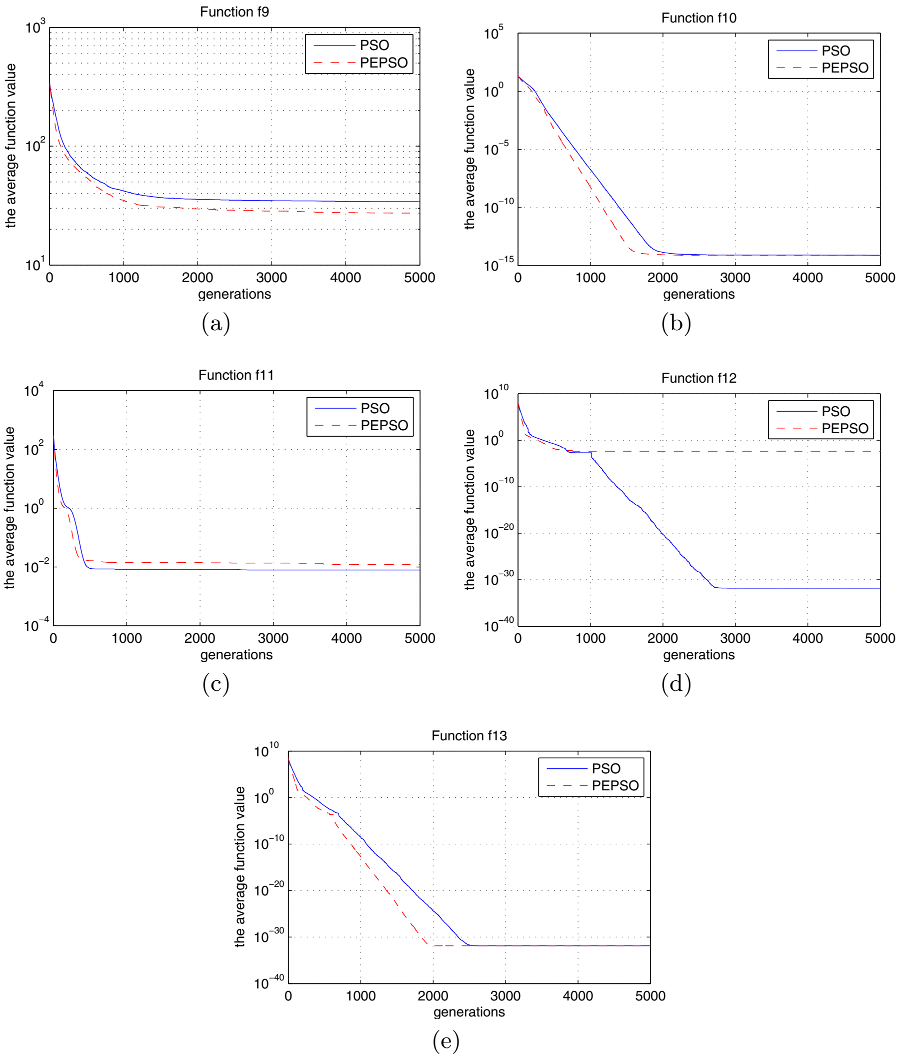
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 2. The mean fitness values versus generations over 50 runs on f_1-f_8



(a)

(b)

(c)

(d)

(e)

Fig. 3. The mean fitness values versus generations over 50 runs on f_9-f_{13}

$[0.25, 0.75]$ and the search range of c_1 and c_2 is $[1.5, 2.5]$; each range is divided into 20 subranges; to reduce the computational cost, we set $c_1 = c_2$ in the experiments and thus the parameter space is divided into $20^2 = 400$ grids.

The population size for both PSO and PEPSO is 200 and the algorithms will stop after 5000 generations. The decision vector dimensions of all the test problems are set to be $n = 30$. Each algorithm is executed independently for each instance for 50 times.

4.3 Results and Analysis

Table 2 shows the means and standard devisions of PSO and PEPSO on the 13 problems over 50 runs after 1000, 2000, and 3000 generations. Figs 2 and 3 illustrate the mean fitness values versus generations over 50 runs on all the test problems.

From the results, we can see that

- For f_1 - f_5, f_7, and f_9, PEPSO outperforms PSO not only on the converge speed but also on the quality of the obtained solutions.
- For f_6, f_{10}, and f_{13}, both PSO and PEPSO obtain similar results. However, PEPSO converges faster than PSO.
- For f_8, f_{11}, and f_{12}, PSO shows better performance than PEPSO.

The only difference between PSO and PEPSO is that PEPSO adaptively tunes its parameters in (3). The results indicate that for most of the test instances, adaptively online tuning strategy is better than setting the parameters offline. However, for some problems, PEPSO works worse than PSO. The reason might be that the probability model in PEPSO converges to local optimal solutions, i.e., the probabilities of some bad parameter vectors are much higher than those of good parameter vectors.

5 Conclusion and Future Work

In this paper, we proposed a PSO algorithm with an adaptively parameter tuning strategy by an EDA. The proposed algorithm, PEPSO, was compared with a general PSO algorithm on 13 widely used test instances. The preliminary results indicated that for most of the test problems, PEPSO performed better than PSO, either in convergence speed or in both convergence speed and solution quality. Although the adaptive strategy (EDA) still needs some parameters, it leads to the improvements of solution quality and convergence speed.

The research on adaptively tuning EA parameters is still in its very infancy and our work presented in this paper is also rather preliminary. Much work remains to be done in the future, for example, designing more efficient probability model update strategies and comparing PEPSO with other PSO algorithms [12] [13] [14].

Acknowledgement

This work is supported by National Science Foundation of China (No. 60773119) and Program for New Century Excellent Talents in University (No. NCET-08-0193).

References

1. De Jong, K.: Parameter setting in eas: a 30 year perspective. In: Parameter Setting in Evolutionary Algorithms, pp. 1–18 (2007)
2. Eiben, A.E., Michalewicz, Z., Schoenauer, M., Smith, J.E.: Parameter control in evolutionary algorithms. In: Parameter Setting in Evolutionary Algorithms, pp. 19–46 (2007)
3. Zhou, A., Zhang, Q., Jin, Y.: Approximating the set of pareto optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. IEEE Transactions on Evolutionary Computation 13(5), 1167–1189 (2009)
4. Yen, G.G., Lu, H.: Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation. IEEE Transactions on Evolutionary Computation 7(3), 253–274 (2003)
5. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation 10(6), 689–699 (2006)
6. Smit, S., Eiben, A.: Comparing parameter tuning methods for evolutionary algorithms. In: Proceedings of the 2009 IEEE Congress on Evolutionary Computation, pp. 399–406. IEEE, Los Alamitos (2009)
7. Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions I. binary parameters. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)
8. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Dordrecht (2002)
9. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: 6th International Symposium on Micromachine and Human Science, Nagoya, Japan, pp. 39–43 (1995)
10. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2001)
11. Yao, X., Liu, Y., Liang, K.-H., Lin, G.: Fast evolutionary algorithms. In: Advances in evolutionary computing: theory and applications, pp. 45–94. Springer, Inc., New York (2003)
12. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation 10(3), 281–295 (2006)
13. de Oca, M.A.M., Stützle, T., Birattari, M., Dorigo, M.: Frankenstein's pso: A composite particle swarm frankenstein's pso: A composite particle swarm optimization algorithm. IEEE Transactions on Evolutionary Computation 13(5), 1120–1132 (2009)
14. Zhan, Z.-H., Zhang, J., Li, Y., Chung, H.-H.: Adaptive particles warm optimization. IEEE Transactions on Systems, Man and Cybernetics, Part B (2009)

The Ant Colony Optimization Algorithm for Multiobjective Optimization Non-compensation Model Problem Staff Selection

Ryszard Tadeusiewicz[1] and Arkadiusz Lewicki[2]

[1] AGH University of Science and Technology, Krakow, Poland
[2] University of Information Technology and Management in Rzeszow, Poland
rtad@agh.edu.pl, alewicki@wsiz.rzeszow.pl

Abstract. This paper describes proposal for the application to modify the Ant Colony Optimization for multiobjective optimization non-compensation model problem staff selection. After analyzing the combinatorial problem involving multicriterial process of recruitment and selection model, it proposed non-compensating its solution using the modified ACO heuristic strategy. This shows that the lack of opportunities to receive appropriate the resulting matrix is related to the accurate prediction of the decision at an acceptable as satisfactory for implementation only available deterministic algorithms.

Keywords: ACO, metaheuristics strategy, heuristic decision-making system, task for recruitment and selection of employees.

1 Introduction

Written for more than 250 000 years in human nature a process of continuous search for different and equally optimal solutions and to make the final choice among them has led not only to the rapid development of high-tech electronic and information technology, but also contributed to the ongoing influx of vast amounts of unstructured and important information. Its processing and exchange and does not affect just about every activity of society, but also a fundamental determinant of the competitiveness of each company. It is in fact correct and the ability to quickly identify, acquire, use and management of new knowledge is a key success factor in many areas. Currently constructed and implemented information systems, linked to the creation of new, previously existing pieces of knowledge, defined as decision support systems should therefore also:

- support the analysis of decision-making processes,
- assist the selection process, meeting the elements for creating criterial constraints posed solutions,
- perfect design tools for teaching decision-making.

High complexity and the need for diffusion of information and knowledge in such systems, however, make the currently available deterministic solutions appear to be

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 44–53, 2010.

no longer sufficient. In the process of obtaining and analyzing information in them are not so used analytical methods, which usually provide solutions nonglobal, but the methods of artificial intelligence such as neural networks or genetic algorithms. These systems, however, have significant shortcomings. Genetic algorithms in solving a problem can not guarantee a global solution, and only provide a sufficiently good solution within a given time. They are a usually a compromise between searching the entire area of feasible solutions, and optimizing the local community. Moreover, these algorithms are characterized by the uncontrolled growth of trees on several occasions solutions. Neural networks are a true modern tool with unique features, such as learning ability and generalization of acquired knowledge, with the result that they acquire the ability to draw conclusions even on the basis of incomplete data. But they too are not free from defects. One of the most important is the lack of a multi-step reasoning, which in turn is associated with a lack of use of intermediate solutions.

Conducted by the authors of this article tests [1], [2] show that, but you can find alternatives to currently used. This alternative can be applied appropriately modified the Ant Colony Optimization Algorithm and programming. An example of the complex, multiple criteria decision problem, whose solution should be of key significance for each company [3], [4] is the problem of recruiting and selecting employees.

2 The Approach

Regardless of the method of carrying out the process of recruitment and selection of the final step should always return the expected information in the form of the desired data. This problem is multidimensional, because of the multiplicity of contexts, with which we are dealing here speaking of the forecasting ability of candidates to carry out their tasks. Forecasting should be related not only because of the reasonable and fair evaluation, but also to the maximization of the quality of the whole process, taking into account while also minimizing costs, which can be written as a function of view expressed by the formula (1):

$$\mathrm{F(x)} = \sum_{i}^{n}\sum_{j}^{m} x_{ij} * \sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij}x_{ij} \rightarrow \max, \tag{1}$$

for:

$$\forall_{l\in<1,2,..,o>} \sum_{i=1}^{n}\sum_{j=1}^{m} c_{il} \geq k_{lj} \tag{2}$$

and

$$\forall_{i\in<1,2,..,n>} \sum_{j=1}^{m} x_{ij} \tag{3}$$

where:

i – represents a candidate for the job j,

l - represents possession predisposition, representing the desired criterion for a single or group of positions

n - number of applicants on the job,

m - maximum number of posts to the cast,

o - maximum number of criterial features,

z_{ij} - means choosing the best candidate for the position j,

x_{ij} - takes the value 1, where the selection of the candidate or the value 0, if it is rejected,
c_{il} - predisposition owned by the candidate i,
k_{lj} - criterion for selecting candidates for the position j,
max - maximizing the number of positions filled with the greatest profit in the form of skills competency.

The objective function described by the formula (1) therefore represents the global optimum searches related not only to the best choice for a single position, but also to maximize the number of associations with the candidate positions.

The studies [1], [2] indicate that the right approach in this regard may be applicable mechanisms chemotaktyki stud, represented by the Ant Algorithms [5-12]. In carrying out implementation of these algorithms can be made for modification to the form of improved (Ant Colony Optimization), which takes into account both the new update rule osmotic medium of communication and experience developed through new strategies for exploration, so that the transition to the next state of the problem is always assured in the form of a compromise choice between operation associated with the accepted criteria, within the meaning already introduced restrictions on the initialization stage, and exploration based on the heuristic function of probability functions. In this way not only get a guarantee of quality solutions increase with increasing solution space, but also thanks to new rules, not only global but also local upgrade path pheromones increase the likelihood of finding a solution at least close to accurate. The task of such an Ant Colony Optimization decision support system based on the heuristics described mechanisms, so the appointment will be a discrete set of feasible solutions, so that you can obtain the extreme criterion function, described by formula (1). As the analysis of the problem [1], [4] however, it will be possible only when based on the available data repository acquired knowledge about the degree of fulfillment of each of the possible limitations of the stand in the company leading the recruiting process. It is therefore considered and implemented in a non-deterministic system of positive feedback mechanisms should be linked with the appropriate algorithm based on the knowledge acquisition strategy and optimum search strategies for a given objective function.

In the proposed algorithm, an ant colony consisting of mr ant in each new cycle should be randomly arranged in sections that are complex document index expressions, as long as the information that the state of a given expression is already determined was not previously stored in global memory. Then, each worker performs a transfer to one of the available set of definitions of vertices criterial features described by parameters representing the number of characters of a phrase, and indicate the importance of expression. The aim of this movement is determined in accordance with rule (4):

$$\mathrm{R}= \begin{cases} \arg\max\limits_{j\in\Omega}\{\tau_j(t)\cdot[\phi_{oj}]^{\beta}\},\ when\ \ q\le q_0 \\ S, \qquad \text{otherwise} \end{cases} \tag{4}$$

where: τ_j- is the intensity of pheromone trail, located on the top element of the set representing the description of each attribute criterion, β- parameter allows the relative importance of weight control following expressions, q – random number from the interval <0; 1>, q_0 – decisive factor determining how the next node transition, Ω - memory cycle of ants (local memory), ϕ - weight of the expression, while S– node is drawn with probability:

$$S = \begin{cases} \dfrac{\tau_j(t)\cdot[\phi_{oj}]^{\beta}}{\sum\limits_{j\in\Omega}[\tau_j]\cdot[\phi_{oj}]^{\beta}} & ,for \quad j\in\Omega \\ 0 & ,for \quad j\notin\Omega \end{cases}, \tag{5}$$

If the expression is drawn in accordance with the comparator element of the set is updated also features an array of indicators, which was the element of value ϕ *100. This process is repeated iteratively up to as much as the number of elements in the set under consideration. Implementation of all modeled cycles followed by updating the pheromone trail is associated with return of the array indices of the defining abilities of the candidate.

With expertise in both the requirements and have the expected degree of suitability, skills and competence of the candidates aspiring for the job should be recruited to find the best set of solutions, so that you can get to maximize the profit function (1). This ensures that the algorithm consists of two main sequence of instructions:

- the sequence of construction of the matrix gains, and
- construction of a sequence of matrix solutions stand.

The effort of moving each of the mr ants departing from the nest to a food source (representing one of the vacant posts in the company) located in the set S is also undertaken by successive ants and only if it is associated with reward in the form of profit iterative. At the same profits in line with the scale of values is understood here as a value greater than 0 and equal to at least 1. In other cases, the transition is marked as unattractive, as in the following involves the rejection of such a solution, also by the other ants. In each subsequent iteration cycle ants were randomly placed in the points representing the elements of S shall select only one candidate for which profits transition $z_{ij} > 0$, according to the rule:

$$r = \begin{cases} \arg\max\limits_{i\in\Omega}\{\tau_{ji}(t)\cdot[z_{ji}]^{\beta}\}, & when \quad q\leq q_0 \\ P, & \text{otherwise} \end{cases}, \tag{6}$$

where: τ_{ji}- is the intensity of pheromone trail, located on the edge of the connecting set S with the set K, β - parameter allows to control the relative importance of information about the cast return to the i-candidate for the j-th position, q – random number from the interval <0; 1>, q_0 – decisive factor determining how the next node transition, *Ω* - set of vertices not yet visited by ant, while *P* – vertex drawn with probability:

$$\mathrm{P_{ij}}\,(\mathrm{t}) = \begin{cases} \dfrac{\tau_{ji}(t)\cdot[z_{ji}]^{\beta}}{\sum\limits_{h\in\Omega}[\tau_{jh}]\cdot[z_{jh}]^{\beta}} & ,for \quad j\in\Omega \\ 0 \quad ,for & j\notin\Omega \end{cases}, \tag{7}$$

updating the local content of pheromone on the edges of the transition, which before the start of the first cycle were given only a limited value initial τ_0, in accordance with agreed objectives. After each cycle, followed by the global update. It concerns not only the association, which pointed to a solution so far the best quality, but also of related searches the structure of the set of candidates, because this can eliminate the possibility of local minima remain blocked in the association and receive a set of vertices of a set of candidates and available positions for as many as the multiplicity and largest total for a profit.

3 Results

Therefore, the research methods being adopted should always seek to coordinate how to deal with the stated objective of experiments carried out, therefore, the methodology of research, which is on the one hand the result of the assumptions made and approaches to solve the problem, while on the other drawn from the empirical knowledge experimental knowledge of the area, allowed to deliberate and conscious planning and conducting research processes, so that you can perform analysis of the quality of the solutions proposed heuristic.

In this purpose built, and then used a research tool in the form of decision-making support system with implemented an ant optimization algorithms modifications. This system has been designed with multi-layer architecture in order to ensure that the processing of all relevant information characterizing multicriterial process of recruitment and selection, conducted by the company, regardless of its size, scope of business, whether staff selection criteria adopted.

To conduct experiments related to the testing system was selected two groups of data sets containing objects such as profiles and application documents bench candidates. The first group of test objects of the recruitment process and selection of employees related to abstract data, selected so that you can make the optimal selection of system parameters, and then check both the usefulness and quality of heuristic solutions set. Its structure has allowed tests for the following model situations:

- The number of candidates is greater than the number of available jobs in the process of recruitment and selection is a possibility that the data that you can make all staffing positions and maximize the return Competence,
- The number of candidates is greater than the number of job vacancies in the recruitment process, but even so the selection method of compensation due to the failure to comply with certain requirements of criterion should exclude the possibility of staffing all positions,

- The number of candidates equals the number of available jobs in the recruitment process, but nevertheless it is possible to selecting data from the limits criterion that all available positions were given cast,
- The number of candidates is the same as the number of vacancies in the recruitment process, but even so the selection method of compensation due to the failure to comply with certain requirements of criterion should exclude the possibility of staffing all positions,
- The number of candidates is less than the number of vacant posts, but you can offer a selection of staff to obtain the largest possible gain competence, determining how to cast the largest number of posts.

Each set (FTM1, FTM2, FTM3, FTM4, FTM5) consisted of a number of items, so you can easily perform the verification of the correctness of results returned for the above-defined model for different scenarios combinatorial problem.

The second group of data (marked as collections of FA, FB, FC and FD) is made available to the authors by the Office of Career University of Computer Science and Management in Rzeszow actual data 4 selected companies Polish economic sector, which in 2008 carried out the recruitment process in Podkarpackie more than one post, and then to the university revealed the results for statistical purposes. These reports were used to compare the quality taken by the company with the optimum global decision proposed by the system. In this process, however, were taken into account not only sets the requirements which are criterial constraints defined by the company, but also those where the pursuit of objectivity in staff selection process created under the public tools to create the correct profile for the vacant position. This approach was designed to compare the results obtained, and then determine their degree of convergence with the decisions taken.

In view of the fact that the ants do not have memory longer to harvest more than 18 components, at the time resolve to pay an acceptable as a satisfactory experiments on fine-tuning the process parameters used research tool always assumed the existence of two lists TABU:

- List ($TABU_K$), which is memory of the ants have already visited vertex set of K, and
- List ($TABU_S$) used to store information about the area already examined solutions in the form of completed graph edge.

Studying the behavior of the system for sequentially changing values, the number of ants as three times the size of the set representing the position filled, and the number of iterations the system was established on the 120th Each time, to ensure the quality of the results was carried out 20 independent experiments.

The final stage of this research was the analysis of the results of a properly modified the number of iterations of the system. In this way, the results obtained in the form of optimal solutions, respectively for 18, 75, 120, 300 and 600 running iterations of the system. Subsequently, several studies were performed to investigate the quality of the proposed solution in dealing with the actual data held. The obtained results of these experiments illustrate the sequence fig.1, fig.2, fig.3, fig.4, fig.5.

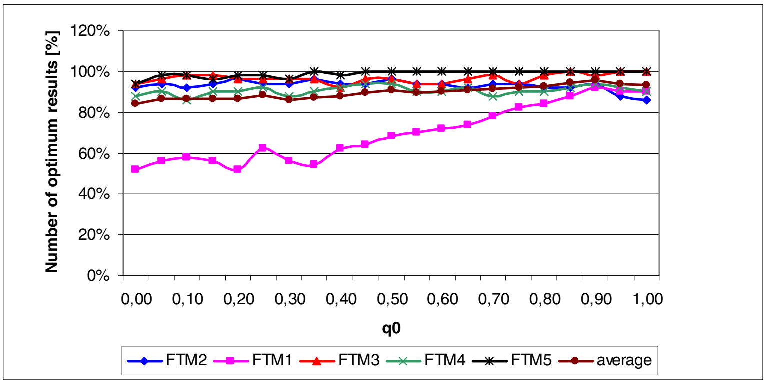


Fig. 1. Dependence as a result the percentage of optimal solutions generated by the system from the parameter being changed q0 experiment with regard to harvesting.

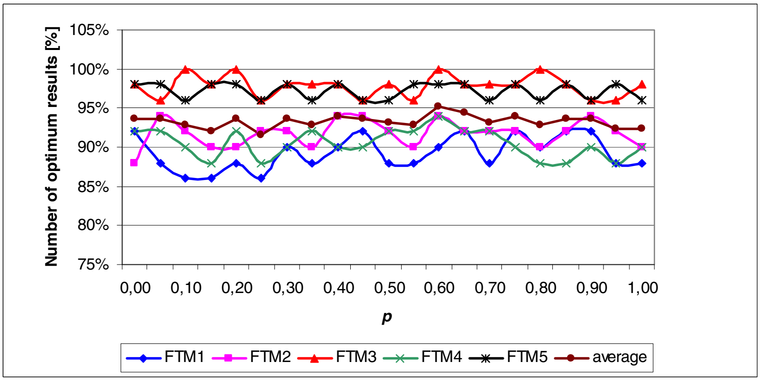


Fig. 2. Dependence as a result the percentage of optimal solutions generated by the system from the parameter being changed ρ experiment with regard to harvesting

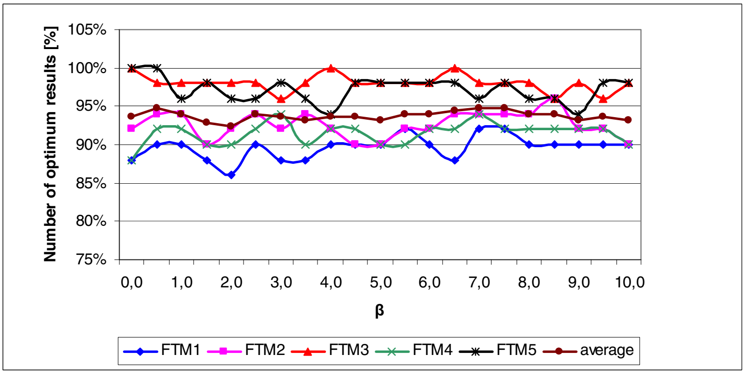


Fig. 3. Dependence as a result the percentage of optimal solutions generated by the system from the parameter being changed β experiment with regard to harvesting

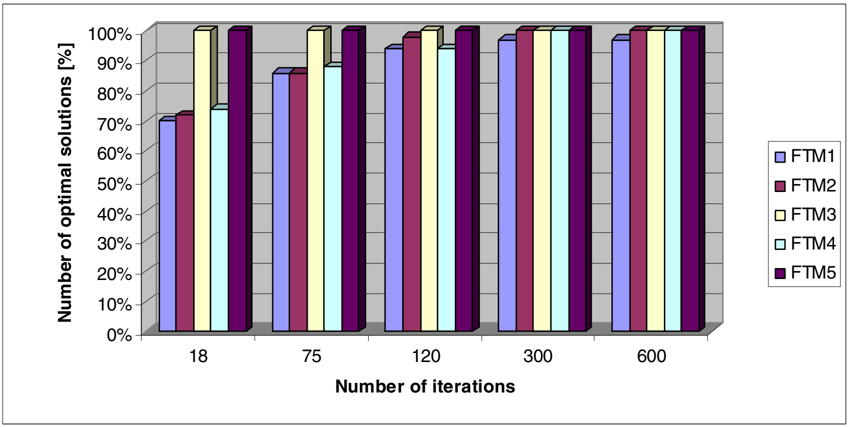


Fig. 4. The percentage of optimal solutions depending on the number of iterations of the test system

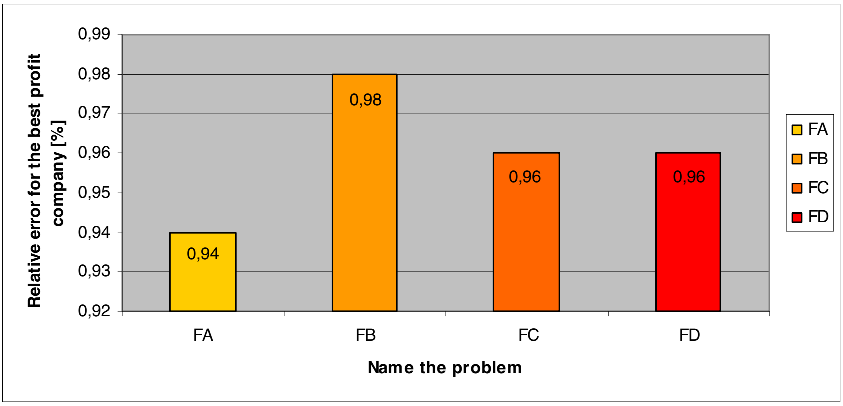


Fig. 5. Relative error value specified in relation to the exact solution for a set of FA, FB, FC and FD

4 Conclusion

Parameter, which, as studies have shown the greatest impact on the quality of obtained solutions is the parameter q0. Experimentally determined the best value, equal to 0.9 indicates that for the ants to solve combinatorial problems defined previously staff selection is much greater exploration of the uncharted areas still possible to obtain solutions than the alternative and the exercise of some already known sources, guaranteeing a profit. No less a total rejection of the possibility of cooperative ants and provide each other with information on discoveries made in the case of large data sets made the returned results were significantly worse than those of the population benefiting from synergies (positive feedback). This proves only that the same non-deterministic dynamics of this mechanism, at the time when we can reach many local extreme enough anymore. Then use the collective knowledge, which is used in such a way not only will not lead to a rapid convergence of the results achieved, but a positive impact on the competitiveness of individuals striving to achieve even better results. Thus, only the tactics of the population using a continuous search for new and better solutions, subject to the collective pattern of colonial collaboration has the opportunity to achieve success and find a good solution, regardless of the complexity of the tasks being solved, the process of supporting human resource management.

When considering the value of another factor ρ, accounting for evaporation rate of the information represented by the available info-chemical substance, which helped us to achieve the best results, we can say that it should be neither too low nor too high, and its optimum is 0.6. It allows us the one hand, the effective collection of knowledge and an efficient and indispensable mechanism for long-term memory, on the other, to avoid stagnation, a situation in which ants are often the only intelligence lead other individuals of the same colony. This parameter is also an additional feature in the full context of the adopted tactics associated with updating the local memory of the entire colony set up for life. The proposed algorithmic approaches because ants not only mean the best solution in the cycle, but also all the solutions found unsatisfactory. This is carried out but only locally. In this way, an updated footprint, less in the next iteration of the coefficient ρ is much weaker than the other good solutions, and the stronger the initialization value, imposed on the edges have not yet explored. Therefore we can say that he has been the role of the accelerator, because acting in this way reduces the time to find satisfactory solutions.

Considered a parameter that could also affect the global optimum of a parameter β, which defines the relative importance between the acquired chemical information and data about the possible return of transition edges of the graph still unused consideration of stochastic methods. Its relatively large, amounting to up to 7.0 because the value indicates that a larger number of cases is more preferred range of new solutions, rather on the basis of rewards in the form of a profit than a trace of osmotic.

Analyzing the results of the experiments, fine-tuning the coefficients proposed and implemented strategies, in the form of a modified ACO heuristic mechanism can say that these experiments not only allowed the proper identification and selection of optimum parameters of the system, but also allowed the study and drafting of its effectiveness in assisting decision-making for different types of problems related to recruitment and selection of employees. In comparison with the data representing the real problems, it appears that such a mechanism is able to generate solutions that may seem satisfactory, and even better than those that are related only to the decision of man.

References

1. Lewicki, A.: Use the ant colony optimization algorithms to build the decision-making multicriterion system for recruitment and selection of employees. In: Dissertation, AGH, Krakow (2009)
2. Lewicki, A., Tadeusiewicz, R.: The recruitment and selection of staff problem with an Ant Colony System, Backgrounds and Applications. In: Advances in Intelligent and Soft Computing, vol. 2. Springer, Heidelberg (2010)
3. Jassim, R.K.: Competitive Advantage through the Employees, CCH, Australia (2007)

4. Yakubovich, V.: Stages of the Recruitment Process and the Referrer's Performance Effect, Informs, Maryland (2006)
5. Dorigo, M., Socha, K.: An introduction to ant colony optimization. Technical Report TR/IRIDIA/2006-010 (2006)
6. Decastro, L., Von Zuben, F.: Recent Developments In Biologically Inspired Computing. Idea Group Publishing, Hershey (2004)
7. Azzag, H., Monmarché, N., Slimane, M., Venturini, G., Guinot, C.: AntTree: A new model for clustering with artificial ants. In: IEEE Congress on Evolutionary Computation, Canberra. wolumen, vol. 4, pp. 2642–2647. IEEE Press, Los Alamitos (2003)
8. Handl, J., Knowles, J., Dorigo, M.: Ant-based clustering and topographic mapping. Artificial Life 12(1) (2005)
9. Pang-Ning, T.: Introduction to Data Mining. Addison Wesley Publication, Reading (2006)
10. Sendova-Franks, A.: Brood sorting by ants: two phases and differential diffusion. Animal Behaviour (2004)
11. Abbass, H.A., Hoai, N.X., McKay, R.I.: AntTAG.: A new method to compose computer using colonies of ants. In: Proceedings of the IEEE Congress on Evolutianory Computation, Honolulu, vol. 2 (2002)
12. Dowsland, K., Thompson, J.: Ant colony optimization for the examination scheduling problem. Journal of the Operational Research Society, 426–439 (2005)

A New Self-adaption Differential Evolution Algorithm Based Component Model

Shen Li and Yuanxiang Li

State Key Lab. of Software Engineering, School of Computer Science,
Wuhan University, Wuhan 430072, China
lanceleesh@qq.com
yxli62@yahoo.com.cn

Abstract. Finding a solution to constrained optimization problems (COPs) with differential evolution (DE) is a promising research issue. This paper proposes a novel algorithm to improve the original mutation and selection operators of DE. It explored some benefits from the component model and self-adaption mechanism, while solving the constrained optimization problems. Six benchmark functions about constraint problems are used in the experiment to evaluate the performance of the proposed algorithm. The experiment results demonstrate its effectiveness compared with other the current state-of-the art approaches in constraint optimization such as KM, SAFF and ISR.

Keywords: Constrained optimization problems, differential evolution, feasible component breeding pool, self-adaption.

1 Introduction

Differential evolution (DE) is a floating-point encoding evolutionary algorithm characterized by its simplicity, rapidness and high efficiency. This algorithm has received widespread concern regarding its potential for solving unconstrained optimization problems. However, in the real-world applications, most practical optimization problems involve constraints. How to solve constrained optimization problems (COPs) with DE becomes a promising research issue.

Like other stochastic methods, DE applied in constrained optimization problems lacks an explicit mechanism to bias the search in feasible region. So it should combine with constraint-handling techniques. Various constraint handling methods have been proposed and mainly grouped into several categories, including penalty function method, special representations and operators, repair method, separate objective and constraints, and hybrid methods, etc. Penalty function method has been the most frequently used technique. Although the principal idea of penalty method is conceptually simple, the appropriate penalty parameters or weighting factors for each constraint are still difficultly assigned without enough problem specific information. Based on this point mainly, several approaches have been designed to avoid this hand-tuning of the penalty factors, such as static, dynamic, annealing, adaptive, co-evolutionary, and death

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 54–63, 2010.

penalties. It's known that adaptive penalty methods are very promising for constrained optimization, since they can make use of information obtained during the search to adjust their own parameters [1]. The aim of special representations and operators [2][3] is to preserve the feasibility of solutions at all times and to avoid generating a large number of infeasible solutions. Many approaches of this sort locate at special constrained problems and limit theirs applications. Repair methods [4], which are normally used in combinatorial optimization problems, are good choices to make feasible an infeasible individual at least at a low computational cost. Various percentage replacement rules of the repaired chromosomes have been reported. However, a strong bias may be introduced in the repairing procedure and do harm to the search of the global feasible solution. There are several approaches which handle constraints and objectives separately. The most representative approaches based on this idea are: satisfying constraints based on a lexicographic order [5], the superiority of feasible points [6][7] and the methods based on multi-objective optimization concepts. Other hybrid methods combine evolutionary computation techniques with deterministic procedures for constrained optimization problems, mainly related to differential evolution (DE), particle swarm optimization (PSO), cultural algorithm (CA), etc. This paper focuses on the last type of techniques.

Based on the literatures referring to usage of DE for solving constrained problems [8][9][10][11], mainly various selection operators have been applied for constraint handling. As a current state-of-the-art selection method, stochastic ranking[12] can maintain the promising infeasible solutions, which is quite beneficial to the problems with disconnected feasible regions and the ones with the optimal in the feasible region boundaries. The satisfying results in [12] have provided an intuitional evidence for this [8]. Therefore, this paper devises an improved self-adaption selection strategy with a balance between the preservation of feasible solutions and maintenance of population diversity. Different from common criteria in traditional constraint-handling techniques, selection criteria will be adjusted dynamically according to the population optimization process, simultaneously many measurement values of the current population are adopted in this self-adaptive selection operator so that infeasible individuals are decided adaptively their states (retained or discarded). The original mutation operator is also improved and combined with feasible component breeding pool to implement the transplantation on feasible components information. On the basis of component model in [13], the interrelationship between each dimensional component of solution and constraints is revealed. A novel measurement of feasible component is defined and feasible component breeding pool collects various feasible characteristics of each dimensional component. Due to the mutation in DE operated on each dimension, the three operated variables selected from the breeding pool will effectively direct the transformation of the infeasible component into the feasible one at a small cost.

The remainder of this paper is organized as follows.Section II shortly summarizes the main operation of basic differential evolution algorithm. Section III describes the improved mutation operator and self-adaption selection strategy for constrained optimization. In Section IV, experimental results of our approach on some well-known benchmark functions are presented. Finally, in Section V, we draw our conclusions and provide some possible paths for future research.

2 Differential Evolution Algorithm

As with all evolutionary optimization algorithms, differential evolution algorithm is a population-based stochastic search technique. At present, there are several variants of DE. In this paper, the general version used throughout our approach was the DE/rand/1/bin scheme, which is briefly described in the following.

2.1 Problem Statement

In general, the most representative description about the multi-constraint nonlinear programming problem can be expressed as follows:

$$minimize \qquad f(X), \qquad X = (x_1, x_2, ..., x_n) \in R^n \tag{1}$$

where $f(X)$ is the objective function, $X \in S \cap F$,and S is a n-dimension rectangle space in R^n bounded by the parametric constraints as

$$l(i) \le x_i \le u(i), \qquad 1 \le i \le n \tag{2}$$

The feasible region F is defined by a set of m additional liner or nonlinear constraints($m \ge 0$)

$$F = \{X \in R^n | g_j(X) \le 0, h_k(X) = 0, 1 \le j \le q < k \le m\} \tag{3}$$

m is the number of inequality constraints and $m - q$ is the number of equalities (in both cases, constraints can be either linear or non-linear). For an inequality constraint which satisfies $g_j(X) = 0 (1 \le j \le q)$, then we will say that is active at X. And all equality constraints are considered active at all points in feasible region F.

In constrained optimization problems, the constraint violation value of a decision vector is usually the sum of each constraint violation. Then the constraint violation function of m constraints is defined as $G(X) = \sum_{j=1}^{m} max[0, g_j(X)]$.

2.2 Basic DE Algorithm Framework

Differential evolution is an evolutionary algorithm originally proposed by Price and Storn [14][15], whose main design emphasis is real parameter optimization. DE mainly differs from traditional EA algorithms in the aspect of generating new vectors by adding the weighted difference vector between two population members to a third member. DE has several schemes to generate new vectors. In this section, the simplest and most popular DE/rand/1/bin differential evolution method is presented. In the basic DE framework, the algorithm firstly generates initial population ranged in the search space, and then each individual updates themselves followed by the three operators — mutation, crossover and selection.

Mutation. For each target vector $\boldsymbol{X}_i$=($x_{i1}, x_{i2}, ..., x_{iD}, D$=$problemdimension$), a mutation vector $\boldsymbol{V}_i = (v_{i1}, v_{i2}, ..., v_{iD})$ is generated according to $v_{ij} = x_{aj} + F \cdot (x_{bj} - x_{cj})$ Where $i = \{1, 2, ..., N_p\}$, N_p is population size. Random indexes a, b, c are integer, mutually different, and also chosen to be different from the running index i. $F \in [0, 2]$ is a real constant which determines the amplification of the added differential variation of $(x_{bj} - x_{cj})$.

Crossover. The crossover operator applied in DE not only extends the variety of the new individuals but also increases the diversity of the population. After mutation, the crossover operation forms the final trial vector on the basis of the current vector and its corresponding mutation vector. It defines the following trial vector:

$$u_{ij} = \begin{cases} v_{ij} \; if \; rand(0,1) \leq CR \; or \; j = k; \\ x_{ij} \; if \; rand(0,1) > CR \; or \; j \neq k; \end{cases} \quad i = \{1, 2, ..., N_p\} \; and \; j\text{=}\{1, 2, ..., D\} \tag{4}$$

$CR \in (0, 1)$ is the predefined crossover probability constant. $rand(0, 1)$ generates a uniformly distributed number between 0 and 1. $k \in \{1, 2, ..., D\}$ is a random parameter index, which ensures that at least one parameter of U_i is always selected from the mutated vector V_i.

Selection. The selection rule of basic DE algorithm is the famous greedy criterion. Compare the target vector X_i with its corresponding trial vector U_i, the vector with the smaller fitness value is chosen as a member of next generation population.

3 Our Proposed Algorithm

In this paper, our proposed algorithm improved the original mutation and selection operators of DE. It explored some benefits from the component model and self-adaption mechanism, while solving the constrained optimization problems. A pseudo-code of our approach (called Self-adaption differential evolution algorithm based component model) is shown as follow:

```
program Inflation (Output)
  begin
    Generate randomly initial population;
    Evaluate initial population;
    Initialize the feasible component breeding pool;
    Do
      For each individual in the current population
         Apply the improved mutation operator;
         Apply the crossover operator;
         Evaluate the new generated individuals Check feasibility
         of each dimensional component and each individual;
         Apply the improved self-adaption selection operator;
```

```
        Update the feasible component breeding pool;
      End for
    Until the termination condition is achieved;
  end.
```

The following section would introduce the definition of feasible component, the forming method of the feasible component breeding pool, how to combine the mutation in DE with the feasible component breeding pool to implement the information transplantation, and the details of improved self-adaption selection scheme.

3.1 Mutation Based on Feasible Component

For the characters of diverse information extracted from individuals, literature [13] presented a component-based model. It takes each dimensional component as a unit to extract feasible (or infeasible) information. The extracted information also can be expressed as the interrelationship between each dimensional component and constraints. Only several components of an individual appeared in each constraint condition, so the variation of the related components would decide whether to meet the feasibility of each constraint condition. In the literature [13], if a certain dimensional component of individuals appears in several constraint conditions, and all these relating constraints meet the feasibility requirement, then the component can be defined as a feasible component, otherwise, it is an infeasible component. Note that feasible component is not only included in feasible individuals but also in infeasible individuals. And then the infeasible component should merely be included in infeasible individuals.

After taking dimensional component as a unit, a lot of valuable information can be collected from individuals whether feasible or infeasible. The role of the feasible component breeding pool is to store this characteristic information of all the feasible components and be ready for the generation of new feasible components. Owing to different constrained optimization problems, the number of feasible component breeding pools is different and is decided by the dimensionality of the constrained problem. In other words, each dimensional breeding pool will conserve many individuals which contain the relevant dimensional feasible component. If the ith dimensional component of an individual is feasible, the ith breeding pool includes this individual. Simultaneously, if the individual includes many various dimensional feasible components, it will be conserved in many relevant feasible component breeding pools. The example on the forming method of feasible component breeding pool is displayed in Figure 1.

As is known, the original mutation operation operated on each dimension and generated the new mutation vector variables by combining three individuals randomly selected from the current population. For that much valuable information of all the feasible components is stored in each dimensional feasible component breeding pool, the improved mutation operator would randomly select each dimensional variable of the three individuals from the relevant dimensional breeding pool. This change would achieve the transplantation on feasible components information and enhance the utilization of feasible components of infeasible individuals.

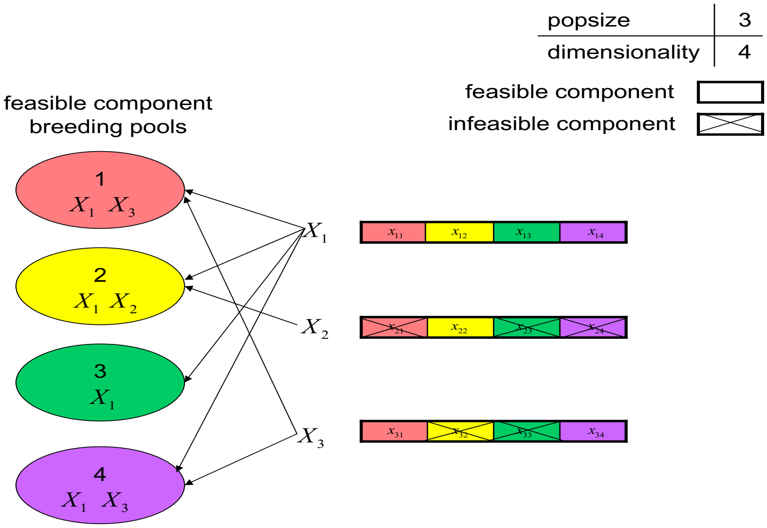

popsize 3
dimensionality 4
feasible component
infeasible component
feasible component breeding pools
1
X_1 X_3
2
X_1 X_2
3
X_1
4
X_1 X_3
X_1
X_2
X_3
x_{11} x_{12} x_{13} x_{14}
x_{21} x_{22} x_{23} x_{24}
x_{31} x_{32} x_{33} x_{34}

Fig. 1. An example on feasible component breeding pools

3.2 Self-adaption Selection

Different from common criteria in traditional constraint-handling techniques, the improved selection operation will be adjusted dynamically according to the population optimization process, simultaneously a new parameter λ is used to measure the feasibility proportion of the current population, so that infeasible individuals are decided adaptively their states (retained or discarded).

In this paper, the selection operation would compare the target vector X_i with its corresponding trial vector U_i according to the following criteria. In the current population, supposing f_{max}, f_{min} respectively is the maximum or minimum objective function value of all the feasible individuals, f_{gmax}, f_{gmin} respectively the maximum or minimum objective function value of all the infeasible individuals, G_{max}, G_{min} the maximum or minimum constraint violations of all the infeasible individuals.

1. If there is no feasible individual in the current population and $\lambda = 0$, the selection operator compares X_i with U_i according to the following formula, and then selects the individual with the smaller function value as the member of next generation population.

$$f'(\cdot) = \frac{(f(\cdot) - f_{gmin})}{(f_{gmax} - f_{gmin})} + \frac{(G(\cdot) - G_{min})}{(G_{max} - G_{min})} \tag{5}$$

2. If there are many feasible individuals in the current population and $0 \leq \lambda \leq 1$, the comparison formulations can be described as follow.
   (a) X_i and U_i are both feasible individuals, the comparisons among them are based only on their objective function values.

(b) X_i and U_i are both infeasible individuals, when $(f(X_i) - f_*)(f(U_i) - f_*) < 0$, the vector with the smaller objective function value is chosen, otherwise the vector with smaller constraint violation value is chosen. Thereinto, $f_* = \lambda f_{min} + (1 - \lambda) f_{max}$.

(c) X_i is a feasible individual and U_i an infeasible individual, compare the value between $f(X_i)$ and $f_* + G(U_i)$, the vector with smaller value is chosen as the member of next generation population.

(d) X_i is an infeasible individual and U_i a feasible individual, compare the value between $f_* + G(X_i)$ and $f(U_i)$, the vector with smaller value is chosen.

Owing to a constraint-handling technique will inevitably experience two phases (Whether there are feasible individuals in the current population), the selection operator are designed various comparison criteria. When some feasible individuals exist in the current population, the parameter of the feasibility proportion is considered in the comparison formulations. Analyzing the selection criteria b)c)d) in the phase two, if the feasibility proportion of the current population has a larger value, thereby the probability increases for infeasible individuals to survive into the next population. On the contrary, if the feasibility proportion has a lower value, feasible individuals would be selected with a higher probability. Obviously, these behaviors reflect the adaptive feature of our approach. Furthermore, in view of these criteria, many properties can be summarized. The comparisons among feasible individuals are based only on their objective function values. However, both the objective function values and the degree of constraint violations should be taken into account when comparing feasible individuals with infeasible individuals or comparing among infeasible individuals.

4 Example Study

4.1 Test Functions and the Experimental Conditions

In order to evaluate the performance of our proposed algorithm, six benchmark functions about constraint problems are used in the experiment, respectively g01,g04,g06,g07,g09,g10 in literature[10] are shown as $f_1 - - - f_6$. For each problem, total 30 independent runs of the algorithm were performed in order to assess the feasibility, performance and robustness of the approach. The parameters in our examples are set as follows: the population size $N_p = 100, F = 0.65, R = rand(0,1)$, the maximum generation $MaxG = 6000$, the tolerance for equality constraint violation $\varepsilon = 0.0001$. Table 1 summarizes the results performed by our algorithm, including the known optimal solutions for each test problem, the best, mean, worst objective function values and the standard deviations.

4.2 Experimental Results and Discussions

As for the same test functions, the proposed algorithm was compared with other the current state-of-the art approaches in constraint optimization, which are

Table 1. Comparing our algorithm(SADE-CM) with respect to KM, SAFF and ISR

FUN.		f_1	f_2	f_3	f_4	f_5	f_6
Opt.		**-15.000**	**-30665.539**	**-6961.814**	**24.306**	**680.630**	**7049.248**
Best	KM	-14.786	-30664.500	-6952,100	24.620	680.910	7147.900
	SAFF	**-15.000**	-30665.500	-6961.800	24.480	680.640	7061.340
	ISR	**-15.000**	**-30665.539**	**-6961.814**	**24.306**	**680.630**	**7049.248**
	SADE-CM	**-15.000**	-30665.518	-6961.807	24.367	680.648	7052.316
Mean	KM	-14.708	-30655.300	6342.600	24.826	681.160	8163.600
	SAFF	**-15.000**	-30665.200	-6961.800	26.580	680.720	7627.890
	ISR	**-15.000**	**-30665.539**	**-6961.814**	**24.306**	**680.630**	7049.250
	SADE-CM	**-15.000**	-30665.500	-6958.723	24.381	680.655	7054.109
Worst	KM	-14.615	-30645.900	-5473.900	25.069	683.180	9659.300
	SAFF	**-15.000**	-30663.300	-6961.800	28.400	680.870	8288.790
	ISR	**-15.000**	**-30665.539**	**-6961.814**	**24.306**	**680.630**	7049.270
	SADE-CM	**-15.000**	-30665.307	-6955.583	24.413	680.776	7055.838
St.dev	KM	—	—	—	—	—	—
	SAFF	00E+00	4.85E-01	00E+00	1.14E+00	5.92E-02	3.73E+02
	ISR	00E+00	1.10E-11	1.90E-12	6.30E-05	3.20E-13	3.20E-03
	SADE-CM	00E+00	2.06E-02	2.25E+00	1.87E-02	7.07E-02	1.30E+00

Koziel and Michalewicz's [16] GA (KM), Farmani and Wright's [17] denoted as SAFF and the improved stochastic ranking (ISR).

As shown in Table 1, the optimal solution of each constraint problem is marked in bold and ISR consistently found the optimal solutions for other five test functions except f_6. Although the ISR method is one of the most competitive algorithms known to date for constrained optimization and it is involved DE framework with a certain probability, some degree of uncertainty in the ISR is caused by the increase and setting of parameters. For test functions f_1, f_2, f_4, f_6our proposed algorithm SADE-CM achieved better solutions than the KM and SAFF algorithms on the best, mean, worst objective function values. SADE-CM can find a better near-optimal solution than SAFF for functionf_3, but the stability of our algorithm is still needed to improve. As shown in Table 1, other all algorithms except KM can consistently converge to the optimal solutions for f_1 which includes both inequality and equality constraints. It is observed from that each standard deviation is always 00E+00. For functions f_2, f_3, f_4, f_5, the solutions achieved by SADE-CM are only different from the known optimal solutions at 00E-02 magnitude. Therefore, our algorithm has the potential to solve constrained optimization problems and doesn't increase the extra parameters to be adjusted. As for the compared four algorithms on function f_6, only ISR has produced the known optimal solution. SADE-CM has achieved the second best solutions and has better capability than KM and SAFF.

In summary, we can conclude that the proposed algorithm in this paper can find similar or even better solutions than KM and SAFF. Although the accuracy of the optimal solution is slightly worse than ISR, SADE-CM combined with component-based model is easy to implement without extra parameters to be adjusted.

5 Conclusion

Constrained optimization problem has become an important research issue in the disciplines of science and engineering application. On the basis of component model, each dimensional component is taken as a unit. Simultaneously, some characteristic information of all the feasible components can be collected from individuals whether feasible or infeasible and be stored in various dimensional feasible component breeding pools. Our proposed algorithm improved the original mutation operator in DE and randomly selected each dimensional variable of the three individuals from the relevant dimensional breeding pool to operate mutation. This improvement would achieve the transplantation on feasible components information and enhance the utilization of feasible components of infeasible individuals. The original select operator in DE is also improved by self-adaption constraint handling techniques. Different from common criteria in traditional constraint-handling techniques, selection criteria will be adjusted dynamically according to optimization process of the population, simultaneously many measurement values of the current population are adopted in this self-adaptive selection operator so that infeasible individuals are decided adaptively their states (retained or discarded). The new approach is compared against other optimization techniques in several benchmark functions. The results obtained show that the proposed algorithm in this paper can find similar or even better solutions than some compared algorithms. Although the accuracy of the optimal solution is slightly worse than ISR, our algorithm combined with component-based model is easy to implement without extra parameters to be adjusted and has the potential on validity, robustness and precision.

References

1. Wang, Y., Cai, Z., Zhou, Y., Zeng, W.: An Adaptive Tradeoff Model for Constrained Evolutionary Optimization. IEEE Transactions on Evolutionary Computation 12(1), 80–92 (2008)
2. Michalewicz, Z., Janikow, C.Z.: Handling Constraints in Genetic Algorithms. In: The 4th International Conference on Genetic Algorithms (ICGA 1991), pp. 151–157. Morgan Kaufmann Publishers, California (1991)
3. Schoenauer, M., Michalewicz, Z.: Evolutionary Computation at the Edge of Feasibility. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 245–254. Springer, Heidelberg (1996)
4. Michalewicz, Z., Nazhiyath, G.: Genocop III: A co-evolutionary algorithm for numerical optimization with nonlinear constraints. In: 2nd IEEE Conference Evolutionary Computation, vol. 5, pp. 647–651. IEEE Press, Los Alamitos (1995)
5. Schoenauer, M., Xanthakis, S.: Constrained GA Optimization. In: The 5th International Conference on Genetic Algorithms (ICGA 1993), pp. 573–580. Morgan Kauffman Publishers, California (1993)
6. Powell, D., Skolnick, M.M.: Using genetic algorithms in engineering design optimization with non-linear constraints. In: The 5th International Conference on Genetic Algorithms (ICGA 1993), pp. 424–431. Morgan Kauffman Publishers, California (1993)

7. Deb, K.: An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering 186(2/4), 311–338 (2000)
8. Zhang, M., Luo, W., Wang, X.: Differential Evolution with Dynamic Stochastic Selection for Constrained Optimization. Information Sciences 178(15), 3043–3074 (2008)
9. Mezura-Montes, Velazquez-Reyes, J., Coello Coello, C.A.: Promising infeasibility and multiple offspring incorporated to differential evolution for constrained optimization. In: GECCO, pp. 225–232 (2005)
10. Runarsson, T.P., Yao, X.: Search biases in constrained evolutionary optimization. J. IEEE Trans. Evolutionary Computation 35(2), 233–243 (2005)
11. Mezura-Montese, E., Colleo, C.A.C., Morales, T.: Simple feasibility rules and differential evolution for constrained optimization. In: The 3rd Mexican International Conference on Artificial Intelligence, pp. 707–716. Springer, Heidelberg (2004)
12. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. J. IEEE Trans. Evolutionary Computation 4(3), 284–294 (2000)
13. Wu, Y., Li, Y., Xu, X.: A Novel Component-Based Model and Ranking Strategy in Constrained Evolutionary Optimization. In: Huang, R., Yang, Q., Pei, J., Gama, J., Meng, X., Li, X. (eds.) Advanced Data Mining and Applications. LNCS, vol. 5678, pp. 362–373. Springer, Heidelberg (2009)
14. Storn, R., Price, K.: Minimizing the real functions of the ICEC 1996 contest by differential evolution. In: IEEE International Conference on Evolutionary Computation, pp. 842–844. IEEE Press, Nagoya (1996)
15. Storn, R., Price, K.: Differential evolution-A simple and efficient adaptive scheme for global optimization over continuous spaces. University of California, Berkeley (2006)
16. Koziel, S., Michalewicz, Z.: Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. J. Evolutionary Computation 7, 19–44 (1999)
17. Farmani, R., Wright, J.A.: Self-adaptive fitness formulation for constrained optimization. J. IEEE Trans. Evolutionary Computation 7(5), 445–455 (2003)

A Novel Interpolation Method Based on Differential Evolution-Simplex Algorithm Optimized Parameters for Support Vector Regression

Dongmei Zhang, Wei Liu, Xue Xu, and Qiao Deng

Department of Computer Science, China University of Geosciences,
Wuhan, Chian
jjielee@163.com

Abstract. Support Vector Machine (SVM) is a machine learning method based on Structual Risk Minimization (SRM). In traditional SVM, different selections of hyper-parameters have a significant effect on the forecast performance. Differential evolution algorithm (DE) is a rapid evolutionary algorithm based on the real-code, which can avoid the local optimization by the differential mutation operation between individuals. Simplex searching algorithm is a direct searching algorithm which solves nonconstraint nonlinear programming problems. This paper introduces the application of SVM in the spatial interpolation in geosciences field. It proposes a new method of optimizing parameters of SVM based on DE algorithm and simplex algorithm. Firstly, DE algorithm is used to obtain the initial value of simplex algorithm, then the simplex local searching strategy is applied to optimize SVR parameters for the second time. Moreover, the spatial interpolation simulation is conducted on the standard dataset of SIC2004. The case study illustrates that the proposed algorithm has higher forecast accuracy and proves the validity of the method.

Keywords: spatial interpolation; Support Vector Machine; parameter optimization; differential evolution algorithm (DE); simplex algorithm.

1 Introduction

In geology, meteorology fields, there will be few data can be collected through field surveys, and the spatial distribution of those data is scattered and irregular, even the few sample numbers fail to meet the research so that spatial interpolation is needed to interpolate the space and supply the lacked data in the whole unknown region. The frequently-used methods in spatial interpolation are Kriging algorithm, discrete smooth interpolation (DSI) method, inverse distance weight method, minimum curvature, nearest neighboring interpolation, radial basis function, the artificial neural network (ANN) method, and the fractal interpolation, etc[1]. However, the geometric and function method fails to give theoretical error estimate and meet the accuracy of interpolation; and Kriging method has a demand of the data for meet a certain spatial distribution law[2], but the artificial neural network (ANN) method has a tendency to smooth the actual data values, and therefore it underestimates extreme values, as seen in the second dataset.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 64–75, 2010.

Support Vector Machine (SVM), proposed by Vapnik and his partner in 1990, is a nonlinear Machine Learning with small sample based on Structual Risk Minimization (SRM) with the characteristics such as excellent learning ability, strong promotion ability, and no special requirements for data, etc[3-5]. In recent years, SVM is applied to spatial interpolation in geology, but its result of interpolation is relevant to the kernel function of SVM and the selection of hyper-parameters[6-8]. The most common kernel function of SVR is Radial Basis Function (RBF)[9], where the error penalty parameter C, radial basis parameter σ, and the insensitive loss function ε have a big influence on the performance of SVR, hence one of the urgent problems of the application of SVR is to find out the C, σ, ε quickly and efficiently.

Based on the difference between groups, Differential Evolution (DE) is a rapid evolutionary algorithm proposed by Rainer Storn and Kenneth Price in 1995[10-11]. Compared with basic genetic algorithm, DE algorithm, has the characteristics of having easy structure, easy to implement, fast and robust, adopts real-code, and the unique mutation operation can help DE to avoid the local optimization. Simplex algorithm is a direct method of solving the nonconstraint nonlinear programming problems with the adcantages of few numbers of iterations and fast convergence speed, and the disadvantages of being sensitive to initial point and falling into the local optimization easily. In this paper, in view of the advantages and disadvantages of the two algorithms, a composite algorithm based on DE algorithm and Simplex algorithm is proposed, besides, the standard dataset of SIC2004 competition is adopted as the study object, and the experimental result indicates that the composite method has better forcasting accurancy and stronger promotion ability.

2 A Brief Overview of SVR

The basic theory of SVR is defined as follows: Assume that given the independent and identically distributed sampletraining dataset: $\{(x_i, y_i), i = 1, 2, ..., l\}$, where $x_i \in R^d$ is a d-dimensional column vector, which indicates the d-dimensional attribute value (AV) of the i sample, and $y_i \in R$ indicates the target value of the i sample. The definition of ε-insensitive loss function is defined as follow:

$$|y - f(x)|_\varepsilon = \begin{cases} 0 & |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon & |y - f(x)| > \varepsilon \end{cases} \tag{1}$$

The purpose of the learning is to construct a function $f(x)$ that makes the difference between the prediction and target value is less than the given error ε, and at the same time, guarantees the VC-dimension of the function is at the least. Through the non-linear transformation Φ(x), SVR maps the sample data set to another high dimensional feature space, and then construct a regression function in this feature space as follows:

$$f(x) = w^T \cdot \Phi(x) + b \tag{2}$$

To find out the w、b under the condition of (1) when minimizing the trust scope, slack variables ξ_i, ξ_i^* are utilized, and translate solving the function into the following quadratic programming problem:

$$\begin{cases} Min \dfrac{1}{2} w^T w + C\sum_{i=1}^{l} (\xi_i - \xi_i^*) \\ s.t. \quad y_i - w \cdot \phi(x_i) - b \le \varepsilon + \xi_i \\ \qquad w \cdot \phi(x_i) + b - y_i \le \varepsilon + \xi_i^* \\ \qquad \xi_i \ge 0 \\ \qquad \xi_i^* \ge 0 \qquad i = 1, 2, \cdots, l \end{cases} \tag{3}$$

Where C>0 is the error penalty parameter, which indicates the degree of penalty when beyond the ε pipeline.

The solving of formula (3) can be translated into the dual problem (4) through Lagrange multipliers as follows:

$$\begin{cases} \underset{\alpha、\alpha^*}{Max}[-\dfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) \\ \qquad -\varepsilon\sum_{i=1}^{l} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*)] \\ s.t. \quad \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 \\ \qquad 0 \le \alpha_i \le C \\ \qquad 0 \le \alpha_i^* \le C \end{cases} \tag{4}$$

Solving the dual problem can get the regression function as follows:

$$f(x) = \sum_{x_i \in SV} (\alpha_i - \alpha_i^*)K(x_i, x) + b \tag{5}$$

Where SV indicates the support vector.

The kernel function used in this paper is Gaussian RBF:

$$K(x, x_i) = \exp(-\frac{\| x - x_i \|^2}{\sigma^2}) \tag{6}$$

3 Differential-Simplex Composite Algorithm

3.1 Differential Evolution Algorithm

Differential Evolution algorithm (DE) is a nonconstraint direct optimizing algorithm based on the difference between groups, proposed by Rainer Storn and Kenneth Price in 1995 to solve the chebyshev polynomial problem [12-13]. It adopts the real-code, the basic thought of DE is to obtain the intermediate population through recombinant by making use of the difference between individuals in population, and the new generation of population through the competition between parents and offsprings. Since there is no requirement for initial value, DE gets the characteristics of fast convergence, simple structure and robustness, etc.

3.1.1 Initializing the Population

Form the solutions of optimization problem as D-dimensional solution vector, in which each solution vector is a basic individual in the population, and use formula (7) to generate the first generation of population, which can be distributed in solution space randomly.

$$\begin{gathered} x_{ij}(0) = x_i^L + rand_{ij}(0,1)(x_{ij}^U - x_{ij}^L), \\ i = 1, 2, \cdots, NP; \; j = 1, 2, \cdots, D \end{gathered} \tag{7}$$

Where $rand_{ij}(0,1)$ is the random number between [0,1], and x_{ij}^U、x_{ij}^L indicate the upper bounds and lower bounds of the j variable respectively.

3.1.2 The Mutation Operation

The mutation operation of DE is done by zooming the difference value between two arbitary target individual vectors in the population, and adding the D-value into the third individual vector to form a new variable[14]. For the target vector of the G generation, the j variable of mutation vector is produced according to the formula (8) as follows:

$$v_{ij}(G+1) = x_{r1j}(G) + F * (x_{r2j}(G) - x_{r3j}(G)) \tag{8}$$

Where $x_{r1j}(G)$, $x_{r2j}(G)$, $x_{r3j}(G)$ are three different individuals randomly picked from population, and F is the zoom factor whose value range is [0,1].

3.1.3 The Crossover Operation

The aim of the crossover operation is to increase the diversity of populations, and jump out of the local optimization. The new individual is produced according to the formula (9):

$$u_{ji}(G+1) = \begin{cases} v_{ji}(G+1), r_j \le CR \parallel j = rn_i \\ x_{ji}(G+1), r_j \le CR \,\&\, j \ne rn_i \end{cases} \tag{9}$$

Where $r_j \in [0,1]$ is the random number of the corresponding j variable of the individual, $CR \in [0,1]$ crossover factor, and $rn_i \in [1,D]$ is the corresponding coefficient of the i vector.

3.1.4 The Selection Operation

Compare the mutation vector $u_i(G+1)$ with the target vector $x_i(G)$, and select those individuals whose value of fitness function is smaller as the next generation.

$$x_i(G+1) = \begin{cases} u_i(G) & if(f(u_i(G)) \le f(x_i(G))) \\ x_i(G) & if(f(u_i(G)) \ge f(x_i(G))) \end{cases} \tag{10}$$

3.1.5 Fitness Function

To enhance the promotion ability of models, 5-folds cross-validation strategy is used in this paper, and formula (11) is the adopted fitness function:

$$fitness = \frac{1}{5}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{11}$$

Where $\hat{y}_i$, y_i are the true value and target value of the sample data respectively.

3.2 The Simplex Searching Strategy

Simplex searching algorithm is a directly searching algorithm of local optimization, whose core ideology is that the best vertex X_W, the worst vertex X_B and another vertex X_o in simplex are used to estimate the gradient, the direction of which is from the worst vertex to the centroid X_C of the other vertices besides the worst one, and then find out a good point to replace the worst vertex through reflection, expansion, compression and the whole contraction, etc, to construct a new simplex and iterate in a proper order. The computational formula of computing the reflection point X_R, expansion point X_E, and compression point X_S is defined in (12) [15-16]. Fig.1 shows the estimation of direction gradient of simplex, and Fig.2 presents the relation among the reflection point, expansion point, and compression point.

$$\begin{aligned} X_R &= X_C + \alpha(X_C - X_W) \quad & \alpha \le 1 \\ X_E &= X_C + \gamma(X_R - X_C) \quad & \gamma > 1 \\ X_S &= X_C + \beta(X_R - X_C) \quad & \beta < 1 \end{aligned} \tag{12}$$

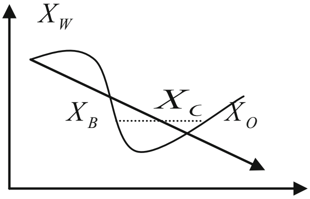


Fig. 1. The estimation of direction gradient of simplex

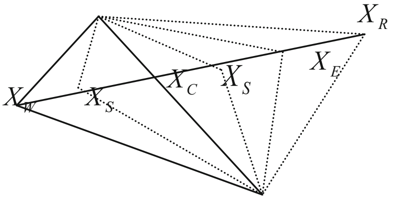


Fig. 2. The reflection point, expansion point, and compression point

3.3 The Flow Sheet of DE Algorithm-Simplex Method Optimizing Parameter of SVR in Two Ways

The flow sheet of the whole DE-Simplex optimizing algorithm in the experiment of this paper is shown as Fig.3:

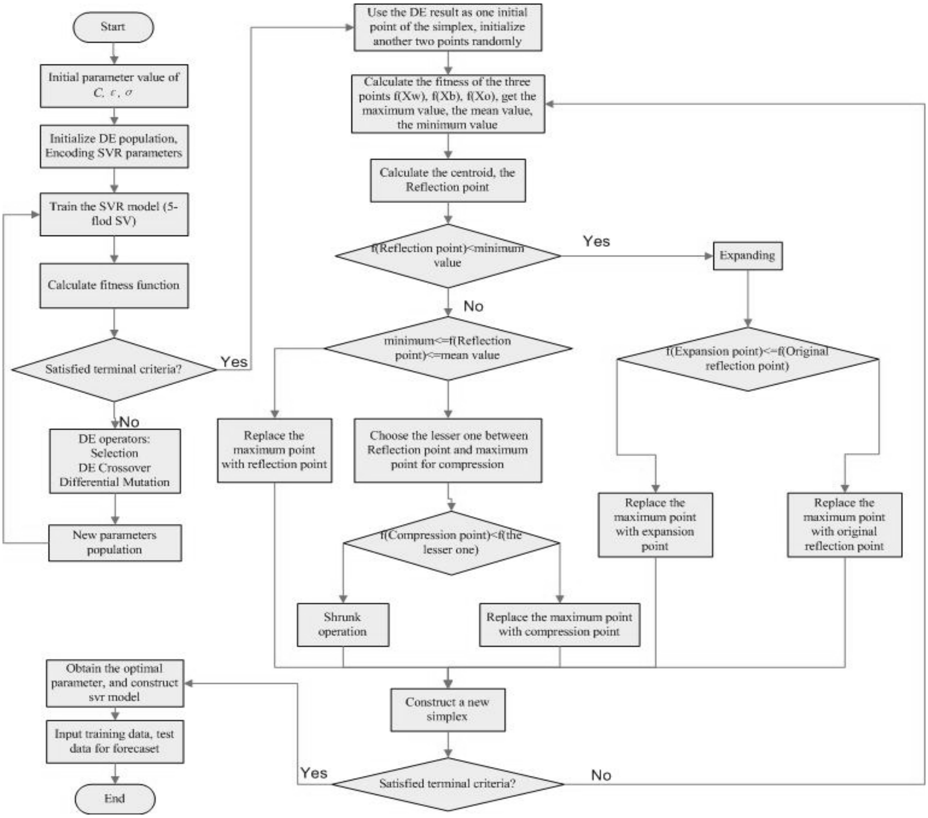

Start
Initial parameter value of C, ε, σ
Initialize DE population, Encoding SVR parameters
Train the SVR model (5-flod SV)
Calculate fitness function
Satisfied terminal criteria?
Yes
No
DE operators: Selection DE Crossover Differential Mutation
New parameters population
Use the DE result as one initial point of the simplex, initialize another two points randomly
Calculate the fitness of the three points f(Xw), f(Xb), f(Xo), get the maximum value, the mean value, the minimum value
Calculate the centroid, the Reflection point
f(Reflection point)<minimum value
Yes
Expanding
No
minimum<=f(Reflection point)<=mean value
f(Expansion point)<=f(Original reflection point)
Replace the maximum point with reflection point
Choose the lesser one between Reflection point and maximum point for compression
Replace the maximum point with expansion point
Replace the maximum point with original reflection point
f(Compression point)<f(the lesser one)
Shrunk operation
Replace the maximum point with compression point
Obtain the optimal parameter, and construct svr model
Input training data, test data for forecaset
End
Construct a new simplex
Yes
Satisfied terminal criteria?
No

Fig. 3. The flow sheet of the whole DE-Simplex algorithm optimizing parameter of SVR

4 Simulation

In this paper, the training dataset of normal and emergency data is used for training, while the corresponding testing dataset of normal and emergency data is used for testing. The implementation environment of the DE-Simplex-SVR spatial interpolation algorithm is: Interl(R) Core(TM)2 Duo CPU P7350 2.0GHz, 2048 RAM, Windows Vista operating system and the development environment of VC6.0, and the Libsvm kit, developed by Lin Chih-Jen[17], the associated professor in National Taiwan University is used.

4.1 The Setting of Dataset and Parameter

To avoid the numerical calculating problem caused by the different data measurement and value range, all data used in this experiment is normalized to the range of [-1,1]. The initial parameter of DE algorithm, Simplex method and SVR, and the value ranges of the optimizing parameter of SVR C、ε、σ are set up in table 1.

Table 1. The initial parameter of DE algorithm, Simplex method and SVR, and the optimizing parameter of SVR

The parameter of DE		The parameter of Simpex		The initial parameter of SVR		The value range of C、ε、σ	
Numbers of individual	20	Reflection coefficient	1	C	1	C	[0.1-50]
Optimizing algebra	80	Expansion coefficient	1.7	ε	0.1	ε	[0.0001-20]
Selection strategy	DE Selection	Compression coefficient	0.3	σ	0.1	σ	[0.1-32]
Mutation probability	0.9	Optimizing algebra	100				
Mutation coefficient	0.5	Given error	0.001				

4.2 Experiment Result and Analysis

Through the testing on the experimental dataset of SIC2004[18], table 2 below gives the average minimum, maximum, mean, median and standard deviation of 10 times' experiments on SIC2004 dataset using mixed algorithm.

Table 3 gives the forcasting error.

It can be seen from the results of table 2 and table 3 that the minimum and maximum on the normal dataset and the emergency dataset of the new mixed optimizing algorithm is closer to the target value than the ordinary DE optimizing algorithm, and the standard deviation is smaller. Moreover, the calculating MAE (mean relative error), ME (mean error), RMSE (root mean square error) on the normal dataset of the new mixed optimizing algorithm is smaller than the ordinary DE algorithm, while on the emergency dataset, because of the irregularity of few emergency data, the relative error of the forcasting result is larger, which leads to three larger final evaluation criterion.

Table 2. Comparison of the statistical results of two dataset (nSV/h)

N=808	Min.	Max.	Mean	median	std.dev.
Observed(the normal dataset)	57.0	180.0	98.0	98.8	20.0
Estimates by DE(the normal dataset)	70.538	123.648	96.980	99.782	13.481
Estimates by DE-Simplex (the normal dataset)	70.438	123.965	97.214	100.086	13.360
Observed(the emergency dataset)	57.0	1528.2	105.4	99.0	83.7
Estimates by DE(the emergency dataset)	70.518	123.319	97.937	100.478	14.229
Estimates by DE-Simplex (the emergency dataset)	68.889	126.773	97.705	100.133	14.099

Table 3. Comparison of the forcasting error of two algorithms

Data sets	Algorithm	MAE	ME	Pearson's r	RMSE
The normal dataset	DE	9.462	-1.038	0.776	12.827
The normal dataset	DE-Simplex	9.359	-0.803	0.782	12.680
The emergency dataset	DE	16.468	-7.480	0.274	81.266
The emergency dataset	DE-Simplex	16.761	-7.712	0.278	81.370

Correlation coefficient indicates the linear correlation degree among variables, and the closer the correlation coefficient is to one, the better the degree of correlation is. It can be seen from the experiment that through DE-Simplex algorithm, the obtained correlation coefficient is closer to one than through ordinary DE algorithm, which shows the validity of the proposed DE-Simplex approach in this paper.

Fig. 4 below shows the target value, the estimating value of DE, and the scatter graph of the estimating value of DE-Simplex on the normal dataset and the emergency dataset.

Fig. 5 gives the isoline maps of the target value and estimating value of DE and DE-Simplex on the normal dataset and the emergency dataset.

Through the comparison of the scatter graph and isoline map on the normal dataset and the emergency dataset , it is obviously that the calulating accurancy of DE-Simplex algorithm is higher, and its estimating effect is fitter to the target value, and it can reflect the distribution trend of the whole data more successful.

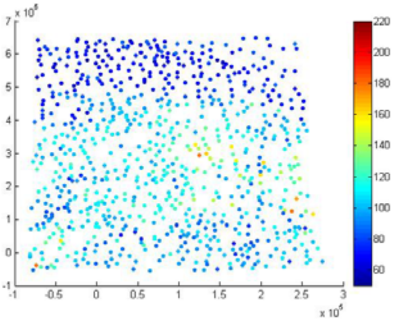


Fig. 4(a). The scatter graph of target value on normal data

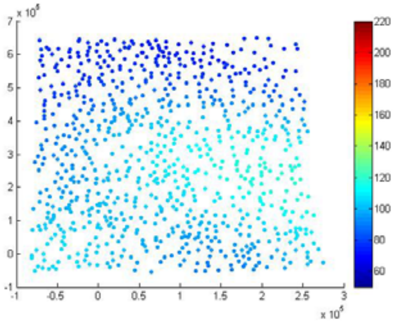


Fig. 4(b). The scatter graph of estimating value of DE algorithm on normal data

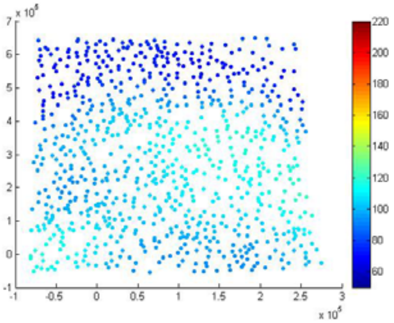


Fig. 4(c). The scatter graph of estimating value of DE-Simplex on normal data

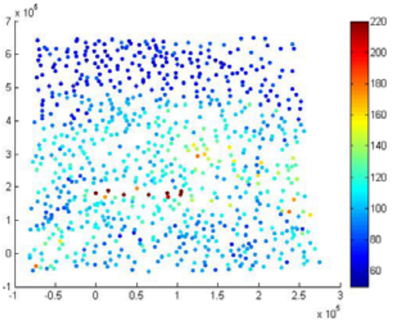


Fig. 4(d). The scatter graph of target value on emergency data

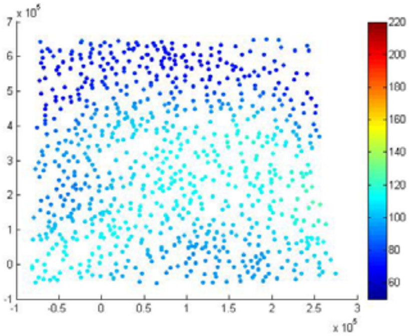


Fig. 4(e). The scatter graph of estimating value of DE on emergency data

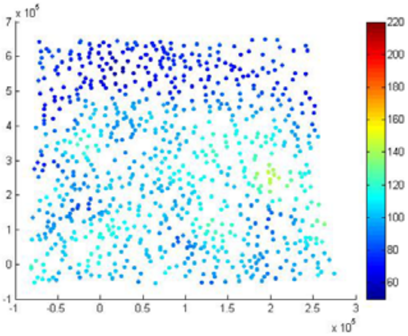


Fig. 4(f). The scatter graph of estimating value of DE-Simplex on emergency data

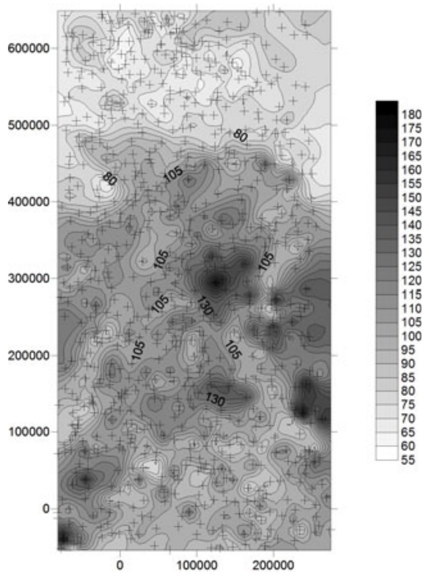


Fig. 5(a). The isoline map of target value on normal data

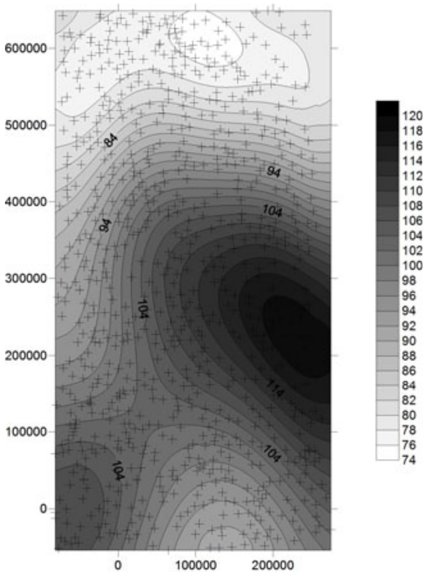


Fig. 5(b). The isoline map of estimating value of DE on normal data

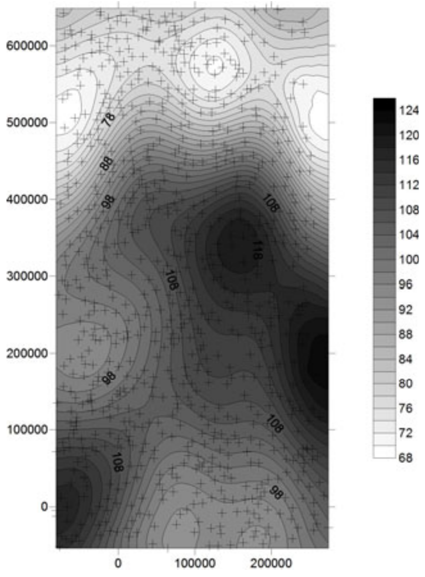


Fig. 5(c). The isoline map of estimating value of DE-Simplex on normal data

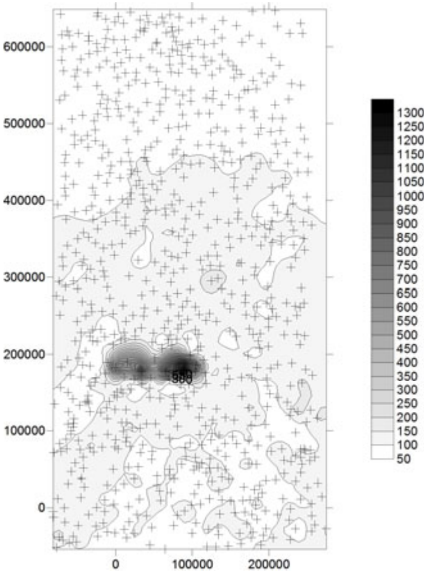


Fig. 5(d). The isoline map of target value on emergency data

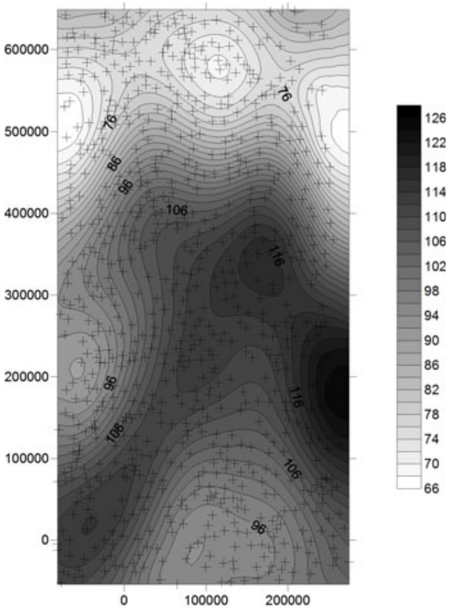


Fig. 5(e). The isoline map of estimating value of DE on emergency data

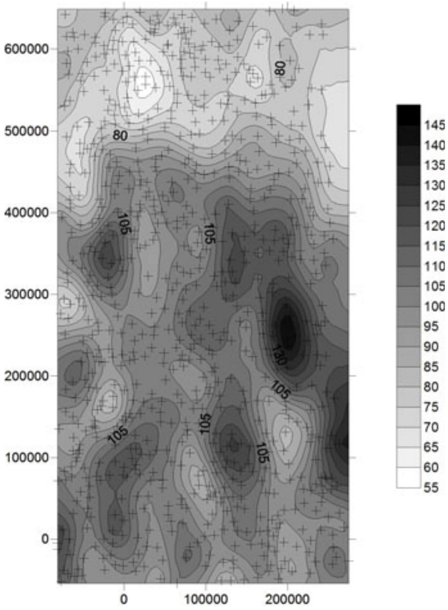


Fig. 5(f). The isoline map of estimating value of DE-Simplex on emergency data

5 Conclusion

In this paper, a spatial interpolation method of two-level optimizing SVR hyper-parameters based on Differential evolution-Simplex algorithm is proposed. This approach determines the approximate optimal parameter which regarded as the initial value of Simplex, by differential algorithm firstly, and then the strong simplex local optimizing ability is applied to the optimization parameter. In addition, the 5-folds cross-validation strategy is adopted as the fitness function in this paper to develop the promotion ability of the training model. In the experimental section, the standard dataset of SIC2004 international spatial interpolation is used to test the performance of the algorithm proposed in this paper, and the result indicates that the estimating value obtained by the two-level DE-Simplex optimizing strategy is closer to the ture value. The emphasis of the next work is how to make further improvement on Simplex algorithm in order to improve the forcasting accurancy and convergence speed.

Acknowledgements

This work is support by Important National Science & Technology Specific Projects(KZ10K820), the Natural Science Foundation of China (Grant No. 40972206 and No.60873107), the National High-Tech Research and Development Plan of China (Grant Nos.863-317-01-04-99, 2008AA12A201), the Special Fund for Basic Scientific Research of Central Colleges, China University of Geosciences (Wuhan).

References

[1] Mallet, J.L.: Discrete smooth interpolation in geometric modeling. Computer Aided Design 24(4), 78–193 (1992)

[2] Pozdnoukhov, A.: Support Vector Regression For Automated Robust Spatial Mapping Of Natural Radioactivity. Applied GIS 1(2), 1–10 (2005)

[3] Wang, W., Zhang, X., Lu, W.: Determination of the spread parameter in the Gaussian kernel for classification and regression. Neurocomputing 55(6), 643–663 (2003)

[4] Yang, D., Tian, Y.-J.: The new data mining method - support vector machine. Science Press, Beijing (2004)

[5] Kwok, J.T., Tsang, I.W.: Linear dependency between and the input noise in support vector regression. IEEE Trans. on Neural Networks 14(3), 544–553 (2003)

[6] Yan, G., Li, C., Ma, G.: SVM parameter selection based on hybrid genetic algorithm. Harbin University 40(5), 688–691 (2008)

[7] Wang, K., Yang, S., Dai, T.: A approach optimize support vector machine parameters using genetic algorithm. Computer Applications and Software 26(7), 109–111 (2009)

[8] Wang, P., Qian, L., Cao, J.: Support vector machine parameter optimization based on ant colony algorithm. Journal of Nanjing University: Natural Science 33(4), 464–468 (2009)

[9] Guo, L., Zhao, C.: Parameter optimization algorithm for support vector machine using 10 fold cross validation 45(8), 55–57 (2009)

[10] Mao, R., Wang, X., Xue, X.: An adaptive differential evolution algorithm. Computer Applications and Software 25(12), 7–8 (2009)

[11] Chen, W., Gong, W., Cai, Z.: Orthogonal differential evolution algorithm for optimization in engineering design. Computer Engineering and Applications 44(18), 230–232 (2008)

[12] King S.: Threats Solutions to Web Services Security. Network Security 2003(9) (2003)

[13] Konstantin Beznosov, D.J., Kawamoto, S., Hartman, B.: Introduction to Web services and their security. Information Security Technical Report 10, 2–14 (2005)

[14] Li Jiewei, Z.C.: A Novel Automatic Parameters Optimization Approach Based on Differential Evolution for Support Vector Regression. In: Advances in Computation and Intelligence, pp. 510–519 (2008)

[15] Feng, X., Tan, G.: Simplex search in the fusion of genetic algorithms. Computer Engineering and Applications 44(18), 30–33 (2008)

[16] He, D., Jiang, L., Wang, F.-L.: Simplex operator based genetic algorithm. Information and Control 30(38), 276–278 (2001)

[17] Chang, C.-C., Lin, C.-J.: Libsvm: a library for support vector machines (2003), `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[18] Dubois, G., Galmarini, S.: Introduction to the Spatial Interpolation Comparison (SIC) 2004 exercise and presentation of the data sets. Applied GIS 1(2), 09-1–09-11 (2005)

A Variant of Differential Evolution Based on Permutation Regulation Mechanism

Dazhi Jiang[1], Hui Wang[2], and Zhijian Wu[2]

[1] Department of Computer Science, Shantou University,
Shantou 515063, China
jiangdazhi111007@tom.com
[2] The State Key Laboratory of Software Engineering, Wuhan University,
Wuhan 430072, China
wanghui_cug@yahoo.com.cn, zjwu9551@sina.com

Abstract. Differential evolution (DE) is a stochastic, population based search method, which has emerged as a powerful tool for solving optimization problems. This paper presents a novel algorithm based on traditional DE and permutation regulation mechanism to enhance the performance of DE. As a kind of enhanced learning strategy, the permutation regulation mechanism, which makes efforts in the evolving, is constructed by rearranging the selected three father vectors. In order to verify the performance of the proposed algorithm, two experiments on some well-known benchmark functions are conducted. Performance compared with other three DE variants confirms that the new algorithm outperforms better in terms of solution accuracy.

Keywords: Differential evolution, permutation regulation mechanism.

1 Introduction

DE, proposed by Storn and Price, is comparatively a newer addition to the class of evolutionary algorithms [1]. As a reliable and effective global optimizer algorithm, DE has been successfully applied in diverse fields. Actually, it has been said that DE is one of the most competitive EAs for continuous optimization.

The behavior of DE is influenced by mutation and crossover operators and by the values of the involved parameters (e.g. F and C_r). During the last decade a lot of papers addressed the problem of finding insights concerning the behavior of DE algorithms. For parameters controlling, Qin and Suganthan [2] presented a self-adptive DE for numerical optimization named as SaDE, which emphasis on the adaptation of parameter settings and strategies in the evolving procedure. Yong et al. [3] introduced a neighborhood search strategy to DE (NSDE) which used Gaussian and Cauchy distributed functions to generate parameter F. Based on SaDE and NSDE, SaNSDE was presented by Yang et al [4]. For new search strategies designing in DE algorithm, Noman and Iba proposed a crossover-based adaptive local search operation for enhancing the performance of DE algorithm which an adaptive hill climbing strategy was employed [5]. An novel opposition-based DE (ODE) algorithm [6], which

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 76–85, 2010.

accelerates the convergence speed of DE, was presented by Rahnamayan et al. ODE uses a orient learning concept, which named as opposition-based learning, to calculate the opposite solutions of current population.

In this paper, we design a new algorithm based on traditional DE according to a special learning strategy in mutation, named as DE based on Permutation Regulation Mechanism (DEPRM). Permutation regulation mechanism, which implies in the procedure of DE mutation, could generate more good candidates compared with traditional DE. The experiments show that, DE with permutation regulation mechanism will improve the efficiency and effectiveness of problem solving.

Organization of this paper is as follows. In Section 2, the traditional DE algorithm is briefly introduced. The Section 3 gives the algorithm DEPRM clearly, and thereafter closely, a simulation is constructed for efficiency verification. Experimental verifications are given in Section 4. The last section is formed by conclusions.

2 A Brief Introduction to Differential Evolution

DE is a population-based, direct, robust and efficient search method. Like other evolutionary algorithms, DE starts with an initial population vector randomly generated in the solution space [10]. Let assume *N* is a constant number which presents the size of population, and *D* is the dimension of parameter vectors. So, the population is expressed as $X_i(t)$, where $i = 1, 2, \cdots, N$, t is the generation. The main difference between DE and other evolutionary algorithms, such as Genetic Algorithm and Particle Swarm Optimization algorithm, is its new generation vectors generating method. In order to generate a new population vectors, three vectors in population are randomly selected, and weighted difference of two of them is added to the third one. After crossover, the new vector is compared with the predetermined vector in population. If the new vector is better than predetermined one, replace it; else, the predetermined vector saved in the next generation's population. For traditional DE, the procedure is illustrated as following:

Mutation: For each vector *i* from generation *t*, a mutant vector $X_i(t+1)$ is defined by

$$X_i(t+1) = X_{r_1}(t) + F(X_{r_2}(t) - X_{r_3}(t)),$$

where $i \in \{1, 2, \cdots, N\}$ and $r_1, r_2, r_3 \in [0, N]$, i , r_1 , r_2 and r_3 are different. The differential mutation parameter *F*, known as scale factor, is a positive real normally between 0 and 1, but it also can take values greater than 1. Simply, larger values for *F* result in higher diversity in the generated population and the lower values in faster convergence.

The mutation strategy mentioned above is notated as DE/rand/1/bin, which is used frequently in literature due to the high performance.

Crossover: Crossover also plays an important role in DE algorithm which increases the diversity of the population. A crossover vector $X_i'(t+1)$ is defined as following:

$$X_i'(t+1) = (X_{i,j}'(t+1), X_{i,j}'(t+1), \ldots, X_{i,j}'(t+1))$$

where $j \in \{1, 2, \cdots, D\}$ and

$$X'_{i,j}(t+1) = \begin{cases} X_{i,j}(t+1), if\ rand(j) \le C_r, \\ X_{i,j}(t), \quad else. \end{cases}$$

Selection: A greedy selection mechanism is used as follows:

$$X_{i,j}(t+1) = \begin{cases} X'_{i,j}(t+1), if\ f(X'_{i,j}(t+1)) < f(X_{i,j}(t)), \\ X_{i,j}(t)\ , \quad else. \end{cases}$$

Without loss of generality, this paper only considers minimization problems. If, and only if, the trail vector $X'_i(t+1)$ is better than $X_i(t)$, then $X_i(t)$ is set to $X'_i(t+1)$; otherwise, the $X_i(t)$ go into the next generation without changed.

3 DE Based on Permutation Regulation Mechanism Algorithm

3.1 The Framework of New Variant DEPRM

In DEPRM, three vectors selected from N_P are arranged according their fitness in landscapes. The evolving framework of DEPRM is constructed as follows.

```
Initialize population of $N_p$ vectors at random, $t=0$
  While stop criterion not met do
    For all vectors $X_i(t)$ in the population do
    Pick at random three distinct vectors from the current
        population $X_{r_1}(t)$, $X_{r_2}(t)$ and $X_{r_3}(t)$, where, $r_1 \ne r_2 \ne r_3$.
    Arrange the three vectors within ascending order ac-
        cording to their fitness. Then,
        $f(X_a(t)) < f(X_b(t)) < f(X_c(t))$, $a,b,c \in \{r_1,r_2,r_3\}$ and $a \ne b \ne c$.
    Create intermediate vector $X_i(t+1)$ with
        $X_i(t+1) = X_a(t) + F(X_b(t) - X_c(t))$.
    Do the crossover and selection operators which are
        same to the traditional DE.
  End while
```

The control parameter C_r is one of the most important issues of DE research. The performance of DE algorithm will be affected apparently within an improper C_r. In practice, a DE user should tentatively select the initial parameter settings for the problem. Then, the trial-and-error method has to be used for fine tuning the parameters further. However, in some cases, the time for finding proper is unacceptably long. In SaDE[2], C_r is adaptively changed instead of taking fixed values to deal with different classes of problems. DEPRM use the quite same method to adjust the C_r in the problems solving. DEPRM accumulates the previous learning experience within a certain generation interval so as to dynamically adapt the value of C_r to a suitable value. Before the evolving, C_r is set to 0.5. For each individual $X_i(t)$, the C_r set as follows:

$$CR_i = \mathrm{N}_i(CRm, 0.1)$$

The C_r values for all individuals keep unchanging for a prefixed generations (5 in DEPRM), and a new set of C_r values is generated using the same equation. During every period, all the C_r values associated with candidates successful enter into the next generation are recorded in a list. After a specified number of generations (25 in DEPRM), the *CRm* will be updated by the mean value of list.

3.2 Analyses of DEPRM and DE with Simulation

In this section, a simple simulation is designed to analyze the effect of DEPRM compared with DE. For a minimum two dimension problem,

$$f(x, y) = \min((x-5)^2 + (y-5)^2),$$

where $x, y = rand(0,10)$, and the global optimal solution is $c(x, y) = (5,5)$. Assume N=100, in initialization step, an especial initialization scheme is used to generate primal individuals. According to this scheme, all the individuals X_i ($i = 1, 2, \cdots, 100$) are located in the coordinates with uniform distribution and the Euclidean distance for particles is $dis(X_i, c) \geq 2.0$. After initialization, the purpose of this simulation, which is concentrating on the production ability of good solutions within mutation and crossover operators of DE and DEPRM, could be depicted. With this orientation, all the vectors are generated and recorded without replacement in the procedure of genetic operating. Furthermore, the new vectors will not participate in evolution in current or later procedure, and the vectors are not adjusted according to the boundary of search space.

Let us start with the probability definitions which are calculated in our simulations.

Definition1: The probability of offspring (o) generated in search space is defined as below:

$$p_o = p[\alpha < dis(o, c) < \beta],$$

where p_o stands for probability function, and α , β stands for upper and lower limits of distance respectively.

Definition2: The probability of offspring (o) generated by operators in problem's domain which are better than the selected fathers ($X_{r_1}(t)$, $X_{r_2}(t)$ and $X_{r_3}(t)$) is defined as follows:

$$pb_o = p[f(o) \succ \{f(X_{r_1}), f(X_{r_2}), f(X_{r_3})\}],$$

where pb_o stands for probability function, and $\succ$ means better than.

The following algorithm implements simulation for 2-dimensional search space.

Framework for Simulation:

```
TRIALS=100.
NUMBERS =20.
INTERVAL =30.
N=100.
```

```
For i=1 to TRIALS
  OIDE[INTERVAL];OIDEPRM[INTERVAL];OTDE[TRIALS];OTDEPRM[TRIALS];OTDEF[TRIALS]//F<0.5;OTDEPRMF[TRIALS]//F<0.5;
  $X_i = (x_i, y_i) = [0,10]$, where $i = 1, 2, \cdots, 100$, and $dis(X_i, c) \ge 2.0$.
  For j=1 to NUMBERS
    For k=1 to N
      $o_{DE} = X'_{i,DE}(t+1)$, $o_{DEPRM} = X'_{i,DEPRM}(t+1)$. Where the offspring $X'_{i,DE}(t+1)$ is generated by DE/rand/1/bin, and $X'_{i,DEPRM}(t+1)$ is generated by DEPRM algorithm mentioned above.
      $OIDE\left[\lfloor 2*dis(o_{DE}, c)\rfloor\right]++$.
      $OIDEPRM\ \left[\lfloor 2*dis(o_{DEPRM}, c)\rfloor\right]++$.
      If $f(o_{DE}) \succ \{f(X_{r_1}), f(X_{r_2}), f(X_{r_3})\}$ then
        OTDE[i]++.
        If (F<0.5) then
        OTDEF[i]++.
        End if
      End if
      If $f(o_{DEPRM}) \succ \{f(X_{r_1}), f(X_{r_2}), f(X_{r_3})\}$ then
        OTDEPRM[i]++.
        If (F<0.5) then
        OTDEPRMF[i]++.
        End if
      End if
    End for
  End for
End for
For every elements in OIDE[i] and OIDEPRM[i], set
        OIDE[i] ← OIDE[i]/( TRIALS* NUMBERS*N),
     OIDEPRM [i] ← OIDEPRM [i]/( TRIALS* NUMBERS*N)
For every elements in OTDE [i], OTDEPRM [i], OTDEF[i] and OTDEPRMF[i], set
           OTDE [i] ← OTDE [i]/( NUMBERS*N),
        OTDEPRM [i] ← OTDEPRM [i]/( NUMBERS*N),
          OTDEF [i] ← OTDEF [i]/( NUMBERS*N),
       OTDEPRMF [i] ← OTDEPRMF [i]/( NUMBERS*N).
```

As seen in Fig 1, for two algorithms, the genetic operators are more likely to generated offspring when interval between 5 and 10. The offspring generated by DEPRM have a higher chance to be closer to the global optimal solution compared with DE. When interval =10, p_{DEPRM} becomes smaller. It is a good performance that DEPRM is apt to generated offspring in the search space where the problem predefined (interval>10 is out of the boundary). This phenomenon could be understood that DEPRM algorithm can generated valid candidates for evaluation, but DE has more offspring wait to be adjusted by the boundary control.

In Fig 2 and Fig 3, the averaged survived times of offspring is curved for explaining the efficiency of DEPRM. In Fig 2, for all experiment trails, DEPRM get a higher probability to generated better solutions than DE apparently. The average probability of pb_{DEPRM} is 0.31706, and 0.20508 for pb_{DE} . In Fig 3, it is obviously that DEPRM generated more and better solutions around itself when F<0.5, and the local search mechanism is make effort in the process. The scheme for generating better offspring in DE is not so diversity compared with DEPRM. DE get a pair chance to generated better offspring whatever the F chose. However, F<0.5 is still performances better than F>0.5, which means local search is more potential for genetic operating. The average of pb_{DEPRM} is 0.20634 when F<0.5, and 0.11072 when F>0.5. The average of pb_{DE} is 0.10833 when F<0.5, and 0.09675 when F>0.5. pb_{DEPRM} is higher than pb_{DE} in F<0.5 and F>0.5.

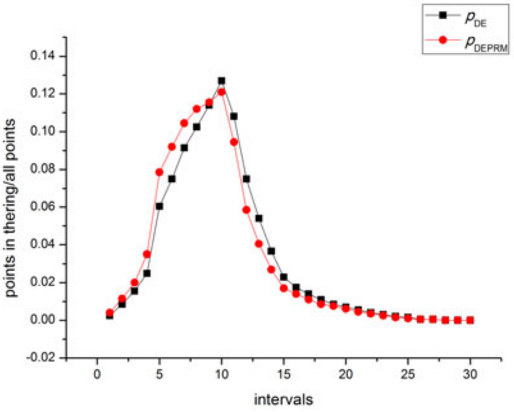


Fig. 1. The output of p_{DE} and p_{DEPRM}

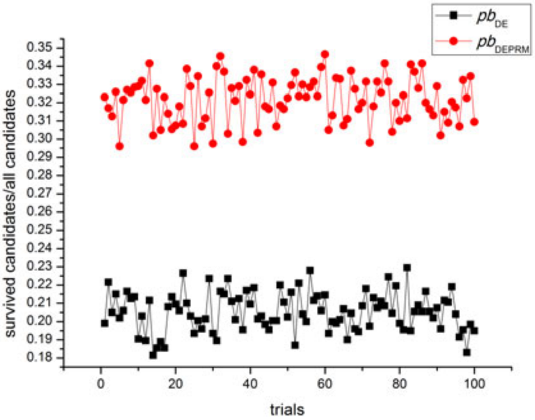


Fig. 2. The output of pb_{DE} and pb_{DEPRM}

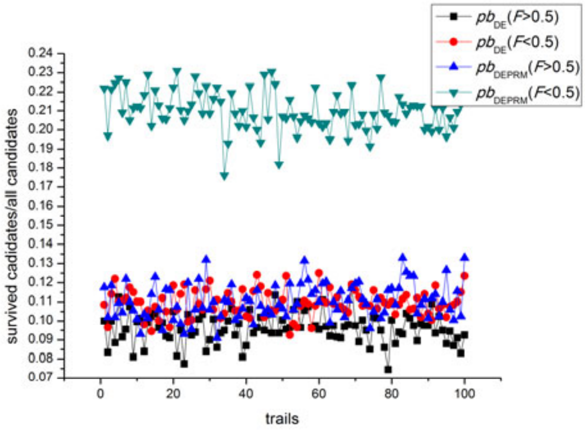


Fig. 3. The output of pbDE (F>0.5 and F<0.5) and pbDEPRM(F>0.5 and F<0.5)

4 Experimental Verification

To verify the performance of DEPRM, one benchmark set is used in the following experiments. All the benchmark functions listed in table 1 have been frequently used in literature. To evaluate the performance of convergence accurately, the maximum number of function calls (MAX_NFC) was employed.

Table 1. The 8 test functions used in the experiments, where D is the dimension of the functions, X is the definition domain, and fmin is the minimum values of the function

Number	Test Functions	D	X	$f_{\min}$
F1	$f(x)=\sum_{i=1}^{D} x_i^2$	30	[-100,100]	0
F2	$f(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	30	[-10,10]	0
F3	$f(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i} x_j)^2$	30	[-100,100]	0
F4	$f(x)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(1-x_i)^2]$	30	[-30,30]	0
F5	$f(x)=\sum_{i=1}^{D}(\lfloor x_i+0.5\rfloor)^2$	30	[-100,100]	0
F6	$f(x)=\sum_{i=1}^{D} ix_i^4+random[0,1)$	30	[-1.28,1.28]	0
F7	$f(x)=-\sum_{i=1}^{D}(x_i\sin\sqrt{\lvert x_i\rvert})$	30	[-500,500]	-12569.5
F8	$f(x)=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	30	[-600,600]	0

Experiment 1:
In Experiment1, DEPRM is compared with some excellent algorithms, which are SaDE, NSDE and SaNSDE. For parameters setting in DEPRM of current experiment, N=100, F=$Rand$(0.0,1.0) .

The average results of 25 independent trials on functions F1 to F8 are summarized in table 2. Especially, the results for three algorithms, SaDE, NSDE and SaNSDE, are cited from Yang's Paper [6].

Table 2. Comparison of DEPRM with SaDE, NSDE and SaNSDE, where Mean Best indicates the average best fitness values

F	MAX_NFC	SaDE Mean Best	NSDE Mean Best	SaNSDE Mean Best	DEPRM Mean Best
F1	150000	7.49E-20	7.76E-16	3.02E-23	**1.19E-42**
F2	150000	6.22E-11	4.51E-10	4.64E-11	**7.48E-25**
F3	150000	1.12E-18	1.06E-14	6.62E-22	**9.51E-40**
F4	500000	2.10E+01	1.24E+01	**4.13E-30**	4.52
F5	150000	**0.0**	**0.0**	**0.0**	**0.0**
F6	150000	7.58E-03	1.20E-02	7.21E-03	**3.92E-03**
F7	150000	**-12569.5**	**-12569.5**	**-12569.5**	**-12569.5**
F8	150000	8.88E-18	6.72E-09	**0.0**	**0.0**

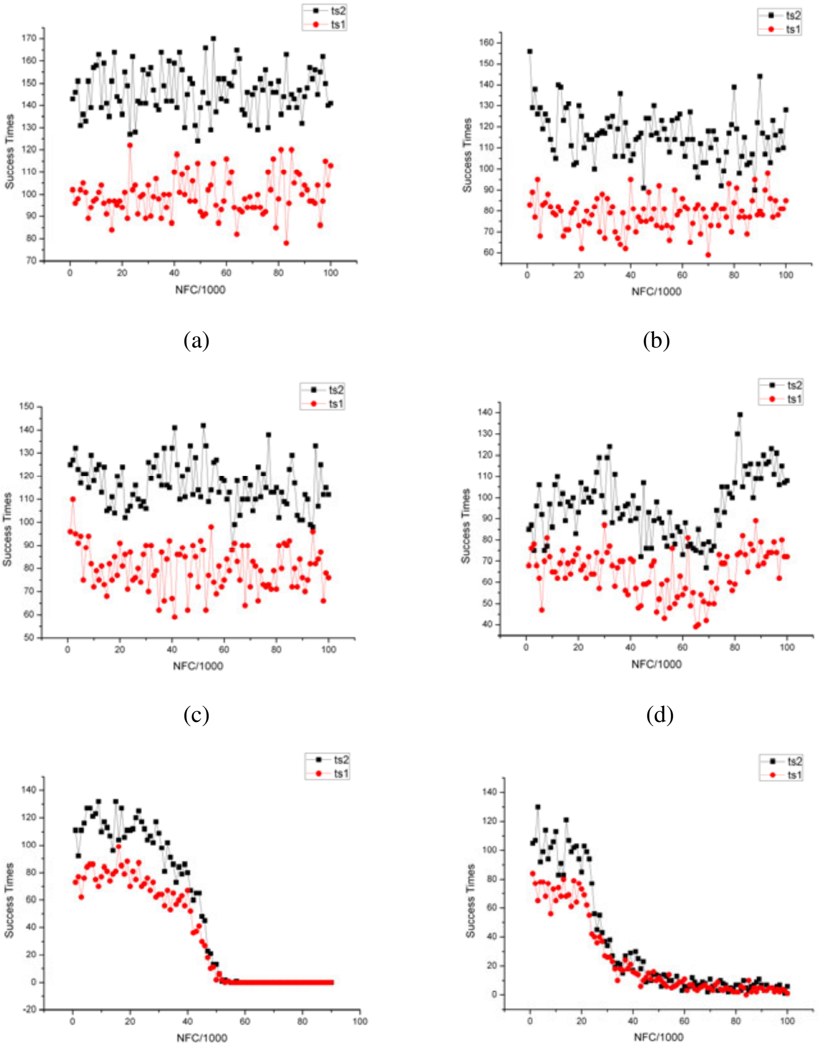
For functions F1-F3, F6, DEPRM achieved better than SaDE, NSDE and SaNSDE. For function F5 and F7, all four algorithms obtain the exactly same result. For function 8, DEPRM, as well as SaNSDE, obtain the best performance. SaNSDE achieved the best in function F4 while the result of DEPRM better than SaDE and NSDE. From the experiment1, it is normal that DEPRM is not suitable for all test functions which can be explained by No-Free-Lunch theorem [7].

Experiment2:
In SaDE, two candidate strategies are used to improve the performance of algorithm [2]. For this purpose, two probabilities are applied to decide which strategy is chose for each individual in current population. What's more, for adaptability, the probabilities are changed in the evolution according to the number of trial vectors successfully entering the next generation and the vectors discard. A strategy with a bigger success times, to some extend indicates that it has more potential to generate good candidates.

In experiment 2, according to SaDE, a hybrid evolutionary algorithm is constructed with two mutation strategies obtained from traditional DE (DE/rand/1/bin) and DEPRM, which focuses on the times the trail vectors successfully entering the next generation. According to this, the performance of DE and DEPRM could be performed clearly. In hybrid algorithm, two mutation strategies generated trail vectors respectively, then compared together with each individual in current population. The success times of trail vectors entering the next generation by applying DE and DEPRM are recorded as $\boldsymbol{ts}_1$ and $\boldsymbol{ts}_2$ respectively. Those two numbers are accumulated within a specified number of function calls (1000 in our experiments). After accumulated period, both of them will be reset for another accumulation step.

All experiments in experiment 2 have been repeated 25 times with different random initialization, for each benchmark function to obtain statistically reliable performance numbers. For parameter settings of hybrid evolutionary algorithm, N=100, D=30, F=$Rand$(0.0,1.0) and C_r = 0.5. MAX_NFC for all the trails are 1E+06.

As seen in the Fig 4, $\boldsymbol{ts}_2$ is frequently bigger than the $\boldsymbol{ts}_1$ obviously (especially in (a), (b), (c), (d) and (h)), which means that mutation operator with permutation regulation in DEPRM has more chance and potential to generate good candidates compared with DE/rand/1/bin.



(a)

(b)

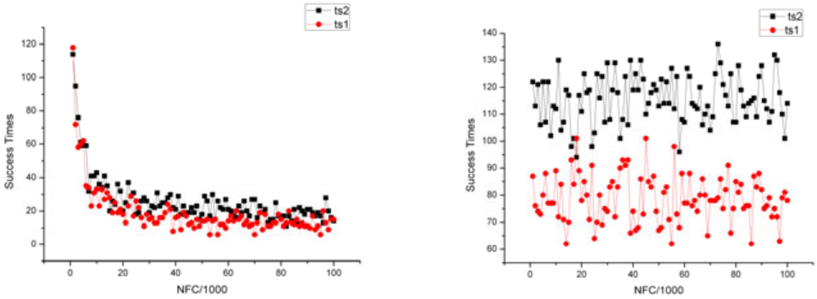
(c)

(d)

(e)

(f)

Fig. 4. The success times of trail vectors entering the next generation by applying DE mutation (ts1) and DEPRM mutation (ts2) for all test functions. X axis represents a period of number of function calls and Y axis represents the success times. (a)F1. (b) F2. (c) F3. (d) F4. (e) F5. (f) F6. (g) F7. (h) F8.



(g) (h)

Fig. 4. (*continued*)

5 Conclusions

In this paper, a new differential evolution, DEPRM is presented. In DEPRM, permutation regulation mechanism is used to adjust the order of vectors in mutation. This mechanism has a high potential to improve performance of optimization algorithm and can be embedded with various mutation strategy of DE. Such tiny regulation which can make so much significant efforts is exciting and delightful for algorithm designer, while useful and helpful for algorithm designing. The future works will focus on the combinations of permutation regulation mechanism and other DE variants, such as DE/best/1/bin, DE/best/2/bin, DE/rand/2/bin, etc.

References

1. Storn, R., Price, K.: Differential Evolution-A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim., 341–359 (1997)
2. Qin, A.K., Suganthan, P.N.: Self-adaptive Differential Evolution Algorithm for Numerical Optimization. In: Proc. of the 2005 IEEE Congress on Evolutionary Computation, pp. 1785–1791 (2005)
3. Yang, Z., He, J., Yao, X.: Making a difference to differential evolution. Advance in Metaheuristics for Hand Optimization, 397–414 (2008)
4. Yang, Z., Tang, K., Yao, X.: Self-adaptive differential evolution with neighborhood search. In: Proc. Congr. Evol. Comput., pp. 1110–1116 (2008)
5. Noman, N., Iba, H.: Accelerating differential evolution using an adaptive local search. IEEE Transactions on Evolutionary Computation, 107–125 (2008)
6. Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A.: Opposition-based differential evolution algorithms. In: IEEE Congress on Evolutionary Computation, pp. 2010–2017 (2006)
7. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 67–82 (1997)

Differential Evolution Based Band Selection in Hyperspectral Data Classification

Xiaobo Liu, Chao Yu, and Zhihua Cai

School of Computer Science, China University of Geoscience,
Lumo Road. 388, 430074, Wuhan, Hubei China
jerrycug@yahoo.com.cn

Abstract. Differential evolution is a new heuristic approach for global optimization method. Hyperspectral data included huge information, how to process the hyperspectral data lack of reliable and effective technique. We presented an evolutional program to select hyperspectral band, which is very fast and certainty, in order to avoid the band felling in local minima, the selected bands are contained in the entire spectral range, and have a good performance in hyperspectral data classification. The classification methods we used are very classic, which are good at the imbalance data, so that the classification accuracy are convincing, for example NBtree, Naive Bayes and J4.8. The results in classification of various kinds of minerals in uranium deposit are better than other methods, such as information entropy.

Keywords: Differential Evolution, Band Selection, Spectral Classification, Information Entropy.

1 Introduction

Remote Sensing technique is one of the most important methods to get the information of the earth's surface, which is developed in 1990s.Hyperspectral sensors generate imagery that captures surface and sub-surface properties, and provide non-invasive and non-intrusive reflectance measurements. While,the hyperspectral data contains very rich information, how to remove the redundancy information, keep the relevant information and improve the model's accuracy are the challenges for us. As hyperspectral sensors acquire images in very close spectral bands, the resulting high-dimensional feature sets contain redundant information. The redundant information resulted in not only waste much processing time by the classification, but also influenced the classification accuracy. A reduction of the dimensionality of the Hyperspectral data overcomes this problem. In order to avoid the bad influences, we select only a dozen well-defined bands to get adequate results, in hyperspcetral data, the spectral bands are generally divided proportionally over the spectral range. Although a method for band selection leads to data compression, we would like to emphasize that the performance objective of data compression is based on data size(communication band width), which is different from classification or discrimination accuracy.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 86–94, 2010.

A more general approach is to treat the hyperspectrum as a continuous function and to select complete waveband settings, to imitate the sensor. In the past, different efforts have been made for suitable band selection. The problem is often defined as a selection of a new basis to describe the spectra by optimizing some criterion like error of reconstruction or discrimination between classes. Price[9] proposed an iterative method to create a local basis with minimal reconstruction error. In[5], the basis matching technique was applied to the band selection problem for target detection. In[14], a PCA-based approach was applied for band selection. In[6] propose two best-bases feature extraction algorithms for classification purposes. In a top-down approach, a binary splitting of the spectrum is iteratively performed, and the mean reflectance values are chosen as features. A bottom-up algorithm builds an agglomerative tree by merging highly correlated adjacent bands and projecting them onto their Fisher direction. Similarly, in[11] propose an iterative method for merging neighboring bands.

Our motivation in this paper is use a simple and efficient heuristic method to solve a typical combinatorial optimization problem. DE as a novel evolutional algorithm very effective in global search, it's genetic operation not only guarantee the results would not fall into local optimal, but also enhance the population's diversity, that means, we can search all the possible bands at the same time, and get the best band combination.

The paper is organized into four sections. Section 2 presents the proposed method applied in band selection. Section 3 presents the experiments and reports the results. In section 4, the results compared with other methods' results and got the conclusions.

2 Relate Work

2.1 Band Selection

Bands selection techniques generally involve both a search algorithm and define the bands range. The search algorithm generates some useful features by taking combinations of the bands, those features are the reasonably located in the high-dimensional feature sets, which reduced the dimensionality largely. Then accord the generated features use methods to define the bands range. In this paper, attention is focused on search algorithm: Differential Evolution (DE).

2.2 Differential Evolution

DE is a parallel direct search method,is a very simple population based, stochastic function minimizer which is very powerful at the same time. DE is the best genetic algorithm approach. Unlike simple GA(Genetic Algorithm) that uses binary coding for representing problem parameters, DE is a simple yet powerful population based, direct search algorithm with the generation-and-test feature for globally optimizing functions using real valued parameters. The DE's advantages are its simple structure, ease of use, speed and robustness. Price & Storn[13] gave the working principle of DE with single strategy. Later on, they suggested ten different strategies of DE[10]. A strategy that works out to be the best for a given problem may not work well when applied for a different problem. What's more, the strategy and key parameters to be adopted for a problem are to be determined by trial & error. The main difficulty with the DE technique, however, appears to lie in the slowing down of convergence as the region of global minimum is approaching. DE is a parallel direct search method which utilizes NP D-dimensional parameter vectors

$$x_i, G; i = 1, 2, ..., NP \tag{1}$$

as a population for each generation G. NP does not change during the minimization process. The initial vector population is chosen randomly and should cover the entire parameter space. As a rule, we will assume a uniform probability distribution for all random decisions unless otherwise stated. In case a preliminary solution is available, the initial population might be generated by adding normally distributed random deviations to the nominal solution $x_{nom;0}$. DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector. Let this operation be called mutation, Figure 2 shows a two-dimensional example that illustrates the different vectors which play a part in the generation of $v_{i,G+1}$. The mutated vector's parameters are then mixed with the parameters of another predetermined vector, the target vector, to yield the so-called trial vector. If the trial vector yields a lower cost function value than the target vector, the trial vector replaces the target vector in the following generation. This last operation is called selection. Each population vector has to serve once as the target vector so that NP competitions take place in one generation[13].

2.3 Classification Method

As a result of band selection, the generated feature set is a default set. What we want is the perfect accuracy, so we use the k-Nearest Neighbor(KNN),Naive Bayes Tree(NBTree), Naive Bayes and J4.8 to perform the classification task, which are classic classification in Data Mining[4].In particular, the proposed algorithm represents valid alternatives to classical algorithms as they allow different tradeoffs between the qualities of selected feature subsets and computational cost. In order to prove our method to be more effective than other methods, we chose two band selection methods to compare with our method, the results showed DE is better than Correlation-based Feature Selection(CFS)and Symmetrical Uncertainty(SU)[8].

3 Our Approach for Band Selection

3.1 Data Pre-process

Aiming at demonstrating the technique, we apply it to the problem of classification of minerals in the uranium area of 701 located in Jiling area Gansu

Provence China. In the paper, we take the Pitchblende and Uranotemnite for example, two minerals have 51 examples, the 2151 bands value which range from 350nm 2500nm. From the hyperspectral database we can know every sample from 350nm to 2500nm(interval 1nm) match with a reflectivity. Every reflectivity from 350nm to 2500nm looked as a feature, so we can get 2151 features. In the 51 samples, because the feature parameters of absorption band location play an decision role, so when we analyze feature parameters we only care about one of reflectivity value is or not locate at the absorption band location. In this paper, we selected Pitchblende and Uranotemnite as the dataset, in the dataset, if does not appear absorption band location we set it as 0, if appear then we set it as 1. Because in the sample, 0 feature is nonsense so we deleted them, as a result the 2151 features reduce to 331 features.

3.2 Based on Differential Evolution Band Selection

Solving the function optimization question still a hot topical in evolution computation. In generally, the function optimization question described as:

$$minimize f(X).X = (x_1, x_2, ..., x_n) \subseteq R^n \tag{2}$$

$Where X \in S, X \subseteq R^n$ as the search space, $f(X)$ is the object function, usually every independent variable x_i must satisfy the bound contain.

$$l_i \leq x_i \leq u_i, i = 1, 2, ..., N \tag{3}$$

In this paper, the object function is the information entropy function, which is based on evaluating each band separately using the information entropy measure $H(\lambda)$ defined below[2].

$$H(\lambda) = -\sum_{i=1}^{m} P_i \log_2 P_i \tag{4}$$

H is the entropy measure, P is the probability density function of reflectance values in a hyperspectral band and m is the number of distinct reflectance values. Generally, if the entropy value H is high then the amount of information in the data is large. Thus, the bands are ranked in the ascending order from the band with the highest entropy value (large amount of information) to the band with the smallest entropy value (small amount of information).For example, if we set three bands'integrated entropy for

$$H(\lambda_1, \lambda_2, \lambda_3) = -\sum_{i=1}^{m} P_{i1,i2,i3} \log_2 P_{i1,i2,i3} \tag{5}$$

like this, compute all possible bands combination's integrated entropy, according to descending order arranged the results, then the best band selection question can be solved.

However, in DE the object function aims at searching the global minimization, the information entropy method aims at the maximum entropy measure, we have to change the information entropy function to

$$H(\lambda) = \sum_{i=1}^{m} P_i \log_2 P_i \tag{6}$$

Through the $H(\lambda)$ we get the combination which has the highest information entropy.

Using the proposed approach we can select some best feature band points to solve the classification, but those points do not classify different minerals accurately enough to in practically, because in the test samples the composition and measurement condition are different, even though the same mineral, the feature bands especially the absorption bands positions are very different. So, if we just use those selected points to perform classification should not have enough practice application value, that why need to expand the feature band datasets accord the best feature band points.

We can know from 2.1, absorption bands position is the main parameter, the best feature values result to the best classification result. The distance between unselected nearby absorption bands positions and the selected best points can serve as a measurement which use to an effect standard to measure the best points applying in classification. The method is:

1. Selection of the best feature band points related to sub-interval
   Firstly, according to the best feature band points define a sub-interval which include the best feature point and other absorption band points in sample. For example, Anm is the best feature of band point, then the selected sub-interval must contain other absorption band points appeared near Anm in all samples. But if the selected feature sub-interval is too large, the correlation between the best band point becomes weakening, at the same time, should divide to nearby feature points or noise points. After some researches and experiences, we choose the 40nm as the sub-interval range, the same mean, the best band point's feature sub-interval is [A-20, A+20].
2. In the sub-interval, search all absorption band points $A_1, A_2, A_3, ...$, according their value divide to left and right around A.
3. Compute the distance between A and $A_1, A_2, A_3, ...$separately, then we get two distance value sets: On the left of A is $l_1, l_2, l_3, ...$ and on the right of A is $r_1, r_2, r_3, ...$
4. Compute the mean value of $l_1, l_2, l_3, ...$ get a value l_a and compute the mean value of $r_1, r_2, r_3, ...$ get a value r_a, at the and we get the best band dataset [A- l_a, A+r_a].

3.3 Hyperspectral Data Classification

The object of hyperspectral data classification is to realize self-discrimination of the ground cover classes accord to the measured hyperspectral data. Without loss of generality, in the paper, we use some mature classification technologies in Data mining, such as:KNN, J4.8, Naive Bayes,NBTree. The experiment data constituted by Measured single mineral spectral data. Due to the Pitchblende and Uranotemnite have the similar physical construction, their spectral curves are very similar, even though visual interpretation[7] which is hard to discrimination. What this paper have chose four classification methods use to classify the Pitchblende and Uranotemnite hyperspectral data, then construct effective models of the two minerals,Figure 3 showed the flowchart of DE applied in band selection for hyperspectral data classification.

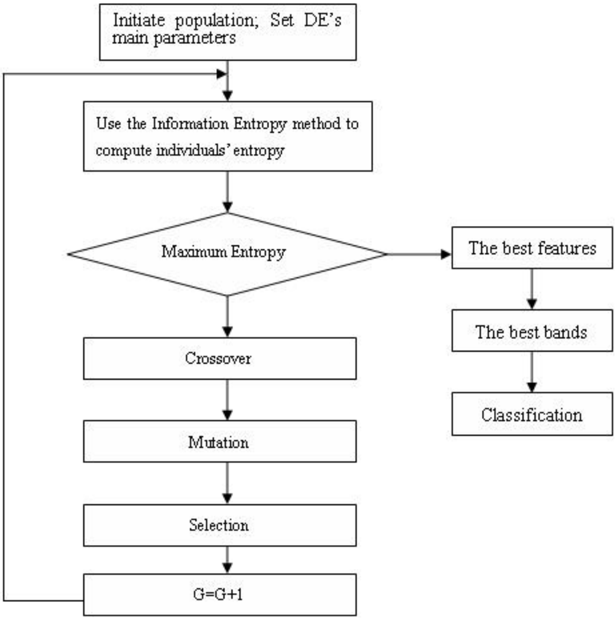

Initiate population; Set DE's main parameters
Use the Information Entropy method to compute individuals' entropy
Maximum Entropy
The best features
The best bands
Classification
Crossover
Mutation
Selection
G=G+1

Fig. 1. The flowchart of DE applied in band selection for hyperspectral data classification

4 Experimental Results

4.1 Experimental Main Parameters Set and Experimental Results

The proposed method for band selection were applied to hyperspectral data classification in mineral discrimination. The method ran on a Dell PC, single Intel Pentium processor and Windows XP operating system.And the main language is C#.

In the method, the parameters in DE are set like this: population P=20, max Generation G=1000, control factor F=0.5, crossover probability Cr=0.9, and the best bands combination M=10, namely, we selected 10 best bands. The method applied in the Pitchblende and Uranotemnite datasets, after ran the DE algorithm we can get 10 feature band's values and the corresponding feature subsets.

Table 1. Ten best features and the corresponding feature subsets

1	2	3	4	5
412nm	418nm	427nm	428nm	431nm
412∽423nm	414∽427nm	420∽434nm	420∽434nm	422∽436nm
6	7	8	9	10
438nm	1357nm	1581nm	1875nm	1898nm
427∽441nm	1354∽1368nm	1574∽1589nm	1866∽1889nm	1891∽1903nm

4.2 Dimensionality Reduction

The aim of selection of few existing bands is to realize reduction of dimensionality. Using the selected 10 bands applied to the original dataset which has 51 samples of two minerals, according to the bands range 350nm 2500nm, total 2151 bands values. We got the new dataset after band selection. In the new dataset, we can see many 0 bands value, which mean that if the original dataset 2151 bands values falling in the 10 selected best bands then display the value, otherwise change to 0. According the method of the 2.2, deleted the 0 features, as the result, we got 331 features, which reduced the dimensionality greatly.

4.3 Classification Results

In order to demonstrate selected bands are more suitable for classification, we chose four methods, such as KNN, J4.8, Naive Bayes, NBtree. in the experiment, every method was tested 10 times and we get the average value.For the purpose of getting a good measure of performance, we used the tenfold cross-validations method[15]. The data is divided randomly into ten parts, in each of which the class is represented in approximately the same proportions as in the full dataset. Each part is held out in turn and the learning scheme trained on the remaining nine-tenths; the its error rate is calculated on the holdout set. Thus the learning procedure is executed a total of ten times, on different training sets (each of which have a lot in common). Finally, the ten error estimates are averaged to yield an overall error estimate.

We use the proposed method to compare with Correlation-based Feature Selection(CFS) method and Symmetrical Uncertainty(SU). The Table 1 show the classification results of the two methods applied in J4.8, Naive Bayes, Naive Bayes Tree. From the Table 1 we can see the classification accuracy based on DE bands selection are all better than the CFS and SU. The main reasons to explain this results are the CFS method based on some features highly correlation. For example, chose one of all features which is the highest correlated with a class, then chose another feature combine with the first one to calculate the correlation of class, if better than other features, then add a good feature to subset, after some iterations, if attain M features(in this paper, M=10), algorithm end. Obviously, the algorithm will exhaust many time, and be easy to falling local minimal. symmetric uncertainty [15] which represents a weighted average of the two uncertainty coefficients,however,it based on the information gain, the results are also worse than DE.

The DE algorithm is a parallel direct search method which conquered those problems. In the algorithm, at the beginning, we will choose 10 features randomly, an compute the information entropy, and keep the best 10 features, all the features through some genetic operations, at the end, get the highest information entropy, as well as get the best 10 features. So this method is quicker than CFS and US, and can find the global optimal results.

Table 2. The classification results use four classification techniques

Classification Method	DE	CFS	SU
KNN	66.6667%	66.6667%	54.7843%
Naive Bayes	**73.94959%**	64.7059%	60.7843%
NBtree	**67.509%**	66.6667%	56.8627%
J4.8	**72.549%**	58.8235%	52.9412%

5 Conclusion

In this paper, we presented a novel evolutionary method DE for band selection in hyperspectral data. The DE algorithm uses a stochastic random search that is not only good convergence properties, simple to understand and to implement, but also particularly easy to work with, having only a few control variables which remain fixed throughout the entire optimization procedure. The method is applied in hyperspectral data is very effective, easy to get the global optimal results, based on the results,use the data mining techniques obviously heighten the classification accuracy. As the development of remote sensing technology, people pay more attention to the hyperspectral data process. In the past, facing up huge hyperspectral data, human just do some simple process by manual, with the artificial intelligence appear, scholars are very interested in this field, and want to apply this technology to the hyperspectral remote sensing, design a new software to automatic extract hyperspectral data information, this also is the new filed that our Data Mining workers to research. This paper is a new try that use the presented algorithm to process hyperspectral data. Although decreased the search time, heightened the classification accuracy, for the purpose of 90 percent classification accuracy, we need to do lots of work in the future.

Acknowledgments

This work was supported by the National High Technology Research and Development Program (863 Program) of China (No. 2009AA12Z117).

References

1. Backer, D.S., Kempeneers, P., Debruyn, W., Scheunders, P.: A Band Selection Technique for Spectral Classification. IEE Geoscience and Remote Sensing Letters 2(3), 319–323 (2005)

2. Bajcsy, P., Groves, P.: Methodology for Hyperspectral Band Seletion. Photogrammetric Engineering and Remote Sensing 70(7), 793–802 (2004)
3. Hughes, G.F.: On the Mean Accuracy of Statistical Pattern Recognizers. IEEE Transactions on Information Theory IT-14, 55–63 (1968)
4. Han, J., Kamber, M., et al.: Data Mining Concepts and Techniques. China Machine Press, Beijing (2002)
5. Karlholm, J., Renhorn, I.: Wavelength Band Selection Method for Multispectral Target Detection. Applied Optics 41(32), 6786–6795 (2002)
6. Kumar, S., Ghosh, J., Crawford, M.M.: Best-bases Feature Extraction Algorithms for Classification of Hyperspectral Data. IEEE Trans. Geosci. Remote Sensing 39(7), 1368–1379 (2001)
7. Peng, W.L., Bai, Z.P., et al.: Introduction of Remote Sensing. Higher Education Press (2002)
8. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Recipes in C. Cambridge University Press, Cambridge (1988)
9. Price, J.C.: Band Selection Procedure for Multispectral Scanners. Applied Optics 33(15), 3281–3287 (1994)
10. Price, K., Storn, R.: Home Page of Differential Evolution (2003), http://www.ICSI.Berkeley.edu/~storn/code.html
11. Riedmann, M., Milton, E.J.: Supervised Band Selection for Optimal Use of Data From Airborne Hyperspectral Sensors. In: IEEE International Geoscience and Remote Sensing Symposium (2003)
12. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
13. Storn, R., Price, K.: Differential EvolutionCA Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997)
14. Wiersma, D.J., Landgrebe, D.: Analytical Design of Multispectral Sensors. IEEE Trans. Geosci. Remote Sensing 18, 180–189 (1980)
15. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Amsterdam (2005)

Diversity Analysis of Opposition-Based Differential Evolution—An Experimental Study

Hui Wang[1], Zhijian Wu[1], Shahryar Rahnamayan[2], and Jing Wang[1]

[1] State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, P.R. China
[2] Faculty of Engineering and Applied Science, University of Ontario Institute of Technology (UOIT), 2000 Simcoe Street North, Oshawa, ON L1H 7K4, Canada
wanghui_cug@yahoo.com.cn, zjwu9551@sina.com,
shahryar.rahnamayan@uoit.ca, wj.jxufe@gmail.com

Abstract. Opposition-based differential evolution (ODE) is a recently proposed DE variant, which has shown faster convergence speed and more robust search abilities than classical DE. The concept of opposition was utilized for the first time in optimization area to propose ODE. It is based on two important steps, generation jumping and elite selection. Some studies have pointed out that the first step improves diversity and provides more potential points to be searched (diversification), while the second step decreases diversity and accelerates convergence speed (intensification). However, there is not any experimental study to support this explanation. In this paper, we present an experimental study to analyze how the diversity changes in ODE. The experimental results confirm the explanation, and show that ODE makes a good balance between generation jumping and elite selection.

Keywords: Differential evolution (DE), Opposition-based DE, Opposition-based learning, Diversity analysis, Global optimization.

1 Introduction

Many real-world problems may be formulated as optimization problems with variables in continuous domains (continuous optimization problems). In the past decades, different kinds of nature-inspired optimization algorithms have been proposed to solve optimization problems, such as Simulated Annealing (SA) [1], Evolutionary Algorithms (EAs) [2], Particle Swarm Optimization (PSO) [3], and Differential Evolution (DE) [4], etc. According to frequently reported experimental studies [5], DE has shown better performance than many other nature-inspired algorithms in terms of convergence speed and robustness over several benchmark functions and real-world problems.

Since the DE algorithm is simple, efficient and easy to implement, it has attracted many researchers to work on improving its performance. Among different kinds of DE variants, opposition-based DE (ODE) [6] is one of the most excellent ones. The ODE employs an opposition-based learning (OBL) concept for opposition-based population initialization and generation jumping, in which current estimate and its

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 95–102, 2010.

opposite estimate are considered at the same time in order to achieve a better approximation for a current candidate solution. It has been proved in [7], an opposite candidate solution has a higher chance to be closer to the global optimum solution than a random candidate solution. In [8], the OBL used in ODE is regarded as a diversity enhancement strategy to explain why ODE converges faster than classical DE. However, it did not give any experimental results to support the explanation. In this paper, we focus on analyzing the diversity of ODE based on experimental studies, and try to explain why ODE shows faster convergence speed and more robust search abilities than classical DE.

The rest paper is organized as follows. In Section 2, the opposition-based DE is briefly introduced. In Section 3, we analyze the diversity of ODE. Section 4 presents the test functions, experimental results and discussions. Finally, the work is concluded in Section 5.

2 Opposition-Based Differential Evolution

Opposition-based Learning (OBL) [9] is a new concept in computational intelligence, and has been proven to be an effective concept to enhance various optimization approaches [10], [11], [12], [13], [14]. When evaluating a candidate solution x to a given problem, simultaneously computing its opposite solution will provide a better chance to find another candidate solution closer to the global optimum.

Opposite Number [6]–Let $x \in$ [a, b] be a real number. The opposite of x is defined by:

$$x^* = a + b - x. \tag{1}$$

Similarly, the definition is generalized to higher dimensions as follows.

Opposite Point [6]–Let $X=(x_1,x_2,...,x_D)$ be a point in a D-dimensional space, where $x_1,x_2,...,x_D \in R$ and $x_j \in [a_j, b_j]$, $j \in \{1,2,...,D\}$. The opposite point $X^* = (x_1^*,x_2^*,...,x_D^*)$ is defined by:

$$x_j^* = a_j + b_j - x_j. \tag{2}$$

By applying the definition of opposite point, the opposition-based optimization can be defined as follows.

Opposition-based Optimization [6]–Let $X=(x_1,x_2,...,x_D)$ be a point in a D-dimensional space (i.e., a candidate solution). Assume $f(X)$ is a fitness function which is used to evaluate the candidate's fitness. According to the definition of the opposite point, $X^* = (x_1^*,x_2^*,...,x_D^*)$ is the opposite of $X=(x_1,x_2,...,x_D)$. If $f(X^*)$ is better than $f(X)$, then update X with X^*; otherwise keep the current point X. Hence, the current point and its opposite point are evaluated simultaneously in order to continue with the fitter one.

Similar to all population-based optimization, two main steps are distinguishable for DE, namely, population initialization and producing new generations by evolutionary operations such as mutation, crossover, and selection. ODE enhances these two steps using the OBL scheme. The original DE is chosen as a parent algorithm and the proposed opposition-based ideas are embedded in DE to accelerate its convergence speed.

For the population initialization, we implement this step as follows.

1) Randomly initialize the population P (N_p).
2) Calculate opposite population OP by

$$OP_{i,j} = a_j + b_j - P_{i,j}, \ . \tag{3}$$

where $i = 1, 2, ..., N_p$; $j = 1, 2, ..., D$, $P_{i,j}$ and $OP_{i,j}$ represent the jth variable of the ith individual of the population and the opposition population, respectively.

3) Select the N_p fittest individuals from $\{P \cup OP\}$ as initial population.

For the second step, ODE employs an opposition-based generation jumping as follows. By applying a similar approach to the current population, the evolutionary process can be forced to jump to a new solution candidate, which ideally is fitter than the current one. Based on a jumping rate J_r, the opposition is conducted on the current population after generating new populations by original DE operators. Then the N_p fittest individuals are selected from the union of the current population and the opposite population. Unlike the population initialization, generation jumping calculates the opposite population based on dynamic boundaries.

$$OP_{i,j} = \mathrm{MIN}_\mathrm{j}^\mathrm{p} + \mathrm{MAX}_\mathrm{j}^\mathrm{p} - P_{i,j}, \ . \tag{4}$$

where MIN_j^p and MAX_j^p are the minimum and maximum values of the jth dimension in current search space, respectively.

```
              Algorithm 1: ODE Algorithm
Begin
  Opposition-based Population initialization;
  while(BFV > VTR and NFC < MAX_NFC) do
    Execute classical DE algorithm;
    If ($rand(0,1) < J_r$) do
      Generate opposite population $OP$;
      Select $N_p$ fittest individuals from $P$ and $OP$ as
      new population $P$;
    End if
  End while
End
```

The main steps of ODE are presented in Algorithm 1, where P is the current population, OP is the opposite population, $rand(0,1)$ is a uniform random number within [0,1], J_r is the jumping rate, BFV is the best fitness value so far, VTR is the value-to-reach [6], NFC is the number of evaluations, and MAX_NFC is the maximum number of evaluations.

3 Diversity Analysis of ODE

In DE, large population size N_p will improve the search abilities of DE to explore more potential regions. But this will slow down the convergence speed. In order to increase the number of potential points to be searched while staying with a lower population size gives rise to the various strategies for diversity enhancement [8]. Opposition-based DE (ODE) is a typical example for diversity enhancement, which uses either the mutant vector obtained in the usual way or its opposite point based on a jumping rate J_r. Once the opposite population is generated, there are 2*N_p individuals available, N_p from the current population and another N_p from the opposite population. Based on an elite selection mechanism, the N_p fittest individuals are selected from those 2* N_p individuals to form the next generation population.

The ODE contains two important steps, generation jumping and elite selection. The first step can be regarded as a diversity enhancement mechanism, which increases the number of potential points to be searched. More potential points are generated to explore more regions. This will be helpful to improve the robustness of ODE. The second step is an elite section, which usually decreases diversity and speeds up convergence because only the best individuals are retained. Higher diversity can make convergence slower but it can increase robustness, while lower diversity can make convergence faster but it easily suffers from premature convergence. From the presented experimental results in [6], ODE obtains faster convergence speed and more robust search abilities than classical DE. It shows that ODE might make a good balance between generation jumping and elite selection. In this paper, we will focus on analyzing how ODE adjusts the diversity from the view of experimental studies.

4 Experimental Verification

4.1 Test Functions

In Rahnamayan's study [6], there were 58 benchmark functions used to verify the performance of ODE. In this paper, only the first 30 functions (f_1-f_{30}) are used (for the rest 28 functions, we got similar conclusions). For the descriptions of these functions, please refer to [6].

4.2 Results

In order to analyze the effects of the two steps, generation jumping (opposition) and elite selection, on the diversity, we divide the two steps into three states, before opposition, after opposition, and after selection. Then we calculate the diversity for each state, and observe the changes of diversity in these states. Let *DivBO*, *DivAO*, and *DivAS* represent the diversity of the three states, respectively (see Fig. 1).

The diversity of population is calculated as follows [15]:

$$Diversity(t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \sqrt{\sum_{j=1}^{D} \left(X_{i,j}(t) - \overline{X_j(t)} \right)^2} , \tag{5}$$

where $X_{i,j}(t)$ is the *j*th value of the *i*th individual in population at generation *t*, and $\overline{X_j(t)}$ is defined by

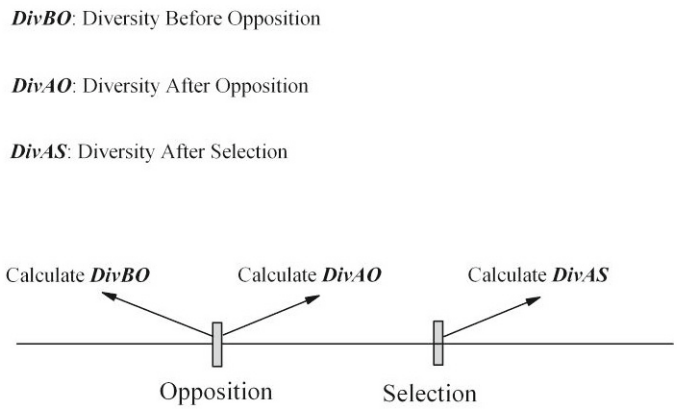

DivBO: Diversity Before Opposition
DivAO: Diversity After Opposition
DivAS: Diversity After Selection
Calculate *DivBO*
Calculate *DivAO*
Calculate *DivAS*
Opposition
Selection

Fig. 1. Calculating diversity in different states

$$\overline{X_j(t)} = \frac{\sum_{i=1}^{N_p} X_{i,j}(t)}{N_p}. \tag{6}$$

Table 1 shows the average results of *DivBO*, *DivAO* and *DivAS* over the evolution on the 30 test functions (The *DivAO* is calculated for the double sized population, 2**N*$_p$). As seen, for all test functions, *DivAO* > *DivBO* and *DivAS* < *DivAO* are satisfied. The former inequality means that the diversity increases after generation jumping (opposition), and the latter one indicates that the diversity decreases after the elite selection. The first step is beneficial for diversification and improving the robustness, while the second step focuses on decreasing diversity and speeding up the convergence (intensification). That is why GODE shows faster convergence and more robust performance than classical DE.

Besides the average diversity analysis, we also calculate the probability of diversity increasing after generation jumping (*Prob_Div_Inc_AO*) and diversity decreasing after elite selection (*Prob_Div_Dec_AS*). The *Prob_Div_Inc_OP* and *Prob_Div_Dec_Sel* are computed as follows.

$$Prob\_Div\_Inc\_AO = \frac{Num\_Div\_Inc\_AO}{Num\_OP}, \tag{7}$$

$$Prob\_Div\_Dec\_AS = \frac{Num\_Div\_Des\_AS}{Num\_OP}, \tag{8}$$

where *Num_Div_Inc_AO* is the number of diversity increasing after generation jumping (it means that *Div_AO* > *Div_BO*), *Num_Div_Dec_AS* is the number of diversity decreasing after selection (it means that *Div_AS* < *Div_AO*), and *Num_OP* is the number of oppositions.

Table 1. The average results of *DivBO*, *DivAO* and *DivAS*

Functions	*DivBO*	*DivAO*	*DivAS*
f_1	1.268	**1.312**	1.189
f_2	1.191	**1.236**	1.114
f_3	16.949	**17.928**	16.795
f_4	0.0268	**0.02817**	0.02534
f_5	1.9893	**2.2335**	1.9658
f_6	146.15	**150.611**	137.30
f_7	1.3837	**1.4122**	1.3487
f_8	3.62709	**3.7891**	3.4074
f_9	1.05907	**1.23316**	0.9195
f_{10}	1.39196	**1.62557**	1.2717
f_{11}	21.1816	**21.6471**	19.457
f_{12}	0.163412	**0.174889**	0.1507
f_{13}	0.450783	**0.52986**	0.45443
f_{14}	1.13873	**1.34012**	1.1723
f_{15}	2.2847	**2.3771**	2.1457
f_{16}	49.656	**51.0923**	50.085
f_{17}	0.0684	**0.1137**	0.0689
f_{18}	0.157298	**0.1821**	0.1570
f_{19}	2.4949	**2.8431**	2.4886
f_{20}	3.1566	**3.9480**	3.243
f_{21}	1.132	**1.211**	1.092
f_{22}	18.958	**19.204**	17.628
f_{23}	68.5354	**70.5751**	64.149
f_{24}	0.19908	**0.20393**	0.19825
f_{25}	1.98192	**2.57376**	1.99292
f_{26}	2.08961	**2.40776**	1.90771
f_{27}	1.4323	**1.83425**	1.3224
f_{28}	1.41144	**1.6711**	1.29114
f_{29}	20.1733	**24.2225**	19.35
f_{30}	0.504458	**0.509575**	0.32736

Table 2 shows the average results of *Prob_Div_Inc_AO* and *Prob_Div_Dec_AS* over 30 runs. It can be seen that both the generation jumping and the elite selection increases and decreases diversity with very high average probabilities (more than 99%). The results demonstrate that the generation jumping always increases the diversity and provides more chances to explore more potential regions, while the elite selection could decreases the diversity each time and accelerate convergence speed. It also proves that *Div_AO* > *Div_BO* and *Div_AS* < *Div_AO* are satisfied.

Table 2. The average results of *DivBO*, *DivAO* and *DivAS*

Functions	*Prob_Div_Inc_AO*	*Prob_Div_Dec_AS*
f_1	100%	100%
f_2	100%	100%
f_3	94%	91%
f_4	100%	100%
f_5	100%	100%
f_6	100%	100%
f_7	100%	100%
f_8	100%	100%
f_9	89%	100%
f_{10}	100%	100%
f_{11}	100%	100%
f_{12}	100%	100%
f_{13}	100%	100%
f_{14}	100%	100%
f_{15}	100%	100%
f_{16}	100%	100%
f_{17}	100%	100%
f_{18}	100%	99.7%
f_{19}	100%	100%
f_{20}	100%	100%
f_{21}	100%	100%
f_{22}	100%	100%
f_{23}	100%	100%
f_{24}	100%	100%
f_{25}	100%	100%
f_{26}	93%	100%
f_{27}	100%	100%
f_{28}	100%	100%
f_{29}	100%	94%
f_{30}	100%	100%
Average	**99.2%**	**99.5%**

5 Conclusion

In this paper, we present an experimental study on diversity analysis of ODE. In ODE, there are two important steps, generation jumping and elite selection. The first step is beneficial for increasing diversity and exploring more promising regions, while the second one is helpful to speed up convergence. Although these two steps are incompatible, ODE makes a good balance between them towards searching candidate solutions. The experimental verification supports our explanations well. However, we only present an experimental study to analyze the diversity of ODE. That may not be enough to explain the advantages of ODE. Some theoretical analysis will be conducted in our future work.

Acknowledgments. This work was supported by the Jiangxi Province Science & Technology Pillar Program (No.: 2009BHB16400), and the National Natural Science Foundation of China (No.: 61070008).

References

1. Kirkpatrick, S., Gelatt, C.D., Vecchi, P.M.: Optimization by Simulated Annealing. Science 220, 671–680 (1983)
2. Back, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Publisher, New York (1996)
3. Storn, R., Price, K.: Differential Evolution–A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997)
4. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of International Conference on Neural Networks, vol. IV, pp. 1942–1948. IEEE Press, Piscataway (1995)
5. Vesterstrom, J., Thomsen, R.: A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems. In: Proc. Congress on Evolutionary Computation, Portland, vol. 2, pp. 1980–1987 (2004)
6. Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A.: Opposition-Based Differential Evolution. IEEE Transaction on Evolutionary Computation 12(1), 64–79 (2008)
7. Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A.: Opposition versus Randomness in Soft Computing Techniques. Applied Soft Computing, 906–918 (2008)
8. Storn, R.: Differential Evolution Research – Trends and Open Questions. In: Chakraborty, U.K. (ed.) Advances in Differential Evolution. SCI, vol. 143, pp. 1–31 (2008)
9. Tizhoosh, H.R.: Opposition-Based Learning: A New Scheme for Machine Intelligence. In: Proceedings of International Conference on Computational Intelligence for Modeling Control and Automation, Vienna, Austria, pp. 695–701 (2005)
10. Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A.: Opposition-Based Differential Evolution Algorithms. In: Proceedings of Congress on Evolutionary Computation, pp. 2010–2017. IEEE Press, Vancouver (2006)
11. Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A.: Opposition-Based Differential Evolution for Optimization of Noisy Problems. In: Proceedings of Congress on Evolutionary Computation, pp. 1865–1872. IEEE Press, Vancouver (2006)
12. Wang, H., Liu, Y., Zeng, S.Y., Li, H., Li, C.H.: Opposition-Based Particle Swarm Algorithm with Cauchy Mutation. In: Proceedings Congress on Evolutionary Computation, pp. 4750–4756. IEEE Press, Singapore (2007)
13. Rahnamayan, S., Wang, G.G.: Solving Large Scale Optimization Problems by Opposition-Based Differential Evolution (ODE). Transactions on Computers 7(10), 1792–1804 (2008)
14. Wang, H., Wu, Z.J., Rahnamayan, S., Kang, L.S.: A Scalability Test for Accelerated DE Using Generalized Opposition-Based Learning. In: Proceedings of International Conference on Intelligent System Design and Applications, Pisa, Italy, pp. 1090–1095 (2009)
15. Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. Wiley & Sons, Chichester (2005)

Hybrid Differential Evolution Algorithm with Chaos and Generalized Opposition-Based Learning

Jing Wang[1,2], Zhijian Wu[1], and Hui Wang[1]

[1] State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, China
[2] School of Software & Communication Engineering,
Jiangxi University of Finance and Economics, Nanchang 330013, China
wj.jxufe@gmail.com

Abstract. This paper presents a hybrid differential evolution (DE) algorithm based on chaos and generalized opposition-based learning (GOBL). In this algorithm, GOBL strategy transforms current search space into a new search space with a random probability, which provides more opportunities for the algorithm to find the global optimum. When the GOBL strategy isn't executed, the chaotic operator, like a mutation operator, will be introduced to help the DE to jump out local optima and improve the global convergence rate. Simulation results show that this hybrid DE algorithm can electively enhance the searching efficiency and greatly improve the searching quality.

Keywords: differential evolution, generalized opposition-based learning, chaos, evolutionary algorithm.

1 Introduction

Differential Evolution (DE) is an effective robust optimization algorithm which was proposed by Rainer Storn and Kenneth Price in 1995[1, 2]. It became one of the fastest stochastic algorithms quickly during the first international IEEE evolutionary algorithm race held in Japan[3] because of the simpler evolutionary operator and fewer control parameters. However, according to No-Free-Lunch theorem [4], it also has insurmountable shortcomings, the slower convergence rate in latter periods, even failing to local extremes. It's search performance depends on balancing the ability of global exploration and local development.

In this paper, an enhanced DE, based on generalized opposition-based learning (GOBL) and chaotic operator, is proposed to accelerate the convergence rate of classical DE. The GOBL was introduced in our previous work [5] which presented a general model for opposition-based learning (OBL). Accelerated DE by OGBL and Chaos has been called CGODE in this paper.

The paper is organized as follows. In Section 2, the classical DE algorithm is briefly reviewed. The GOBL technique and chaotic operator are presented in Section 3. Section4 gives an implementation of the proposed algorithm, CGODE. In Section 5, the testing of the proposed methods through benchmark problems are carried out and the experimental results are compared. Finally, the work is summarized in Section 6.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 103–111, 2010.

2 The A Brief Review of Differential Evolution

DE is a population-based stochastic search algorithm, and has been successfully applied to solve complex problems including linear and nonlinear, unimodal and multimodal functions. It has been shown that DE is faster and more robust on these functions than many other evolutionary algorithms [6].

There are several variants of DE [3], where the most popular variant is shown by "*DE*/*rand*/1/*bin*" which is called classical version. The proposed algorithm is also based on this DE scheme. Let us assume that $X_i(t)(i = 1, 2, \ldots, N_p)$ is the *i*th individual in population P(t), where *ps* is the population size, *t* is the generation index, and *P*(*t*) is the population in the *i*th generation. The main idea of DE is to generate trial vectors. Mutation and crossover are used to produce new trial vectors, and selection determines which of the vectors will be successfully selected into the next generation.

Mutation-For each vector $X_i(t)$ in Generation *t*, a mutant vector V is generated by

$$V_i(t) = X_{i1}(t) + F\big(X_{i2}(t) - X_{i3}(t)\big), i1 \neq i2 \neq i3 \tag{1}$$

where i =1, 2,..., N_p and *i*1, *i*2, and *i*3 are mutually different random integer indices within [1, N_p]. The population size N_p should be satisfied N_p≥4 because *i*, *i*1, *i*2, and *i*3 are different. F∈ [0, 2] is a real number that controls the amplification of the difference vector ($X_{i2}(t)$–$X_{i3}(t)$).

Crossover-Like genetic algorithms, DE also employs a crossover operator to build trial vectors by recombining two different vectors. The trial vector is defined as follows:

$$U_i(t) = (U_{i1}(t), U_{i2}(t), \ldots, U_{in}(t)) \tag{2}$$

where j =1, 2,...,n and

$$U_{ij}(t) = \begin{cases} V_{ij}(t), \text{if } rand_j(0,1) \leq CR \text{ or } j = l \\ X_{ij}(t), \text{Otherwise} \end{cases} \tag{3}$$

CR ∈ (0, 1) is the predefined crossover probability, and $rand_j$ (0, 1) is a random number within (0, 1) for the *j*th dimension, and l ∈ {1, 2,…,*n*} is a random parameter index.

Selection-A greedy selection mechanism is used as follows:

$$X_i(t) = \begin{cases} U_i(t), \text{if } f\big(U_i(t)\big) < f(X_i(t)) \\ X_i(t), \text{Otherwise} \end{cases} \tag{4}$$

Without loss of generality, this paper only considers minimization problem. If, and only if, the trial vector $U_i(t)$ is better than $X_i(t)$, then $X_i(t)$ is set to $U_i(t)$; otherwise, the $U_i(t)$ is unchanged.

3 The Related Strategies

3.1 Generalized Opposition Based Learning

In our previous work, we presented a new evolutionary technique to transform solutions in current search space to a new search space, we called it generalized opposition-based learning strategy (GOBL) [5]. This method transforms individual in current search space to a new search space. At the same time, by simultaneously evaluating the solutions in current search space and transformed space, it can improve the chance to find solutions more closely to the global optimum than current solutions.

The GOBL strategy is described briefly as follows.

Let $X = (X_{1,}X_{2,} \dots, X_{n,})$ be a solution in an n-dimensional space, then $X = (X_{1,}^{*} X_{2,}^{*} \dots, X_{n,}^{*})$ is the corresponding solution of X in the transformed search space. This new strategy is defined by equation (5).

$$X_{ij}^{*} = k[a_j(t) + b_j(t)] - X_{ij} \tag{5}$$

where X_{ij} is the jth vector of the ith solution in the population, $X_{i\ j}^{*}$ is the transformed solution of X_{ij}, k can be set as a random number within [0,1]. $a_j(t)$ and $b_j(t)$ are the minimum and maximum values of the jth dimension in current search space respectively, and t=1,2,…, indicates the generations, i=1,2,…, indicates the size of the population, j=1,2,…, indicates the dimensions.

Assume $f(X)$ is a fitness function which is used to evaluate the solution's fitness. If $f(X^{*})$ is better than $f(X)$, then update X with X^{*}; Otherwise keep the current solution X. Hence, the current solution and its transformed solution are evaluated simultaneously, and the better one will be selected.

When the STS is conducted, new solutions in transformed population (*TP*) are calculated according to the equation (5). The *PopSize* means the size of population. The specific steps are as follows.

(1) Calculate the *fitness* of solutions in current population *P*.
(2) Calculate the new solutions in transformed population *TP* by equation (1).
(3) Select the *PopSize* fittest individuals from $P \cup TP$ as the new current population.

3.2 Chaotic Operator

Chaos is a universal nonlinear phenomenon with stochastic property, ergodic property and regular property, whose ergodicity can be used as a kind of mechanism for optimization to effectively avoid the search being trapped in local optimum, so that chaos has been a novel and promising tool for global optimization [7].

Chaotic mapping can be regarded as a component of effective optimization algorithms relying on its university, randomcity and sensitivity dependence on the initial conditions. In this paper, logistic map, one of the simplest map which was brought to the attention of scientists by May [8] that appears in nonlinear dynamics of biological population evidencing chaotic behavior, is employed for hybrid DE. The equation is defined as follows:

$$CX_{n+1} = \alpha CX_n(1 - CX_n) \tag{6}$$

In this equation, X_n is the nth chaotic number where n denotes the iteration number. Obviously, $X_n \in (0,1)$ under the conditions that the initial $X_0 \in (0,1)$, and that $X_0 \in \{0.0, 0.25, 0.5, 0.75, 1, 0\}$, $\alpha = 4$ have been used in the experiments. Literature [9] proposed a chaotic local search (CLS) strategy to enhance the performance of particle swarm optimization. Based on the concept of CLS, we propose the chaotic operator as follows.

(1) Mapping the variables X_i^n among the intervals $(X_{min,i}, X_{max,i})$ to chaotic variables CX_i^n located in the interval (0,1) by using the following equation.(i=1,2,…,D, D means the dimensions)

$$CX_i^n = (X_i^n\text{-}X_{min,i})/(X_{max,i} - X_{min,i}) \tag{7}$$

(2) Produce the next generation of chaotic variable CX_i^{n+1} by equation(6)

(3) Mapping the chaotic variables CX_i^{n+1} to the variables X_i^{n+1} by following equation.

$$X_i^{n+1} = X_{min,i} + CX_i^{n+1}(X_{max,i} - X_{min,i}) \tag{8}$$

(4) Calculate the *fitness* of new solutions with variables X_i^{n+1}

(5) Select the *PopSize* fittest individuals from current variables X_i^{n+1} and current population as the new current population.

4 Implementation of the Hybrid Differential Evolution Algorithm

The DE algorithm search performance depends on balancing the ability of global exploration and local development. However, when the balance is broken, the searching stagnates easily, and the population will easily fall into local optima especially in latter periods. Under these circumstances, the current search space hardly contains the global optimum. So it is difficult for the current population to achieve better solutions. In this part, GOBL strategy is embedded in DE algorithm and improves the performance of the algorithm, and the strategy transforms current search space into a new search space with a random probability, which provides more opportunities for the algorithm to find the global optimum and accelerates convergence speed. When the GOBL strategy isn't executed, the chaotic operator, like a mutation operator, will be introduced to help the DE jumping out local optima and provides more opportunities to find global optimum.

The main steps of CGODE are given as follows.

Algorithm: STS-EGT

```
Begin
  Initialize the population P(t), P(t)={$x_1(t)$, $x_2(t)$,…,$x_n(t)$}, $x_i(t) \in S$;
  Execute GOBL operation and form TP(t) by equation (5);
  Evaluate the population P(t)∪ TP(t);
  Calculate evaluation times;
  Select Popsize fittest individuals from P(t)∪ TP(t) as current population P(t)
  while (NE ≤ $MAX_{FES}$)
    if(rand(0,1)<$p_s$)
      Execute GOBL operation and form TP(t) by equation (5)
      Evaluate the population TP(t);
      Calculate evaluation times;
      Select Popsize fittest individuals from P(t)∪ TP(t) as next population P(t+1)
    else
      Execute DE operation by equation (1), (3), (4);
      Evaluate the population P(t);
      Calculate evaluation times;
      Select Popsize/5 fittest individuals with a elite strategy;
      Execute chaotic operator and form chaotic population CP(t);
      Evaluate the population CP(t);
      Calculate evaluation times;
      Select Popsize fittest individuals from P(t) ∪ CP(t) as next population P(t+1)
    end if
  end while
  Output $x_{best}$ and f($x_{best}$);
End
```

Where P is the current population, TP is the transformed population after using GOBL, and CP is the population formed by chaotic operator, *Popsize* is the population size, NE is the current evaluation times, and MAX_{FES} is the maximum number of evaluations. The value of *k* in equation(5) is related to the characteristics of given problems. Different values of *k* will result in different search spaces which may cover the global optimum(s). It is also very important to set the value of p_s which determines the implementation probability of the GOBL strategy. In this paper, *k* and *ps* in CGODE are empirical studies. We will focus on the investigations of *k*, *ps* in future work.

5 Experiments and Analysis

5.1 Parameter Settings and Measures

For the experiments, the following eleven scalable benchmark problems have been considered:

1. *f*1-*f*6 of the CEC'2008 Special Session and Competition on Large Scale Global Optimization test suite[10].
2. Schwefel's Problem 2.22 (*f*7), Schwefel's Problem 1.2 (*f*8), Extended f10 (*f*9), Bohachevsky (*f*10), and Schaffer (*f*11), see [11] for their descriptions.

All the experiments are conducted 25 times with different random seeds, and the best and average results throughout the optimization runs are recorded. All the functions used in this paper are to be minimized.

The settings for the CGODE algorithms are:

- Population size:100
- Dimension size: 50
- The probability p_s : 0.1
- Maximum number of function evaluations: MAX_{FES}=5000*D
- The value of *k* in the GOBL: *rand* (0,1)
- Mutation Strategy: DE/rand/1/bin (classical DE)

5.2 Fixed-Evaluations Results

Table1 lists the average best function value of 25 independent runs between CGODE, GODE and DE. It can be obviously seen that CGODE is better than GODE and DE, in most case.

Table 1. Comparsion among DE, GODE, CGODE on f_1–f_{11}

Fuctions	CGODE	GODE	DE
*f*1	0	0	0
*f*2	11.2016	0.066749	0.086734
*f*3	20.4644	31.2262	30.8011
*f*4	0	6.57E-15	0.099496
*f*5	0	0	0
*f*6	9.33E-15	1.32E-14	1.04E-14
*f*7	1.81E-28	1.16E-21	1.88E-18
*f*8	1.37E-19	1.09E-07	0.074929
*f*9	4.77E-12	5.27E-08	2.12E-06
*f*10	5.75E-56	5.28E-42	4.39E-34
*f*11	5.04E-12	3.25E-08	2.09E-06

5.3 Convergence Rate Analysis

The results of Convergence process among DE, GODE[12], CGODE, are given by table2-5. The "FES" means the current evaluation times, and the corresponding value is also recorded. Seen from the results, the Convergence rate of CGODE is almost faster than other algorithms all the time. The CGODE algorithm not only produce the global optimal value but also give faster convergence rate. Many statistical measures as following tables justify the superiority of the proposed methods.

Table 2. The Comparsion of convergence among DE, GODE, CGODE on f_1–f_3

F		f1			f2			f3	
FES	CGODE	GODE	DE	CGODE	GODE	DE	CGODE	GODE	DE
25000	4.50803	19.7923	11.8943	26.1708	45.1609	44.7104	15500	63906.3	15514.4
50000	0.000305	0.004301	0.001908	12.2441	18.6414	20.1613	45.6259	64.727	53.6482
75000	2.08E-08	1.06E-06	2.30E-07	6.77497	7.53939	9.02714	40.2283	44.7074	44.1611
100000	1.25E-12	2.68E-10	3.35E-11	4.54024	3.31712	3.7285	36.5647	41.9578	42.007
125000	1.14E-16	5.06E-14	4.00E-15	3.79194	1.40897	2.07497	32.9064	39.4428	40.7629
150000	6.59E-21	1.74E-17	9.81E-19	3.51296	0.580783	1.28755	29.2883	37.6897	38.6731
175000	7.32E-25	4.56E-21	1.82E-22	3.38229	0.258867	0.707761	26.2471	35.7106	37.8596
200000	0	8.91E-25	2.22E-26	3.34916	0.127583	0.376777	22.8253	33.5307	36.9011
225000	0	1.58E-30	0	3.33529	0.055848	0.20531	19.1658	31.3827	34.9901
250000	0	0	0	3.32847	0.024362	0.097423	16.003	29.7896	32.8271

Table 3. The Comparsion of convergence among DE, GODE, CGODE on *f*4–*f*6

F		f4			f5			f6	
FES	CGODE	GODE	DE	CGODE	GODE	DE	CGODE	GODE	DE
25000	142.481	197.317	176.673	1.06917	1.12895	1.17009	1.51608	2.72093	2.32246
50000	54.8417	107.124	110.481	0.000612	0.006831	0.006489	0.006696	0.020012	0.011048
75000	23.3213	66.5803	64.5264	3.44E-08	1.98E-06	1.06E-06	4.13E-05	0.000267	0.000155
100000	2.58999	42.8694	38.1436	1.06E-12	4.46E-10	2.11E-10	3.31E-07	4.29E-06	1.70E-06
125000	0.000116	13.5121	12.022	0	9.06E-14	4.23E-14	2.97E-09	7.10E-08	1.75E-08
150000	7.88E-09	0.43756	0.047404	0	0	0	1.91E-11	9.16E-10	2.38E-10
175000	5.92E-13	8.57E-05	8.83E-06	0	0	0	2.35E-13	1.99E-11	2.38E-12
200000	0	2.36E-08	9.53E-10	0	0	0	1.82E-14	2.81E-13	4.31E-14
225000	0	4.80E-12	1.60E-13	0	0	0	7.55E-15	2.18E-14	1.47E-14
250000	0	0	0	0	0	0	7.55E-15	1.47E-14	1.11E-14

Table 4. The Comparsion of convergence among DE, GODE, CGODE on *f*7–*f*9

F		f7			f8			f9	
FES	CGODE	GODE	DE	CGODE	GODE	DE	CGODE	GODE	DE
25000	0.218817	0.969229	2.88135	11.2702	363.109	12250	40.7939	86.0561	101.099
50000	8.85E-05	0.005345	0.027127	0.107183	16.2524	3384.55	2.66447	14.16	19.2822
75000	4.64E-08	2.40E-05	0.000289	3.26E-06	2.55915	561.922	0.196599	2.24479	3.75671
100000	2.72E-11	9.33E-08	2.34E-06	7.23E-10	0.059635	139.946	0.016241	0.250071	0.86402
125000	3.78E-14	6.12E-10	2.30E-08	8.18E-13	0.001805	55.4984	0.000199	0.028268	0.242043
150000	2.34E-17	1.65E-12	2.75E-10	6.74E-15	3.02E-05	9.08529	8.97E-06	0.002177	0.03899
175000	1.55E-20	6.24E-15	3.16E-12	1.38E-17	2.71E-06	3.28119	2.74E-07	4.39E-05	0.00618
200000	4.59E-24	2.75E-17	3.01E-14	1.21E-20	1.07E-07	0.731416	9.63E-09	2.28E-06	0.00038
225000	3.23E-27	1.17E-19	3.20E-16	6.47E-23	5.11E-09	0.225289	4.34E-10	1.29E-07	3.03E-05
250000	4.33E-30	6.03E-22	3.21E-18	7.17E-25	2.63E-10	0.061028	8.76E-12	7.44E-09	2.89E-06

Table 5. The Comparsion of convergence among DE, GODE, CGODE on *f*10–*f*11

F	*f*10			*f*11		
FES	CGODE	GODE	DE	CGODE	GODE	DE
25000	0.132803	4.69878	22.4229	25.7176	78.4845	94.5316
50000	1.23E-07	0.000204	0.00893	1.15032	12.4045	19.7042
75000	8.80E-14	1.24E-08	1.11E-06	0.064832	1.89072	4.49351
100000	4.70E-21	3.49E-13	1.89E-10	0.002076	0.407044	0.956396
125000	4.00E-27	7.46E-19	2.39E-14	3.96E-05	0.054753	0.24096
150000	2.40E-33	1.50E-23	2.92E-19	1.13E-06	0.008043	0.040614
175000	2.76E-40	5.48E-28	3.24E-23	2.31E-08	0.000318	0.006354
200000	8.98E-47	4.68E-33	4.81E-27	3.76E-10	1.73E-05	0.000361
225000	1.00E-52	1.18E-37	4.60E-31	8.19E-12	1.14E-06	2.96E-05
250000	6.86E-59	4.05E-42	5.36E-35	2.36E-13	8.06E-08	2.40E-06

5.4 Summaries of Experiment Results

From experiment results on the eleven benchmark functions, we can get the following conclusions:

1) DE is a highly efficient algorithm, but the convergence rate for some special functions usually reduces in latter periods, even failing to local extremes.
2) Through transforming the search space, GOBL strategy, which contains space transformation technique, can get the algorithm out from local optimum, and ultimately achieve the optimal solution. Particularly, for the functions with high-dimensionalities and many minima, the effects would be more obvious.
3) Introducing chaotic operator with ergodicity, irregularity and the stochastic property into DE could improve the global convergence obviously. It can be helpful to escape more easily from local minima than can be done through the traditional DE.

5 Conclusions

In this paper, a novel hybrid DE algorithm named CGODE which embed the GOBL strategy and chaotic operator is proposed. The proposed approach not only performs exploration by using the population-based evolutionary searching ability of DE with GOBL, but also performs exploitation by using the chaotic operator. The proposed CGODE is superior in term of searching quality, efficiency and robustness on initial conditions. Besides, to achieve the same searching quality, CGODE is much faster than traditional DE and GODE .

In future research works, we shall focus on how to apply this novel hybrid algorithm to solve more practical problems.

Acknowledgments. This work was supported by the Jiangxi Province Science & Technology Pillar Program (No.: 2009BHB16400), and the National Natural Science Foundation of China (No.:61070008).

References

1. Storn, R., Price, K.: Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces. University of California, Berkeley (1995)
2. Lampinen, J.: A bibliography of differential evolution algorithm[EB/ OL] (2002-10-14), http://www.lut.fi/~jlampinen/debiblio.htm
3. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
4. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1, 67–82 (1997)
5. Wang, H., Wu, Z.J., Liu, Y., Wang, J., Jiang, D., Chen, L.: Space transformation search: A new evolutionary technique. In: World Summit on Genetic and Evolutionary Computation (2009)
6. Vesterstrom, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: Proc. Congr. Evol. Comput., vol. 2, pp. 1980–1987 (2004)
7. Wang, L., Zheng, D.Z., Li, Q.S.: Survey on Chaotic Optimization Methods. Journal of Computing Technology and Automation 20 (2001) (Chinese)
8. May, R.: Simple mathematical models with very complicated dynamics. Nature 261, 459–467 (1976)
9. Liu, B., Wang, L., Jin, Y.H.: Improved particle swarm optimization combined with chaos. Journal of Chaos, Solitons & Fractals 25, 1261–1271 (2005)
10. Tang, K., Yao, X., Suganthan, P.N., MacNish, C., Chen, Y.P., Chen, C.M., Yang, Z.: Benchmark Functions for the CEC 2008 Special Session and Competition on LargeScale Global Optimization, Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China (2007), http://nical.ustc.edu.cn/cec08ss.php
11. Herrera, F., Lozano, M.: Workshop: Evolutionary Algorithms and other Metaheuristics for Continuous Optimization Problems - A Scalability Test (2009), http://sci2s.ugr.es/programacion/workshop/Scalability.html
12. Wang, H., Wu, Z.J., Rahnamayan, S., Kang, L.S.: A Scalability Test for Accelerated DE Using Generalized Opposition-Based Learning. In: Ninth International Conference on Intelligent Systems Design and Applications (2009)

Self-adapting Differential Evolution Algorithm with Chaos Random for Global Numerical Optimization

Ming Yang[1], Jing Guan[2], Zhihua Cai[1], and Lu Wang[3]

[1] School of Computer Science, China University of Geosciences,
Wuhan, 430074, China
yangming0702@gmail.com
[2] Institute for Pattern Recognition and Artificial Intelligence,
Huazhong University of Science and Technology, Wuhan, 430074, China
g_jing0414@yahoo.com.cn
[3] Department of Computer Engineering, Ordnance Engineering College,
Shijiazhuang, 050003, China

Abstract. Choosing the proper control parameters for DE is quite difficult because the best settings for the control parameters can be different for different functions. In this paper, the proposed self-adaptive method is an attempt to determine the values of control parameters *F* and *CR*. In this method, the adjusting of *F* and *CR* associates with fitness of individuals and the new values are Chaos random numbers. The experiment results show that this algorithm can attain better solutions than other algorithms for multimodal functions.

Keywords: differential evolution, self-adapting, chaos random, global numerical optimization.

1 Introduction

Differential Evolution (DE) is a simple yet powerful evolutionary algorithm (EA) for global optimization introduced by Price and Storn [1]. The DE algorithm has gradually become more popular and has been used in many practical cases, mainly because it has demonstrated good convergence properties and is principally easy to understand [2].

DE has three control parameters: amplification factor of the difference vector—*F*, crossover control parameter—*CR*, and population size—*NP*. The original DE algorithm keeps all three control parameters fixed during the optimization process. However, there still exists a lack on knowledge of how to find reasonably good values for the control parameters of DE, for a given function [3].

Although DE algorithm has been shown to be a simple and powerful for optimizing continuous functions, users are still faced with the problem of preliminary testing and hand-tuning of the evolutionary parameters prior to commencing the actual optimization process [4]. Algorithms with suitable parameters can get good solution, while unsuitable parameters can only get bad solution. DE is also sensitive to the parameters setting. Choosing suitable parameter values is, frequently, a problem-dependent task and requires previous experience of the user. Despite its crucial importance, there is no

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 112–122, 2010.

consistent methodology for determining the control parameters [5]. As a solution, self-adaptation has proved to be highly beneficial for automatically and dynamically adjusting evolutionary parameters, such as crossover and mutation rates. Self-adaptation is usually used in Evolution Strategies (ES) [6]. Self-adaptation enables an evolutionary strategy to adapt itself to any general class of problem, by reconfiguring itself accordingly, and dose this without any user interaction [7, 8].

Ali and Törn in [9] proposed new versions of the DE algorithm, introducing an auxiliary population of NP individuals alongside the original population and a rule for calculating the control parameter F automatically. Y.-C. Jiao [10] proposed a modification of the DE algorithm, applying a number-theoretic method to generate the initial population, and using simplified quadratic approximation with the three best points. A method for gradually reducing population size is proposed in [11], which improved the efficiency and robustness of the algorithm and can be applied to any variant of a DE algorithm.

Liu and Lampinen [3] proposed a version of DE, where the mutation control parameter and the crossover control parameter are adaptive. Teo in [4] made an attempt at self-adapting the population size parameter, in addition to self-adapting crossover and mutation rates. Brest et al. in [12] proposed a DE algorithm, using a self-adapting mechanism on the control parameters *F* and *CR*. Qin and Suganthan in [13] proposed the Self-adaptive Differential Evolution algorithm (SaDE), where the choice of learning strategy and the two control parameters *F* and *CR* do not require pre-defining. During evolution, suitable learning strategy and parameter settings are gradually self-adapted, according to the learning experience. Brest et al. [14] reported performance comparison of certain selected DE algorithms, which use different self-adapting or adapting control parameter mechanisms. M. Montes et al. [15] conducted a comparative study of DE variants. They proposed a rule for changing control parameter *F* at random from interval [0.4, 1.0] at generation level. They used different values of control parameter *CR* for each problem. The best *CR* value for each problem was obtained by additional experimentation. J. Tvrdik in [16] proposed a DE algorithm where competition between different control parameter settings was used. Liu and Lampinen [3] proposed a fuzzy adaptive differential evolution (FADE) algorithm, which dynamically controls DE parameters *F* and/or *CR*, and converged much faster than the traditional DE, particularly when the dimensionality of the problem is high or the problem concerned is complicated [3].

In this paper, the parameter control technique is based on the self-adaptation of two parameters (*F* and *CR*), associated with the fitness of individual in evolutionary process, and using Chaos random number to adapt *F* and *CR*. The main goal here is to produce a flexible DE without hand-tuning of the evolutionary parameters.

2 Differential Evolution and Related Work

2.1 Original DE Algorithm

There are several variants of DE [1, 17]. In this paper, we use the DE scheme which can be classified using notation as *DE/rand/1/bin* strategy.

A set of D optimization parameters is called an individual. It is represented by a D-dimensional parameter vector. A population consists of NP parameter vectors $\mathbf{x}_{i,G}$, i=1,2,…,NP, where G denotes one generation, NP is the number of members in a population.

According to Storn and Price [1, 17], DE have three operations: mutation, crossover and selection.

The crucial idea behind DE is a scheme for generation trial parameter vectors. Mutation and crossover are used to generate new vectors (trial vectors), and selection then determines which of the vectors will survive into the next generation.

A. Mutation

For each target vector $\mathbf{x}_{i,G}$, a mutation vector $\mathbf{v}$ is generated according to

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r1,G} + F(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}),\ \ r_1 \neq r_2 \neq r_3 \neq i \tag{1}$$

with randomly chosen indexes r_1, r_2, $r_3 \in [1, NP]$. Note that indexes have to be different from each other and from the running index i so that NP must be at least four. F is a real number ($F \in [0, 2]$) that controls the amplification of the difference vector ($\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}$).

B. Crossover

The target vector is mixed with the mutated vector, using the following scheme, to yield the trial vector

$$\mathbf{u}_{i,G+1} = (\mathbf{u}_{1i,G+1}, \mathbf{u}_{2i,G+1}, \ldots, \mathbf{u}_{Di,G+1}) \tag{2}$$

where

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1}, & if\ \ rand(j) \leq CR \text{ or } j = rn(i) \\ x_{ji,G}, & if\ \ rand(j) > CR \text{ and } j \neq rn(i) \end{cases} \tag{3}$$

for j=1,2,…,D, $rand(j) \in [0, 1]$ is the jth evaluation of a uniform random generator number. CR is the crossover constant $\in [0, 1]$, which has to be determined by the user. $rn(i) \in \{1,2,\ldots,D\}$ is a randomly chosen index which ensures that $\mathbf{u}_{i,G+1}$ gets at least one element from $\mathbf{v}_{i,G+1}$. Otherwise, no new parent vector would be produced and the population would not alter.

C. Selection

A greedy selection scheme is used

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G+1}, & if\ f(\mathbf{u}_{i,G+1}) < f(\mathbf{x}_{i,G})\ for\ minimization\ problems \\ \mathbf{x}_{i,G}, & otherwise \end{cases} \tag{4}$$

for j=1,2,…,D, if, and only if, the trial vector $\mathbf{u}_{i,G+1}$ yields a better cost function value than $\mathbf{x}_{i,G}$, then $\mathbf{x}_{i,G+1}$ is set to $\mathbf{u}_{i,G+1}$; otherwise, the old value $\mathbf{x}_{i,G}$ is retained.

2.2 Related Work

A self-adaptive DE refers to the self-adapting mechanism on the control parameters, proposed by Brest et al. [12]. This self-adapting mechanism uses the already exposed '*rand/1/bin*' strategy (Equation (1)).

In [12] a self-adaptive control mechanism was used to change the control parameters *F* and *CR* during the run. The third control parameter *NP* was kept unchanged.

Each individual in the population was extended using the values of these two control parameters (Fig.1). Both of them were applied at individual level. The better values for these (encoded) control parameters lead to better individuals, which, in turn, are more likely to survive and produce offspring and, hence, propagate these better parameter values.

New control parameters $F_{i,G+1}$ and $CR_{i,G+1}$ were calculated as follows:

$$F_{i,G+1} = \begin{cases} F_l + rand_1 \cdot F_u & if\ rand_2 < t_1 \\ F_{i,G}, & otherwise \end{cases} \tag{5}$$

$x_{1,1,G}$	$x_{1,2,G}$	...	$x_{1,D,G}$	$F_{1,G}$	$CR_{1,G}$
$x_{2,1,G}$	$x_{2,2,G}$	...	$x_{2,D,G}$	$F_{2,G}$	$CR_{2,G}$
...	...	...	...	...	...
$x_{NP,1,G}$	$x_{NP,2,G}$	...	$x_{NP,D,G}$	$F_{NP,G}$	$CR_{NP,G}$

Fig. 1. Self-adapting control parameters *F* and *CR* are encoded into the individual. The vector of each individual $\mathbf{x}_{i,G}$ is extended by the values of two parameters: $F_{i,G}$ and $CR_{i,G}$.

$$CR_{i,G+1} = \begin{cases} rand_3 & if\ rand_4 < t_2 \\ CR_{i,G}, & otherwise \end{cases} \tag{6}$$

They produce control parameters *F* and *CR* in a new vector. $rand_j, j \in \{1,2,3,4\}$ are uniform random values$\in[0, 1]$. t_1 and t_2 represent the probabilities of adjusting control parameters *F* and *CR*, respectively. The new *F* takes a value from [0.1, 1.0] in a random manner. The new *CR* takes a value from [0, 1]. $F_{i,G+1}$ and $CR_{i,G+1}$ are obtained before the mutation is performed. So they influence the mutation, crossover and selection operations of the new vector $\mathbf{x}_{i,G+1}$.

This self-adaptive DE algorithm tested on benchmark optimization problems. The results show that the algorithm, with self-adaptive control parameter settings, is better or at least comparable to the standard DE algorithm and FADE algorithm [3].

3 Self-adapting Differential Evolution Algorithm with Chaos Random

In section 2.2, the self-adaptive DE has two obvious defects:

(1) The adjusting of $F_{i,G}$ and $CR_{i,G}$ is determined by fixed probabilities t_1 and t_2. At generation *G*, suppose that $F_{i,G}$ and $CR_{i,G}$ are suitable for individual *i*, while $F_{j,G}$ and $CR_{j,G}$ are not suitable for individual *j*, but $F_{i,G}$ and $CR_{i,G}$ are adjusted with the same probability. The suitable and unsuitable $F_{i,G}$ and $CR_{i,G}$ should be distinguished and dealt differently.

(2) The new values of $F_{i,G}$ and $CR_{i,G}$ are determined by uniform random number. From Fig.2, it can be seen that uniform random numbers generated by computer programming are mainly within [0.3, 0.7], and can not cover points all over the value domain.

In this paper, the adjusting of $F_{i,G}$ and $CR_{i,G}$ associates with fitness of individuals and the new values are Chaos random numbers.

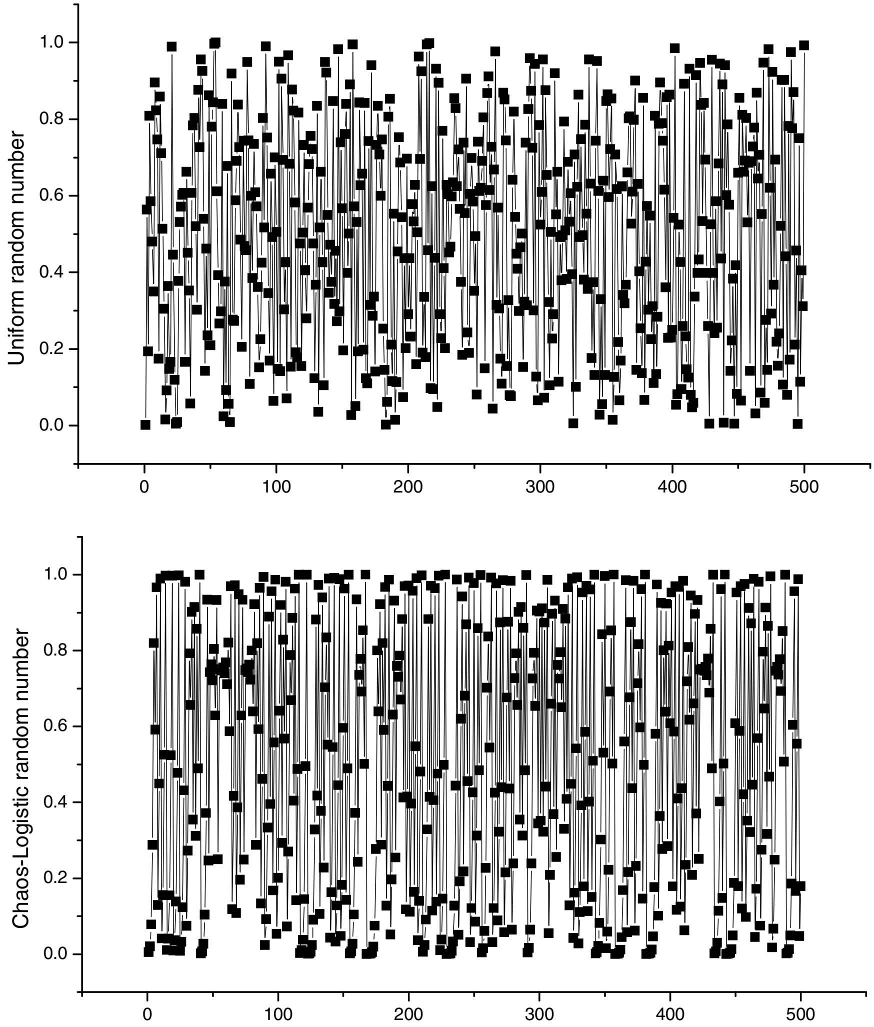


Fig. 2. Comparison between Chaos-Logistic and uniform random number sequence (500 numbers). Uniform random numbers generated by computer programming are mainly within [0.3, 0.7], and can not cover points all over the value domain. But Logistic random has a better performance.

3.1 Chaos Random—Logistic Function

Chaos is seemingly a kind of irregular motions, which has random-like behavior without any random factors in the deterministic non-linear system (randomness of the inner algorithm framework). The greatest characteristic of chaotic system is that it is very sensitive to the initial value, and the future behavior of system is unpredictable in long term.

In this paper, the chaos function is Logistic function [18], which is a classical chaos system:

$$r_{n+1} = \mu r_n \left(1 - r_n\right), n = 0,1,2,... \tag{7}$$

where μ is a parameter. It can generate a serial sequence when the first number $r_0 \in$ (0,1) is randomly generated and μ is specific. The Logistic appears different characteristic with different values of μ (see Fig. 3). When μ=4, Logistic system is in the Chaos state and $r_n \in (0,1)$ is chaos sequence which can search the points all over the solution space. And From Fig.2, it can be seen that Logistic random has a better performance than uniform random, considering the distribution of numbers over the whole random space.

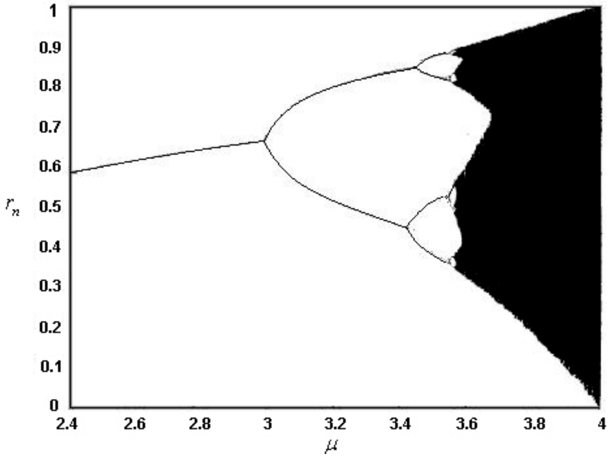


Fig. 3. How Logistic enters chaos. The Logistic appears different characteristic with different values of μ. When μ=4, Logistic system is in the chaos state.

3.2 Self-adapting Parameters—New Version of DE Algorithm

View the two obvious defects of self-adaptive DE in section 2.2, we proposed a new self-adapting DE algorithm with Chaos random (SaDEC), in which adjusting of $F_{i,G}$ and $CR_{i,G}$ associates with fitness of individuals and the new values are Chaos random numbers:

$$F_{i,G+1} = \begin{cases} F_l + LRand_1 & if\ rand_1 < 1/\left(t + \left|\Delta f_i\right|\right) \\ F_{i,G}, & otherwise \end{cases} \tag{8}$$

$$CR_{i,G+1} = \begin{cases} LRand_2 & if\ rand_2 < 1/\left(t+\left|\Delta f_i\right|\right) \\ CR_{i,G}, & otherwise \end{cases} \tag{9}$$

where *rand*$_j$, $j\in\{1,2\}$ are uniform random values$\in[0, 1]$, *LRand*$_j$, $j\in\{1,2\}$ are Logistic random numbers, $\Delta f_i = f\left(\mathbf{x}_{i,G}\right) - f\left(\mathbf{x}_{i,G-1}\right)$, and t is a fixed value avoiding divisor is zero.

If $F_{i,G}$ and $CR_{i,G}$ were suitable for individual i in the evolutionary process, the value of $\left|\Delta f_i\right|$ would be large, and the adaptation of $F_{i,G}$ and $CR_{i,G}$ would be smaller probability of occurrence than the case of unsuitable $F_{i,G}$ and $CR_{i,G}$ for individuals. The new values of $F_{i,G}$ and $CR_{i,G}$ produced by Chaos random could be suitable with larger probability than uniform number. Usually, F_l=0.1, t=10.

4 Experiments

4.1 Benchmark Functions

Twenty-three benchmark functions from [19] were used to test the performance of our DE algorithm to assure a fair comparison. The benchmark functions are given in Table 1. N denotes the dimensionality of the test problem, S denotes the ranges of the variables, and $f_{\min}$ is a function value of the global optimum. A more detailed description of each function is given in [19] and [20], where the functions were divided into three classes: functions with no local minima, many local minima, and a few local minima.

Function f_1-f_{13} are high-dimensional problems. Functions f_1-f_5 are unimodal. Function f_6 is the step function which has one minimum and is discontinuous. Function f_7 is a noisy quadratic function. Functions f_8-f_{13} are multimodal functions where the number of local minima increases exponentially with the problem dimension [19]. Function f_{14}-f_{23} are low-dimensional functions which have only a few local minima [19]. Yao *et al.* [19] described that, for multimodal functions, the final results are much more important since they reflect the algorithm's ability to escape from poor local optima and locate a good near-global optimum.

Table 1. Benchmark functions. N denotes the dimensionality of the test problem, S denotes the ranges of the variables, and $f_{\min}$ is a function value of the global optimum.

Test function	N	S	f_{min}
$f_1(x)=\sum_{i=1}^{N} x_i^2$	30	$[-100, 100]^N$	0
$f_2(x)=\sum_{i=1}^{N}\lvert x_i\rvert+\prod_{i=1}^{N}\lvert x_i\rvert$	30	$[-10, 10]^N$	0
$f_3(x)=\sum_{i=1}^{N}\left(\sum_{j=1}^{i} x_j\right)^2$	30	$[-100, 100]^N$	0
$f_4(x)=\max_i\left\{\lvert x_i\rvert, 1\le i\le n\right\}$	30	$[-100, 100]^N$	0
$f_5(x)=\sum_{i=1}^{N-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	$[-30, 30]^N$	0
$f_6(x)=\sum_{i=1}^{N}\left(\lfloor x_i+0.5\rfloor\right)^2$	30	$[-100, 100]^N$	0
$f_7(x)=\sum_{i=1}^{N} i x_i^4+random[0,1)$	30	$[-1.28, 1.28]^N$	0

Table 1. (*continued*)

$f_8(x)=\sum_{i=1}^{N}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	$[-500, 500]^N$	-12569.5
$f_9(x)=\sum_{i=1}^{N}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	30	$[-5.12, 5.12]^N$	0
$f_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N}x_i^2}\right)$ $-\exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right)+20+e$	30	$[-32, 32]^N$	0
$f_{11}(x)=\frac{1}{4000}\sum_{i=1}^{N}x_i^2-\prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	$[-600, 600]^N$	0
$f_{12}(x)=\frac{\pi}{N}\begin{Bmatrix}10\sin^2(\pi y_i)\\+\sum_{i=1}^{N-1}\{(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]\}+(y_N-1)^2\end{Bmatrix}$ $+\sum_{i=1}^{N}u(x_i,10,100,4)$	30	$[-50, 50]^N$	0
$f_{13}(x)=0.1\begin{Bmatrix}\sin^2(3\pi x_1)+\sum_{i=1}^{N-1}\{(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]\}\\+(x_N-1)^2[1+\sin^2(2\pi x_N)]\end{Bmatrix}$ $+\sum_{i=1}^{N}u(x_i,5,100,4)$	30	$[-50, 50]^N$	0
$f_{14}(x)=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right]^{-1}$	2	$[-65.536, 65.536]^N$	0.998004
$f_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	4	$[-5, 5]^N$	0.0003075
$f_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	$[-5, 5]^N$	-1.0316285
$f_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2$ $+10\left(1-\frac{1}{8\pi}\right)\cos(x_1)+10$	2	[-5,10]×[0, 15]	0.398
$f_{18}(x)=\left[1+(x_1+x_2+1)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]$ $\times\left[30+(2x_1-3x_2)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$	2	$[-2, 2]^N$	3
$f_{19}(x)=-\sum_{i=1}^{4}\left\{c_i\exp\left[-\sum_{j=1}^{4}a_{ij}\left(x_j-p_{ij}\right)^2\right]\right\}$	4	$[0, 1]^N$	-3.86
$f_{20}(x)=-\sum_{i=1}^{4}\left\{c_i\exp\left[-\sum_{j=1}^{6}a_{ij}\left(x_j-p_{ij}\right)^2\right]\right\}$	6	$[0, 1]^N$	-3.32
$f_{21}(x)=-\sum_{i=1}^{5}\left[(\boldsymbol{x}-\boldsymbol{a}_i)(\boldsymbol{x}-\boldsymbol{a}_i)^{\mathrm{T}}+c_i\right]^{-1}$	4	$[0, 10]^N$	-10.1532
$f_{22}(x)=-\sum_{i=1}^{7}\left[(\boldsymbol{x}-\boldsymbol{a}_i)(\boldsymbol{x}-\boldsymbol{a}_i)^{\mathrm{T}}+c_i\right]^{-1}$	4	$[0, 10]^N$	-10.4029
$f_{23}(x)=-\sum_{i=1}^{10}\left[(\boldsymbol{x}-\boldsymbol{a}_i)(\boldsymbol{x}-\boldsymbol{a}_i)^{\mathrm{T}}+c_i\right]^{-1}$	4	$[0, 10]^N$	-10.5364

4.2 Experimental Results

We applied our new version of self-adaptive DE (SaDEC), existing self-adaptive DE (SaDE) in [12] and (original) DE in [1] to a set of benchmark optimization problems. All the three algorithm use the DE scheme which can be classified using notation as *DE/rand/1/bin* strategy.

The initial population was generated uniformly at random in the range, as specified in Table 1.

Throughout this paper, the parameters of algorithms are set as the following:

(1) Population size: *NP*=50;
(2) Original DE algorithm: *F*=0.5, *CR*=0.9. Our decision for using those values is based on proposed values from literature [1, 3, 9, 21];
(3) The values of control parameters of SaDE in [12]: $F_{i,0}$=0.5, $CR_{i,0}$=0.9, t_1=0.1, t_2=0.1, F_l=0.1, F_u=0.9;
(4) The values of control parameters of SaDEC: $F_{i,0}$=0.5, $CR_{i,0}$=0.9, F_l=0.1, *t*=10;
(5) The stop condition of the three algorithms is the same function evaluating number for each function (see Table 2).

Table 2. Experimental results, averaged over 50 independent runs, of SaDEC, SaDE, DE algorithms. "Mean Best" indicates average of minimum obtained and "Std Dev" stands for standard deviation.

F	*Func. Eval.*	*SaDEC* *Mean Best (Std Dev)*	*SaDE* *Mean Best (Std Dev)*	*DE* *Mean Best (Std Dev)*
f_1	100000	2.8×10^{-23} (4.7×10^{-23})	2.0×10^{-24} (1.6×10^{-24})	2.8×10^{-26} (1.2×10^{-25})
f_2	100000	2.6×10^{-14} (1.6×10^{-14})	8.8×10^{-15} (4.5×10^{-15})	1.5×10^{-14} (1.1×10^{-14})
f_3	500000	1.5×10^{-14} (2.6×10^{-14})	6.7×10^{-10} (1.3×10^{-9})	1.2×10^{-22} (6.7×10^{-22})
f_4	500000	1.1×10^{-12} (6.4×10^{-12})	1.9×10^{-16} (9.7×10^{-16})	11.0 (5.3)
f_5	500000	1.6×10^{-1} (7.8×10^{-1})	1.1×10^{-15} (6.7×10^{-15})	16.5 (7.2)
f_6	17000	**0 (0)**	0 (0)	0.26 (0.63)
f_7	500000	1.7×10^{-3} (7.0×10^{-4})	1.5×10^{-3} (4.0×10^{-4})	1.4×10^{-3} (4.2×10^{-4})
f_8	75000	**-12560.0 (32.5)**	-12475.1 (187.9)	-7091.8 (1420.2)
f_9	140000	**0 (0)**	6.2 (4.3)	25.0 (19.9)
f_{10}	500000	**3.9×10^{-15} (3.2×10^{-30})**	3.9×10^{-15} (3.2×10^{-30})	3.9×10^{-15} (3.2×10^{-30})
f_{11}	500000	6.4×10^{-4} (2.2×10^{-3})	0 (0)	2.8×10^{-3} (6.9×10^{-3})
f_{12}	500000	**3.0×10^{-17} (1.9×10^{-32})**	3.0×10^{-17} (1.9×10^{-32})	0.014513578 (6.3×10^{-2})
f_{13}	500000	**2.9×10^{-17} (2.5×10^{-32})**	2.9×10^{-17} (2.5×10^{-32})	0.0734875 (5.1×10^{-1})
f_{14}	3000	**0.9980038 (5.6×10^{-16})**	0.9980038 (5.6×10^{-16})	0.998180304 (1.2×10^{-3})
f_{15}	22000	**0.000307486 (3.8×10^{-19})**	0.00030748601 (4.6×10^{-11})	0.000307558728 (3.4×10^{-7})
f_{16}	3000	**-1.031628 (4.5×10^{-16})**	-1.0316279 (4.2×10^{-7})	-1.031628 (4.5×10^{-16})

Table 2. (*continued*)

f_{17}	6000	**0.3978874 (2.2×10^{-16})**	0.397887414 (7.6×10^{-8})	0.3978874 (2.2×10^{-16})
f_{18}	3000	**3 (0)**	3.00000124 (5.9×10^{-6})	3 (0)
f_{19}	2500	**-3.862782 (1.8×10^{-15})**	-3.86278122 (1.9×10^{-6})	-3.862782 (1.8×10^{-15})
f_{20}	13000	-3.31961714 (0.017)	-3.321995 (2.2×10^{-15})	-3.30772784 (0.04)
f_{21}	8500	**-10.1532 (1.3×10^{-14})**	-10.1531992 (4.4×10^{-6})	-9.90274464 (1.3)
f_{22}	7000	**-10.40294 (0)**	-10.4029342 (2.4×10^{-5})	-10.40294 (0)
f_{23}	10000	-10.42919474 (0.76)	-10.40294 (0)	-10.53641 (5.4×10^{-15})

Table 2 shows the results obtained by SaDEC, SaDE and DE algorithms. Relative to SaDE and DE, the better or equal results got by SaDEC for each function are in boldface.

For the unimodal functions f_1-f_5, expect f_4 and f_5, DE is little better than SaDE and SaDEC. For functions f_1-f_4, SaDE and SaDEC have almost same performance. The reason of worse performance of SaDEC than SaDE, for function f_5, is that there are two runs in which the individuals of SaDEC are almost the same and can not get better individual.

For functions f_6 and f_7, SaDEC and SaDE can get better results than DE.

For the multimodal functions with many local minima, i.e., f_8-f_{13}, it is clear that the best results are obtained by SaDEC, except f_{11}. Interestingly, for function f_8, SaDEC and SaDE can get optimal solutions most runs with 50 runs, and in a few runs, SaDEC and SaDE fall into local optimum, which lead to they have poor values of "Mean Best" and "Std Dev". For function f_8, DE can not get any optimal solution with all 50 runs, and falls into local optimal. This indicates that DE, SaDE and SaDEC do not have good stability.

For the functions f_{14}-f_{23} with only a few local minima, all the three algorithms were able to find optimal solutions.

5 Conclusions

Choosing the proper control parameters for DE is quite a difficult task because the best settings for the control parameters can be different for different functions. In this paper, the proposed self-adaptive method is an attempt to determine the values of control parameters F and CR. In this method, the adjusting of $F_{i,G}$ and $CR_{i,G}$ associates with fitness of individuals and the new values are Chaos random numbers.

Our self-adaptive DE algorithm has been implemented and tested on benchmark optimization problems taken from literature. The results show that our algorithm can get better solutions than other algorithms for multimodal functions, because the Chaos random is more likely to cover points all over the value domain.

References

1. Storn, R., Price, K.: Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optimiz. 11, 341–359 (1997)
2. Liu, J., Lampinen, J.: On setting the control parameter of the differential evolution method. In: Proc. 8th Int. Conf. Soft Computing (MENDEL 2002), pp. 11–18 (2002)
3. Liu, J., Lampinen, J.: A fuzzy adaptive differential evolution algorithm. Soft Comput. Fusion Found. Methodol. Appl. 9(6), 448–462 (2005)
4. Teo, J.: Exploring dynamic self-adaptive populations in differential evolution. Soft Comput—Fusion Found. Methodol. Appl. 10(8), 673–686 (2006)
5. Maruo, M.H., Lopes, H.S., Delgado, M.R.: Self-adapting evolutionary parameters: Encoding aspects for combinatorial optimization problems. In: Raidl, G.R., Gottlieb, J. (eds.) EvoCOP 2005. LNCS, vol. 3448, pp. 155–166. Springer, Heidelberg (2005)
6. Bäck, T.: Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, New York (1996)
7. Bäck, T.: Adaptive business intelligence based on evolution strategies: some application examples of self-adaptive software. Inf. Sci. 148, 113–121 (2002)
8. Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.): Handbook of evolutionary computation. Institute of Physics Publishing and Oxford University Press, Oxford (1997)
9. Ali, M.M., Törn, A.: Population set-based global optimization algorithms: some modifications and numerical studies. Comput. Oper. Res. 31(10), 1703–1725 (2004)
10. Jiao, Y.-C., Dang, C., Leung, Y., Hao, Y.: A modification to the new version of the price's algorithm for continuous global optimization problems. J. Global Optim. 36(4), 609–626 (2006)
11. Brest, J., Maučec, M.S.: Population size reduction for the differential evolution algorithm. App. Intell. (29), 228–247 (2008)
12. Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans. Evol. Comput. 10(6), 646–657 (2006)
13. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: The 2005 IEEE congress on evolutionary computation, CEC 2005, pp. 1785–1791 (2005)
14. Brest, J., Bošković, B., Greiner, S., Žumer, V., Sepesy Maučec, M.: Performance comparison of self-adaptive and adaptive differential evolution algorithms. Soft Comput—Fusion Found. Methodol. Appl. 11(7), 617–629 (2007)
15. Mezura-Montes, E., Velázquez-Reyes, J., Coello Coello, C.A.: A comparative study of differential evolution variants for global optimization. In: GECCO, pp. 485–492 (2006)
16. Tvrdík, J.: Competitive differential evolution. In: MENDEL 2006, 12th International Conference on Soft Computing, pp. 7–12 (2006)
17. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential evolution: A practical approach to global optimization. Springer, New York (2005)
18. Heinx, O.P.: Chaos and Fractals. Springer, New York (1992)
19. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Trans. Evol. Comput. 3(2), 82–102 (1999)
20. Lee, C.Y., Yao, X.: Evolutionary programming using mutations based on the Lévy probability distribution. IEEE Trans. Evol. Comput. 8(1), 1–13 (2004)
21. Vesterstroem, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: Proc. IEEE Congr. Evolutionary Computation, Portland, pp. 1980–1987 (2004)

A Concurrent-Hybrid Evolutionary Algorithms with Multi-child Differential Evolution and Guotao Algorithm Based on Cultural Algorithm Framework

Xia Li[1], Kunqi Liu[1,2], Lixiao Ma[1], and Huanzhe Li[1]

[1] Department of Computer Science, Shijiazhuang University of Economics,
Shijiazhuang, 050031
[2] School of Computer, China University of Geosciences, Wuhan, 430074
lixiaruihua@163.com

Abstract. This paper proposes a multi-child differential evolutionary algorithm(MCDE), and forms a concurrent-hybrid evolutionary algorithm by integrating the MCDE algorithm and Guotao algorithm based on variable searching subspace(VSSGT) into the culture algorithm framework. Numerical experiment results indicate that the performance of the proposed algorithm is better than that of MCDE, Differential Evolution algorithm(DE) and VSSGT, and better than that of the DE with double trial vectors based on Boltzmann mechanism.

Keywords: Differential evolutionary algorithm, Multi-child Differential evolutionary algorithm, Guotao algorithm based on variable searching subspace, Cultural algorithm, Concurrent-Hybrid evolutionary algorithm.

1 Introduction

How to design the efficient evolutionary algorithm is one of the basic questions in the field of the evolutionary computation[1][2].There has a basic way to resolve this question, by integrating the existing evolutionary algorithms into other algorithms[3][4]. The earliest findings of this aspect was traced back to the combination of the Genetic Algorithm(GA) and the local search[4], and now it has developed into Memetic algorithm[5], this work has greatly promoted the development of evolutionary algorithm. In fact, we can obtain a new Hybrid evolutionary algorithm by combining evolutionary algorithm with another with local search ability, such as the combination of the Evolutionary Algorithm and Simulated Annealing Algorithm[6][7], the Evolutionary Algorithm and the GA[6][8][9], and so on. These efforts show that the integration of two kinds of evolutionary algorithms has become a new way to design new evolutionary algorithm.

The Cultural Algorithm[10] was invented by Professor Robert G. Reynolds, University of Michigan, in 1994. It is a global optimization algorithm by simulating the evolutionary process of human culture. In fact, it is a framework of Concurrent-Hybrid evolutionary algorithm integrating two evolutionary algorithms. In 2005, Ai Jing bo and Teng Hongfei proposed the Cultural based PSO Algorithm by integrating

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 123–133, 2010.

the PSO and the Genetic Algorithm into this framework, this algorithm is good at solving the layout question[11].

DE[12] and Guotao Algorithm[13][14] are two well-known algorithm in field of evolutionary computation, and also hot algorithm on evolutionary algorithm at home and broad. For solving function optimization problems, Guotao convergences at a slow rate (especially for solving high dimensional function optimization problems), and DE can search fast, but is easy to fall into local optimum[16]. To make up for deficiencies in the two algorithms, inspired by the idea of hybrid algorithm, this paper proposes MCDE and Concurrent-Hybrid Evolutionary Algorithms with MCDE and VSSGT based on Cultural Algorithm framework (CHEA). Numerical experimental results show that the performance of CHEA is better than MCDE and VSSGT, and better than boDE in [15].

The content of this paper is as follows: Section II discusses MCDE and VSSGT, Section III discusses CHEA with MCDE and VSSGT based on Cultural Algorithm Framework, Section IV gives the results of simulating experiments and the analysis, the last section summarizes the paper and points out the ideas for future work.

2 The Multi-child Differential Evolution Algorithm and the Guotao Algorithm Based on Variable Searching Subspace

2.1 The Multi-child Differential Evolution Algorithm

The DE is proposed by Rainer Storn and Kenneth Price in 1995[17], it is a kind of real-coded evolutionary algorithm. Its basic idea is that using the difference of the individuals of the current population to obtain the mutation population, then using this population and its parent population to obtain the new crossover population. Afterwards, the next generation population is gotten by competing between the new population and its parent population. To further improve the speed of DE, this paper proposes MCDE. Its basic idea is that generating a mutation sub-population, and selecting an individual with optimal fitness value from the mutation sub-population to participate in the crossover operation, and also generating a crossover sub-population, and selecting an individual of optimal fitness value to participate in the selection operation. The size of mutation and crossover sub-population is set according to the different problems (usually between 1 and 5).

The MCDE using the DE/rand/1/bin scheme is described as follows:

(1) Initialization

Set the parameters N (the size of the population), Pc(Crosser probability), StepMax(the max evolutionary generation), EvoNum(the size of mutation sub-population), CroNum(the size of crossover sub-population). Select the initial population randomly: $X(t)=(X_1(t), X_2(t),\ldots, X_N(t))$□ $t=0$, each $X_i(0)$ has n-dimensional vector, and find the individual of optimal fitness value $X_{best}(0)$.

(2) Mutation

For each individual of the t^{th} population, obtain the mutation sub-population SE_k by mutate operation, $1 \le k \le EvoNum$. The generating method of the each individual of the SEk is as follows: Select randomly three individuals $X_a(t)$, $X_b(t)$, $X_c(t)$, $a \ne b \ne c$,

SE_k= X_a(t)+F(X_b(t)- X_c(t)), F is a real and constant factor which controls the amplification of the differential variation (X_b(t)- X_c(t)). Find fitness value of the each individual, and chose the individual SEbest with the optimal fitness value, set V_i(t)= SE_{best}.

(3) Crossover

For each individual of the tth population, obtain the Crossover sub-population SC_k by crossover operation, 1≤k≤CroNum. The generating method of the each individual of the SC_k is as follows: Generate a random number r, r∈[0,1], if r>Pc, then SC_k [j]= V_i(t)[j], else SC_k [j]= X_i(t)[j]. Find fitness value of each individual, and chose the individual SC_{best} with the optimal fitness value, set H_i(t)= SC_{best}.

(4) Selection

Select the individual with the optimal fitness value from the individual H_i(t) and the individual X_i(t) as the individual X_i(t+1).

(5) Judge the ending conditions

Find the individual X_{best}(t+1) with the optimal fitness value from the (t+1)-th population, if meet the precision requirement or t=StepMax, then the algorithm is over, output the X_{best}(t+1) and its optimal fitness value, else set t=t+1, and turn to the second step.

2.2 Guotao Algorithm Based on Variable Searching Subspace

Guotao Algorithm was proposed by Guo tao in 1999 which combined the polycross method and population climbing method[18]. It generates the new individual by extending a few individual of the subspace. And it goes out of the bad individual, and keeps the diverse of the population. The speed of convergence is slow.

The size of the subspace is invariable in standard Guotao algorithm. It searches in the same dimension subspace, whether the result is good or bad. The size of searching space is wide and needs a lot of calculation, when the population closes to the optimal result. Therefore, this paper uses VSSGT algorithm, which shrinks the size of the subspace according to quality of the result.

VSSGT algorithm is described as follows:

(1) Initialization

Set the parameters N(the size of the population), M(the size of the subspace). Select the initial population randomly: X(t)=(X_1(t), X_2(t),…, X_N(t)) t=0, where X(0) is the initial population, X_i(0) is n-deimentional vector. Find the individual of optimal fitness value X_{best}(0) and the individual of worst fitness value X_{worst}(0).

(2) Search subspace

Select m individuals (X1(t),X2(t),…,Xm(t)) from X(t) randomly, and form the subspace $V(t)=\left\{X_i(t) \mid X_i(t)=\sum_{i=1}^{m} a_i X_i(t)[j], 1\le j\le n\right\}$, where $\sum_{i=1}^{m} a_i = 1$, $-0.5 \le a_i \le 1.5$.

(3) Selection

Select an individual $\overline{X}$ from subspace V randomly, if the $\overline{X}$ is better than X_{worst}(t), then X_{worst}(t+1)= $\overline{X}$.

(4) Jude the end conditions

If the $X_{worst}(t+1)$ is better than $X_{best}(t+1)$, then the algorithm is end, else if $|X_{best}(t+1)-X_{worst}(t+1)|<\eta$ (in this paper $\eta=10^{-6}$) and M>3, then M=M−1, and turn to the second step.

3 Concurrent Hybrid Evolution Algorithm

3.1 The Framework of the Cultural Algorithm

Cultural Algorithm is an evolutionary algorithm simulating the evolutionary process of human culture[19]. Fascination of the algorithm is its double evolutionary mechanism. Cultural Algorithm has a population space and a belief space. Population space is mainly responsible for the evolutionary operation, and calculates the individual, and provides the good individuals to the belief space. Belief space updates own knowledge sources. Belief space accepts the good individuals from population space, and influences the population space by influence operation. Fig. 1 shows the framework of the Cultural Algorithm.

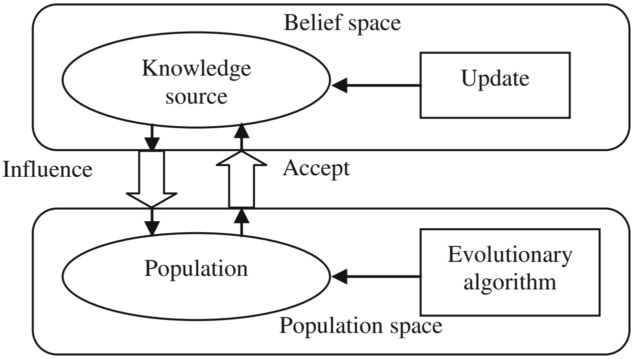


Fig. 1. The framework of the Cultural Algorithm

Under the framework of Cultural Algorithm, the population space and the belief space has own population, and uses different evolutionary algorithm to evolve own population. In the process of evolution, the two space exchange the information regularly, and influence each other. And then, the framework of Cultural Algorithm becomes a framework of Concurrent-Hybrid evolutionary algorithm integrating two evolutionary algorithms. It is also a Co-evolution Algorithm that integrating two evolutionary algorithm.

3.2 Concurrent Hybrid Evolution Algorithm with Multi-child Differential Evolution Algorithm and Guotao Algorithm Based on Variable Searching Subspace Based on Cultural Algorithm Framework

In the above framework of Concurrent-Hybrid evolutionary algorithm, if the population space and the belief space use two different evolutionary algorithms that can make up each other's deficiencies, then the Concurrent-Hybrid evolutionary algorithm can co-evolve. Guotao algorithm convergences at a slow rate (especially for solving high dimensional function optimization problems), but it can keep the diversity of the population. DE can search fast, but it is easy to fall into local optimum. To make up for deficiencies in the two algorithms, this paper proposes the CHEA that combining DE and Guotao Algorithm into the framework of Cultural Algorithm. In order to improve the performance of this CHEA, the population space uses MCDE that proposed in this paper, the belief space uses VSSGT.

The accept operation of the framework has fixed ratio acceptance, dynamic acceptance and fuzzy acceptance□ and so on. This paper uses the fixed ratio accept operation. MCDE in population space selects the best individual to displace the worst individual per AcceptStep generations (AcceptStep=10).

The influence operation has different roles in different step. In the early, the population evolves fast in own space, and now doesn't need big influence by belief space. In the late, the diversity of the population space is bad. And it needs the influence by belief space to extend searching space, and improve the ability of global searching. In this paper, some good individuals in belief space displace the same number bad individuals (the number is 50% in this paper) per InfluenceStep generations. The InfluenceStep can calculate by following formula.

$$InfluenceStep = BaseNum + \frac{MaxStep - CurrentStep}{MaxStep} * DevNum \tag{1}$$

The MaxStep is the max evolutionary generation (MaxStep=100000 in this paper). The CurrentStep is the current evolutionary generation, BaseNum and DevNum are constant, BaseNum=15 and DevNum=100. So, the less influence on the population space in the early, the more influence in the late.

The CHEA proposed in this paper is described as follows:

(1) Initialization

Set the parameters N (the size of the population), the max generations MaxStep=100000, the current generation CurrentStep=1. Initial the initial population randomly and the relational parameters.

(2) Evolution

The population space uses MCDE to evolve the population. The belief space uses VSSGT to evolve the population.

(3) Accept operation

If CurrentStep%AcceptStep=0, then the best individual of the population space displaces the worst individual of the belief space, else turn to the 4th step.

(4) Influence operation

If CurrentStep%InfluenceStep=0, then the 50% good individual of the belief space displace the same number bad individual of the population space, else turn to the 5th step.

(5) Jude the end conditions

If the optimal fitness value of the population space meets the precision requirement or the max generation equals StepMax, then the algorithm is end, and output the best individual and its optimal fitness value, else set CurrentStep=CurrentStep+1, and turn to the second step.

4 Experiment and Result Analysis

4.1 Test Functions

This paper chooses 16 test functions from [15]. These test functions concludes the unimodal and multimodal function, functions with correlated and uncorrelated variables, noisy function and function with plateaus. The dimension of the function is from 2 to 30. Table 1 lists the test functions including their range and the optimal value.

Table 1. Test Functions

Test Functions	**Dim**	**Range**	**Optimal value**
$f_1(\vec{x}) = \sum_{i=1}^{n} x_i^2$	30	$-5.12 \le x_i \le 5.12$	$f_1(\vec{0}) = 0$
$f_2(\vec{x}) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} x_i$	30	$-10 \le x_i \le 10$	$f_2(\vec{0}) = 0$
$f_3(\vec{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_i \right)$	30	$-100 \le x_i \le 100$	$f_3(\vec{0}) = 0$
$f_4(\vec{x}) = \max \lvert x_i \rvert, 1 \le i \le n$	30	$-100 \le x_i \le 100$	$f_4(\vec{0}) = 0$
$f_5(\vec{x}) = \sum_{i=1}^{n} (100 \cdot (x_{i+1} - (x_i)^2)^2 + (x_i - 1)^2)$	30	$-30 \le x_i \le 30$	$f_5(\vec{1}) = 0$
$f_6(\vec{x}) = \sum_{i=1}^{n} \left(\left\lfloor x_i + \frac{1}{2} \right\rfloor \right)^2$	30	$-100 \le x_i \le 100$	$f_6(\vec{p}) = 0,$ $-\frac{1}{2} \le p < \frac{1}{2}$
$f_7(\vec{x}) = \left(\sum_{i=1}^{n} (i+1) \cdot x_i^4 \right) + rand\ [0,1)$	30	$-1.28 \le x_i \le 1.28$	$f_7(\vec{0}) = 0$
$f_8(\vec{x}) = \sum_{i=1}^{n} -x_i \cdot \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	$-500 \le x_i \le 500$	$f_8(42\vec{0}.97) =$ -12569.5
$f_9(\vec{x}) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30	$-5.12 \le x_i \le 5.12$	$f_9(\vec{0}) = 0$
$f_{10}(\vec{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	$-32 \le x_i \le 32$	$f_{10}(\vec{0}) = 0$
$f_{11}(\vec{x}) = \frac{1}{4000}\left(\sum_{i=1}^{n} x_i^2\right) + \left(\prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i+1}}\right)\right) + 1$	30	$-600 \le x_i \le 600$	$f_{11}(\vec{0}) = 0$
$f_{12}(\vec{x}) = \left(\frac{1}{500} + \sum_{j=1}^{25} \left(j + 1 + \sum_{i=1}^{2} (x_i - a_{ij})^6 \right)^{-1} \right)^{-1}$	2	$-65.54 \le x_i \le 65.54$	$f_{12}(-3\vec{1}.95)$ $= 0.998$
$f_{13}(\vec{x}) = \sum_{i=1}^{11} \left(a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right)^2$	4	$-5 \le x_i \le 5$	$f_{13}(0.19, 0.19,$ $0.12, 0.14)$ $= 0.0003075$
$f_{14}(\vec{x}) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x$	2	$-5 \le x_i \le 5$	$f_{14}(-0.09, 0.71)$ $= -1.0316$
$f_{15}(\vec{x}) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	2	$-5 \le x_i \le 15$	$f_{15}(9.42, 2.47)$ $= 0.398$
$f_{16}(\vec{x}) = \{1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)$ $\{30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27.$	2	$-2 \le x_i \le 2$	$f_{16}(1.49e-5, 1)$ $= 3$

4.2 Experimental Environment and Parameters Settings

(1) Experimental environment settings

CPU : Intel P4, 3GHz dual-core, Memory : 512M, OS : Microsoft Windows XP, Program Language : Microsoft Visual C++ 6.0。

(2) Parameters of algorithm settings

The size of population is 100 in MCDE, DE, and VSSGT, the max generation is 100000, F=0.5 and Pc=0.9 in MCDE and DE. Set EvoNum=3, CroNum=3 in MCDE. Set the size of subspace M=10, $-0.5 \le a_i \le 1.5$.

The parameter of CHEA is set the same as MCDE and VSSGT. But the size of mutation subspace and crossover subspace is set according to the different problems, and usually value between 1 and 5. The size of subspace in VSSGT is set between 3 and 10. And, the size of population in CHEA is 100 in most functions, but the size is 400 in 3 functions in order to improve the quality of the result.

4.3 Comparison of the Experimental Results

This paper calculates and compares the results of the above test functions by CHEA, MCDE, DE, VSSGT and paper [15].

In order to test the stability of the algorithm, every algorithm runs 30 times. Table 2 lists the experimental results of the 16 test functions by using four different algorithms. The content concludes test functions, the generation obtaining the best result, the average optimal fitness value and the standard unbiased variance.

CHEA convergences with faster rate, and keep the diversity of population. The best result is found by CHEA. The result of MCDE is better too. DE finds the best result of some functions. MCDE and DE convergences also fast, but doesn't convergence for some function in the late. VSSGT is slower. And the result is worse. In compare with [15], CHEA proposed in this paper improves in the speed of convergence and the quality of result.

For test functions f_1, $f_{2,}$ f_3, CHEA finds the best solution 0. The convergence speed of CHEA is slower than the algorithm in [15], but the accuracy is increased. MCDE and DE have a better result. But the result of VSSGT is worse.

For test functions f_4, CHEA finds the best result 0. The accuracy of result is same with in [15], but the speed of convergence is slower. MCDE, DE and VSSGT have worse results.

Test functions f_5, f_8—f_{11} are 5 multimodal functions. The f_5 is Rosenbrock function. Rosenbrock function is a classical and complex optimal problem, it is hard to find its global best result. The f_8 is Schwefel function. Its local best points are all far away from its global best result. The f_9 is Rastrigin function, has lots of local optimal point. The f_{10} is Ackley function, and have many local optimal points, but finds the global best result easily in 5 functions. The f_{11} is Griewank function. It is a correlated variables function, and hard to find the best result. The 5 functions are mainly used to verify the algorithm's population diversity. The experimental results show that CHEA finds the best result of f_5, f_8 and f_{11}, and the speed of convergence is faster than the [15]. MCDE is better for f_{11} and f_5, but is worse for f_8. The result of DE and VSSGT is very bad for f_5, f_8 and f_{11}. For f_9, f_{10}, the result's quality of CHEA is better than [15], but the generations is more than [15]. MCDE is worse for f_9. DE and VSSGT don't find the best result. For f_{10}, MCDE and DE are worse, but VSSGT don't find the best result.

Table 2. Results for four Algorithms on Test Functions

Test Function	Generation	CHEA	MCDE	DE	VSSGT
		Mean Best (Std-dev)	Mean Best (Std-dev)	Mean Best (Std-dev)	Mean Best (Std-dev)
f_1	8000	0 (0)	2.75E-206 (2.77E-206)	4.09E-91 (4.34E-91)	1.59E+03 (3.53E+02)
f_2	15000	0 (0)	1.53E-193 (1.30E-193)	5.43E-85 (4.24E-85)	1.69E+01 (2.01E+00)
f_3	7500	0 (0)	1.84E-191 (2.30E-191)	5.28E-83 (6.25E-83)	3.67E+05 (9.53E+04)
f_4	30000	0 (0)	4.25E-01 (7.74E-01)	4.59E-01 (6.12E-01)	1.34E+01 (2.15E+00)
f_5	3000	0 (0)	3.86E-28 (4.60E-28)	3.34E-01 (2.24E-01)	1.01E+02 (1.92E+01)
f_6	200	0 (0)	1.33E-01 (2.24E-01)	2.98E+02 (6.91E+01)	1.62E+04 (3.07E+03)
f_7	5000	1.19E-03 (4.00E-04)	3.27E-02 (5.22E-03)	3.59E-02 (7.74E-03)	1.85E-02 (6.67E-03)
f_8	4000	-1.26E+04 (5.34E-12)	-5.99E+03 (1.64E+02)	-7.51E+03 (1.10E+03)	-5.35E+03 (3.32E+02)
f_9	6000	1.99E-01 (3.21E-01)	5.62E+00 (1.68E+00)	1.57E+02 (7.06E+00)	8.83E+01 (8.62E+00)
f_{10}	3000	2.96E-15 (1.16E-30)	5.05E-12 (1.51E-12)	1.05E-04 (2.53E-05)	9.34E+00 (6.15E-01)
f_{11}	1000	0 (0)	0 (0)	3.33E-08 (6.24E-08)	1.91E+01 (3.97E+00)
f_{12}	100	9.98E-01 (4.30E-16)	9.98E-01 (4.30E-16)	9.98E-01 (4.30E-16)	6.50E+00 (2.98E+00)
f_{13}	1000	3.07E-04 (1.59E-19)	8.15E-04 (2.62E-04)	8.70E-04 (2.15E-04)	7.88E-04 (1.01E-04)
f_{14}	100	-1.03E+00 (4.30E-16)	-1.03E+00 (4.30E-16)	-1.03E+00 (4.30E-16)	-7.65E-01 (1.89E-01)
f_{15}	100	3.98E-01 (5.37E-17)	3.98E-01 (5.37E-17)	3.98E-01 (5.37E-17)	7.19E-01 (2.22E-01)
f_{16}	50	3 (0)	3 (0)	3 (0)	1.50E+01 (6.52E+00)

Test functions f_6 is a function with plateaus. CHEA can find the best solution at 200 generation, and CHEA is very steady. Its speed of convergence is faster than [15]. MCDE is worse. DE and VSSGT don't find the best result.

Test functions f_6 is a function with noisy. CHEA can find the best result at 5000 generation. Its speed of convergence is slower, but the quality of the result is better than [15]. MCDE, DE and VSSGT are worse for f_6.

Test functions f_{12}, f_{15} and f_{16} are simple unimodal functions. The dimension is lower. CHEA finds the best result for all test functions. CHEA is faster and more steady than [15] for f_{12}, f_{15} and f_{16}. The best result is found by MCDE and DE also, but the result of VSSGT is worse.

CHEA convergences the global best point for f_{13}. But the result of MCDE, DE and VSSGT is worse.

Test functions f_{14} is a camel back function with six peaks. CHEA, MCDE and DE can find the best result at 100 generation. The results are same as [15]. But the result of VSSGT is worse.

The performance of CHEA is better. Take the test function f_1, f_7, f_{10} and f_{15} for instance, the following gives the best fitness curves figures of 4 algorithms in Figure 2, Figure 3, Figure 4 and Figure 5.

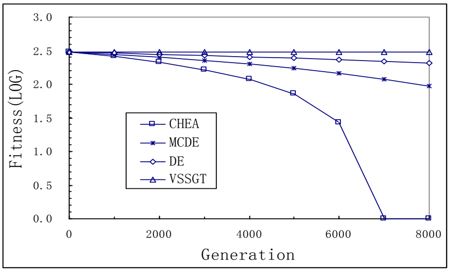


Fig. 2. Best fitness curves of the f_1

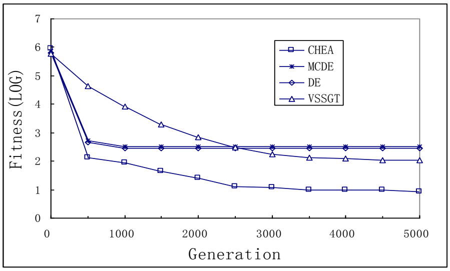


Fig. 3. Best fitness curves of the f_7

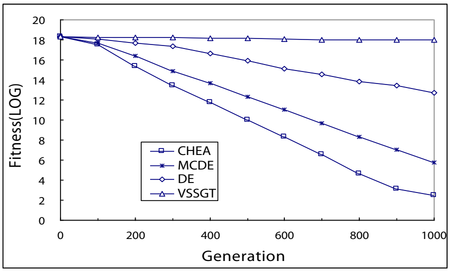


Fig. 4. Best fitness curves of the f_{10}

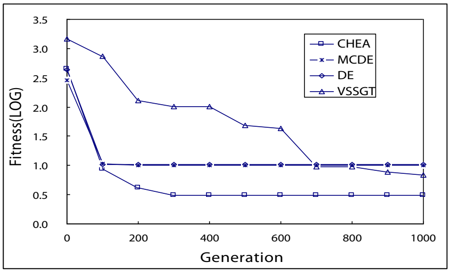


Fig. 5. Best fitness curves of the f_{15}

For f_1, f_7, f_{10} and f_{15}, the speed of convergence in CHEA is faster than MCDE, DE and VSSGT. For f_1 and f_7, the speed of convergence in MCDE is also faster than DE and VSSGT. For f_7 and f_{15}, the speed of convergence in MCDE and DE are fast, but algorithms are almost no convergence in the late. So, the result's quality is worse than VSSGT. But CHEA can overcome the shortcomings of MCDE, in the late, and improves the diversity of population by VSSGT. Therefore, CHEA can guarantee the convergence speed, but also ensure the diversity of population.

5 Conclusion

This paper proposes a Concurrent-Hybrid Evolutionary Algorithm by combining the MCDE and VSSGT algorithm into the Cultural Algorithm. Numerical experimental results indicate that the performance of CHEA is better than MCDE, DE, VSSGT and boDE in [15]. In the future, we will mainly focus on the following.

(1) We will do some works on theory researches on CHEA algorithm.

(2) We will use CHEA to solve the complex optimation problems, such as optimal problem with constraints, multi-objective optimal problem and Dynamic Optimal problem.

Acknowledgments. This work is supported by the National Natural Science Foundation of China(Grant No. 60772196) and the Special Fund for Basic Scientific Research of Central Colleges (No. CUG090109). The authors sincerely thank the teachers in evolutionary computation seminar of Shijiazhuang University of Economics, they give many constructive comment to our work.

References

1. Liu, K.-q., Kang, L.-s., Zhao, Z.-z.: The Brief Report of Research on Cognizing the subarea of Evolutionary Computation (I). Computer Science 36(7) (2009)
2. Liu, K.-q., Kang, L.-s., Zhao, Z.-z.: The Brief Report of Research on Cognizing the subarea of Evolutionary Computation (II). Computer Science 36(8) (2009)
3. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. California Institute of Technology, Pasadena, California, USA, Tech. Rep. Caltech Concurrent Computation Program, Report 826 (1989)
4. John, J.: Grefenstette, Lamarckian learning in multi-agent environments. In: Proc. Fourth Intl. Conf. of Genetic Algorithms, pp. 303–310. Morgan Kaufmann, San Mateo (1991)
5. Krasnogor, N., Jim Smith, A.: Tutorial for Competent Memetic Algorithms:Model, Taxonomy and Design Issues. IEEE Transactions on Evolutionary Computation 10(6), 472–488 (2006)
6. Yong, L., Li-shan, K.: The Annealing evolution algorithm as function optimizer. Parallel Computing 21, 389–400 (1995)
7. Kunqi, L.: Differential Evolution Algorithm Based on Simulated Annealing. In: Kang, L., Liu, Y., Zeng, S. (eds.) ISICA 2007. LNCS, vol. 4683, pp. 120–126. Springer, Heidelberg (2007)
8. Wang, L., Jiao, L.: A novel genetic algorithim based on immunity. In: Proceedings of the 2000 IEEE International Symposium on Circuits and Systems(ISCAS), pp. 385–388 (2000)
9. Zhang, Q., Sun, J., Tsang, E.: Evolutionary Algorithm with Guided Mutation for the Maximum Clique Problem. IEEE Transaction on Evolutionary Computation 9(2), 192–200 (2005)
10. Reynolds, R.G.: An Introduetion to Cultural Algorithms. In: Sebalk, Fogel, A.V., River Edge, J. (eds.) Proceedings of the 3th annual Conference on Evolution Programming, pp. 131–136. World Scientific Publishing, NJ (1994)
11. Jingbo, A., Hongfei, T.: Cultural based Particle Swarm Optimization Algorithm with Application. Liaoning, Dalian Uni. of Tech. (2005)

12. Storn, R., Price, K.: Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization (11), 341–359 (1997)
13. Tao, G., Kang, L.-s.: A new evolutionary algorithm for function optimization. Wuhan University Journal of Nature Sciences 4(4), 409–414 (1999)
14. Kang, Z., Li, Y., Liu, P., Kang, L.-s.: An all-purpose evolutionary algorithm for solving nonlinear programming problems. Journal of computer research and development 39(11) (2002)
15. Wu, Z., Huang, H.: A differential evolution algorithm with double trial vectors based on Boltzmann mechanism. Journal of NanJing University (Natural Sciences) 44(2) (2008)
16. Vesterstrom, J., Thomsen, R.: A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems. In: Proceedings of the 2004 Congress on Evolutionary Computation, pp. 1980–1987 (2004)
17. Storn, R., Price, K.: Differential evolution – a simple and dfficient adaptive scheme for global optimization over continuous spaces. Technical report, International Computer Science Institute, Berkley (1995)
18. Tao, G., Kang, L.-s.: A New Evolutionary Algorithm for Function Optimization. Journal of WuHan University (Natural Sciences) 4(4), 409–414 (1999)
19. Guo, Y.-n., Wang, H.: Overview of cultural algorithms. Computer Engineering and Applications 45(9), 41–46 (2009)

A Hierarchical Distributed Evolutionary Algorithm to TSP

Chengjun Li[1], Guangfu Sun[1], Dongmei Zhang[1], and Songhu Liu[2]

[1] School of computer, China University of Geosciences,
Wuhan 430074, China
[2] China Construction Bank Wuhan Audition Branch 6th Department,
Wuhan 430015, China
cuglicj@126.com

Abstract. A hierarchical distributed evolutionary algorithm (hdEA) to TSP is proposed in this paper to enhance the performance of evolution algorithm (EA). This hdEA has a two-level migration combination in which global migrations and local ones are both in ring topology. Moreover, to simplify the settings, the ratio of local migrations and global ones is used to replace interval of these two kinds of migrations. In experiments using several instances from TSPLIB, the outcomes of this hdEA and those of the dEA with ring topology were compared. The results show the comprehensive advantage of the hdEA.

Keywords: hierarchical distributed evolutionary algorithm, two-level migration, TSP.

1 Introduction

Traveling salesman problem (TSP) is a famous NP-hard problem discussed in many fields. It can be defined as follow: Let $G=(V,A)$ be a graph, where $V = \{v_1, \ldots\ldots, v_n\}$ is a vertex (or node) set and $A = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\}$ is an edge set, with a non-negative distance (or cost) matrix $C = (c_{ij})$ associated with A. The problem is said to be a symmetric if $c_{ij} = c_{ji}$ for all $(v_i, v_j) \in A$, and asymmetric if this is not necessarily the case .The TSP consists of finding the shortest Hamiltonian cycle in G, (or circuit in the asymmetric case) which is often simply called a tour. This problem attracts many researchers of different fields and have many applications.

Evolutionary algorithms (EAs) have been employed to solve this problem for years. However, using EAs, only the best solutions of small scale data sets of TSP can be obtained. According to the experimental results, the efficient EA was purposed in [1] can only get the best solution of some instances whose scale is less than 280. Thus, it is not enough to merely use simple EA to solve TSP of large scale.

* Supported by the National Natural Science Foundation of China under Grant No. 40972206.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 134–139, 2010.

In the past a few decades, there exists a tradition in using parallel models of evolutionary algorithms (EAs) to obtain higher efficiency or better results. Among types of parallel EAs (PEAs), the distributed EAs (dEAs) are very popular optimization procedures. A dEA is a multi-subpopulation model performing sparse exchanges of individuals among the elementary subpopulations. This model can be readily implemented in distributed memory MIMD computers, which provides one main reason for its popularity. Moreover, some hybrid algorithms in which a two-level approach of parallelization named hierarchical distributed EAs (hdEAs) are undertaken in the literature[3-5]. The experimental results showed that the solutions of hdEAs are better than those of dEA if the number of subpopulations is same and much enough. For the same reasons, the combination that there are several farms of distributed algorithms with a still higher level dEA making migrations among connected farms is the most widely used among these hybrid algorithms[2].

Distributed evolutionary algorithms (dEAs) have been used to enhance the solutions of TSP. There are some papers about this topic, but there is still a large space to get improved parallel algorithms. In this paper, a hierarchical evolutionary algorithm to TSP is proposed to get better solutions of larger scale TSP instances. The result of experiments showed that its performance is better than that of a dEA that have the same number of subpopulations.

The remain of this paper was organized as follow: dEA was introduced in section 2 and hdEA in section 3. In section 4, a set of experiments were carried on in the same condition to compare the outcomes of these two kinds of algorithm to TSP and an analysis was done about the results. The last section is a conclusion.

2 Distributed Evolutionary Algorithm

Though a dEA have a variety of parameters and settings, the kind of a dEA is mainly determined by its migration strategy. A migration strategy must define the island topology, when migration occurs, which individuals are being exchanged, the synchronization among the subpopulations, and the kind of integration of exchanged individuals within the target subpopulations[2]. Two ideas are behind a distributed model: exploring different search areas via different subpopulations, and maintaining diversity within populations thanks to the exchange of individuals[6].

Here gives the some definitions of the parallel settings and parameters in dEAs.

Migration topology: The topology rule of sending and receiving individuals among subpopulations in every time migration.

Migration strategy: The rule of selecting which individuals in subpopulations to be emigrated and replaced.

Migration size: The number of individuals to be migrated

Migration interval: certain number of generations between two migrations

A dEA may be run according to Fig. 1. In this Fig., master is only one while slaves have multiple ones. The task of former is to spawn slaves and correct results coming form them then send the results back to them according to a given topology.

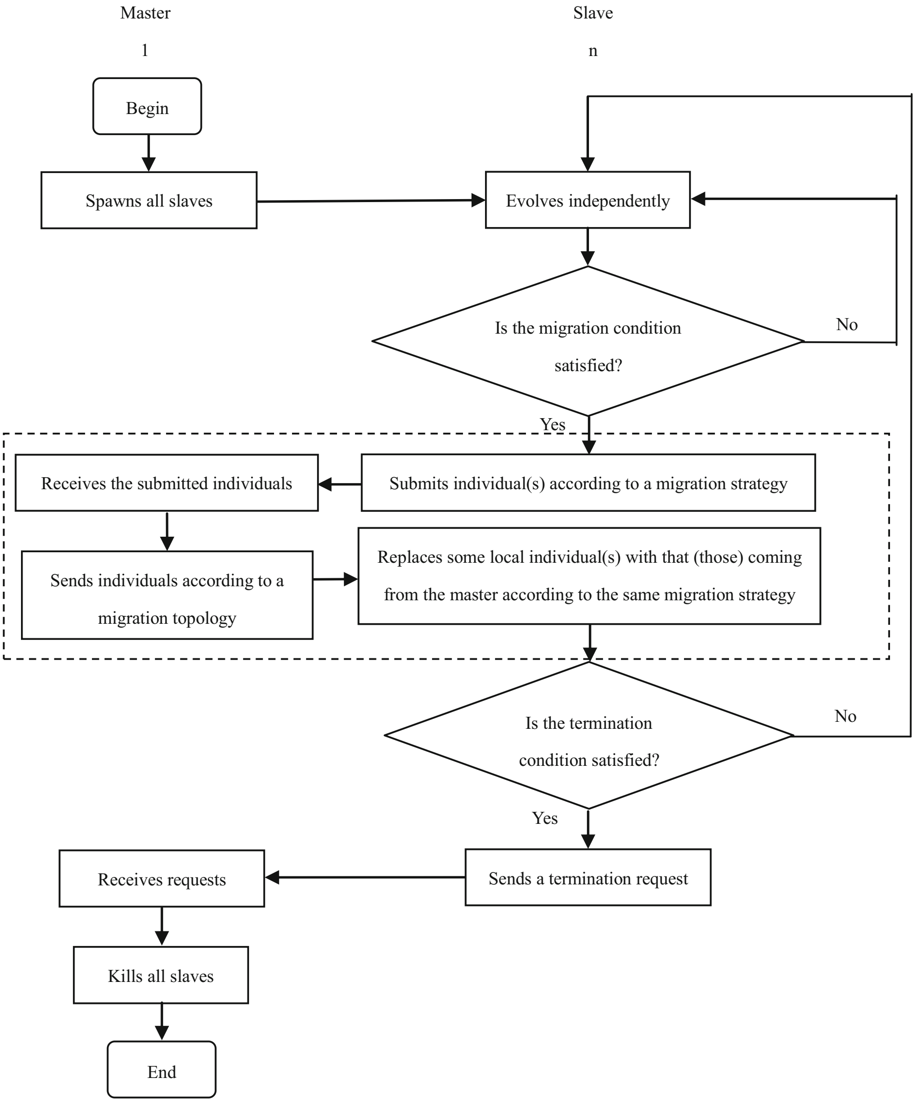

Master
1
Slave
n
Begin
Spawns all slaves
Evolves independently
Is the migration condition satisfied?
No
Yes
Receives the submitted individuals
Submits individual(s) according to a migration strategy
Sends individuals according to a migration topology
Replaces some local individual(s) with that (those) coming from the master according to the same migration strategy
Is the termination condition satisfied?
No
Yes
Receives requests
Sends a termination request
Kills all slaves
End

Fig. 1. Flow chart of a distributed evolutionary algorithm

3 Hierarchical Distributed Evolutionary Algorithm

Definition 3.1. Group: The set of all subpopulation lying in a same local migration of a hdEA.

HdEAs belong to dEAs but have more complicated migrations which are divide into local migrations and global ones. In generally, after local migration being executed many times, global migration will be executed once. During the course of local migrations, the slaves in a same group exchange their individuals according to local topologies. However, in global migrations, a part of individuals of each group are send to other groups and replace some local individuals according to global topologies.

The effect of global migration is shown as below. In Fig. 2, P denote the whole population. G denote a group in P. S denote a subpopulation. I denote a individual.

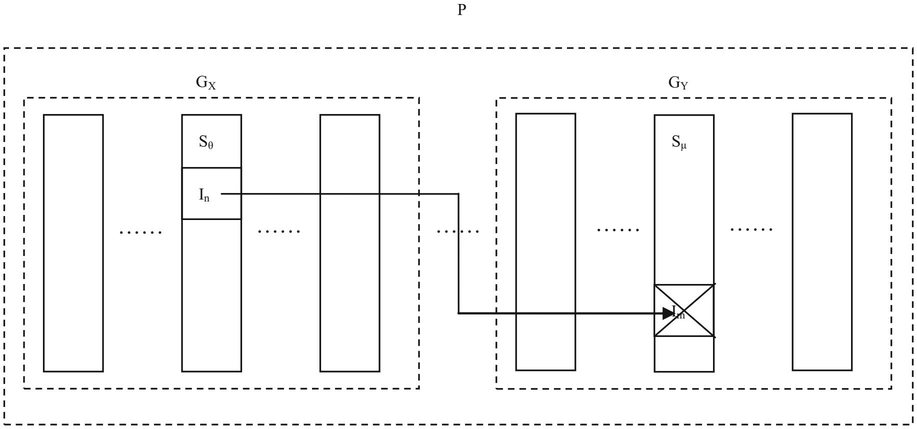


Fig. 2. Sketch map of individual global migration

In one time global migration, when I_n is immigrated to S_μ which located in G_Y of P, it is going to replace the I_m in S_μ.

The existence of these two different migrations is the key feature of hdEAs, since they establish the real hierarchy between the basic dEAs and the hdEA[3].

The behavior of traditional dEAs can be improved with hdEA model in the following ways[3]:

1>Better performance for each node. In the hdEAs, every node is a dEA instead of a simple subpopulation.

2>The search may be carried out in different levels. Every basic dEA develops an independent search and collaborates, through the global migrations, with others basic dEAs on a global search of the hdEA.

Although these combinations of local migrations and global ones may give rise to interesting and efficient new algorithms, they have the drawback of having even more new parameters to account for the more complex topological structure[2]. Hence, the core task of build a hdEA is to confirm the parameters and settings combination.

4 A Hierarchical Distributed Evolutionary Algorithm to TSP

4.1 Parallel Parameters and Settings

To get better solutions of larger scale TSP instances, a hdEA were employed to enhance the performance of EA. In experiments, the outcomes of this hdEA and that of a dEA based on the same EA proposed in [1] were compared. In this comparison, these two algorithm used the same number of subpopulation and have the same other parameters and rules as below.

To decrease the complexity of confirming the parameters and settings combination, a novel way is used in this paper. That is, instead of setting the interval of local migrations and global ones separately, a new parameter that the ratio of these two kinds of migrations can be used. Once the migration interval and the ratio of these two kinds of migrations are confirmed, the interval of local migrations and global ones are both confirmed.

All the parallel parameters and settings are selected by experiments and listed in Table 1.

Table 1. Parallel parameters and settings

Size of subpopulation	100 individuals
Number of subpopulation	8×8/64 (hdEA/ dEA)
Migration size	1
Migration topology	ring
Ratio of local migration times to global migration time	19:1 (hdEA)
Emigration-replacement method	Random-random
Termination condition of algorithm	The best solution is in submitted individuals or the submitted individuals are all the same during 100 times migration

4.2 Comparison of Algorithms

In Table 2, ① denoted hdEA based on ring topology both in global migration and local migration while ② denoted dEA based on ring topology.

Table 2. Comparison between outcomes of the two algorithms

Dataset	Algorithm/ Executed times	Migration interval	Best outcome	Average	Standard deviation	T-test (①-② and ③-④)
a280	①/10	2000	2579	2579.0	0.00	-
	②/10		2579	2579.0	0.00	
lin318	①/10	50000	42029	42029.0	0.00	-
	②/10		42029	42029.0	0.00	
pcb442	①/50	200000	50808	50893.2	41.02	-8.94
	②/50		50912	50930.8	9.36	
d657	①/50	300000	48923	48950.8	20.24	-25.10
	②/50		48980	49042.2	30.27	
U724	①/50	200000	41916	41938.4	15.23	-47.53
	②/50		42030	42097.4	29.80	

As above mentioned, the simple EA in [1] can only get the best solution of a280. These results show that the dEA and the hdEA have better performance. In Table 2, the results showed that this hdEA can get better solutions than the dEA. Comparison of variances shows the new algorithms is more stable and the T-test also shows the full advantages of the hdEA.

4.3 Speed-Up Ratio Experiment of This hdEA

In experiment, lin318 instance was run 20 times and the average run time was used to compute speed up ratio. The data in table 3 show this hdEA can get a super-linear speed-up ratio.

Table 3. Results of speed up ratio experiment

Number of hosts	1	2	4	8	16	32
Speed up ratio	1.00	1.88	3.11	6.25	11.10	25.90

5 Conclusion

The experiments carried on this paper showed that this hdEA to TSP can get better solutions. Moreover, it is a feasible approach to use the ratio of these two kinds of migrations replace interval of local migrations and global ones to simplify the settings. There is a conclusion that hdEA is a better selection than dEA if they is the same in their subpopulation and other common parameters and settings. In fact, the parallel parameters or settings have nothing to do with special EAs and can be widely used. No matter what is the problem, hdEA can be employed to get much better performance than a simple EA. In this way, the fact that a hdEA is a short cut to get better solutions for some hard problems to EA even to dEA was proofed. Next work will focus on how to setup the parallel rules and parameters to make most of hdEA.

References

1. Cai, Z.H., Peng, J.G., Gao, W., Wei, W., Kang, L.S.: An improved evolutionary algorithm for the Traveling Salesman Problem. Chinese Journal of Computers 28, 823–828 (2005)
2. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. IEEE Transactions on Evolutionary Computation 6, 443–462 (2002)
3. Liakopoulos, P.I.K., Kampolis, I.C., Giannakoglou, K.C.: Grid enabled, hierarchical distributed metamodel-assisted evolutionary algorithms for aerodynamic shape optimization. Future Generation Computer Systems 24(7), 701–708 (2008)
4. Garai, G., Chaudhuri, B.: A distributed hierarchical genetic algorithm for efficient optimization and pattern matching. Pattern Recognition 40(1), 212–228 (2007)
5. Herrera, F., Lozano, M., Moraga: Hierarchical Disturbuted Genntic Algorithms. International Journal of Intelligent Systems 14, 1099–1121 (1999)
6. Fernández, F., Tomassini, M., Vanneschi, L.: Studying the influence of communication topology and migration on distributed genetic programming. In: Miller, J., Tomassini, M., Lanzi, P.L., Ryan, C., Tetamanzi, A.G.B., Langdon, W.B. (eds.) EuroGP 2001. LNCS, vol. 2038, pp. 51–63. Springer, Heidelberg (2001)
7. Fernández, F., Tomassini, M., Punch III., W.: Experimental study of multipopulation parallel genetic programming. In: Poli, R., Banzhaf, W., Langdon, W.B., Miller, J., Nordin, P., Fogarty, T.C. (eds.) EuroGP 2000. LNCS, vol. 1802, pp. 283–293. Springer, Heidelberg (2000)
8. Cantú-paz, E., Goldberg, D.: On the scalability of parallel genetic algorithms. Evolutionary Computation 7, 429–449 (1999)

A New Distributed Node Localization Scheme Using a Mobile Beacon

Sheng Xiao, Changfeng Xing, and Zhangsong Shi

Electronic Engineering College, Naval University of Engineering, Wuhan,
430033 Hubei, China
xiaosheng1983417@163.com

Abstract. Aimed at the node localization in WSNs, a distributed localization scheme based on a mobile beacon was proposed in this paper. In the scheme, the mobile beacon promulgates its location information periodically, and nodes do not need to communicate with each other so that the energy consummation is depressed and the computation quantity is small. The scheme can be used in the nonlinear condition and thus is suitable for large WSNs which are deployed outdoors. Simulation results show that this scheme performs well under reasonable parameters.

Keywords: Wireless Sensor Networks, sensor localization, Mobile beacon, Unscented Kalman Filter.

1 Introduction

In wireless sensor networks (WSNs), Sensor localization information can be used in the self organization and configuration of networks in deciding where events take place. So, node localization is one of the key supporting technologies.

However, configuring each node with GPS receivers or manual is unrealistic because of constrains of resource, cost and application of environmental. It is very important to study the characteristics of positioning mechanism in WSNs [1]. Because of the nodes are resource constrained and random deployed, communications are vulnerable to interference, so the algorithm must be self-organization, energy efficiency, etc., and have good fault tolerance.

In this paper, a distributed localization scheme based on a mobile beacon is proposed. The scheme does not need to add extra hardware in the nodes, communication and computation are small. Simulation results show that the algorithm has good positioning accuracy and can be applied to large-scale deployment of outdoor occasions very well.

2 Related Work

At present, the positioning algorithm can be divided into two categories: range-based and range-free. The former use trilateralation, triangulation, or the maximum likelihood

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 140–147, 2010.

estimate method to locate nodes after measuring the distance or angle between nodes. RSSI、TOA、TDOA、AOA are the typical methods. It is more accurate than range-free method, but need more computation cost and power consumption. Range-free method utilize the network connectivity, such as Centroid、DV-Hop and MDA-MAP, to achieve node localization. It need less power and cost though the positioning accuracy is lower [2-5].

When range-based method is used, nodes can be divided into two categories: unknown nodes and beacon nodes. Since beacon nodes are equipped with GPS or other positioning device, they can obtain their own coordinates and scattered in the sensor network deployment area. In such positioning method, the beacon mode and arrangement of node play an important role in the final positioning accuracy greatly. As the cost of the beacon node is much higher than ordinary nodes, the cost, computational load and communication load of the entire network will increase if there are many beacon nodes.

Therefore, it was suggested that one mobile beacon or a few mobile beacons be used to aid positioning. In this way, mobile beacon nodes move in the region and send their current location information after a certain distance or time interval, unknown nodes complete their own position estimates according to the received information. So, the number of beacon nodes and the network cost is effectively reduced, and it is very practical. Sichitiu and Ramadurai [6] utilized a single mobile beacon combined with RSSE to estimate the node location for the first time in literature. Galstyan [7] proposed a new method in which the character that "perpendicular bisector of chord through the center of the circle" is used.

This paper put a large-scale outdoor sensor network which is deployed by a mobile beacon (aircraft, vehicle or robot) as the background; integrate the ranging technical and unscented kalman filtering (UKF), to get node position relying on a single mobile beacon. UKF filtering can effectively solve the nonlinear problem and reduce the measurement error. The simulation results show that the algorithm is less affected by distance error and is ideal for outdoor large-scale random deployment occasion.

3 Distributed Localization Using a Mobile Beacon

In our algorithm, the rough location of unknown node by the signal strength of mobile beacon is estimated firstly, then the unknown node tracks the mobile beacon using UKF algorithm. As the mobile beacon periodically promulgates its location information, the location information of the unknown node can be continuously updated online so that the position accuracy can be improved.

3.1 System Model

We assume that the position of the node does not change after deployment, so the state equation and measurement equation can be presented as:

$$\mathrm{x}_i(n+1)=\mathrm{x}_i(n)+\mathrm{w}_i(n) \quad 1\leq \mathrm{i} \leq 3 \tag{1}$$

$$\mathrm{y}(n)=\mathrm{g}(\mathrm{x}(n))+\mathrm{v}(n) \tag{2}$$

$\mathrm{w}_i(n)$ is the position of the node that has subtle disturbance. $\mathrm{g}(\cdot)$ is the observation vector and $\mathrm{v}(n)$ is the observation noise. Further more, Formula (2) can be expressed as:

$$y(n)=\sqrt{(\Delta x(n))^2+(\Delta y(n))^2+(\Delta z(n))^2}+v(n) \tag{3}$$

$\Delta x(n)$ ($\Delta y(n)$, $\Delta z(n)$)is the distance between the beacons and unknown nodes in the x (y, z) axis at the present time, $v(n)$ is the error caused by the wireless communication, which is assumed to be a mean zero normal white noise. We supposed that $\Delta z(n)$ equals h because mobile beacon is assumed to move in a horizontal plane.

3.2 UKF Filter

UKF is a typical filtering method which can be used in nonlinear system. It approaches the probability density function (PDF) and the estimated mean can be accurate to the second-order Taylor expansion. Compared to the particle filter (PF), UKF select a good sample set using a series of systems of random variables to represent the posterior probability density, so a higher certainty can be obtained.

(1) Unscented transform.

Firstly, we compute Sampling points and the Corresponding weights:

$$\begin{cases} \mathrm{x}_0 = \overline{X} & i=0 \\ \mathrm{x}_i = \overline{X} + (\sqrt{(n_x+k)\rho_x})_i & i=1,\dots,n_x \\ \mathrm{x}_{i+n_x} = \overline{X} - (\sqrt{(n_x+k)\rho_x})_i & i=1,\dots,n_x \end{cases} \tag{4}$$

$$\begin{cases} W_0 = k/(n_x+k) & i=1,\dots, n_x \\ W_i = 1/[2(n_x+k)] & i=1,\dots, n_x \\ W_{i+n_x} = 1/[2(n_x+k)] & i=1,\dots, n_x \end{cases} \tag{5}$$

K is the Scale parameter, $(\sqrt{(n_x+k)\rho_x})_i$ is the root mean square matrix of the first row or line of $(n_x+k)\rho_x$, and n_x is the state vector dimension.

Then, sampling points propagate through a nonlinear function:

$$\mathrm{y}_i = g(\mathrm{x}_i) \quad i=0,\dots,2n_x \tag{6}$$

The estimated value and covariance estimates of y can be presented as:

$$\overline{\mathrm{y}} = \sum_{i=0}^{2n_x} W_i \mathrm{y}_i \tag{7}$$

$$\mathrm{P}(\mathrm{y}) = \sum_{i=0}^{2n_x} W_i (\mathrm{y}_i - \overline{\mathrm{y}})(\mathrm{y}_i - \overline{\mathrm{y}})' \tag{8}$$

(2) Filtering Model

The algorithm utilizes the WLSE to obtain a rough position of the unknown node (the initial state value). In order to speed up the convergence rate of UKF, we can set a large initial state error covariance matrix.

$\hat{\mathrm{x}}(k|k)$ is the state estimate vector and $\mathrm{P}(k|k)$ is the state estimate covariance. The corresponding δ points and their corresponding weights can be calculated by the formula (4) and (5). According to equation of state, we can get one step prediction:

$$\xi_i(k+1|k)=f(k,\xi_i(k|k)) \tag{9}$$

According to the formula (6) and (7), the state estimate and the state prediction covariance can be predicted as:

$$\hat{\mathrm{x}}(k+1|k)=\sum_{i=0}^{2n_x} W_i\xi_i(k+1|k) \tag{10}$$

$$P(k+1|k)=\sum_{i=0}^{2n_x} W_i\Delta X_i(k+1|k)\Delta X_i'(k+1|k)+Q(k) \tag{11}$$

The measurement covariance and the measurement state vector covariance are:

$$P_{zz}=R(k+1)+\sum_{i=0}^{2n_x} W_i\Delta Z_i(k+1|k)\Delta Z_i'(k+1|k) \tag{12}$$

$$P_{xz}=\sum_{i=0}^{2n_x} W_i\Delta X_i(k+1|k)\Delta Z_i'(k+1|k) \tag{13}$$

Status updating and covariance updating can be expressed as:

$$\hat{X}(k+1)=\hat{X}(k+1|k)+K(k+1)(Z(k+1)-\hat{Z}(k+1|k)) \tag{14}$$

$$P(k+1)=P(k+1|k)-K(k+1)P_{zz}(Z(k+1)K'(k+1) \tag{15}$$

$$K(k+1)=P_{xz}P_{zz}^{-1} \tag{16}$$

4 Algorithm Flow

The flow of the proposed algorithm is shown in Figure 1:

The sub-process can be illustrated like this:

(1) The mobile beacon send its location information periodically and unknown node receive the information passively.

(2) After unknown node have received location information (more than 3), the RSSI set, observations set and coordinates set of the beacon signal are established.

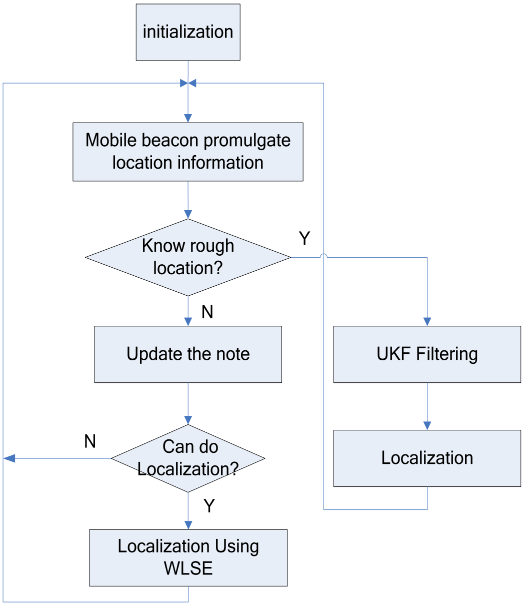


Fig. 1. Algorithm flow chart

(3) The approximate location of unknown nodes using WLSE is calculated.
(4) The exact position of unknown node using UKF algorithm is calculated.

We considered two main points in choosing the mobile beacon path: Any unknown node can receive 3 different beacon packets at least to ensure that the node can be located; Shorten the path length to reduce the time required for positioning and decrease the cost of the node.

Figure 2 shows a sample application.

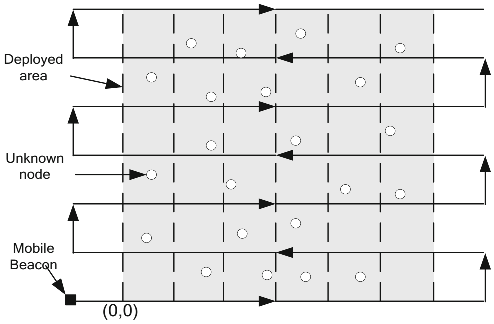


Fig. 2. Application example

5 Simulation and Analysis

5.1 Environment and Parameter Settings

In this paper, we consider the outdoor deployment of WSNs so the Propagation path loss model of the signal can be expressed in the following formula:

$$\mathrm{PL}(d_0)=32.44+10k\lg(d_0)+10\lg(f) \tag{17}$$

f is the frequency, k is the path attenuation factor. We assume d_0 is 1m to facilitate the calculation.

We assumed there are random 100 nodes in a $200\times200\mathrm{m}$ area, the movement route of the mobile beacon is shown in Figure 1. The transmission distance of the mobile Beacon is 30m. All of the following simulation results are the average of 30 times. The parameters of the simulation are shown in the following table:

Table 1. Simulation parameter settings

parameter	Value
Broadcasting interval (s)	1
Ranging error	10%
Beacon speed(m/s)	5
Nodes number	200

5.2 Analysis

We compare the performance of our algorithm and the Monte Carlo localization (MCL) algorithm in several different simulation scenarios.

In our first simulation scenario, we compare the RMSE of location, with the number of beacon position varying from 4 to 7. The performance results of three cases are shown in Figure 3. It can be seen that the proposed algorithm is better than MCL under the same conditions. Refer to the large-scale outdoor deployment; the unknown node can obtain much location information of mobile beacon using less additional cost, so the algorithm has significant advantages.

Figure 4 shows the changes of positioning accuracy. We assume that there is normal ranging error. It can be seen that the positioning accuracy of our algorithm is better than MCL algorithm under the same conditions.

Figure 5 shows the CPU time of our algorithm and MCL algorithm. As connectivity increases, the CPU time using MCL algorithm has also increased rapidly, while the CPU time using our algorithm remains unchanged ultimately because there are no communication between nodes.

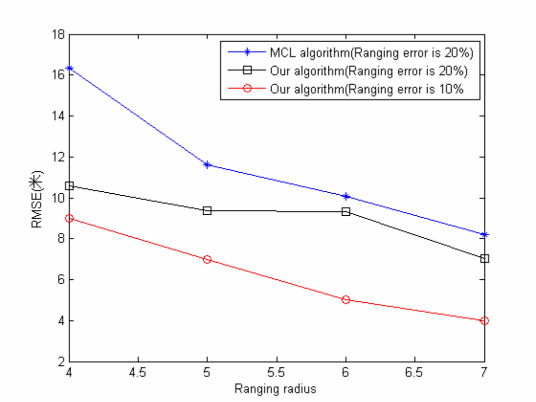


Fig. 3. Number of beacon location VS Average error

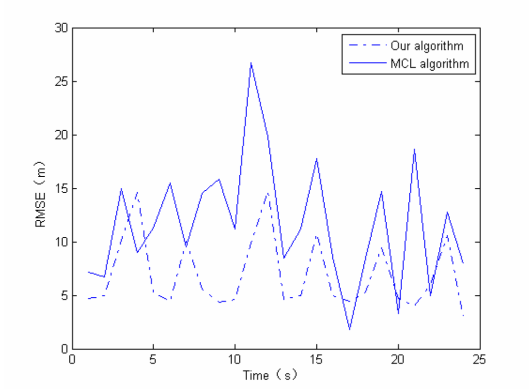


Fig. 4. The average error of two schemes

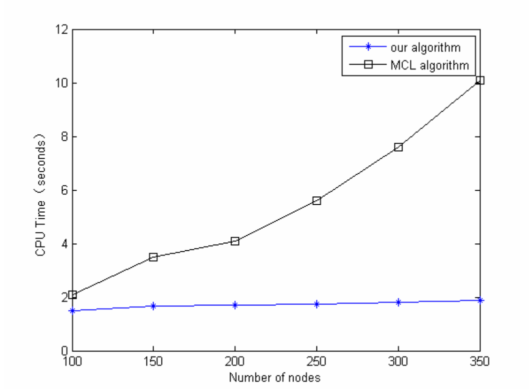


Fig. 5. CPU Time VS Nodes Number

6 Conclusion

We proposed a mobile-based distributed localization scheme in WSNs. In this scheme, Unknown nodes do not need to communicate with each other so the energy consumption is reduced and the calculation is not complicated. Simulation results show that our scheme can effectively improve the accuracy of determining nodes.

References

1. Akyildiz, I.F., Su, W.: Wireless Sensor Network: A Survey. Computer Networks 38(4), 393–422 (2002)
2. Fu-Bao, W., Long, S., Feng-yuan, R.: Self-Localization Systems and Algorithms for Wireless Sensor Networks. Journal of Software 16(5), 1148–1157 (2005)
3. Nicolescu, D., Nath, B.: Ad Hoc Positioning Systems (APS). In: Proc. of the 2001 IEEE Global Telecommunications Conf., pp. 2926–2931. IEEE Communications Society, Los Alamitos (2001)
4. Niculescu, D., Nath, B.: DV Based Positioning in Ad Hoc Networks. Journal of Telecommunication Systems 22(1-4), 267–280 (2003)
5. Bahl, P., Padmanabhan, V.N.: RADAR.: An In-building RF-based User Location and Tracking System. In: Proceedings of the 19th Annual Joint Conference on IEEE Computer and Communications Societies, pp. 775–784. IEEE Press, Aviv (2000)
6. Sichitiu, M.L., Ramadurai, V.: Localization of Wireless Sensor Networks with a Mobile Beacon. In: Proc. of the IEEE Int'l. Conf. on Mobile Ad-Hoc and Sensor Systems, pp. 174–183. IEEE Computer Society, Los Alamitos (2004)
7. Galstyan, A., Krishnamachari, B., Lerman, K.: Distributed online localization in sensor networks using mobile target. In: Proceeding of the international Symposium on information Processing Sensor Networks, Berkeley, pp. 61–70 (2004)

A Novel Multi-Population Genetic Algorithm for Multiple-Choice Multidimensional Knapsack Problems

Qian Zhou[1,2] and Wenjian Luo[1,2]

[1] Nature Inspired Computation and Applications Laboratory, School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, Anhui, China
[2] Anhui Key Laboratory of Software in Computing and Communication, University of Science and Technology of China, Hefei 230027, Anhui, China
zhou0924@mail.ustc.edu.cn, wjluo@ustc.edu.cn

Abstract. In this paper, a novel Multi-Population Genetic Algorithm (MPGA) is proposed to solve the Multiple-choice Multidimensional Knapsack Problem (MMKP), a kind of classical combinatorial optimization problems. The proposed MPGA has two evolutionary populations and one archive population, and can effectively balance the search biases between the feasible space and the infeasible space. The experiment results demonstrate that the proposed MPGA is better than the existing algorithms, especially when the strength of constraints is relatively strong.

Keywords: Combinatorial Optimization, Genetic Algorithm, Multiple-choice Multidimensional Knapsack Problem.

1 Introduction

The Knapsack Problem is NP-Complete, and the Multiple-choice Multidimensional Knapsack problem (MMKP) is a variant of Knapsack problems with multiple constraints and multiple choice [1]. In the MMKP, there are *n* classes of items. For the class *i*, there are r_i items, and each item *j* has one nonnegative value v_{ij} and a nonnegative weight vector $W_{ij} = \left(w_{ij}^1, w_{ij}^2, \ldots, w_{ij}^l\right)$, where *l* means the number of the resource constraints. The formula of the MMKP can be described as follows [1].

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} \sum_{j=1}^{r_i} v_{ij} x_{ij} \\
s.t. \quad & \sum_{i=1}^{n} \sum_{j=1}^{r_i} w_{ij}^k x_{ij} \le b_k \ , \qquad k \in \{1, \ldots, l\}, \\
& \sum_{j=1}^{r_i} x_{ij} = 1, \qquad i \in \{1, \ldots, n\}, \\
& x_{ij} \in \{0,\ 1\}, \qquad i \in \{1, \ldots, n\},\ \ j \in \{1, \ldots, r_i\}.
\end{aligned}$$

In the MMKP, only one item can be selected from each class. When the *j*th item in the *i*th class is picked into the knapsack, x_{ij} is 1. Otherwise, x_{ij} is 0. The objective of the

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 148–157, 2010.

MMKP is that the sum of the values of selected items is maximized. Meanwhile, the resource constraints should be satisfied.

Some algorithms have been proposed to solve the MMKP, and there are some benchmarks of the MMKP [2]. Moser et al. [3] firstly proposed a heuristic algorithm for the MMKP, which is based on the Lagrange Multipliers method. Based on the concept of the aggregate necessary resource proposed by Toyoda [4] for the MDKP (the Multidimensional Knapsack Problem), Khan et al. [5] proposed an improved algorithm for MMKP. Hifi and his colleagues proposed a guided local search algorithm in [1], and a reactive local search algorithm which can effectively escape from local optima in [6]. Furthermore, Cherfi and Hifi proposed a column generation algorithm to solve the MMKP in [7], and proposed three versions of a hybrid heuristic algorithm for the relatively large-scale MMKP in [8]. By constructing convex hulls to reduce the search space, Akbar et al. [9] proposed a method to find the near-optimal solution of the MMKP. In [10], Sbihi proposed an exact algorithm for the MMKP, which adopted a branch-and-bound procedure and the best-first search strategy. Parra-Hernandez et al. [11] proposed an efficient heuristic algorithm, i.e. the HMMKP, which transformed a MMKP to a MDKP.

In this paper, a novel multi-population genetic algorithm is proposed for the MMKP with many constraints. There are three populations in the MPGA. The MPGA is good at keeping the balance between feasible individuals and infeasible individuals. Experimental results demonstrate that the MPGA has better performance than the previous algorithms.

The rest of this paper is organized as follows. Section 2 presents some related work of the MMKP. Section 3 describes the proposed multi-population genetic algorithm. Some experiments are done in Section 4. Section 5 includes some discussion. Finally, in Section 6, this paper is summarized briefly.

2 Related Work

So far, most traditional methods for the MMKP are heuristic algorithms, including Lagrange Multipliers [3], branch-and-bound [7, 8, 10], the guided local search [1], and so on. However, Parra-Hernandez and his colleagues [11] have pointed out that these heuristic algorithms have two defects: (1) when the constraints of the MMKP are relatively strong, the heuristic algorithms are easily trapped into the local optimum, even cannot find a feasible solution. (2) The time costs needed by these heuristic algorithms are relatively large.

Genetic algorithms (GAs) [12] are a kind of global search algorithms based on natural selection, and are effective for combinatorial optimization problems. The GAs have been widely used to solve the Knapsack problems. For example, a GA is applied to solve the MDKP in [13], and the time cost of the GA for the MDKP is very short. However, to our best knowledge, except the work in [14], there is no other work that uses the GAs to solve the MMKP. In [14], Liu proposed a simple genetic algorithm for the MMKP. The simple GA in [14] used the criterion by Deb [15] to compare the candidate individuals. The criterion [15] is: (1) if both individuals are feasible, the individual with a higher fitness value is better. (2) If one individual is feasible and another is infeasible, the feasible one is better. (3) If both are infeasible, the individual with a lower constraint violation is better. Obviously, the GA only with such a criterion cannot effectively balance the search between the feasible space and infeasible space very well. When the level of constraints becomes stronger, the GA cannot guarantee to find a better feasible solution.

In this paper, a Multi-Population Genetic Algorithm (MPGA) is proposed to solve the MMKP. The MPGA can balance the search biases between the feasible space and the infeasible space. In the next section, the MPGA is introduced in detail.

3 The Proposed MPGA

3.1 The Fitness Function

For convenience, the fitness function is introduced firstly. The length of an individual is the same as the number of classes in the MMKP. An individual has two fitness values [14], i.e. the value-fitness and the violation-fitness.

The value-fitness of an individual indicates that the sum of the values of items which are picked into the knapsack. The greater of the value-fitness, the better the individual is.

The violation-fitness of an individual indicates the violated level of constraints. The violated-fitness is set to $\frac{1}{k}\sum_{i=1}^{k}\frac{w_i}{b_i}$, where k is the number of the constraints, and w_i indicates the weight sum on the ith resource constraints [14]. Especially, if the weight sum of selected items on the ith resource constraint is no greater than b_i, $w_i = 0$. Therefore, if an individual is feasible, its violation-fitness is 0. If one individual is infeasible, the smaller of the violation-fitness, the better the individual is.

3.2 The Algorithm Description

The MPGA includes two evolutionary populations and one archive population at each generation t, i.e. ε_t, ε_t^c and ε_t^m. The evolutionary population ε_t is used to find the global optimal of the MMKP, and the evolutionary population ε_t^c is used to find the feasible solutions, while the archive population ε_t^m only memorizes the best solutions ever evolved. The interaction of three populations is shown in Fig. 1.

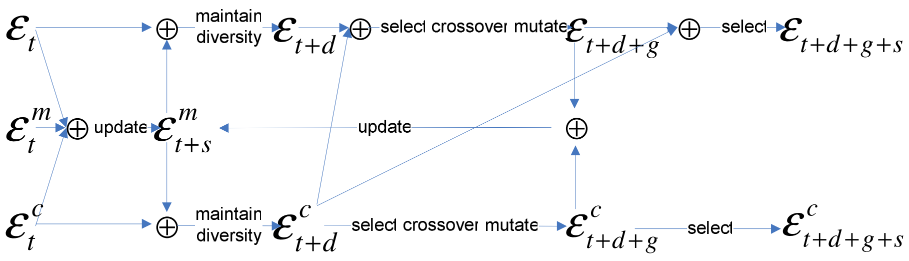


Fig. 1. The interaction of three populations

Algorithm 1. The process of the MPGA

1: Initialize three populations of ε_t , ε_t^m and ε_t^c randomly, and let *t*=0. The size of ε_t is *N*, and the size of ε_t^c is also *N* and set ε_t^m to be empty. As for ε_t^m , in the course of evolutionary iterations, the number of feasible individuals is no greater than 2*N*, and the number of infeasible ones are also no greater than 2*N*. Therefore, the maximal size of ε_t^m is 4*N*.

2: Evaluate ε_t , ε_t^m and ε_t^c , and update ε_t^m to ε_{t+s}^m . Firstly, calculate the value-fitness and the violation-fitness of each individual. Secondly, the best 2*N* feasible individuals from ε_t , ε_t^m and ε_t^c are selected and memorized in ε_{t+s}^m , and the best 2*N* infeasible individuals in terms of the violation-fitness are also saved. Therefore, a middle population ε_{t+s}^m is generated. Noted that the fitness value (i.e. the value-fitness and the violation-fitness) of each individual in ε_{t+s}^m is different from others.

3: Generate the middle population ε_{t+d} and ε_{t+d}^c . If ε_t converges to a local optimum, ε_t is emptied, and the best individuals in ε_{t+s}^m is copied into ε_t to generate ε_{t+d} . Otherwise, set $\varepsilon_{t+d} = \varepsilon_t$. Meanwhile, the same operation is done on ε_t^c to generate ε_{t+d}^c . Let φ denotes the set of feasible individuals in ε_t^c .

4: Generate the middle population ε_{t+d+g} with selection, crossover and mutation. According to the criterion by Deb [15], the individuals are selected by the binary tournament selection from ε_{t+d} and φ . The crossover is the two-point crossover. As for the mutation operator, for each class items, the item that makes the violation-fitness minimum is selected. If the violation-fitness does not change, the item with the largest value is selected. Noted that the violation-fitness of all feasible individuals is equal to zero.

5: Generate ε_{t+d+g}^c with selection, crossover and mutation. Selection is done on only infeasible individuals in ε_{t+d}^c , and the binary tournament selection in terms of the violation-fitness is adopted. Both crossover and mutation operators are the same as Step 4. The size of ε_{t+d+g}^c is $\left|\varepsilon_{t+d}^c - \varphi\right|$.

6: Update ε_{t+s}^m by selecting the best individuals from ε_{t+s}^m , ε_{t+d+g} and ε_{t+d+g}^c according to the value-fitness and the violation-fitness.

7: Generate $\varepsilon_{t+d+g+s}$ by selecting best individuals from ε_{t+d+g} and φ . Once some feasible individuals are found, the best one is kept in $\varepsilon_{t+d+g+s}$ and never lost.

8: Generate $\varepsilon_{t+d+g+s}^c$ by selecting the best $\left|\varepsilon_{t+d}^c - \varphi\right|$ individuals from ε_{t+d+g}^c and infeasible ones in ε_{t+d}^c , and randomly generating $\left|\varphi\right|$ individuals. The selection is in terms of the violation-fitness.

9: If the stop condition is not satisfied, let $\varepsilon_t = \varepsilon_{t+d+g+s}$, $\varepsilon_t^c = \varepsilon_{t+d+g+s}^c$, $\varepsilon_t^m = \varepsilon_{t+s}^m$, *t*=*t*+1, and go to step 2.

According to Fig. 1, the detail process of the MPGA is described in Algorithm 1. The primary process of the MPGA is shown in Algorithm 1.

As for Step 3 in Algorithm 1, more details are given as follows.

(1) When the number of individuals with identical value-fitness and violation-fitness in ε_t (or $\varepsilon_t^c - \varphi$) is no less than $N-5$ (or $N-|\varphi|-3$), or the number of individuals with different value-fitness and violation-fitness is less than 3, the population ε_t (or $\varepsilon_t^c - \varphi$) is regarded as being trapped into the local optimum. When the population ε_t (or $\varepsilon_t^c - \varphi$) is being trapped into the local optimum, ε_t (or $\varepsilon_t^c - \varphi$) is emptied, and the individuals in ε_{t+s}^m are copied into ε_t (or $\varepsilon_t^c - \varphi$).

(2) Suppose the number of feasible individuals in ε_{t+s}^m is δ, and the feasible individuals in ε_{t+s}^m are decrementally ranked according to the value-fitness, and the infeasible individuals are incrementally ranked according to the violation-fitness. If $\delta \geq 5$, the feasible individuals in ε_{t+s}^m are copied into ε_t circularly. If $\delta < 5$, other $(5-\delta)$ best infeasible individuals are selected from ε_{t+s}^m, and these five individuals are copied into ε_t circularly. As for $\varepsilon_t^c - \varphi$, the best infeasible individuals in ε_{t+s}^m are copied into $\varepsilon_t^c - \varphi$.

(3) If the best feasible individual is not improved after ε_t is emptied and occupied by the feasible individuals in ε_{t+s}^m for three times, and $\delta \geq 3$, and *t* is even, and the number of individuals with the same value-fitness is greater than a random number γ between 3 and 7, only γ individuals are kept. Meanwhile, other individuals are replaced by feasible individuals in ε_{t+s}^m randomly.

4 Experiments

4.1 Experimental Settings

In this section, since the number of constraints of the MMKP instances adopted in [11, 14] is relatively large, the MPGA is compared with the HMMKP [11] and the simple GA [14] with a series of experiments. The test instances (i.e. mknapcb7, mknapcb8 and mknapcb9) can be obtained from [16]. For each instances, three different values of *f* is set, where *f* means the constraint strength [11].

The experimental results of the HMMKP is from [11]. The settings of the GA are the same as [14], where the population size is 100, and the maximal evolutionary generation is 2000. For fair, in the MPGA, *N*=50, and the number of maximal evolutionary generation is 2000 for each run. The crossover probability p_c is 0.8. Each experiment runs 50 times independently.

4.2 Experimental Results

The experimental results are given in Table 1, Table 2, Table 3.

Table 1. Experimental results on Mknapcb7

f	Mknapcb7	HMMKP	GA[14]			MPGA		
		[11]	MAX	AVE	MIN	MAX	AVE	MIN
0.9	0	17510	18627	18535.8	18524	18627	**18549.1**	18524
	1	17948	18081	18077.0	17981	18081	**18081.0**	18081
	2	17049	17688	**17688.0**	17688	17688	**17688.0**	17688
	3	17833	17935	**17935.0**	17935	17935	**17935.0**	17935
	4	17315	18550	18532.8	18483	18550	**18548.7**	18483
	5	18318	18707	18639.3	18608	18707	**18705.1**	18614
	6	18045	18141	18135.8	18107	18141	**18140.6**	18119
	7	17301	18122	**18122.0**	18122	18122	**18122.0**	18122
	8	17563	18881	18821.9	18799	18881	**18869.1**	18803
	9	16691	17286	**17263.7**	17184	17286	17239.1	17184
0.84	0	15617	17447	17396.3	17335	17482	**17443.1**	17351
	1	15951	17025	17008.8	16855	17120	**17026.9**	17025
	2	14080	16655	16473.4	16292	16655	**16607.8**	16421
	3	14876	16986	16913.9	16805	17041	**16991.5**	16846
	4	15595	17531	**17531.0**	17531	17531	**17531.0**	17531
	5	15791	17742	17664.0	17546	17742	**17671.6**	17602
	6	15484	17408	17323.0	17250	17425	**17394.9**	17385
	7	14963	16837	16782.5	16692	16985	**16855.9**	16774
	8	16160	17763	17751.9	17625	17763	**17759.5**	17616
	9	13098	16325	16235.1	16131	16325	**16323.6**	16289
0.8	0	N/A	15597	15333.5	14946	15799	**15559.6**	15404
	1	N/A	15816	15599.4	15488	15783	**15724.6**	15665
	2	N/A	15469	14954.5	14698	15503	**15410.3**	14951
	3	N/A	15869	15722.6	15381	15869	**15822.3**	15709
	4	N/A	16300	16211.9	15694	16300	**16300.0**	16300
	5	N/A	16742	16535.4	16425	16742	**16742.0**	16742
	6	N/A	16045	15889.7	15700	16401	**16088.8**	15966
	7	N/A	15029	14883.5	14701	15290	**15015.0**	14836
	8	N/A	16564	16511.0	16401	16564	**16559.9**	16541
	9	N/A	14863	14732.8	14567	15033	**14862.1**	14772

From Table 1, Table 2 and Table3, it can be observed that both the GA and the MPGA perform significantly better than the HMMKP [11] in all cases.

When *f* is 0.9, both the GA and the MPGA have similar results, and the MPGA is a little better than the simple GA.

Furthermore, the smaller *f* is, the better the MPGA performs. That is to say, the MPGA is significantly better than both HMMKP and the simple GA when the resource constraint is relatively strong.

Especially, the GA fails to find a feasible solution sometimes. For example, the simple GA sometimes cannot find a feasible solution for three instances of the Mknapcb9 when *f*=0.73. However, the MPGA can always find a better feasible solution.

Table 2. Experimental results on Mknapcb8

f	**Mknapcb8**	**HMMKP [11]**	**GA [14]**			**MPGA**		
			MAX	**AVE**	**MIN**	**MAX**	**AVE**	**MIN**
0.9	0	45493	45982	45981.0	45931	45982	**45982.0**	45982
	1	47130	47291	47278.4	47234	47291	**47291.0**	47291
	2	45390	45673	45643.0	45596	45673	**45658.6**	45616
	3	45810	45806	45804.2	45804	45810	**45808.0**	45804
	4	45270	45361	45334.3	45295	45361	**45358.8**	45317
	5	46611	46611	**46611.0**	46611	46611	**46611.0**	46611
	6	46064	46375	**46375.0**	46375	46375	**46375.0**	46375
	7	45450	45491	45472.2	45459	45491	**45486.1**	45459
	8	47156	47159	**47159.0**	47159	47159	**47159.0**	47159
	9	45859	46149	46148.6	46139	46149	**46148.8**	46139
0.8	0	38523	43230	42980.0	42706	43487	**43209.9**	42872
	1	41185	44696	44459.9	44261	44843	**44629.4**	44450
	2	41259	43237	42938.4	42666	43365	**43108.3**	42806
	3	40066	42825	42545.9	42343	43011	**42778.7**	42513
	4	38262	43124	42898.3	42688	43308	**43040.2**	42726
	5	39670	43277	42949.0	42600	43486	**43258.2**	42993
	6	38547	43675	43415.6	43088	43799	**43574.6**	43330
	7	39445	42218	41987.7	41603	42520	**42225.2**	41957
	8	40954	44485	44173.5	43955	44713	**44446.4**	44006
	9	39834	43426	43132.1	42693	43582	**43380.2**	43062
0.75	0	N/A	38149	37576.9	36895	38933	**38391.6**	37519
	1	N/A	39720	38732.8	37572	40105	**39704.1**	38946
	2	N/A	39000	38208.3	37687	39147	**38798.0**	38392
	3	N/A	38435	37873.5	37349	38826	**38442.9**	38013
	4	N/A	38508	37850.3	37059	39207	**38728.0**	38201
	5	N/A	37444	36264.8	35156	37829	**37000.3**	35777
	6	N/A	37536	36587.9	35356	38475	**37592.2**	36747
	7	N/A	37833	37464.8	37054	38322	**37925.2**	37632
	8	N/A	38910	38009.4	36874	39647	**38911.9**	38240
	9	N/A	36936	36112.4	35363	37380	**36745.5**	35689

5 Discussions

For the MMKP, how to balance the search between the feasible space and the infeasible space is significantly important. In the MPGA, multiple populations are adopted to keep the balance. Experimental results demonstrate that the MPGA has better performance.

The MPGA proposed in this paper has three populations. Two populations are used to find the feasible solutions, and the better feasible solutions, respectively. The other population is used to memorize the best solutions ever generated. Obviously, it is very important to study the interactions among these populations. Different interactive ways may result in different performance. To improve the performance of the MPGA for the MMKP with strong constraints, more efficient cooperation mechanism among these populations should be studied.

Because of page limitation, experiments are just tested on mknapcb7, mknapcb8 and mknapcb9 when *N*=50. Suppose the whole size of both ε_t and ε_t^c is fixed, different sizes of ε_t (or ε_t^c) may result in different performance. Different settings of the parameters should be tested in the future.

Additionally, there are many resource constraints in the MMKP. All constraints should be satisfied in the experiments. However, in the real-word applications, some constraints should be satisfied absolutely, while others should be satisfied as much as possible. Because the MPGA has multiple populations, it seems appropriate to solve such problems. Anyway, it is very interesting to study whether the proposed MPGA can deal with such problems. It will be done in the future.

Table 3. Experimental results on Mknapcb9

f	**Mknapcb9**	**HMMKP [11]**	**GA [14]**			**MPGA**		
			MAX	**AVE**	**MIN**	**MAX**	**AVE**	**MIN**
0.9	0	92021	92025	**92025.0**	92025	92025	**92025.0**	92025
	1	92371	92371	**92371.0**	92371	92371	**92371.0**	92371
	2	93396	93396	93391.9	93384	93396	**93395.8**	93387
	3	91815	91815	91807.4	91798	91816	**91813.3**	91800
	4	93317	93317	93316.6	93295	93317	**93317.0**	93317
	5	91547	91551	**91550.4**	91547	91551	**91550.4**	91549
	6	91480	91480	**91480.0**	91480	91480	**91480.0**	91480
	7	91672	91681	91657.0	91645	91681	**91662.8**	91646
	8	93149	93149	**93149.0**	93149	93149	**93149.0**	93149
	9	93528	93531	93530.0	93507	93531	**93531.0**	93531
0.75	0	74927	82742	81578.7	80681	83268	**82676.5**	81838
	1	73570	80125	78947.1	77715	81222	**80528.5**	79791
	2	74739	81493	80523.3	79458	82815	**81928.0**	80756
	3	69813	80288	79201.7	77916	81672	**80910.2**	80141
	4	74323	82549	81187.4	80028	83572	**82779.4**	81719
	5	74303	81634	80940.3	79892	82815	**82049.1**	81237
	6	72018	78658	77265.0	75917	79760	**78902.2**	77990
	7	73777	79235	78178.4	77115	80493	**79865.2**	79055
	8	74376	80717	79827.0	78840	81970	**81262.5**	80557
	9	73496	81877	80893.4	79514	83236	**82535.4**	81536
0.73	0	N/A	77443	75357.0	73681	78584	**77801.9**	76725
	1	N/A	73197	71614.3	70363	74877	**73897.2**	72803
	2	N/A	75731	36528.3	0	76904	**75552.8**	73675
	3	N/A	73322	71832.4	70024	75335	**74259.5**	72440
	4	N/A	76382	75142.7	74055	78389	**77104.6**	75914
	5	N/A	76117	74812.3	72618	78314	**77110.2**	75709
	6	N/A	71820	67155.3	0	73175	**71956.2**	69992
	7	N/A	72643	71172.7	68969	74434	**73268.5**	71354
	8	N/A	75431	73764.9	72038	76600	**75811.7**	73799
	9	N/A	74605	59521.0	0	77046	**75486.8**	73698

6 Conclusions

In this paper, a novel multi-population genetic algorithm is proposed to solve the MMKP, which has two evolutionary populations and one archive population. These three populations can effectively balance search biases between feasible space and infeasible space. Several experiments are done to test the proposed algorithm. The experimental results demonstrate that the proposed multi-population genetic algorithm has a better performance than the HMMKP and the simple GA, especially when the strength of constraints is relatively strong.

Acknowledgments. This work is partly supported by the National Natural Science Foundation of China (NO. 60774075).

References

1. Hifi, M., Michrafy, M., Sbihi, A.: Heuristic Algorithms for the Multiple-Choice Multidimensional Knapsack Problem. Journal of the Operational Research Society 55(12), 1323–1332 (2004)
2. Han, B., Leblet, J., Simon, G.: Hard Multidimensional Multiple Choice Knapsack Problem, an Empirical Study. Computers and Operations Research 37(1), 172–181 (2010)
3. Moser, M., Jokanovic, D.P., Shiratori, N.: An Algorithm for the Multidimensional Multiple-Choice Knapsack Problem. IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences E80A(3), 582–589 (1997)
4. Toyoda, Y.: A Simplified Algorithm for Obtaining Approximate Solution to Zero-One Programming Problems. Management Science 21(12), 1417–1427 (1975)
5. Khan, S., Li, K.F., Manning, E.G., Akbar, M.M.: Solving the Knapsack Problem for Adaptive Multimedia Systems. Studia Informatica Universalis 2, 157–178 (2002)
6. Hifi, M., Michrafy, M., Sbihi, A.: A Reactive Local Search-Based Algorithm for the Multiple-Choice Multi-Dimensional Knapsack Problem. Computational Optimization and Applications 33(2-3), 271–285 (2006)
7. Cherfi, N., Hifi, M.: A Column Generation Method for the Multiple-Choice Multi-Dimensional Knapsack Problem. Computational Optimization and Applications 46(1), 51–73 (2008)
8. Cherfi, N., Hifi, M.: Hybrid Algorithms for the Multiple-Choice Multi-Dimensional Knapsack Problem. Int. J. Operational Research 5(1), 89–109 (2009)
9. Akbar, M.M., Rahman, M.S., Kaykobad, M., Manning, E.G., Shoja, G.C.: Solving the Multidimensional Multiple-Choice Knapsack Problem by Constructing Convex Hulls. Computers & Operations Research 33(5), 1259–1273 (2006)
10. Sbihi, A.: A Best First Search Exact Algorithm for the Multiple-Choice Multidimensional Knapsack Problem. Journal of Combinatorial Optimization 13(4), 337–351 (2007)
11. Parra-Hernandez, R., Dimopoulos, N.J.: A New Heuristic for Solving the Multichoice Multidimensional Knapsack Problem. IEEE Transactions on Systems Man and Cybernetics Part A-Systems and Humans 35(5), 708–717 (2005)
12. Chen, G., Wang, X., Zhuang, Z., Wang, D.: Genetic Algorithm and Its Applications. Posts & Telecommunications Press, Beijing (1995) (in Chinese)

13. Raidl, G.R.: An Improved Genetic Algorithm for the Multiconstrained 0-1 Knapsack Problem. In: Proceedings of the 1998, IEEE International Conference on Evolutionary Computation, pp. 207–211 (1998)
14. Liu, B.: Evolutionary Algorithms for the Multiple-Choice Multidimensional 0-1 Knapsack Problem. Undergraduate Thesis, School of Computer Science and Technology, University of Science and Technology of China (2008) (in Chinese)
15. Deb, K.: An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering 186(2-4), 311–338 (2000)
16. `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files`

Comparative Analysis for *k*-Means Algorithms in Network Community Detection

Jian Liu

LMAM and School of Mathematical Sciences, Peking University,
Beijing 100871, P.R. China
dugujian@pku.edu.cn

Abstract. Detecting the community structure exhibited by real networks is a crucial step toward an understanding of complex systems beyond the local organization of their constituents. Many algorithms proposed so far, especially the group of methods in the k-means formulation, can lead to a high degree of efficiency and accuracy. Here we test three k-means methods, based on optimal prediction, diffusion distance and dissimilarity index, respectively, on two artificial networks, including the widely known ad hoc network with same community size and a recently introduced LFR benchmark graphs with heterogeneous distributions of degree and community size. All of them display an excellent performance, with the additional advantage of low computational complexity, which enables the analysis of large systems. Moreover, successful applications to several real world networks confirm the capability of the methods.

Keywords: Community structure, *k*-means, Optimal prediction, Diffusion distance, Dissimilarity index.

1 Introduction

In recent years we have seen an explosive growth of interest and activity on the structure and dynamics of complex networks [1, 2]. This is partly due to the influx of new ideas, particularly ideas from statistical mechanics, to the subject, and partly due to the emergence of interesting and challenging new examples of complex networks such as the internet and wireless communication networks. Network models have also become popular tools in social science, economics, the design of transportation and communication systems, banking systems, powergrid, etc, due to our increased capability of analyzing these models. Since these networks are typically very complex, it is of great interest to see whether they can be reduced to much simpler systems. In particular, much effort has gone into partitioning the network into a small number of clusters [3–18], which are constructed from different viewing angles comparing different proposals in the literature. On a related but different front, recent advances in computer vision and data mining have also relied heavily on the idea of viewing a data set or an image as a graph or a network, in order to extract information about the important features of the images or more generally, the data sets [19, 20].

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 158–169, 2010.

In a previous paper [13], an approach to partition the networks based on optimal prediction theory proposed by Chorin and coworkers [21, 22] is derived. The basic idea is to associate the network with the random walker Markovian dynamics [23], then introduce a metric on the space of Markov chains (stochastic matrices), and optimally reduce the chain under this metric. The final minimization problem is solved by an analogy to the traditional k-means algorithm in clustering analysis [24]. Another work [7] is also along the lines of random walker Markovian dynamics, then introduce the diffusion distance on the space of nodes and identify the geometric centroid in the same framework. This proximity reflects the connectivity of nodes in a diffusion process. Under the same framework [6], a dissimilarity index for each pair of nodes is proposed, which one can measure the extent of proximity between nodes of a network and signify to what extent two nodes would like to be in the same community. They can motivate us to solve the partitioning problem also by k-means algorithms [24] under these two measures [16].

We will compare the above three algorithms in k-means formulation based on optimal prediction, diffusion distance and dissimilarity distance, respectively. From the numerical performance to the artificial networks: the ad hoc network with 128 nodes and the LFR benchmark with 1000 nodes, we can see that the three k-means methods identify the community structure during with a high degree of accuracy, while they also produce little different. Moreover, applications to four real word social networks, including the karate club network, the dolphins network, the political books network and the SFI collaboration network, confirm the differences among them.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the three type of k-means algorithms and corresponding framework. In Section 3, we apply the algorithms to six representative examples mentioned before. Finally we make the conclusion in Section 4.

2 The Framework of k-Means Algorithms for Network Partition

2.1 The k-Means Based on Optimal Prediction

In [13], a new strategy for reducing the random walker Markovian dynamics based on optimal prediction theory [21, 22] is proposed. Let $G(S, E)$ be a network with n nodes and m edges, where S is the nodes set, $E = \{e(x, y)\}_{x,y \in S}$ is the weight matrix and $e(x, y)$ is the weight for the edge connecting the nodes x and y. We can relate this network to a discrete-time Markov chain with stochastic matrix p with entries $p(x, y)$ given by

$$p(x, y) = \frac{e(x, y)}{d(x)}, \qquad d(x) = \sum_{z \in S} e(x, z), \tag{1}$$

where $d(x)$ is the degree of the node x [7, 23, 25]. This Markov chain has stationary distribution

$$\mu(x) = \frac{d(x)}{\sum_{z\in S} d(z)} \tag{2}$$

and it satisfies the detailed balance condition [13].

The basic idea in [13] is to introduce a metric for the stochastic matrix $p(x,y)$

$$\|p\|_\mu^2 = \sum_{x,y\in S} \frac{\mu(x)}{\mu(y)} |p(x,y)|^2 \tag{3}$$

and find the reduced Markov chain $\tilde{p}$ by minimizing the distance $\|\tilde{p}-p\|_\mu$. For a given partition of S as $S = \cup_{k=1}^N S_k$ with $S_k \cap S_l = \emptyset$ if $k \neq l$, let $\hat{p}_{kl}$ be the coarse grained transition probability from S_k to S_l on the state space $\mathbb{S} = \{S_1, \ldots, S_N\}$. This matrix can be naturally lifted to the space of stochastic matrices on the original state space S via

$$\tilde{p}(x,y) = \sum_{k,l=1}^N \mathbf{1}_{S_k}(x)\hat{p}_{kl}\mu_l(y), \tag{4}$$

where $\mathbf{1}_{S_k}(x) = 1$ if $x \in S_k$ and $\mathbf{1}_{S_k}(x) = 0$ otherwise, and

$$\mu_k(x) = \frac{\mu(x)\mathbf{1}_{S_k}(x)}{\hat{\mu}_k}, \qquad \hat{\mu}_k = \sum_{z\in S_k} \mu(z). \tag{5}$$

Based upon this formulation, we can find the optimal $\hat{p}_{kl}$ for any fixed partition. With this optimal form $\hat{p}_{kl}$, we further search for the best partition $\{S_1, \cdots, S_N\}$ with the given number of communities N by minimizing the optimal prediction error

$$\begin{aligned}\min_{\{S_1,\cdots,S_N\},\hat{p}_{kl}} J = \left\|\tilde{p} - p\right\|_\mu^2 &= \sum_{x,y\in S} \frac{\mu(x)}{\mu(y)} \Big[\tilde{p}(x,y) - p(x,y)\Big]^2 \\ &= \sum_{x,y\in S} \frac{\mu(x)}{\mu(y)} p^2(x,y) - \sum_{k,l=1}^N \frac{\hat{\mu}_k}{\hat{\mu}_l}\hat{p}_{kl}^2. \end{aligned} \tag{6}$$

A direct calculation shows that the minimizer of (6) is one in which $\hat{p}$ is given by

$$\hat{p}_{kl} = \sum_{x\in S_k, y\in S_l} \mu_k(x)p(x,y). \tag{7}$$

It can be checked that (7) is a stochastic matrix and $\hat{\mu}$ in (5) is an equilibrium distribution for the Markov chain on $\mathbb{S}$ with transition matrix (7). Furthermore, it is easy to see that (7) satisfies a detailed balance condition with respect to $\hat{\mu}$. A variant of k-means algorithm can be used to handle (6). Given an initial partition $\{S_k^{(0)}\}_{k=1}^N$, for $n \geq 0$, use

$$S_k^{(n+1)} = \left\{x : k = \arg\min_l D(x, S_l^{(n)})\right\} \tag{8}$$

to update the new state, where

$$D(x, S_k) = \sum_{l=1}^{N} \sum_{y \in S_l} \mu(x)\mu(y) \left(\frac{p(x,y)}{\mu(y)} - \frac{\hat{p}_{kl}}{\hat{\mu}_l} \right)^2 . \tag{9}$$

This is the theoretical basis for constructing the k-means algorithm for the community structure of complex networks in [13], which is considered to address this optimization issue which guarantees convergence towards a local minimum.

2.2 The k-means Based on Diffusion Distance

The main idea of [7, 16] is to define a system of coordinates with an explicit metric that reflects the connectivity of nodes in a given network and the construction is also based on a Markov random walk on networks. The diffusion distance $D(x, y)$ between x and y is defined as the weighted L^2 distance

$$D^2(x, y) = \sum_{z \in S} \frac{(p(x,z) - p(y,z))^2}{\mu(z)}, \tag{10}$$

where the weight $\mu(z)^{-1}$ penalize discrepancies on domains of low density more than those of high density. This notion of proximity of nodes reflects the intrinsic geometry of the set in terms of connectivity of the nodes in a diffusion process. The transition matrix p has a set of left and right eigenvectors and a set of eigenvalues $1 = \lambda_0 \geq |\lambda_1| \geq \cdots \geq |\lambda_{n-1}| \geq 0$

$$p\varphi_i = \lambda_i \varphi_i, \quad \psi_i^T p = \lambda_i \psi_i^T, \quad i = 0, 1, \cdots, n-1. \tag{11}$$

Note that $\psi_0 = \mu$ and $\varphi_0 \equiv 1$. We also have $\psi_i(x) = \varphi_i(x)\mu(x)$. Let q be the largest index i such that $|\lambda_i| > \delta|\lambda_1|$ and if we introduce the diffusion map

$$\Psi : x \longmapsto \begin{pmatrix} \lambda_1 \varphi_1(x) \\ \vdots \\ \lambda_q \varphi_q(x) \end{pmatrix}, \tag{12}$$

then the diffusion distance $D(x, y)$ can be approximated to relative precision δ using the first q non-trivial eigenvectors and eigenvalues

$$D^2(x, y) \simeq \sum_{i=1}^{q} \lambda_i^2 \Big(\varphi_i(x) - \varphi_i(y) \Big)^2 = \|\Psi(x) - \Psi(y)\|^2. \tag{13}$$

The geometric centroid $c(S_k)$ of community S_k is defined as

$$c(S_k) = \sum_{x \in S_k} \frac{\mu(x)}{\hat{\mu}(S_k)} \Psi(x), \quad k = 1, \cdots, N, \tag{14}$$

where $\hat{\mu}(S_k) = \sum_{x \in S_k} \mu(x)$ [7]. Here $c(S_k)$ may not belong to the set $\{\Psi(x)\}_{x \in S}$. In order to obtain representative centers of the communities that belong to the node set S, we introduce the diffusion center $m^D(S_k)$ by

$$m^D(S_k) = \arg\min_{x \in S_k} \|\Psi(x) - c(S_k)\|^2, \quad k = 1, \cdots, N. \tag{15}$$

2.3 The *k*-Means Based on Dissimilarity Index

In [6, 16], a dissimilarity index between pairs of nodes is defined, which one can measure the extent of proximity between nodes of a network. Suppose the random walker is located at node x. The mean first passage time $t(x, y)$ is the average number of steps it takes before it reaches node y for the first time, which is given by

$$t(x,y) = p(x,y) + \sum_{j=1}^{+\infty}(j+1) \cdot \sum_{z_1,\cdots,z_j \neq y} p(x,z_1)p(z_1,z_2)\cdots p(z_j,y). \tag{16}$$

It has been shown that $t(x, y)$ is the solution of the linear equation

$$[I - B(y)] \begin{pmatrix} t(1,y) \\ \vdots \\ t(n,y) \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \tag{17}$$

where $B(y)$ is the matrix formed by replacing the y-th column of matrix P with a column of zeros [6]. The difference in the perspectives of nodes x and y about the network can be quantitatively measured. The dissimilarity index is defined by the following expression

$$\Lambda(x,y) = \frac{1}{n-2}\left(\sum_{z \in S, z \neq x,y} \Big(t(x,z) - t(y,z)\Big)^2 \right)^{\frac{1}{2}}. \tag{18}$$

If two nodes x and y belong to the same community, then the average distance $t(x, z)$ will be quite similar to $t(y, z)$, therefore the network's two perspectives will be quite similar. Consequently, $\Lambda(x, y)$ will be small if x and y belong to the same community and large if they belong to different communities. The center $m(S_k)$ of community S_k can be defined as

$$m(S_k) = \arg\min_{x \in S_k} \frac{1}{|S_k|} \sum_{y \in S_k, y \neq x} \Lambda(x,y), \quad k = 1, \cdots, N, \tag{19}$$

where $|S_k|$ is the number of nodes in community S_k. This is an intuitive and reasonable idea for us to choose the node reached others in the same community with the minimum average dissimilarity index as the center of S_k.

3 Experimental Results

3.1 Ad Hoc Network with 128 Nodes

We apply our methods to the ad hoc network with 128 nodes. The ad hoc network is a typical benchmark problem considered in many papers [4, 6, 13–18]. Suppose we choose $n = 128$ nodes, split into 4 communities containing 32 nodes each. Assume pairs of nodes belonging to the same communities are linked with probability

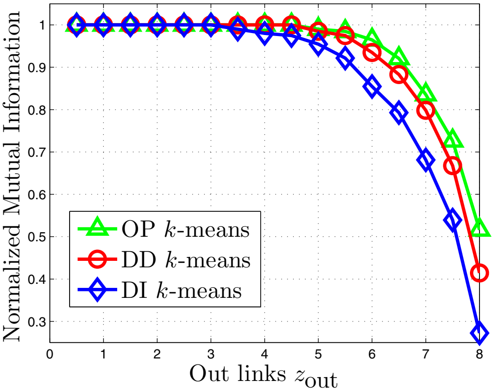


Fig. 1. Test of the three k-means algorithms on the ad hoc network with respect to the normalized mutual information defined in (21). Each point corresponds to an average over 20 graph realizations.

p_{in}, and pairs belonging to different communities with probability p_{out}. These values are chosen so that the average node degree, d, is fixed at $d = 16$. In other words p_{in} and p_{out} are related as

$$31p_{\text{in}} + 96p_{\text{out}} = 16. \tag{20}$$

Here we naturally choose the nodes group $S_1 = \{1 : 32\}, S_2 = \{33 : 64\}, S_3 = \{65 : 96\}, S_4 = \{97 : 128\}$.

Testing an algorithm on any graph with built-in community structure also implies defining a quantitative criterion to estimate the goodness of the answer given by the algorithm as compared to the real answer that is expected. For reviews of similarity measures see [4, 9–18]. In the first tests of community detection algorithms, one used a measure called fraction of correctly identified nodes [4, 13–18], but it is not well defined in some cases when a detected community is a merger of two or more real communities, so a more precise measure, which is called the normalized mutual information, is more appropriate [9–12]. It is based on defining a confusion matrix M, where the rows correspond to the real communities, and the columns correspond to the found communities. The member of M, M_{kl} is simply the number of nodes in the real community k that appear in the found community l. The number of real communities is denoted N_r and the number of found communities is denoted N_f, the sum over row k of matrix M_{kl} is denoted M_k and the sum over column l is denoted M_l . A measure of similarity between the partitions, based on information theory, is then

$$NMI(\mathbb{S}_r, \mathbb{S}_f) = \frac{-2\sum_{k=1}^{N_r}\sum_{l=1}^{N_f} M_{kl}\log(\frac{nM_{kl}}{M_k M_l})}{\sum_{k=1}^{N_r} M_k \log(\frac{M_k}{n}) + \sum_{l=1}^{N_f} M_l \log(\frac{M_l}{n})}. \tag{21}$$

We change z_{out} from 0.5 to 8 and look into the corresponding normalized mutual information produced by the three methods, which is shown in Figure 1. It seems that OP k-means performs better than the two others, especially for the more diffusive cases when z_{out} is large.

3.2 The LFR Benchmark

The LFR benchmark [10–12] is a special case of the planted partition model, in which groups are of different sizes and nodes have different degrees. The node degrees are distributed according to a power law with exponent γ; the community sizes also obey a power law distribution, with exponent β. In the construction of the benchmark graphs, each node receives its degree once and for all and keeps it fixed until the end. In this way, the two parameters p_{in} and p_{out} of the planted partition model in this case are not independent. Once the value of pin is set one obtains the value of pout and vice versa. It is more practical to choose as independent parameter the mixing parameter μ, which expresses the ratio between the external degree of a node with respect to its community and the total degree of the node. Of course, in general one may take different values for the mixing parameter for different nodes, but we will assume, for simplicity, that μ is the same for all nodes, consistently with the standard hypotheses of the planted partition model. A realization of the LFR benchmark, with 500 nodes and parameters $\mu = 0.1, \gamma = 2, \beta = 1, \langle k \rangle = 20$, corresponding to 11 communities represent by different colors is shown in Figure 2(a).

In Figure 3, we show what happens if one operates the three k-means methods on the benchmark, for $n = 1000$ and the average degree $\langle k \rangle = 20$. The four panels correspond to four pairs for the exponents (γ, β) = (2,1), (2,2), (3,1), (3,2). We have chosen combinations of the extremes of the exponents' ranges in order to explore the widest spectrum of network structures. Each curve shows the variation of the normalized mutual information with the mixing parameter μ. In general, we can infer that the k-means type methods give good results.

3.3 The Karate Club Network

This network was constructed by Wayne Zachary after he observed social interactions between members of a karate club at an American university [26]. Soon after, a dispute arose between the clubs administrator and main teacher and the club split into two smaller clubs. It has been used in several papers to test the algorithms for finding community structure in networks [3–6, 13–18]. The partitioning results are shown in Figure 2(b). The three kinds of k-means algorithms produce the same results, which seem consistent with the original structure of the network.

3.4 The Dolphins Network

The dolphins network is an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand [27]. The network was compiled from the studies of the dolphins, with ties between dolphin pairs being established by observation of statistically significant frequent

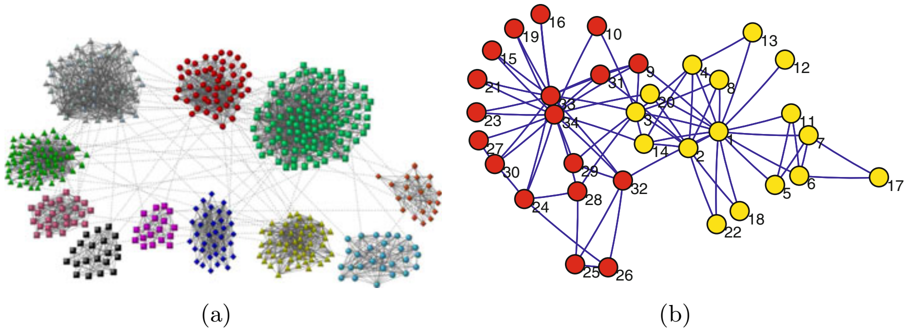


Fig. 2. (a)A realization of the LFR benchmark, with 500 nodes, corresponding to 11 communities represent by different colors. Here $\mu = 0.1, \gamma = 2, \beta = 1, \langle k \rangle = 20$. (b)The community structure for the karate club network. The three kinds of *k*-means algorithms produce the same results.

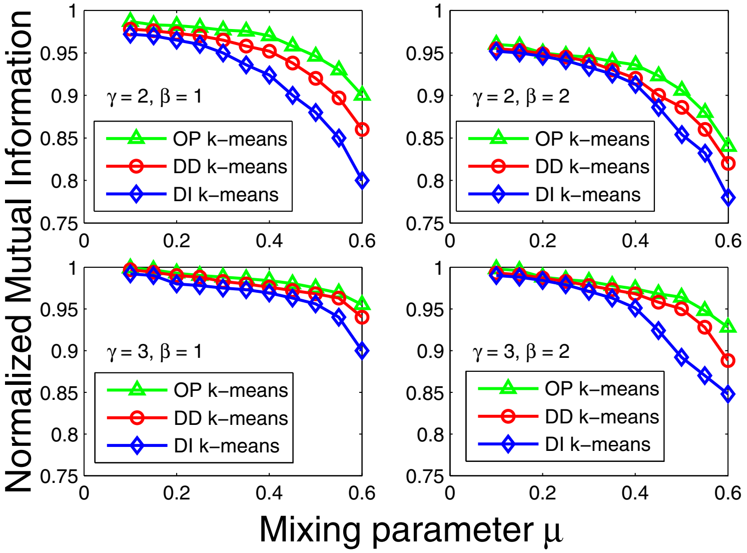


Fig. 3. Test of the three *k*-means methods on the LFR benchmark. The number of nodes $n = 1000$ and the average degree $\langle k \rangle = 20$. The results clearly depend on all parameters of the benchmark, from the exponents γ and β to the mixing parameter μ. Each point corresponds to an average over 20 graph realizations.

association [4]. The results obtained by our methods are shown in Figure 4. According to the results, the network seems splitting into two large communities by the green part and the larger one and the larger one keeps splitting into a few smaller communities, represent by different colors. The split into two groups appears to correspond to a known division of the dolphin community [28]. The

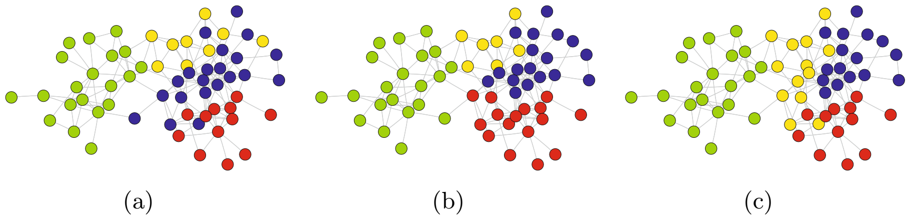


Fig. 4. The community structure for the dolphins network, corresponding 4 clusters represent by different colors. (a)OP *k*-means; (b)DD *k*-means; (c)DI *k*-means. The three kinds of *k*-means algorithms produce a little different.

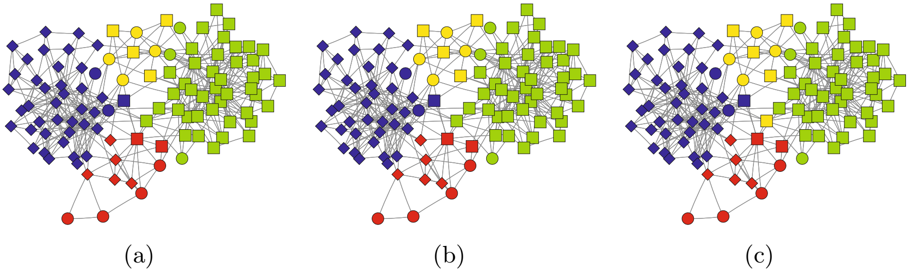


Fig. 5. The community structure for the dolphins network, corresponding 4 clusters represent by different colors. (a)OP *k*-means; (b)DD *k*-means; (c)DI *k*-means. The three kinds of *k*-means algorithms produce nearly the same.

subgroupings within the larger half of the network also seem to correspond to real divisions among the animals that the largest subpart consists almost of entirely of females and the others almost entirely of males.

3.5 The Political Books Network

We consider the network of books on politics, which are assigned based on a reading of the descriptions and reviews of the books posted on Amazon [5]. In this network the nodes represent 105 recent books on American politics bought from the on-line bookseller Amazon.com, and the edges join pairs of books that are frequently purchased by the same buyer, as indicated by the feature that customers who bought this book also bought these other books. As shown in Figure 5, nodes have been given whether they are conservative(box) or liberal(diamond), except for a small number of books which are neutral(ellipse). The results are shown in Figure 5. We find four communities denoted by different colors. It seems that one of these communities consists almost entirely of liberal books and one almost entirely of conservative books. Most of the neutral books fall in the two remaining communities. Thus these books appear to form communities of copurchasing that align closely with political views.

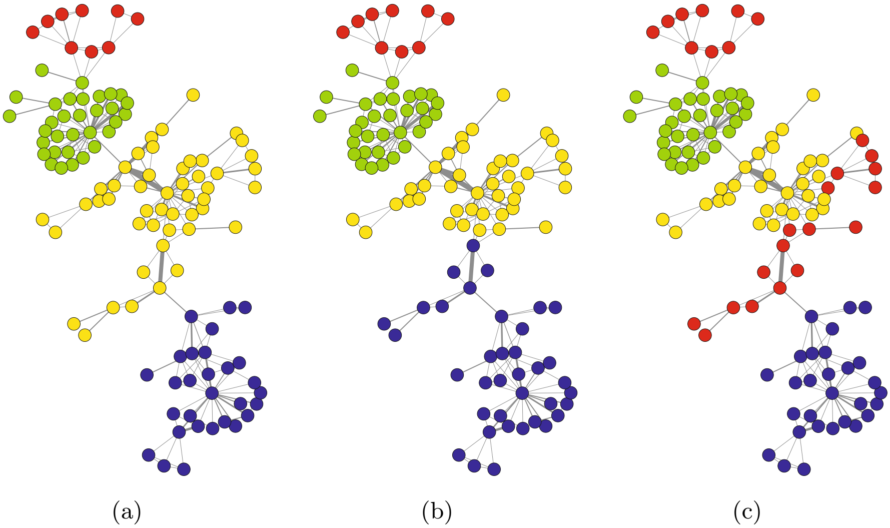


Fig. 6. The community structure for the dolphins network, corresponding 4 clusters represent by different colors. (a)OP *k*-means; (b)DD *k*-means; (c)DI *k*-means. The three kinds of *k*-means algorithms produce a little different, while the result obtained by OP *k*-means seems most reasonable.

3.6 The SFI Collaboration Network

We have applied the three *k*-means method to a collaboration network of scientists at the Santa Fe Institute, an interdisciplinary research center in Santa Fe, New Mexico [3, 6]. The 271 vertices in this network represent scientists in residence at the Santa Fe Institute during any part of calendar year 1999 or 2000, and their collaborators. A weighted edge is drawn between a pair of scientists if they coauthored one or more articles during the same time period. In Figure 6, we illustrate the results from the application of our algorithm to the largest component of the collaboration graph (which consists of 118 scientists). We find that the algorithms split the network into a few strong communities, with the divisions running principally along disciplinary lines. The community at the top of the figure (red) is the least well defined, and represents a group of scientists using agent-based models to study problems in economics and traffic flow. The next community (green) represents a group of scientists working on mathematical models in ecology, and forms a fairly cohesive structure The largest community (yellow) is a group working primarily in statistical physics, and seems sub-divided into several smaller groups. In this case, each sub-community seems to revolve around the research interests of one dominant member. The final community at the bottom of the figure (blue) is a group working primarily on the structure of RNA. It too can be divided further into smaller subcommunities, centered once again around the interests of leading members.

4 Conclusions

In this paper, we test three k-means methods, based on optimal prediction, diffusion distance and dissimilarity index, respectively, on two artificial networks, including the widely known ad hoc network with same community size and a recently introduced LFR benchmark graphs with heterogeneous distributions of degree and community size. All of them have an excellent performance, with the additional advantage of low computational complexity, which enables one to analyze large systems. They identify the community structure during iterations with a high degree of accuracy, with producing little different. Moreover, successful application to several real world networks confirm the capability among them and the differences and limits of them are revealed obviously.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant 10871010 and the National Basic Research Program of China under Grant 2005CB321704. The author thanks Professor S. Fortunato for providing the softwares on generating the LFR benchmarks, Professor M.E.J. Newman for providing the data of the karate club network, the dolphins network and the political books network, Professor Haijun Zhou and Professor Junhua Zhang for providing the data of SFI collaboration network.

References

1. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. 74(1), 47–97 (2002)
2. Newman, M., Barabási, A.L., Watts, D.J.: The structure and dynamics of networks. Princeton University Press, Princeton (2005)
3. Girvan, M., Newman, M.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA 99(12), 7821–7826 (2002)
4. Newman, M., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E 69(2), 026113 (2004)
5. Newman, M.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. USA 103(23), 8577–8582 (2006)
6. Zhou, H.: Distance, dissimilarity index, and network community structure. Phys. Rev. E 67(6), 061901 (2003)
7. Lafon, S., Lee, A.: Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, Graph Partitioning, and Data Set Parameterization. IEEE Trans. Pattern. Anal. Mach. Intel. 28, 1393–1403 (2006)
8. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. Phys. Rev. E 72, 027104 (2005)
9. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. J. Stat. Mech. 9, P09008 (2005)
10. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E 78(4), 046110 (2008)
11. Weinan, E., Li, T., Vanden-Eijnden, E.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Phys. Rev. E 80(1), 016118 (2009)

12. Li, T., Liu, J., Weinan, E.: Community detection algorithms: a comparative analysis. Phys. Rev. E 80(5), 056117 (2009)
13. Weinan, E., Li, T., Vanden-Eijnden, E.: Optimal partition and effective dynamics of complex networks. Proc. Natl. Acad. Sci. USA 105(23), 7907–7912 (2008)
14. Li, T., Liu, J., Weinan, E.: Probabilistic Framework for Network Partition. Phys. Rev. E 80, 026106 (2009)
15. Liu, J.: Detecting the fuzzy clusters of complex networks. Patten Recognition 43, 1334–1345 (2010)
16. Liu, J., Liu, T.: Detecting community structure in complex networks using simulated annealing with k-means algorithms. Physica A 389, 2300–2309 (2010)
17. Liu, J.: An extended validity index for identifying community structure in networks. In: Zhang, L., Lu, B.-L., Kwok, J. (eds.) ISNN 2010. LNCS, vol. 6064, pp. 258–267. Springer, Heidelberg (2010)
18. Liu, J.: Finding and evaluating fuzzy clusters in networks. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) Advances in Swarm Intelligence. LNCS, vol. 6146, pp. 17–26. Springer, Heidelberg (2010)
19. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intel. 22(8), 888–905 (2000)
20. Meilă, M., Shi, J.: A random walks view of spectral segmentation. In: Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics, pp. 92–97 (2001)
21. Chorin, A.J., Kast, A.P., Kupferman, R.: Unresolved computation and optimal predictions. Comm. Pure Appl. Math. 52(10), 1231–1254 (1999)
22. Chorin, A.J.: Conditional expectations and renormalization. Multi. Model. Simul. 1, 105–118 (2003)
23. Lovasz, L.: Random walks on graphs: A survey. Combinatorics, Paul Erdos is Eighty 2, 1–46 (1993)
24. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2001)
25. Chung, F.: Spectral Graph Theory. American Mathematical Society, Rhode Island (1997)
26. Zachary, W.: An information flow model for conflict and fission in small groups. J. Anthrop. Res. 33(4), 452–473 (1977)
27. Lusseau, D.: The emergent properties of a dolphin social network. Proceedings of the Royal Society B: Biological Sciences 270, 186–188 (2003)
28. Lusseau, D., Schneider, K., Boisseau, O., Haase, P., Slooten, E., Dawson, S.: The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Behavioral Ecology and Sociobiology 54(4), 396–405 (2003)

Framework for Distributed Evolutionary Algorithms in Computational Grids

Steffen Limmer and Dietmar Fey

Friedrich Alexander University Erlangen Nürnberg,
Department of Computer Science 3,
Martensstr. 3, 91058 Erlangen, Germany
{steffen.limmer,dietmar.fey}@informatik.uni-erlangen.de

Abstract. In the recent years an increasing number of computational grids have been built, providing an unprecedented amount of computational power. Based on their inherent parallelism, Evolutionary Algorithms are well suited for distributed execution in such grids. Unfortunately, there are several challenges concerning the usage of a grid infrastructure (e.g. the synchronization and submission of jobs and file transfer tasks). In this paper we present a new framework which makes a Globus based grid easily accessible for Evolutionary Algorithms and takes care of the parallelization. The usability is demonstrated by the example of an Evolutionary Algorithm for the Traveling Salesman Problem.

Keywords: Distributed Evolutionary Algorithm, Computational Grid, Globus Toolkit, Traveling Salesman Problem.

1 Introduction

Evolutionary Algorithms (EAs) are a widely-used strategy [1] for solving numerous optimization problems. The main argument for the use of EAs is that they produce acceptable results in a rather short time, compared to exact methods. But, depending on the problem size, EAs can require a compute time of several days or longer, too. For that reason a lot of effort was put in designing parallel versions of EAs ([2], [3]). There are several libraries and toolboxes that enable the execution of parallel EAs, like ParadisEO [4], GAlib [5], JGAP [6] or DREAM [7]. These build on different technics for the parallel execution, namely MPI, PVM, PThreads, Peer-to-Peer or Client-Server principle over TCP/IP.

During the past few years more and more computational grids [8] emerged by combining available computing resources, in particular in the academic domain. This happens for the purpose of providing users with a high amount of computing power via a higher utilization of the envolved resources. These resources are generally spread over multiple administrative domains. A grid middleware acts as a link between the resources. The most popular grid middleware is the Globus Toolkit 4 [9].

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 170–180, 2010.

The communication costs in a grid may be high when compared to clusters, but since the subprocesses of a parallel EA are mostly independent from each other, grids pose an adequate platform for parallel EAs. However, the realization requires a considerable effort. Particularly the following issues bear challenges at the design and implementation:

- The exact form of the parallelization to use
- The submission of jobs and file transfer tasks to the underlying grid infrastructure
- Fault detection and handling
- Identification of appropriate resources for the execution of subtasks (monitoring and scheduling)
- Synchronization of subtasks

These tasks turned out to be in general independent from the actual problem to be solved. We present a framework which makes the huge computational power of computational grids easily accessible for the application of EAs. The framework allows the execution of distributed EAs in a Globus based grid environment and frees the user from dealing with the above-mentioned problems. Thus the user can concentrate on his optimization problem, leaving the grid aspects to the framework.

This paper is organized as follows. Section 2 discusses some related work. Section 3 descibes the parallelization methods used by the framework and Section 4 gives an overview about the framework. Next, benchmark results are presented and finally, Section 6 concludes, and outlines some future work.

2 Related Work

Two related projects resemble ours the most: ParadisEO-CMW [10] and JG^{2}A [11]. The first one is an extension to the previously mentioned ParadisEO framework. Thus it provides a rich set of parallelization strategies. But it is only intended for grids consisting of multiple condor pools combined via flocking [12]. This form of grid is not as popular as Globus based grids and it is questionable if it can be seen as a grid in the commonly accepted meaning at all.

JG^{2}A uses the Globus Toolkit 4 grid middleware, but requires Condor as underlying scheduler on the different sites. This makes it inflexible because in general other local resource managers than Condor are used at different computing sites of a grid and these sites may be under different administrative control (which makes it hard to enforce the deployment of Condor on all those sites). Another drawback of JG^{2}A is that it implements only two simple forms of parallelization: (a) multiple execution of the same EA with different settings (e.g. mutation rate or crossover rate) and (b) distribution of the fitness evaluations for the individuals of a population.

To the best of our knowlege, there exists no previous framework or library for distributed EAs in generic Globus based grids.

3 The Parallelization

The common parallelization strategies proposed in the literature can be classified in three different approaches:

1. Parallelization of a single fitness evaluation
2. Distribution of all the fitness evaluations
3. The island model

The first approach is very problem dependent and so it is outside of our focus, since the framework should be as generic as possible. However, it can be used in conjuction to the framework's parallelization over the grid. The user is responsible for providing an executable for the fitness evaluation (more details are given in section 4.1) and that might be parallelized for example with OpenMP or MPI.

The second approach bases on the parallel computation of the fitness values of the several individuals while the rest of the EA is executed in serial by a master. That kind of parallelization is provided by the framework. Obviously this algorithm yields the same results as a pure serial algorithm. Nevertheless this form of parallelization makes only sense if the calculation of a fitness value requires a large computation time. Otherwise the time spent for the overhead emerging from the parallelization (file transfers, waiting periods of pending jobs at a local batch system and so on) may turn out to be bigger than the time saved.

The third variant is significantly more coarse grained than the previous ones. It employs multiple populations (the "islands"). A start population is decomposed in several equally sized subpopulations and for each of these subpopulations a serial EA is run. These runs are carried out simultaneously. The intention is that in this way the search space is examined more widely spread than with only one

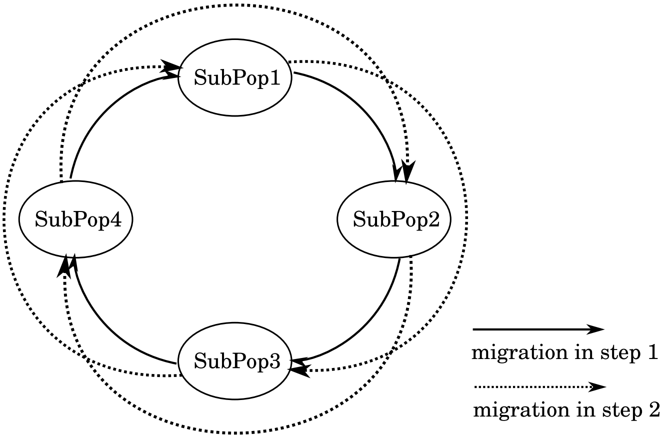


Fig. 1. Four subpopulations in ring topology. In the first migration step individuals are migrated from one subpopulation to its respective neighbor. In the second migration step individuals are migrated to the after next subpopulation and so on.

population, reducing the risk of getting stuck in a local optimum. At multiple points of time good individuals migrate from one subpopulation to another where they may dominate the other individuals if their fitness is good enough. Thereby the subpopulations are supposed to converge towards the global optimum.

According to several papers (e.g. [13] and [14]) for many problems the multi population algorithm, even executed in serial, performs as well as the usual algorithm without decomposition or even better. That means it produces comparably good or better results in the same amount of computation time - the decomposition has no harmful impact on the quality of the algorithm.

For the implementation of such a multi population algorithm some decisions have to be taken: When shall the migrations take place, which individuals should migrate and between which subpopulations should one perform the migration. For the framework we use a simple, yet efficient strategy that works well for most problems where a multi population approach is applicable: two migration steps are always seperated by a fixed number g of generations. So we apply a synchronous migration strategy. Optionally at every migration step the n best individuals of the subpopulation or k random individuals out of the n best are migrated. g, n and if necessary k are set by the user. According to [14], g should be five times the subpopulation size and n should be rather small. Regarding the question between which populations the migrations should be performed, we decided to use a ring topology for the subpopulations. After the first g generations migrations take place between every population and its respective neighbor population. After further g generations there is migration to the population after next and so on, see Figure 1. Since this parallelization approach can be seen as the most suitable for the application in grids, we will concentrate on it in the subsequent sections of this paper.

4 The Framework

The framework handles the execution of an EA following the above-mentioned strategy, using the resources available in the grid. For this purpose it reads at first a user supplied description of the optimization problem and creates corresponding random start populations that are written in separate files. Subsequently it distributes the files together with further required files (see Section 4.1) to available resources and starts there the EAs on the subpopulations. This suboptimizations write their resulting populations inclusive the fitness values to files which are then collected by the framework. Then the framework produces new subpopulations through the described migration on the actual populations and writes them again to files to start the next optimization step. The following sections describe more precisely how the framework works.

4.1 The User Interface

The user has to provide two files for the execution of an optimization: the problem description and an executable. The description file specifies how the individuals of the EA look like, i.e. of how many parameters are they composed, which type the parameters have (floating point, integer, boolean or permutation) and if necessary in which ranges their values can lie. Additionally, some settings for the parallel EA must be defined in the description file: the number of subpopulations, the size of the subpopulations, the number of migrants in an optimization step, how many optimization steps to execute and the number of generations that are computed in one optimization step.

The executable has to carry out the optimization on one sub pupulation. Therefore it must be able to read in the start population from a file. After a certain number of generations it has to write the resulting population with the corresponding fitness values in an output file. For the actual optimization an arbitrary serial EA is sufficent. For the implementation one can employ one of the numerous existing libraries for EAs, e.g. the ones mentioned in the introduction of this paper. The name of the executable and if necessary further required files have to be specified in the already mentioned description file.

When switching from a serial algorithm with one big population to the parallel algorithm with multiple smaller subpopulations one should ensure there is still enough genetic diversity under the individuals despite the smaller population sizes. Depending on the problem it may be advisable to deploy appropriate mechanisms to avoid pre maturing, e.g. adaptive mutation.

4.2 Data and Job Management

The framework is designed for grids based on the Globus Toolkit 4. Globus can be referred to as the de-facto standard for grid middlewares. It provides the basic services for job execution and data transfers in a grid. Most of its services are implemented as web services. These web services are running on every resource in the grid. A resource can here be anything from a single workstation to a whole cluster (in the latter case the services are running on the head node).

Generally there is also a so called GridFTP server running on every resource. It is responsible for data transfers. Apart from a corresponding client, Globus contains a GridFTP library for C. This is used for the implementation of the framework's data management. The tasks of the data management are the creation of working directories on the resources that are used for the computations, the transfer of required in- and output files and finally the deletion of the working directories.

Responsible for the start, monitoring and, if necessary, abortion of jobs in a grid is the web service WS-GRAM. It accepts a job description in which, among others, the executable and its arguments are defined and starts the job accordingly. Jobs are either started directly on the machine where the WS-GRAM is running or are passed to a cluster's local batch system. For every suboptimization that has to be performed the framework creates a job description which is submitted to an available WS-GRAM and monitored, with help of the Java library Java CoG Kit [15].

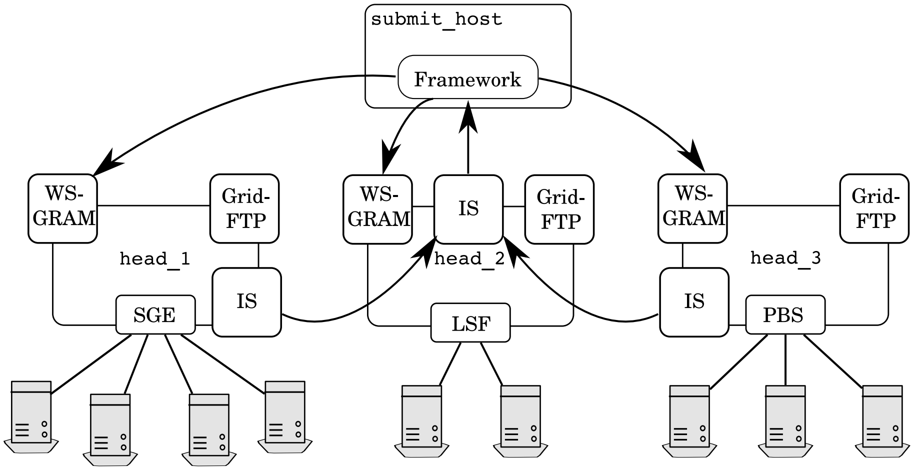


Fig. 2. Sample configuration of a grid, consisting of three clusters. *head_1*, *head_2* and *head_3* are the head nodes of these clusters. All of them run the Globus services (in particular the WS-GRAM, the Index Service (IS) and GridFTP server). The node *submit_host* hosts the framework which is able to send jobs to the clusters via WS-GRAM and retrieve informations from the IS.

4.3 Scheduling

Figure 2 shows a sample grid infrastructure, consisting of three hosts *head_1*, *head_2* and *head_3*. Each of these hosts runs a GridFTP server and a WS-GRAM service. Furthermore the hosts are head nodes of clusters with different local batch systems (Sun Grid Engine, Torque/PBS and Load Sharing Facility). The framework is running on another host, *submit_host*, and is able to submit jobs to each of the other three hosts respectively to the three clusters.

A difficult question is the following: to which host should one submit a job, so that it is finished as soon as possible? First the framework has to determine which resources are currently available. This information is derived from the Index Service (IS), a web service which is part of the monitoring system of Globus. Multiple ISs on different resources can be connected hierarchically to collect all their informations in one central point. In Figure 2, the ISs on *head_1* and *head_3* send their informations to the IS on *head_2*, from which all the information about the grid can thus be queried. But the IS supplies only limited informations, e.g. which hosts are available, which batch systems are running on them and how many free execution nodes are accessible over them. Via additional monitoring software like Ganglia, it can provide further information, e.g. the installed operating system, the amount of free memory or the processor load of the head nodes, on which the IS is installed. But it yields no further information on the execution nodes of the cluster, which are responsible for the actual job execution. Another problem besides the limited informations on the available resources is that the local batch systems perform scheduling, which is unpredictable in advance and which might conflict with a cluster overlapping scheduling strategy. For example, the batch system could keep a job for some reason in its queue although enough nodes are free.

For that reasons the framework has only limited options for scheduling. We decided to use the following strategy: in the first optimization step the jobs are distributed as equally as possible on the available resources, clusters with free nodes are preferred. The suboptimizations work all on populations of the same size and compute the same number of generations. Thus the computation time on two equally powerful resources should be the same. When a suboptimization requires more time on one resource than on another, it is assumed that the first one is less powerful than the second. The framework collects the runtimes of the jobs on the different machines in the first optimization step in a table. In the second step, the jobs are no longer evenly split but according to the table. The table is updated and used for the scheduling decisions in every subsequent optimization step.

Additionally, the user has the possibility to influence the scheduling through statements in the description file of the optimization. There he can explicitly exclude hosts from the scheduling or specify that jobs are only scheduled on hosts with a specific batch system.

4.4 Fault Handling

In a loosely coupled system like a grid the permanent availability of resources can hardly be guaranteed. The failure rate may be rather high. Hence an adequate fault handling is required. The framework continuously monitors the execution states of all submitted jobs. If an error occurs during some step of the execution (e.g. the transfer of the input files), it is first attempted to repeat that step, since it could be just a temporary failure. If the error occurs again, a rescheduling is performed and the job is submitted to another resource. If this fails too, it is assumed that there is a fault in the optimization itself (maybe faulty statements in the description file or an erroneous executable) and the job along with the whole optimization will be aborted. But backups from the subpopulations are created continuously in a persistend database, so the optimization can be resumed at any point of time. This also protects against failures of the framework itself.

In general the populations are at no time held completely in the memory but nearly entirely in the database that is used for the backups. This minimizes the framework's memory demand.

5 Evaluation

For the practical testing of the framework we used the well known Traveling Salesman Problem (TSP). As the EA for solving the TSP we slightly modified the algorithm from [16]. The algorithm encodes the individuals (TSP tours) in path representation (i.e. as permutations) and applies the 2opt local optimization instead of random mutation. For the crossover it introduces a new operator dubbed Greedy Subtour Crossover (GSX). In our tests the algorithm did yield very good results compared to classical mutation and crossover operators for the path representation (like PMX, ERX, OX1, Insertion Mutation or Exchange Mutation [17]). As a benchmark instance we used the TSP *fnl4461* with 4461 cities from the popular library for TSP test problems TSPLIB [18]. In 1994 David Applegate et al. [19] were the first who found an optimal solution for *fnl4461.*

Our setup looks as follows: For the serial executions we used a workstation of our department (with a 2.7 GHz Intel Core i7 quad-core processor and 12 GB memory), in the following referenced as `faui32z`. For the parallel executions another host of our department served as submit host. It has access to two remote clusters (each consisting of machines of different computational power) of two different institutions via Globus. We designate these clusters `gwdg` and `lrz` in the following. The `gwdg` cluster is comperatively highly utilized, resulting in rather long latencies for job executions. The other cluster has a lower utilization and executes jobs generally immediately. To get an impression of the computational power of these three systems, we executed several runs of 500 generations of the EA for the *fnl4461* problem with a population size of 100 on each of them.

Table 1 summarizes the runtime measurements. The runtimes on the clusters do not include the waiting times in the local batch systems. The measurements show that the EA runs in avarage most quickly on `faui32z` and the computational power of a single host of `lrz` is rather weak. Since the clusters contain different types of machines, the runtimes of different executions can vary heavily.

We ran the EA in serial on `faui32z` and in parallel with two different setups of the island model in our test grid, see Table 2 for details. For every optimization method we executed five runs. Figure 3 shows the average progressions for the three methods.

Table 1. Results of the runtime measurements for the serial execution of 500 generations of the TSP EA with 100 individuals for the *fnl4461* problem. All runtimes are given in seconds.

	Runtime (s)		
System	Min.	Avrg.	Max.
faui32z	448	593	693
gwdg	468	643	937
lrz	921	1564	2773

Table 2. The three different EA setups that were used for testing

System	#populations	population size	#generations	Migration	#migrants
faui32z	1	2000	15000	-	0
gwdg & lrz	20	100	15000	after 500 gen.	2
gwdg & lrz	40	50	15000	after 500 gen.	2

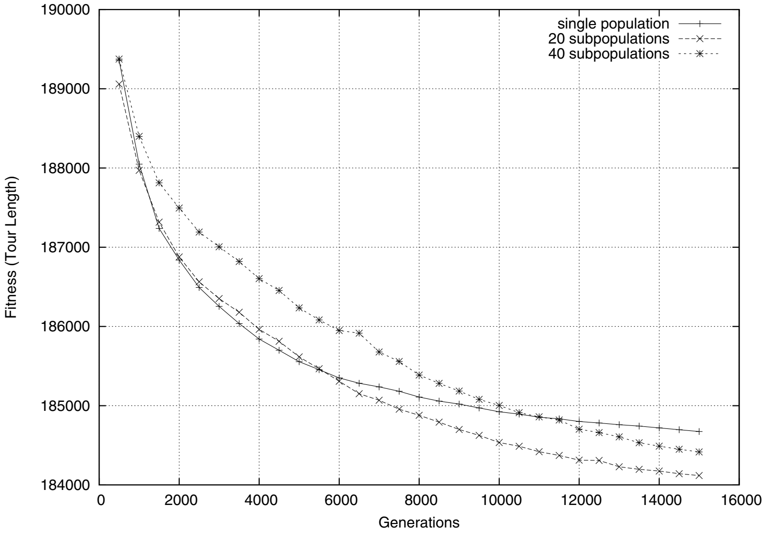


Fig. 3. Average evolutions of the different optimization methods

The algorithm with 20 subpopulations, each consisting of 100 individuals, did yield the best results. Interestingly the single population algorithm is the best up to approx. 5500 generations. Beyond that point it becomes worse than the 20 subpop algorithm and from generation 11000 on even worse than the 40 subpop algorithm which did yield the second best results. The reason for this behavior is probably that the single population algorithm examines the search space more

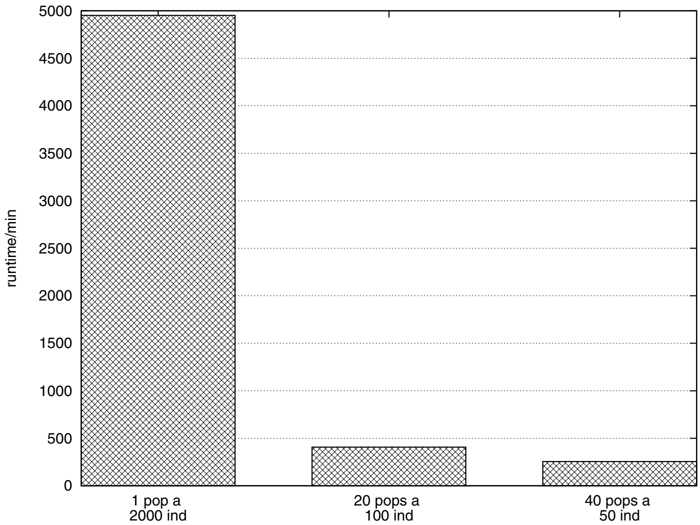


Fig. 4. The avarage runtimes of the different optimization methods

directionally than the multi population algorithms and thus gets stuck in an area of local optima.

The best tour during our tests was produced with the 20 subpop algorithm and has a length of 183979.57 in the euclidean norm. David Applegate et al. used the rounded euclidean norm when they found the optimal solution. Their optimal tour has a length of 182566. Our best solution found has a length of 183734 in rounded euclidean norm, which is only 0.64% worse than the optimum.

Now let us address the runtimes of the different optimizations. Figure 4 shows the average runtimes. The serial algorithm requires approx. 12.2 times as much time as the 20 subpop algorithm and approx. 19.4 times as much as the 40 subpop algorithm. If one considers the overhead of the parallel execution (especially latencies caused by the local batch systems) and the higher computational power of `faui32z` compared to the cluster nodes, the results make a good impression.

6 Conclusion

We have presented a new framework with the intended purpose to exploit existing grid infrastructures based on Globus for the distributed execution of Evolutionary Algorithms. Thus optimizations can benefit from the large available computational power of these grids with only little extra effort. A benchmark illustrates the benefit, in terms of runtime and solution quality, that can be achieved by the framework.

It is planned to add other forms of parallelization (e.g. other topologies or asynchonous migration) to make the framework applicable for more classes of optimization problems. Another intended extension is support for multi objective optimizations. Furthermore the scheduling strategy may be further optimized.

References

1. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
2. Alba, E., Troya, J.M.: A Survey of Parallel Distributed Genetic Algorithms. Complexity 4, 31–52 (1999)
3. Talbi, E.-G., Mostaghim, S., Okabe, T., Ishibuchi, H., Rudolph, G., Coello, C.A.C.: Parallel Approaches for Multiobjective Optimization. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization. LNCS, vol. 5252, pp. 249–372. Springer, Heidelberg (2009)
4. Cahon, S., Melab, N., Talbi, E.-G.: ParadisEO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics. Journal of Heuristics 10(3), 357–380 (2004)
5. Wall, M.: GAlib: A C++ Library of Genetic Algorithm Components, Massachusetts Institute of Technology, http://lancet.mit.edu/ga/dist/galibdoc.pdf
6. JGAP - Java Genetic Algorithms Package, http://jgap.sourceforge.net/

7. Arenas, M.G., Collet, P., Eiben, A.E., Jelasity, M., Merelo, J.J., Paechter, B., Preuß, M., Schoenauer, M.: A Framework for Distributed Evolutionary Algorithms. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 665–675. Springer, Heidelberg (2002)
8. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
9. A Globus Primer - Or, Everything You Wanted to Know about Globus, but Were Afraid To Ask, http://www.globus.org/toolkit/docs/4.0/key/GT4_Primer_0.6.pdf
10. Cahon, S., Melab, N., Talbi, E.-G.: An Enabling Framework for Parallel Optimization on the Computational Grid. In: Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005), vol. 2, pp. 702–709 (2005)
11. Ramírez, M.A., Bernal, A., Castro, H., Walteros, J.L., Medaglia, A.L.: JG2A: A Grid-Enabled Object-Oriented Framework for Developing Genetic Algorithms. COPA (2009)
12. Liu, C., Zhao, Z., Liu, F.: An Insight into the Architecture of Condor - A Distributed Scheduler. In: CNMT International Symposium on Computer Network and Multimedia Technology (2010)
13. Voigt, H.-M., Born, J., Santibañez-Koref, I.: Modelling and Simulation of Distributed Evolutionary Search Processes for Function Optimization. In: Schwefel, H.-P., Männer, R. (eds.) PPSN 1990. LNCS, vol. 496, pp. 373–380. Springer, Heidelberg (1991)
14. Starkweather, T., Whitley, D., Mathias, K.: Optimization using Distributed Genetic Algorithms. In: Schwefel, H.-P., Männer, R. (eds.) PPSN 1990. LNCS, vol. 496, pp. 176–185. Springer, Heidelberg (1991)
15. Laszewski, G.v., Gawor, J., Lane, P., Rehn, N., Russell, M., Jackson, K.: Features of the Java Commodity Grid Kit. In: Concurrency and Computation: Practice and Experience, vol. 14, pp. 1045–1055. John Wiley & Sons, Ltd., Chichester (2002)
16. Sengoku, H., Yoshihara, I.: A Fast TSP Solution using Genetic Algorithm. In: Information Processing Society of Japan 46th Nat'l. Conv. (1993)
17. Larrañaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I., Dizdarevic, S.: Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. Artificial Intelligence Review 13, 129–170 (1999)
18. Reinelt, G.: TSPLIB - A Traveling Salesman Problem Library. ORSA Journal on Computing 3, 376–384 (1991)
19. Applegate, D., Bixby, R., Chvatal, V., Cook, W.: Finding Cuts in the TSP (A Preliminary Report). Research Report, Rice University (1994)

A Review of Tournament Selection in Genetic Programming

Yongsheng Fang[1] and Jun Li[2]

[1] Department of Finance, Anhui Polytechnic University,
Wuhu City, Anhui, P.R. China
[2] Department of Information and Computing Science, Anhui Polytechnic University,
Wuhu City, Anhui, P.R. China
fbmy@ahpu.edu.cn

Abstract. This paper provides a detailed review of tournament selection in genetic programming. It starts from introducing tournament selection and genetic programming, followed by a brief explanation of the popularity of the tournament selection in genetic programming. It then reviews issues and drawbacks in tournament selection, followed by analysis of and solutions to these issues and drawbacks. It finally points out some interesting directions for future work.

1 Motivation

Accurately and objectively identifying the economic situation of students is the most important step of funding poor students in tertiary educational organizations. For effectively implementing the funding policies, it requires a workable, reasonable and scientifically sound assessment approach. Campus OneCard, the platform of Digital Campus, accumulates large volume of complicated consumption data from students. How to mine these consumption data in order to help accurately and objectively identifying the economic situation of students becomes a hot research topic. Genetic programming (GP) [1], one of the metaheuristic search methods in Evolutionary Algorithms (EAs) [2], is based on the Darwinian natural selection theory. Its special characters make it a very attractive algorithm for many real world problems, including data mining, finical prediction, image recognition, and symbolic regression. Therefore, we intend to use GP in the student consumption data mining project. To be able to apply GP in mining the student consumption data successfully, it is necessary to understand the status of the current research results of the most important operator — selection — in GP.

Selection is a key factor of affecting the performance of EAs. Commonly used parent selection schemes in EAs include fitness proportionate selection [3], ranking selection [4], and tournament selection [5]. The most popular parent selection method in GP is tournament selection. Therefore, this paper focus on tournament selection and gives a detailed review of tournament selection in GP in order to provide a thorough understanding of selecting behaviour, features and drawbacks in tournament selection, as well as possible directions for our research work.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 181–192, 2010.

This paper is organised as follows: Section 2 briefly introduces GP; Section 3 introduces standard tournament selection and explains why it is so population in GP, as well as formal modellings which describe its sampling and selection behaviour; Section 4 presents issues related to the sampling strategy in standard tournament selection, together with corresponding analyses and clarifications; Section 5 discusses drawbacks related to selection pressure control in standard tournament selection, followed by corresponding solutions; Section 6 concludes the paper and shows possible directions for future work.

2 Genetic Programming

GP is a technique of enabling a Genetic Algorithm (GA) [3] to search a potentially infinite space of computer programs, rather than a space of fixed-length solutions to a combinatorial optimisation problem. These programs often take the form of Lisp symbolic expressions, called *S-expressions*. The idea of applying GAs to S-expressions rather than combinatorial structures is due originally to Fujiki and Dickinson [6, 7], and was brought to prominence through the work of Koza [1]. The S-expressions in GP correspond to programs which a user seeks to adapt to perform some pre-specified tasks. The fitness of an S-expression may therefore be evaluated in terms of how effectively it performs this task. GP with individuals in S-expressions is referred to as tree-based GP.

GP also has other categorises based on the representations of individual, for instance, linear structure GP [8–11] and graph-based GP [12–15]. Linear structure GP is based on the principle of register machines thus programs can be linear sequences of instructions. Graph-based GP is suitable for the evolution of highly parallel programs which effectively reuse partial results [16].

Briefly, to fulfill a certain task, GP starts with a randomly initialised population of programs. It evaluates each program's performance using a fitness function, which generally compares the program's outputs with the target outputs on a set of training data ("fitness cases"). It assigns each program a fitness value, which in general represents the program's degree of success in achieving the given task. Based on the fitness values, it then chooses some of the programs using a stochastic selection mechanism, which consists of a selection scheme and a selection pressure control strategy. After that, it produces a new population of programs for the next generation from these chosen programs using crossover (sexual recombination), mutation (asexual), and reproduction (copy) operators. The search algorithm repeats until it finds an optimal or acceptable solution, or runs out of resources. A much more comprehensive field guide to GP can be found in [16].

3 Tournament Selection

The standard tournament selection randomly **samples** k individuals with replacement from the current population of size N into a tournament of size k and **selects** the one with the best fitness from the tournament. Therefore, the selection process of the tournament selection consists of two steps: sample and then select. Commonly used tournament sizes are 2, 4 and 7. In general, since the standard breeding process in GP produces one offspring by applying mutation to one parent and produces two offspring by applying crossover to two parents, the total number of tournaments needed is N at the end of generating all individuals in the next generation.

Tournament selection has the following features compared with other selection schemes:

- Its selection pressure can be adjusted easily.
- It is simple to code, efficient for both non-parallel and parallel architecture [17].
- It does not require sorting the whole population first. It has the time complexity $O(N)$ [18].

The last two features make tournament selection very population in GP: 1) GP is very computationally intensive, requiring a parallel architecture to improve its efficiency; 2) It is common to have millions of individuals in a population when solving complex problems [19], thus sorting a whole population is really time consuming. The linear time complexity in tournament selection is therefore very attractive for GP.

As the selection process of standard tournament selection consists of sampling and selecting, there are a large number of research focusing on different sampling and selecting strategies [20–24]. In addition to these practical studies, there are many theoretical studies that model and compare the selection behaviour of a variety of selection schemes [18, 25–29], as well as many dedicated theoretical studies on standard tournament selection [17, 30, 31].

Based on the concept of takeover time [18], Bäck [25] compared several selection schemes, including tournament selection. He presented the selection probability of an individual of rank j in one tournament for a minimisation task[1], with an implicit assumption that the population is wholly diverse (i.e., every individual has distinct fitness value), as:

$$N^{-k}((N-j+1)^k - (N-j)^k) \tag{1}$$

In order to model the expected fitness distribution after performing tournament selection in a population with a more general form, Blickle and Thiele extended the selection probability model in [25] to describe the selection probability of individuals with the same fitness. The model is quite abstract although it is quite elegant. They defined the worst individual to be ranked 1st and introduced the *cumulative fitness distribution*, $S(f_j)$, which denotes the number of individuals with fitness value f_j or worse. They then calculated the selection probability of individuals with rank j as:

$$\left(\frac{S(f_j)}{N}\right)^k - \left(\frac{S(f_{j-1})}{N}\right)^k \tag{2}$$

[1] Therefore the best individual is ranked 1st.

In order to demonstrate the computational savings in backward-chaining evolutionary algorithms, Poli and Langdon [31] calculated the probability that one individual is not sampled in one tournament as $1 - \frac{1}{N}$, then consequently the expected number of individuals not sampled in any tournament as:

$$N\left(\frac{N}{N-1}\right)^{-ky} \tag{3}$$

where y is the total number of tournaments required to form an entire new generation.

In order to illustrate that selection pressure in standard tournament selection is insensitive to population size in general for populations with a more general situation (i.e., some programs have the same fitness value and therefore have the same rank), Xie *et al.* [32] presented a sampling probability model that any program p is sampled at least once in $y \in \{1, ..., N\}$ tournaments as:

$$1 - \left(\left(\frac{N-1}{N}\right)^{N}\right)^{\frac{y}{N}k} \tag{4}$$

and a selection probability model that a program p of rank j is selected at least once in $y \in \{1, ..., N\}$ tournaments as:

$$1 - \left(1 - \frac{\left(\frac{\sum_{i=1}^{j}|S_i|}{N}\right)^k - \left(\frac{\sum_{i=1}^{j-1}|S_i|}{N}\right)^k}{|S_j|}\right)^y \tag{5}$$

where $|S_j|$ is the number of programs of the same rank j.

4 Issues Related to Sampling Strategy

There are two commonly recognised issues in standard tournament selection. These two issues are closely related to each other because they are both caused by the *sampling with replacement* scheme in standard tournament selection.

4.1 Multi-Sampled Issue

One issue is that because individuals are sampled with replacement, it is possible to have the same individual sampled multiple times in a tournament, which is referred as *multi-sampled issue* [33].

To address this issue, Xie *et al.* [33] analysed the other form of tournament selection described in [5] and termed it as *no-replacement tournament selection*. The no-replacement tournament selection samples individuals into a tournament without replacement, that is, it will not return a sampled individual back to the population immediately thus no individual can be sampled multiple times into the same tournament. After the winner is determined, it then returns all individuals of the tournament to the population. Therefore, no-replacement tournament selection is a solution to the multi-sampled issue.

Xie *et al.* [33] gave mathematical models describing the sampling and selection behaviour in no-replacement tournament selection as follows: if D is the event that an arbitrary program is drawn or sampled in a tournament of size k, the probability of D is:

$$P(D) = \frac{k}{N} \tag{6}$$

For a particular program $p \in S_j$, if $E_{j,y}$ is the event that p is selected at least once in $y \in \{1, ..., N\}$ tournaments, the probability of $E_{j,y}$ is:

$$P(E_{j,y}) = 1 - \left(1 - \frac{1}{|S_j|}\left(\frac{\binom{\sum_{i=1}^{j}|S_i|}{k}}{\binom{N}{k}} - \frac{\binom{\sum_{i=1}^{j-1}|S_i|}{k}}{\binom{N}{k}}\right)\right)^y \tag{7}$$

They then used three selection pressure measures, namely *loss of program diversity* [29], *selection frequency* [27], and their own measure *selection probability distribution*, to compare no-replacement tournament selection with standard one based on simulations of four populations with different fitness rank distributions. Their simulation results showed that there exist few difference between the two tournament selection schemes. To further investigate underlying reasons, they presented a model to describe the relationship between population size, tournament size, and confidence level in order to answer the following question:

> “ for a given population of size N, if we keep sampling individuals with replacement, then after how many sampling events, will we have a certain level of confidence that there will be duplicates amongst the sampled individuals?”

They finally showed that for common tournament sizes 4 or less, it is not expected to see any duplicates in anything except very small populations. Even for tournament size 7, it is not expected to see duplicates for populations less than 200. For most common and reasonable settings of tournament sizes and population sizes, the multi-sampled issue *seldom* occurs in standard tournament selection. Further, since duplicated individuals do not necessarily influence the result of a tournament when the duplicates have worse fitness values than other sampled individuals, the probability of significant difference between standard tournament selection and no-replacement tournament selection will be even smaller. Therefore eliminating the multi-sampled issue in standard tournament selection is very unlikely to significantly change the selection performance. They concluded that the multi-sampled issue generally is not crucial to the selection behaviour in standard tournament selection.

4.2 Not-Sampled Issue

The other issue is that because individuals are sampled with replacement in standard tournament selection, it is also possible to have some individuals not sampled at all when using small tournament sizes, which is referred as *not-sampled issue* [34].

This issue was illustrated initially through an experimental work by Gathercole [35]. He showed the selection frequency of each individual and the likelihoods of not-selected and not-sampled individuals in tournament selection of different tournament sizes through 1000 simulations on a sample population of size 50. In his simulation, only one child is produced by crossover or mutation, thus the total number of tournaments required to generate the next entire population is a function of the crossover rate, the mutation rate and the population size, instead of being just the same as the population size. His experimental results are interesting and useful, however it is not clear whether the sample population was fully diverse or not. This issue was then theoretically described by [31].

Two earliest known attempts to address the not-sampled issue are *unbiased tournament selection* by Sokolov and Whitley [24] and *fully covered tournament selection* by Xie [36]. A further detailed analysis of the issue was given recently by Xie *et al.* [34].

In [24], Sokolov and Whitley believed that the potential of better individuals not getting chosen for recombination due to the random sampling is the bias presented in standard tournament selection. Therefore, they developed their unbiased tournament selection that "lines up two different permutations of the population and performs a pairwise comparison" with a constraint, which forces compared individuals to be distinct. Consequently, their method can ensure that every individual is sampled at least once. Tournament size 2 was used to test the unbiased tournament selection on three problems: one with permutation-based solution representation and two under bit encodings. Although the advantage of a generational GA using the unbiased tournament selection varied for different population sizes on the three problems, they concluded that the impact of the bias is significant, and the unbiased tournament selection provides better performance than other selection methods, including standard tournament selection, a rank based selection and fitness proportionate selection.

In [36], Xie also mentioned that since the sampling behaviour in standard tournament selection was random, the individual with bad fitness could be selected multiple times and the individual with good fitness could never be selected. He presented fully covered tournament selection which excludes the individuals that have been selected for next selection to ensure every individual have an equal chance to participate tournaments. He tested the fully covered tournament selection on two symbolic regression problems and concluded that the method is effective, implying that not-sample issue is worth of addressing.

Later, Xie *et al.* provided a detailed algorithm termed *round-replacement tournament selection* which seems based on the fully covered tournament selection and extended the analysis of the not-sampled issue. The algorithm is as follows:

```
Initialise an empty population $T$
while need to generate more offspring do
    if population size $< k$ then
        Refill: move all individuals from the temporary population $T$ to the population $S$
    end if
    Sampling $k$ individuals without replacement from the population $S$
    Select the winner from the tournament
    Move the $k$ sampled individuals into the temporary population $T$
    return the winner
end while
```

They described that in the round-replacement tournament selection any program will be sampled exactly k times during the selection phase thus there is no need to model the sampling probability. They gave a model to describe the selection probability as follows: for a particular program $p \in S_j$, if W_j is the event that p wins or is selected in a tournament of size k, the probability of W_j is:

$$P(W_j) = \frac{\sum_{n=1}^{k} \frac{1}{n} \binom{|S_j|-1}{n-1} \binom{\sum_{i=1}^{j-1} |S_i|}{k-n}}{\binom{N}{k}} \tag{8}$$

They used two measures, namely loss of program diversity and selection probability distribution, to compare round-replacement tournament selection with standard one based on simulations of three populations with different fitness rank distributions. Their simulation results showed that there exist some difference between the two tournament selection schemes. They then tested the effectiveness of the round-replacement tournament selection on the even-6-parity, a symbolic regression, and a binary classification problems using three different tournament sizes, namely 2, 4 and 7. However, the experimental results showed that the improvement of the round-replacement tournament selection is statistically significant *only* when the tournament size is 2 for the symbolic regression and the binary classification problems but practically the differences are small. They finally concluded that although there are some different selection behaviour in the round-replacement tournament selection comparing with the standard tournament selection, the different selection behaviour leads to better GP search results only when tournament size 2 is used for some problems; overall solving the not-sampled issue does not appear to significantly improve a GP system for the given tasks; and the not-sampled issue in the standard tournament selection is not critical.

The clarifications from the literature are very useful. The results show us that in order to improve tournament selection, simply tackling different sampling with or without replacement strategies is not sufficient and should not be our research focus.

5 Drawbacks Related to Selection Pressure Control

5.1 Finer Level Selection Pressure Control

Controlling selection pressure in tournament selection could be done by changing tournament size. However, tournament size can only be an integer number. This drawback limits the ability of tournament selection to influence selection pressure at a coarse level. In order to make tournament selection be able to tune selection pressure at a fine level, Goldberg and Deb [18] presented an alternative tournament selection, which uses an extra probability p. When conducting a tournament between two individuals, the individual with higher fitness value can be selected as a parent with the probability p, while the other is with the probability $1 - p$. By setting p between 0.5 and 1, it is possible to tune the selection pressure continuously between the random selection and the tournament selection with tournament size two. Recently, Hingee and Hutter [37] showed that every probabilistic tournament is equivalent to a unique polynomial ranking selection scheme.

Another attempt to address this drawback is *a fine grained tournament selection* by Filipović *et al.* [21] in the context of GAs for a plant location problem. They argued that standard tournament selection does not allow precise setting of the balance between exploration and exploitation [30]. In their fine grained tournament selection method, the tournament size is not fixed but close to a pre-set value. They claimed that the fine grained tournament selection makes the ratio between exploration and exploitation be able to be set very precise, and that the method solves the plant location problem successfully.

5.2 Automatic and Dynamic Selection Pressure Control

In general, the larger the tournament size, the higher the selection pressure; by using different tournament sizes, the selection pressure can be changed to influence the convergence of the genetic search process. However, it will not work as we expected due to two existing drawbacks during population convergence.

One drawback is when groups of programs having the same or similar fitness values, the selection pressure between groups increases regardless of the given tournament size configuration, resulting in "better" groups dominating the next population and possibly causing premature convergence [38]. The other drawback is when most of programs in population have the same fitness value, the selection behaviour effectively becomes random [39]. Therefore, tournament size itself alone is not adequate for controlling selection pressure.

In fact, according to [38], these drawbacks are part of a more general issue: the evolutionary learning process itself is very dynamic. At some stages, it requires a fast convergence rate (i.e., high parent selection pressure) to find a solution quickly; at other stages, it requires a slow convergence rate (i.e., low parent selection pressure) to avoid being confined to a local maximum. These requirements could be achieved by changing tournament size dynamically in standard tournament selection. However, standard tournament selection is not aware of the dynamic requests. In order to pick correct tournament size, it should collaborate with an extra component that can reveal the underlying dynamics and determine the requests.

The known attempts to implicitly and partially address the drawbacks include *bucket tournament selection* by Luke and Panait [22] and *clustering tournament selection* by Xie *et al.* [40]. They are implicit and partial solutions to the drawbacks because the original goals of the alternatives were not to solve the drawbacks. Instead, the buckets tournament selection is used to apply lexicographic parsimony pressure on parent selection for problem domains where few individuals have the same fitness, and the clustering tournament selection is a consequence of a fitness evaluation saving algorithm which clusters a population using a heuristic called fitness-case-equivalence.

Fortunately, later on, Xie *et al.* re-analysed and re-test their clustering tournament selection explicitly for the drawbacks and adopted two additional population clustering methods, including phenotype-based[2] and genotype-based[3] [38]. They concluded that different population clustering methods have different impact on different problems but the clustering tournament selection can automatically and dynamically adjust selection pressure along evolution as long as the population is clustered properly. They further presented mathematical models and simulation results to explain why the clustering tournament selection is effective [32].

From these research results, we think that clustering tournament selection is a promising direction to improve tournament selection as it is an automatically biased parent selection scheme that is needed by the dynamic evolutionary process: when most of the population are of worse fitness ranks and evolution encounters a danger of missing good individuals, it tends to increase selection bias to better individuals, hoping to drive the population to promising regions quickly; when the population tends to converge to local optima and evolution encounters a danger of losing genetic material, it tends to decrease selection bias to better ones, hoping to keep the population diverse. We also think it would be challenging to choose or develop an appropriate population clustering method for a given problem but it is certainly an interesting research topic.

6 Conclusions

This paper reviewed the current status of research of tournament selection in GP. It clearly showed that different sampling with or without replacement strategies have limited impact on the selection behaviour in standard tournament selection. It also pointed out a promising research direction that is about population clustering and clustering tournament selection. As the student consumption data is very complex, we should follow these research results to implement and/or to improve the tournament selection operator in order to obtain good mining results.

[2] A population is clustered based on fitness value.

[3] A population is clustered based on exact program structure and content.

References

1. Koza, J.R.: Genetic Programming — On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
2. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
3. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
4. Grefenstette, J.J., Baker, J.E.: How genetic algorithms work: A critical look at implicit parallelism. In: Schaffer, J.D. (ed.) Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 20–27. Morgan Kaufmann Publishers, San Francisco (1989)
5. Brindle, A.: Genetic algorithms for function optimisation. PhD thesis, Department of Computing Science, University of Alberta (1981)
6. Fujiko, C.: An evaluation of holland's genetic operators applied to a program generator. Master's thesis, University of Idaho (1986)
7. Fujiko, C., Dickinson, J.: Using the genetic algorithm to generate lisp source code to solve the prisoner's dilemma. In: Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application, pp. 236–240. Lawrence Erlbaum Associates, Inc., Mahwah (1987)
8. Banzhaf, W., Nordin, P., Keller, R., Francone, F.D.: Genetic Programming – An Introduction. In: On the Automatic Evolution of Computer Programs and its Applications. Morgan Kaufmann, San Francisco (1998)
9. Ferreira, C.: Gene expression programming: a new adaptive algorithm for solving problems. Complex Systems 13, 87 (2001)
10. Oltean, M.: Multi-expression programming. Technical report, Babes-Bolyai Univ., Romania (2006)
11. Oltean, M., Grosan, C.: Evolving evolutionary algorithms using multi expression programming. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J.T. (eds.) ECAL 2003. LNCS (LNAI), vol. 2801, pp. 651–658. Springer, Heidelberg (2003)
12. Handley, S.: On the use of a directed acyclic graph to represent a population of computer programs. In: Proceedings of the 1994 IEEE World Congress on Computational Intelligence, Orlando, Florida, USA, vol. 1, pp. 154–159. IEEE Press, Los Alamitos (1994)
13. Hirasawa, K., Okubo, M., Katagiri, H., Hu, J., Murata, J.: Comparison between Genetic Network Programming (GNP) and Genetic Programming (GP). In: Proceedings of the 2001 Congress on Evolutionary Computation, vol. 2, pp. 1276–1282 (2001)
14. Miller, J.F., Job, D., Thomson, P.: Cartesian genetic programming. In: Poli, R., Banzhaf, W., Langdon, W.B., Miller, J., Nordin, P., Fogarty, T.C. (eds.) EuroGP 2000. LNCS, vol. 1802, pp. 131–132. Springer, Heidelberg (2000)
15. Poli, R.: Parallel distributed genetic programming. Technical report, School of Computer Science, University of Birmingham (1996)
16. Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming (2008) (With contributions by J. R. Koza), http://lulu.com, and freely available at http://lulu.com
17. Miller, B.L., Goldberg, D.E.: Genetic algorithms, tournament selection, and the effects of noise. Technical Report 95006, University of Illinois at Urbana-Champaign (1995)

18. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. Foundations of Genetic Algorithms, 69–93 (1991)
19. Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G.: Genetic programming IV: Routine Human-Competitive Machine Intelligence. Kluwer Academic, Dordrecht (2003)
20. Harik, G.R.: Finding multimodal solutions using restricted tournament selection. In: Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 24–31. Morgan Kaufmann, San Francisco (1995)
21. Filipović, V., Kratica, J., Tošić, D., Ljubić, I.: Fine grained tournament selection for the simple plant location problem. In: 5th Online World Conference on Soft Computing Methods in Industrial Applications, pp. 152–158 (2000)
22. Luke, S., Panait, L.: Lexicographic parsimony pressure. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 829–836 (2002)
23. Matsui, K.: New selection method to improve the population diversity in genetic algorithms. In: Proceedings of 1999 IEEE International Conference on Systems, Man, and Cybernetics, pp. 625–630. IEEE, Los Alamitos (1999)
24. Sokolov, A., Whitley, D.: Unbiased tournament selection. In: Proceedings of Genetic and Evolutionary Computation Conference, pp. 1131–1138. ACM Press, New York (2005)
25. Back, T.: Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In: Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 57–62 (1994)
26. Blickle, T., Thiele, L.: A comparison of selection schemes used in evolutionary algorithms. Evolutionary Computation 4(4), 361–394 (1997)
27. Branke, J., Andersen, H.C., Schmeck, H.: Global selection methods for SIMD computers. In: Fogarty, T.C. (ed.) AISB-WS 1996. LNCS, vol. 1143, pp. 6–17. Springer, Heidelberg (1996)
28. Miller, B.L., Goldberg, D.E.: Genetic algorithms, selection schemes, and the varying effects of noise. Evolutionary Computation 4(2), 113–131 (1996)
29. Motoki, T.: Calculating the expected loss of diversity of selection schemes. Evolutionary Computation 10(4), 397–422 (2002)
30. Blickle, T., Thiele, L.: A mathematical analysis of tournament selection. In: Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 9–16 (1995)
31. Poli, R., Langdon, W.B.: Backward-chaining evolutionary algorithms. Artificial Intelligence 170(11), 953–982 (2006)
32. Xie, H., Zhang, M., Andreae, P.: Another investigation on tournament selection: modelling and visualisation. In: Proceedings of Genetic and Evolutionary Computation Conference, pp. 1468–1475 (2007)
33. Xie, H., Zhang, M., Andreae, P., Johnston, M.: An analysis of multi-sampled issue and no-replacement tournament selection. In: Proceedings of Genetic and Evolutionary Computation Conference, pp. 1323–1330. ACM Press, New York (2008)
34. Xie, H., Zhang, M., Andreae, P., Johnston, M.: Is the not-sampled issue in tournament selection critical? In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 3711–3718. IEEE Press, Los Alamitos (2008)
35. Gathercole, C.: An Investigation of Supervised Learning in Genetic Programming. PhD thesis, University of Edinburgh (1998)
36. Xie, H.: Diversity control in GP with ADFs for regression tasks. In: Zhang, S., Jarvis, R.A. (eds.) AI 2005. LNCS (LNAI), vol. 3809, pp. 1253–1257. Springer, Heidelberg (2005)

37. Hingee, K., Hutter, M.: Equivalence of probabilistic tournament and polynomial ranking selection. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 564–571 (2008)
38. Xie, H., Zhang, M., Andreae, P.: Automatic selection pressure control in genetic programming. In: Proceedings of the Sixth International conference on Intelligent Systems Design and Applications, pp. 435–440. IEEE Computer Society Press, Los Alamitos (2006)
39. Gustafson, S.M.: An Analysis of Diversity in Genetic Programming. PhD thesis, University of Nottingham (2004)
40. Xie, H., Zhang, M., Andreae, P.: Population clustering in genetic programming. In: Collet, P., Tomassini, M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) EuroGP 2006. LNCS, vol. 3905, pp. 190–201. Springer, Heidelberg (2006)

Genetic Algorithm for Mixed Chinese Postman Problem

Hua Jiang[1], Lishan Kang[1], Shuqi Zhang[2], and Fei Zhu[3]

[1] Computer School, Wuhan University, Wuhan, Hubei, China
jianghua_whu@163.com
[2] Computer School, Donghu College, Wuhan University, Wuhan, Hubei, China
[3] China Construction Bank Wuhan Audition Branch, Wuhan, China
Jianghua_whu@163.com

Abstract. Mixed Chinese Postman Problem (Mixed CPP) is a NP-Complete problem. This problem has many applications, including route optimization, analyzing interactive system and flow design. This paper presents a genetic algorithm to solve this problem, which has got a good result. Problem-specific genetic operators were designed to keep every individual in population is reasonable and to improve the exploration efficiency. Data experiments show the algorithm is efficient and is better than the existing approximation algorithms.

Keywords: genetic algorithm. Mixed Chinese Postman Problem. NP-Complete.

1 Introduction

A Postman who whishs to travel along every road to deliver mails and come back to postoffice will find the shortest route. This problem is proposed by Kwan Mei-ko in 1962[1], was called Chinese Postman Problem (CPP). If the graph is mixed (that is, contains both directed and undirected edges), then the problem has been shown by Papadimitriou to be NP-Complete[2]. Frederickson has presented a algorithm for mixed Chinese postman problem which has worst-case bound of 5/3[3]. Corberan has presented A GRASP (Greedy Random Adaptive Search Procedures) heuristic for the mixed postman problem[4].

Genetic algorithm was proposed by John Holland[5]. It uses techniques inspired by evolutionary biology such as inheritance, mutation, selection and crossover to find exact or approximate solutions to optimization and search problems. Practice proves that Genetic Algorithm is very effective for NP-complete problem in combinatorial optimization. Jun Byung-Hyun had gave a genetic algorithm for mixed CPP but the algorithm was proven that it can't keep the solution is valid. This paper presents a new algorithm for mixed postman problem and gets good results.

2 Definition

A mixed connected multi-graph $G(V, E, A)$ consists of a set of V of vertices, a multiset E of undirected edges, and multiset A of directed arcs. Each edge e_i=<u, v> is a

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 193–199, 2010.

undirected edge which connects two vertices *u*, *v* in *V* and has positive weight $w(e_i)$. An arc a=<*t*, *h*> is a directed edge from *t* to *h*, both vertices in *V*. We call *t* the tail of arc <*t*, *h*>, and *h* the head of arc <*t*, *h*>. Each arc has positive weight $w(a_i)$. In multi-graph we shall consider graph may have more than one edge or arc between two vertices.

To solve the mixed postman problem we need to find a shortest closed path or circuit that visits every edges and arcs of a mixed connected multi-graph.

The degree of a vertex is the number of edges and arcs incident on the vertex. The outdegree is the number of arcs out of the vertex and the indegree is the number of arcs directed into the vertex.

For the Chinese postman problem on a mixed graph, if and only if 1) the degree of each vertex is even, and 2) for each vertex in the graph, the indegree of arcs adjacent to it is equal the outdegree, then a tour exists[6]. Thus a method of solution for the mixed CPP consists of finding a minimum-cost augmentation of the graph that satisfies the condition 1) and 2) and then identifying the route over the augmented graph. Namely to find set

$E' \subset E,\ A' \subset A$

Minimize:

$$W = \sum w(e') + \sum w(a'), e' \in E', a' \in A',$$

Subject to:

$$\forall v_i \in V, d(v_i) \in E_v, d(v_i)_{in} = d(v_i)_{out}$$

$\bar{G} = G \cup G' = (V, E \cup E', A \cup A')$ is the solution of the Chinese postman problem on a mixed graph. E_v is even set and G' is minimum-cost augmentation of the graph.

3 Genetic Algorithm for Mixed CPP

3.1 The Strategy of Evolving the Solution

We must keep the solution is valid during the evolving process just because it is difficult to find a eulerian cycle in mixed graph. A valid solution contains original graph and its augmentation and satisfies the condition 1) and 2). It is hard to recover whether an individual solution fails to satisfy the condition 1 or 2 in the evolving process. So we must design the genetic operators which can keep the solution valid.

The genetic algorithm in this paper constructed valid solution individuals in initializing the population and keep them valid in evolving process by specially designed genetic operators. So each genetic operator can concentrate on finding the solution with the least cost and the whole algorithm can work efficiently. It is worthwhile even though it took extra effort in designing the genetic operators.

3.2 The Genome of the Individuals

In our algorithm we use the graph $G = (V, E \cup E', A \cup A')$ which contains the route that may be a solution as the individual. We use vector *Edges*=<$edge_0$, $edge_1$, $edge_2$..., $edge_n$,... > to code the multi-graph *G*. Each element in *Edges* is an undirected edge or a directed arc in *G*. The date structure of edge_i is defined as below:

```
struct Type_of_edge
{
  int  VBegin;
  int  VEnd;
  int  Length;
  int  IfEA;
  int  IfAugment;
};
```

The variable *VBegin* is the vertex from which the Euler route go into $edge_i$ and the variable *Vend* is the end vertex of the $edge_i$ in Euler tour. Variable *Length* is the cost of $edge_i$. The $edge_i$ is undirected edge if variable *IfEA* is 0 and is directed arc if *IfEA* is 1. If Variable *IfAugment* equals 1 mean $edge_i$ is belong to the augmentation, and equal 0 mean $edge_i$ is a part of original graph.

3.3 Initialize the Population

As previously stated, we must initialize the population with a group of valid solutions. We can get one valid solution by greedy algorithm but can get only one. A homogeneous population is bad for evolve. So we designed an initialize which can random produce different individuals to fill the population. The initialize algorithm is specified below:

```
Algorithm Individual initialize :
  Input : mixed multigraph $G = (V, E \cup A)$ ;
  Output : mixed multigraph $\bar{G} = (V, E \cup E', A \cup A')$
Procedure :
  1. make multiset $S = E \cup A$ ;
  2. Select two vertices as vstart and vhead, then construct the route from vstart to vhead, and remove the edges and arcs in this route from set S.
  3. while ($S \neq \varphi$)
     If all edges and arcs in set S incident on the vstart then we select one edge or arc <x, y> randomly. Else we randomly select edge or arc <x, y> which doesn't incident on the vstart. Now we construct route from vstart to x randomly and let vhead=y. The edges and arcs appeared in the new constructed route will be removed from the set S.
  4. Make a route from vhead to vstart, now we get an Eulerian cycle.
  5. Set the variable IfRepeat in edges or arcs which repeat in the Eulerian cycle to 1. These edges and arcs form the augmentation of the graph.
End
```

3.4 Crossover

We must make sure the offspring is valid solution when we designed crossover operator. We faced similar problem when we tried to solve travel salesman problem by genetic algorithm. Some crossover operators for TSP, such as PMX(Partially Matched Crossover), will amend the offspring individuals after crossover to make them keep valid. Here we crossover the parents to produce the new solutions follow the same routine.

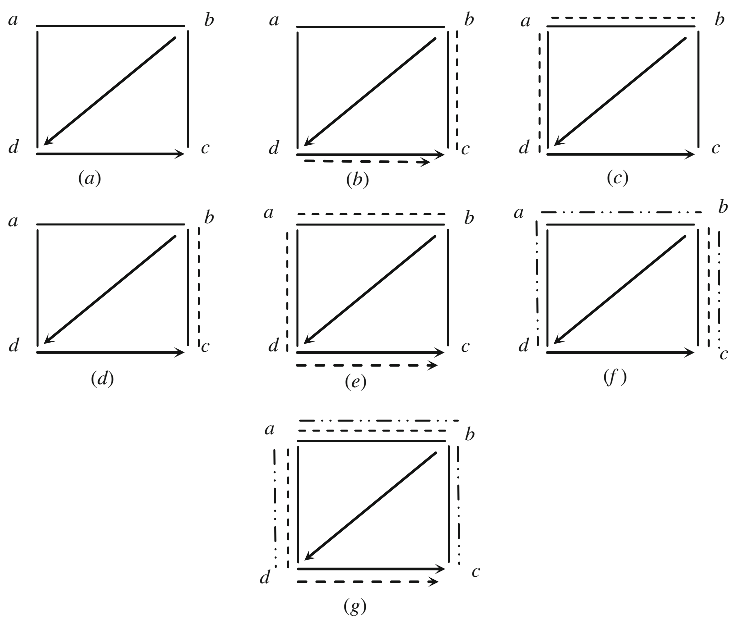


Fig. 1. Figure (*a*) is original graph. Figure (*b*) and (*c*) are two parents. They exchanged augmented edge between vertices *d* and *c* and the get two new solutions shown in figure (*d*) and (*e*). Figure (*f*) and (*g*) show the results after adjustment.

Our crossover operator swaps the augmented edges and arcs which connect two adjacent vertices *v1* and *v2* between two parents. And then the crossover operator will make some additional copies of edges and arcs and orient some edges, so as to make the indegree and ourdegree of each vertex equal. Fig.1 shows the example of crossover.

3.5 Mutation

We designed two different mutate operators in this genetic algorithm. One change the augmented graph and the other change the orientation of the original edge in Eulrain tour.

The augmented graph contains a set of path which comprise of one or more copies of edges and arcs. If we find another path which connect its two end vertices to replace one augmented path the solution is still valid. This kind mutation was shown in Fig.2.

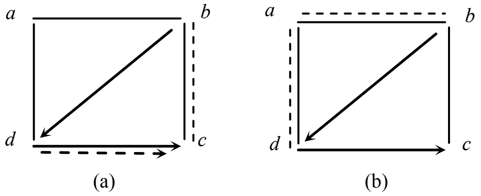


Fig. 2. Figure (a) shows the original solution which had path d→c→b in augmented graph. Figure (b) shows the offspring one after mutation which replaced the augmented graph with path d→a→b.

The edges in mixed graph were oriented in the Eulerian tour. We can also mutate the individuals by changing the direction of the edges. Considering the mutation is local search operator we would change only one edge's direction if there more than one edges between two vertices.

The outdegree and indegree of two vertices didn't equal after we change the direction of the edge <v_1, v_2> and $|d_+(v_i)-d_-(v_i)|=2$. We must find two paths in graph from v_1 to v_2 and add them into augmented graph to make outdegree and indegree of v_1 and v_2 equal again. Fig.3 shows the example.

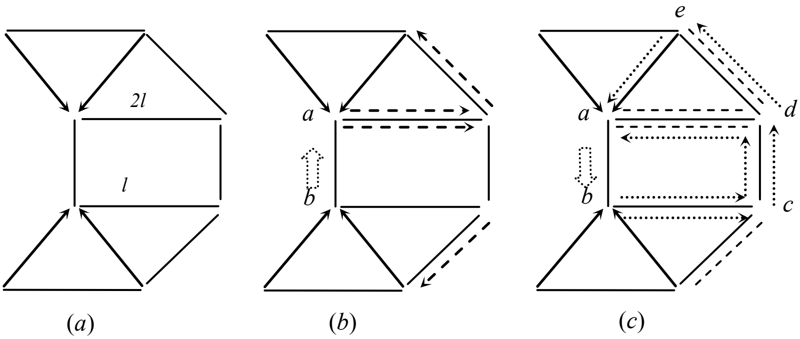


Fig. 3. Figure (a) shows the original solution. Figure (b) shows the augmented graph in which the edge <a, b> is oriented from b to a. The mutation operator reversed its direction and added paths b→c→d→a and b→c→d→e→a in augmented graph as shown in figure (c).

3.6 Redundancy Elimination

The total weights increased after crossover and mutation because these two operators add path to augmented graph to make the solution keep valid. It seems that it is hard to get optimal solution because the weight of the offspring is always more than the weight of the parents. But we can notice that there are some redundant paths in the augmented graph. If we find a circle in the augmented graph and remove it the solution is still valid. Fig.4 shows the redundancy elimination.

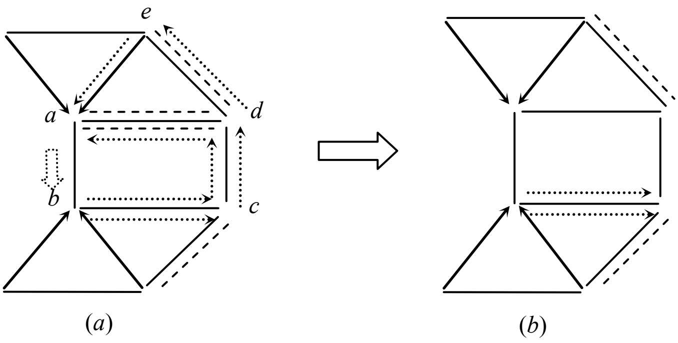


Fig. 4. Figure (*a*) shows the offspring individual in Fig.3. We can find the circles *a→d→e→a* and *c→d→a→d→c*. Edges and arcs in these circles may be removed as redundancies. Figure (*b*) shows the final result.

4 Experiments

We produced some mixed connected multi-graph randomly as benchmark problem and selected MIXED1 algorithm which proposed by Edmonds and Jonson[7] for comparison to test our algorithm.I

In tests the size of population was 30. We employed tournament selection method and stopped the procedure when the best solution had been at a standstill during 1000 iterations.The size of the problems and test results were listed in Table.1.

We can see that the GA won 6 times in 10 tests and broke even with Mixed1 in 4 problems. Figure.5 shows the fourth problem listed in Table.1 and the results from the two algorithms.

Algorithm MIXED1 treated all arcs as edge at first and made an augmented graph to make the degree of all vertices even. Then it modified the augmented graph to make the indegree and outdree of all vertices equal. Almost all approximation algorithm will settled the mixed postman problem in two steps as MIXED1 so that they

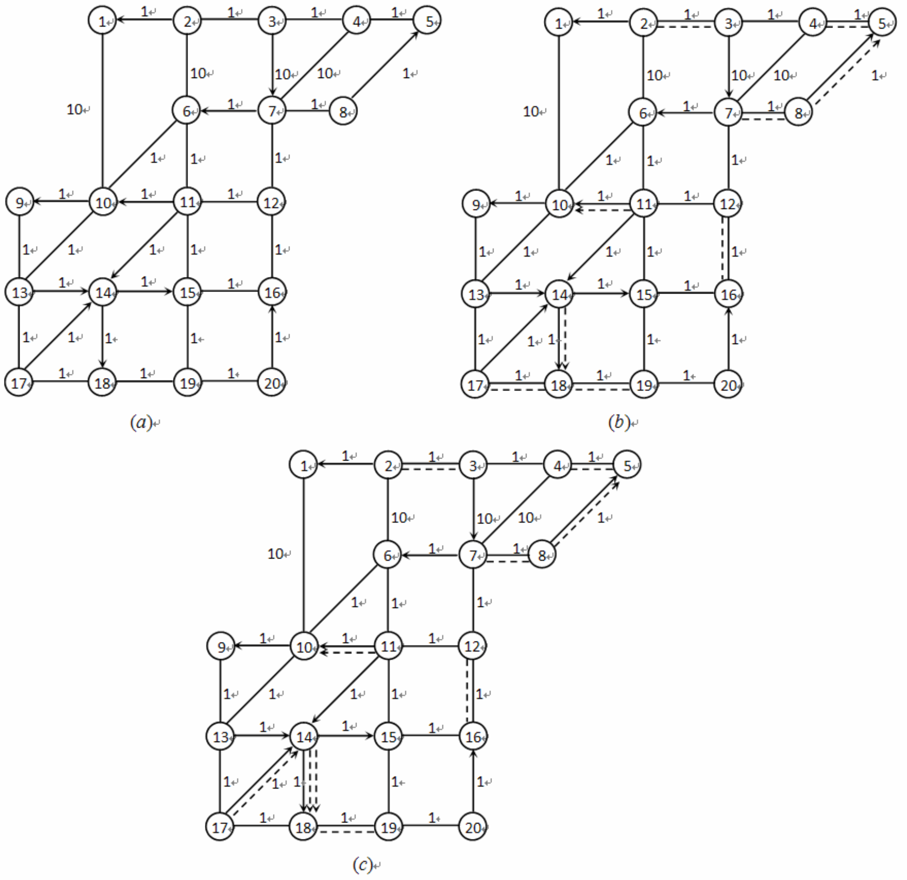


Fig. 5. Figure (a) shows the original graph. Figure (b) shows the GA's result. Figure (c) shows the result from MIXED1.

Table 1. Size of the problems and test results

Problem	Number of arcs	Number of edges	MIXED1(cost)	GA(cost)
1	4	8	35	35
2	8	8	74	68
3	6	8	36	36
4	12	21	79	78
5	21	30	182	176
6	17	15	85	64
7	9	12	48	48
8	11	4	32	32
9	16	3	59	47
10	10	6	43	38

would find the local optimal solution. In the example shown in Fig.5 MIXED1 found the local best. The GA proposed in this paper found the global optimal solution depends on the GA's robust.

5 Conclusion

The result of the experiments presents that our genetic algorithm for mixed postman problem is efficient. The evolve strategy which initialized the population with valid solutions and kept all individuals valid during mutation and crossover guarantee the success of finding global optimal solution.

We also noticed that two genetic operators produced massive redundant edges and arcs and elimination operator wasted too much time to remove these redundancies. The farther research will focus on designing more efficiency operators to improve the algorithm's performance.

References

1. Kuan, M.-g.: Graphic programming using odd or even points. Chinese Mathematics, 237–277 (1962)
2. Papadimitriou, C.H.: On the Complexity of Edge Traversing. Journal of the ACM (JACM) archive 23(3), 544–554 (1976)
3. Fredericson, G.N., Hecht, M.S., Kim, C.E.: Approximation Algorithms for some Routing Problems. SIAM Journal of Computing, 178–193 (1978)
4. Corberan, A., Marti, R., Scnchis, J.M.: A GRASP heuristic for the mixed Chinese Postman Problem. European Journal of Operational Research 142, 70–80 (2002)
5. Holland, J.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. The MIT Press, Cambridge (1975)
6. Harary, F.: Graph Theory. Addison-Wesley, Reading (1970)
7. Edmonds, J., Johnson, E.L.: Matching Euler tours and the Chinese postman. Mathematical Programming 5, 88–124 (1973)

Hybrid Evolutionary Algorithms Design Based on Their Advantages

Guangming Lin[1], Sundong Liu[1], Fei Tang[1], and Huijie Wang[2]

[1] Shenzhen Institute of Information Technology, Shenzhen, China
[2] School of Information Technology and Electrical Engineering Shenzhen University
lingm@sziit.com.cn

Abstract. The search direction and the search step size are two important factors which affect the performance of algorithms. In this paper, we combine Particle Swarm Optimization (PSO) with EP to form two new algorithms namely PSOEP and SAVPSO. The basic idea is to introduce the search direction to the mutation operator of EP and use lognormal self-adaptive strategy to control the velocity of PSO to guide the individual at a faster convergence rate. All of these algorithms are compared to each other with respect to the similarities and differences of their basic components, as well as their performances on seven benchmark problems. Our experimental results show that PSOEP performs much better than all other version of EPs, and SAVPSO performs much better than PSO for the benchmark functions.

Keywords: Hybrid Evolutionary Algorithms, Search Direction, Search Step Size Control.

1 Introduction

Although evolutionary programming (EP) was first proposed as an artificial intelligence approach [1], it has been applied successfully to many numerical and combinatorial optimization problems [2, 3, 4, 18]. Optimization using EP can be explained by two major steps:

1. Mutate the solutions in the current population, and
2. Select the next generation from the mutated and the current solutions.

These two steps can be regarded as a population-based version of the classical generate-and -test method [5], where mutation is used to **generate** the new solutions (offspring) and the selection is used to **test** which of the newly generated solutions should survive to the next generation. Formulating EP as a special case of the generate-and-test method establishes a bridge between EP and other search algorithms, such as evolution strategies (ES), genetic algorithms (GA), simulated annealing (SA), tabu search (TS), Particle Swarm Optimization (PSO) and others, and thus facilitates cross-fertilization amongst different research areas.

One of the difficulties with EP, while solving some of the multi-model optimization problems, it converges very slowly towards an optimal or near-optimal solution. It is well-known that mutation is the key search operator in EP which generates new

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 200–210, 2010.

solutions from the current ones. In [6], a mutation operator based on Cauchy random numbers has been proposed and tested on the suite of 23 functions. The new EP with Cauchy mutation significantly outperforms the classical EP (CEP), which uses Gaussian mutation, on a number of multi-modal functions with many local minima while being comparable to CEP for unimodal and multi-modal function with only a few local minima.

The rest of the paper is organized as following: Section 2 describes the algorithms: IFEP, EP, FEP and PSO. Section 3 describes the implementation of the new PSOEP and SAVPSO algorithms. Section 4 lists benchmark functions use in the experiments, and gives the experimental settings. Section 5 presents and discusses the experimental results. Finally, Section 6 concludes with a summary and a few remarks.

2 Optimization by EPs

A global minimization problem can be formalized as a pair (S, f), where S $\subseteq$ R^n is a bounded set on R^n and f: S $\mapsto$ R is an n-dimensional real-valued function. The problem is to find a point x$_{\min}$ $\in$ S such that f(x$_{\min}$) is a global minimum on S. More specifically, it is required to find an x$_{\min}$ $\in$ S such that

$$\forall x \in S : f(x_{\min}) \le f(x)$$

where *f*(*x*) does not need to be continuous but, it must be bounded. This paper only unconstrained function optimization.

2.1 Classical Evolutionary Programming (CEP)

Fogel [1, 3] and Bäck and Schwefel [7] have indicated that CEP with self-adaptive mutation usually performs better that CEP without self-adaptive mutation for the functions they tested. Hence, CEP with self-adaptive mutation will be investigated in this paper. According to the description by Bäck and Schwefel [7], in this study CEP is implemented as follows:

1. Generate the initial population of μ individuals, and set k=1. Each individual is taken as a pair of real-valued vectors, (x_i, η_i), $\forall i \in \{1,2,...,\mu\}$, where x_i's are variables and η_i's are standard deviations for Gaussian mutations (also known as strategy parameters in self-adaptive evolutionary algorithms).
2. Evaluate the fitness score for each individual $(x_i, \eta_i), \forall i \in \{1,2,...,\mu\}$, of the population based on the objective function $f(x_i)$.
3. Each parent $(x_i, \eta_i), \forall i \in \{1,2,...,\mu\}$, creates a single offspring (x_i', η_i') by: for j=1,2,…n,

$$x_i\text{'(j)} = x_i\text{(j)} + \eta_i\text{(j)}\, N_j(0,1) \quad (2.1)$$

$$\eta_i\text{'(j)} = \eta_i\text{(j)}\exp(\tau\text{'}N(0,1) + \tau N_j(0,1)) \quad (2.2)$$

where x_i(j), x_i'(j), η_i(j) and η_i'(j) denote the j-th component of the vectors x_i, x_i', η_i and η_i', respectively. N(0,1) denotes a normally distributed one-dimensional random number with mean 0 and standard deviation 1. N $_j$ (0,1) indicates that a new random number is generated for each value of j. The τ and τ' are commonly set to $\left(\sqrt{2\sqrt{n}}\right)^{-1}$ and $\left(\sqrt{2n}\right)^{-1}$ [7,8].

4. Calculate the fitness of each offspring $\left(x_i^{'}, \eta_i^{'}\right), \forall i \in \{1,2,...,\mu\}$.
5. Conduct pair wise comparison over the union of parents (x_i , η_i) and offspring $\left(x_i^{'}, \eta_i^{'}\right), \forall i \in \{1,2,...,\mu\}$. For each individual, q opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is no smaller than the opponent's, it receives a "win."
6. Select the μ individuals, out of (x_i , η_i) and $\left(x_i^{'}, \eta_i^{'}\right), \forall i \in \{1,2,...,\mu\}$, that have the most wins to be parents of the next generation.
7. Stop if the halting criterion is satisfied; otherwise, k=k+1 and go to Step 3.

2.2 Fast Evolutionary Programming (FEP)

The one-dimensional Cauchy density function centered at the origin is defined by:

$$f_t(x) = \frac{1}{\pi} \frac{t}{t^2 + x}, \qquad -\infty < x < \infty \tag{2.3}$$

where t>0 is a scale parameter. The corresponding distribution function is

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi} \arctan(\frac{x}{t}).$$

The FEP studied in this paper is exactly the same as CEP described in last section except for Eq. (2.1) which is replace by the following [6]:

$$x_i\text{'(j)}= x_i\text{(j)}+ \eta_i\text{(j)}\ \sigma_j \tag{2.4}$$

Where σ $_j$ is a Cauchy random variable with the scale parameter *t*=1, and is generated a new for each value of *j*. It is worth indicating that we leave Eq.(2.2) unchanged in FEP in order to keep our modification of CEP to a minimum. It is also easy to investigate the impact of the Cauchy mutation on EP when other parameters are kept the same.

2.3 An Improved Fast Evolutionary Programming

Generally, Cauchy mutation performs better when the current search point is away from the global minimum, while Gaussian mutation is better at finding a local optimum in a given region. It would be ideal if Cauchy mutation is used when search points are far away from the global optimum and Gaussian mutation is adopted when search points are in the neighborhood of the global optimum. Unfortunately, the global optimum is usually unknown in practice, making the ideal switch from Cauchy to Gaussian mutation very difficult. Self-adaptive Gaussian mutation [7,2,9] is an excellent technique to partially address the problem. That is, the evolutionary algorithm itself will learn when to "switch" from one step size to another. However, there is room for further improvement to self-adaptive algorithms like CEP or even FEP.

In [19], an improved FEP (IFEP) based on *mixing* (rather than switching) different mutation operators. The idea is to mix different search biases of Cauchy and Gaussian mutation. The implementation of IFEP is very simple. It differs from FEP and CEP only in Step 3 of the algorithm described in Section 2.1. Instead of using Eq. (2.1) for CEP or Eq. (2.4) for FEP alone, IFEP generates two offspring from each parent, one by Cauchy mutation and the other by Gaussian. The better one is then chosen as the offspring. The rest of the algorithm is exactly the same as FEP and CEP. Chellapilla [10] has presented detailed experimental results for comparing different mutation operators in EP.

3 Hybrid Particle Swarm Optimization with EP

3.1 Classical PSO Algorithms

Particle swarm optimization (PSO) algorithm was proposed by Kennedy and Eberhart [20] in 1995, which can be used to solve a wide range of optimization problems. The PSO idea is inspired by natural concepts such as fish schooling, bird flocking and human social relations. PSO algorithm uses a number of particles that constitute a swarm. Each particle traverses the search space looking for the global optimum. In the system, the particles fly in a multidimensional search space. During flight, each particle adjusts its position according to its own experience and the experiences of neighboring particles, making use of the best position encountered by itself and its neighbors.

Assuming the search space is D-dimensional, the swarm's size is N. The i-th particle of the swarm is represented by the D-dimensional vector $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$ and the best particle of the swarm is denoted by index g . The best previous position of the i-th particle is recorded and represented by $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, and the velocity of the i-th particle is $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot r_{i1}(t) \cdot [p_{id}(t) - x_{id}(t)] + c_2 \cdot r_{i2}(t) \cdot [p_{gd}(t) - x_{id}(t)] \tag{3.1}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{3.2}$$

where, $d = 1, 2, \cdots, D$, t is the iteration number; $i = 1, 2, \cdots, N$; w is the inertia weight; c_1 and c_2 are two positive constants, called the cognitive and social parameter respectively; $r_{i1}(t)$ and $r_{i2}(t)$ are two random numbers uniformly distributed within the range [0,1]. The termination criterion for the iterations is determined according to whether the maximum generation or a designated value of the fitness of Pg is reached.

3.2 EP Based on Optimum Search Direction

Exploration ability is related to the algorithm's tendency to explore new regions of the search space; in this stage we should increase the search step size according to the search direction. Exploitation is the tendency to search a smaller region more thoroughly, in this stage; we should reduce the search step size. Researchers in PSO community have analyzed it empirically [22] and theoretically [21], they have shown the particles oscillate in different sinusoidal waves and converging quickly, sometimes prematurely, especially for PSO with small w. They used to believe that inertia weight balances exploration and exploitation in PSO algorithm. We think the factor to balance exploration and exploitation should be the velocity of particle. According to the global optimization search strategy, the search direction and the search step size are two important factors which affect the performance of algorithms. In PSO search, the current global best position *gbest* and the current best position of particles *pbest* determine the search direction. In this paper, we analyze the search direction of EP. We introduce the global best position *gbest* into EP to control the search direction towards the current global optimum. The main steps of the PSOEP algorithm are as follows:

$$x_i\text{'}(j)=x_i(j)+\eta_i(j)\sigma_j + \eta_i(j)\mathrm{N}_j(0,1)\cdot(gBest\text{-}x_i(j)) \qquad (3.3)$$

where $\mathrm{N}_j(0,1)$ indicates that a new random number is generated for each value of j, σ_j is a Cauchy random variable with the scale parameter t=1, and is generated a new for each value of j. *gbest* is the current global optimum.

It is also worth indicating that we leave Eq.(2.2) unchanged in PSOEP in order to keep our modification of CEP to a minimum. It is also easy to investigate the impact of the Cauchy mutation with optimum search direction on EP when other parameters are kept the same.

3.3 Self-Adaptive Velocity PSO (SAVPSO) Algorithm

To design an efficient search algorithm, we consider two important factors: search direction and search step size. When the search points are far away from the global optimum in the initial search stages, the larger search step size would increase the probability of escaping from a local optimum, and if the search direction is right, it also has a higher probability of reaching the global optimum. On the other hand, with the progress of search, the current search points are likely to move closer and closer towards a global optimum. So it is necessary to restrict the step size in the later stages for refinement of solutions. However, it is hard to know in advance whether the search points are far away from the global optimum. Unfortunately, the probability that a randomly generated initial population is very close to a global optimum is quite small in practice. It certainly worth enlarging the search step size in the early stages when we use EAs. In the final stages, the population of EAs tends to converge, and the step size tends to be reduced.

There exist many factors that would influence the convergence property and performance of PSO [20], including selection of ω, c1 and c2; velocity clamping; position clamping; topology of neighborhood; and other factors. Holland discussed the balance between exploration and exploitation that an algorithm must maintain [5].

Exploration ability is related to the algorithm's tendency to explore new regions of the search space; in this stage we should increase the search step size according to the search direction. Exploitation is the tendency to search a smaller region more thoroughly, in this stage; we should reduce the search step size. Researchers in PSO community used to believe that inertia weight balances exploration and exploitation in PSO algorithm. We think the factor to balance exploration and exploitation should be the velocity of particle. In this section we focus on how to control the search step size of PSO. We use lognormal self-adaptive strategy to control the velocity in PSO algorithm. The main steps of the SAVPSO algorithm are as follows:

```
Self-Adaptive Velocity Particle Swarm Optimizer
  Begin
$v(i+1) = \omega \times v(i) + \eta(i) \times (c_1 \times r_1 \times (pb - x(i))$
$+ c_2 \times r_2 \times (gb - x(i)));$
$\eta(i+1) = \eta(i) \times \exp(\tau \delta_1 + \tau' \delta_2);$
$x(i+1) = x(i) + v(i+1);$
end
```

where $0 < \omega < 1$ is a weight determining the proportion of the particle's previous velocity preserved, c_1 and c_2 are two positive acceleration constants, r_1 and r_2 are two uniform random sequences produced from $U(0,1)$, δ_1 and δ_2 are Gaussian random numbers, For Gaussian density function f_G with expectation 0; and variance σ^2 is

$$f_G = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}, \qquad -\infty < x < +\infty$$

τ and τ' are commonly set to $\left(\sqrt{2\sqrt{n}}\right)^{-1}$ and $\left(\sqrt{2n}\right)^{-1}$ [7].

4 Benchmark Functions

The availability of appropriate and standardized sets of benchmark functions is very important for evaluating the performance of any developed algorithms [11]. Seven benchmark functions, from different sources, [2, 7, 12, 13] were used in our experimental studies. The aim of this research is not to show whether PSOEP is better or worse than IFEP, FEP and CEP, instead it is to find when PSOEP is better (or worse) as compared to IFEP, FEP, CEP and why. Wolpert and Macready [14] have shown that under certain assumptions no single search algorithm is better on average for all problems. If the number of test problems is small, it would be very difficult to generalize the claim. Using too small a test set also has the potential risk that the algorithm is biased (optimized) towards the chosen problems, while such bias might not be useful for other problem of interest.

We summarize the benchmark functions which have been used to investigate the behavior of evolutionary algorithms in the continuous parameter optimization domain. The benchmark problems contain functions from simple uni-modal to multimodal with few or many local optima. They range from low to high dimensional and scalable problems. A regular arrangement of local optima, reparability of the objective function, decreasing difficultly of the problem with increasing dimensionality, and potential bias introduced by locating the global optimum at the origin of the coordinate system are identified as properties of multi-modal objective functions which are neither representative of arbitrary problems nor well suited for assessing the global optimization qualities of evolutionary algorithms. The 7 benchmark functions are given in Table 1.

Table 1. The 7 benchmark functions used in our experimental study, where n is the dimension of the function; f $_{\min}$ is the minimum value of the function, and S $\subseteq$ R ^{n}

Test Function	n	S	f_{min}
f_1 (x)= $\sum_{i=1}^{n} x_i^2$	30	$[-100,100]^{n}$	0
f_2 (x)= $\sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10,10]^{n}$	0
$f_3(x) =$ -20exp $\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right)$ - exp $\left(\frac{1}{n} \sum_{i=1}^{n} \cos 2\pi x_i \right)$ +20+e	30	$[-32,32]^{n}$	0
$f_4(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600,600]^{n}$	0
$f_5(x) = -\sum_{i=1}^{5} [(x-a_i)(x-a_i)^T + c_i]^{-1}$	4	$[0,10]^{n}$	-10
$f_6(x) = -\sum_{i=1}^{7} [(x-a_i)(x-a_i)^T + c_i]^{-1}$	4	$[0,10]^{n}$	-10
$f_7(x) = -\sum_{i=1}^{10} [(x-a_i)(x-a_i)^T + c_i]^{-1}$	4	$[0,10]^{n}$	-10

Functions *f*1 and *f*2 are high-dimensional problems, which are uni-modal. Function *f*3 and *f*4 are multi-modal functions where the number of local minima increases exponentially with the problem dimension [12, 13]. Those classes of functions appear to be the most difficult class of problems for many optimization algorithms (including EP). Functions *f*5 to *f*7 are low-dimensional functions which have only a few local minima [13].

5 Experimental Studies and Discussions

5.1 Comparing PSOEP with EPs

In order to carry out a fair comparison among IFEP, FEP and CEP, the population size of IFEP was reduced to half of FEP or CEP in all the following experiments, since each individual in IFEP generates two offspring. However, reducing IFEP's population size by half actually disadvantages IFEP slightly because it does not double the time for any operator (such as selection) other than mutations. Nevertheless, such comparison offers a good and simple compromise.

IFEP was tested in the same experimental setup as FEP and CEP. In all experiments, the same self-adaptive method (i.e., Eq.(2.2)), the same population size, μ=100, the same tournament size, q=10, for selection, the same initial η =3.0, and the same initial population were used for PSOEP, IFEP, FEP and CEP. These parameters were considered as suggested by Bäck and Schwefel [7] and Fogel [2]. The average results of 50 independent runs are summarized in Table 2.

Table 2. Comparing among PSOEP, IFEP, FEP and CEP on function f1 to f7

F	#.of Gen.	PSOEP Mean Best	IFEP Mean Best	FEP Mean Best	CEP Mean Best
f1	1500	3.69×10^{-6}	4.16×10^{-5}	5.72×10^{-4}	1.91×10^{-4}
f2	2000	2.14×10^{-3}	2.44×10^{-2}	7.60×10^{-2}	2.29×10^{-2}
f3	1500	8.38×10^{-4}	4.83×10^{-3}	1.76×10^{-2}	8.79
f4	2000	9.86×10^{-3}	4.54×10^{-2}	2.49×10^{-2}	8.13×10^{-2}
f5	100	-10.15	-6.46	-5.50	-6.43
f6	100	-10.40	-7.10	-5.73	-7.62
f7	100	-10.54	-7.8	-6.41	-8.86

5.2 Comparing SAVPSO with PSO

In order to fairly compare between SAVPSO and PSO, SAVPSO was tested using the same experimental setup as of PSO. In all experiments, the parameters and operators such as the self-adaptive method, the population size is 100, c1=c2=1.5, the initial η =3.0, and the initial population used for SAVPSO and PSO was the same. The average results of 50 independent runs are summarized in Table 3.

Table 3. Comparison between SAVPSO and PSO on function *f*1 to *f*7. All results have been average over 50 runs, where "Mean Best" indicates the mean best function values found in the last generation.

F	#.of Gen.	SAVPSO Mean Best	PSO Mean Best
F1	100	3.57×10^{-7}	1.88×10^{2}
F2	100	9.59×10^{-16}	9.03×10^{3}
F3	100	4.18×10^{-12}	1.05×10^{1}
F4	100	4.44×10^{-17}	1.94×10^{1}
F5	100	-10.15	-5.01
F6	100	-10.40	-4.95
F7	100	-10.54	-4.90

5.3 Discussions

It is very clear from Table 5.2 that SAVPSO has improved PSO's performance significantly for all test functions. For the two uni-modal functions (*f*1 and *f*2), SAVPSO outperformed PSO significantly.

It is very encouraging that SAVPSO is capable of performing much better than PSO for all the test functions. This is achieved through a minimal change to the existing PSO. No prior knowledge or any complicated operators were use, and also no additional parameter was used either. The superiority of SAVPSO also demonstrates the importance of self-adaptive velocity of particles search biases (e.g. "step sizes") in a robust search algorithm.

In fact, the large velocity of particles played a major role in the early stages of evolution, since the distance between the current search points and the global optimum are relatively large on average in the early stages, hence large search step size performed better. However, as the evolution progresses, the distances to the global optimum become smaller and smaller, we should reduce the search step size. The log-normal self-adaptive strategy fit this requirement very well during the whole evolution progress. That is why SAVPSO performs much better than standard PSO. The rapid convergence of SAVPSO is shown in Table 5.2 and support of our explanations.

It is very clear from Table 5.1 that PSOEP has improved EP's performance significantly for all test functions. These results show that PSOEP continues to perform much better than IFEP on multi-modal functions with many minima, and also performs very well on uni-modal functions and multi-modal functions with only a few local minima, which IFEP and FEP face difficulty. PSOEP also achieved much better performance than CEP on these functions.

For the two uni-modal functions where FEP is outperformed by CEP significantly, PSOEP performs better than CEP (*f*1 and *f*2). A close look at the actual average solutions indicates that PSOEP provides better solution than CEP on *f*1 and *f*2 but in smaller magnitude.

For the three Shekel function *f*5 to *f*7, the difference between IFEP and CEP is much smaller than that between FEP and CEP. IFEP has improved FEP's performance significantly on all three functions. It performs better than CEP on *f*5 and *f*6 and worse on *f*7. PSOEP's performance is the best for all the EP versions.

It is very encouraging that PSOEP is capable of performing as well as or better than the better one of IFEP, FEP and CEP for all the test functions. This is achieved through a minimal change to the existing FEP and CEP. No prior knowledge or any complicated operators were used, and also no additional parameter was used either. The superiority of PSOEP also demonstrates the importance of mixing different search biases (e.g. "step sizes") and optimum search direction in a robust search algorithm.

It is that Cauchy mutations played a major role in the population in the early stages of evolution since the distance between the current search points and the global optimum was relatively large on average in the early stages, hence Cauchy mutation performed better. However, as the evolution progresses, the distance became smaller and smaller. Large search step sizes produced by Cauchy mutation tended to produce worse offspring than those produced by Gaussian mutation. The decreasing number of successful Cauchy mutations in those figures illustrates this behavior.

All results have been average over 50 runs, where "Mean Best" indicates the mean best function values found in the last generation.

6 Conclusions and the Future Work

PSOEP uses the idea of mixing search biases to mix Cauchy and Gaussian mutations and the current global optimum search direction. Unlike some switching algorithms which have to decide when to switch between different mutations during search, PSOEP does not need to make such decision and introduces no parameters. PSOEP is robust, assumes no prior knowledge of the problem to be solved, and performs much better than IFEP, FEP and CEP for all the benchmark problems. Future work on PSOEP includes the comparison of PSOEP with other self-adaptive Differential Evolution algorithms [15] and other evolutionary algorithms [16].

SAVPSO uses lognormal self-adaptive velocity of particles in PSO. Unlike some switching algorithms, which have to decide the timing of switching between different velocity of particles in PSO, SAVPSO does not require any switching decision and parameters related to such switching. SAVPSO is robust, assumes no prior knowledge of the problem to be solved, and performs much better than PSO for most benchmark problems.

The idea of PSOEP can also be applied to other evolutionary algorithms to design faster optimization algorithms [17]. For $(\mu+\lambda)$ and (μ, λ) evolutionary algorithms where $\mu<\lambda$, PSOEP would be particularly attractive since a parent has to generate more than one offspring. It may be beneficial if different offspring are generated by different mutations [17].

Acknowledgements

This paper supported by the National Natural Science Foundation of China (No. 6077 2163), the National Natural Science Foundation of Guangdong (No. 9151001002000014), Shenzhen Scientific Research Project (No. SY200806300270A), Research Project of SZIIT (No. BC2009014).

References

1. Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence Through Simulated Evolution. John Wiley & Sons, New York (1966)
2. Fogel, D.B.: System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling. Ginn Press, Needham Heights (1991)
3. Fogel, D.B.: Evolving Artificial Intelligence. PhD thesis, University of California, San Diego, CA (1992)
4. Fogel, D.B.: Applying evolutionary programming to selected traveling samlesman problems. Cybernetics and Systems 24, 27–36 (1993)
5. Yao, X.: An overview of evolutionary computation. Chinese Journal of Advanced Software Research 3(1), 12–29 (1996)
6. Yao, X., Liu, Y.: Fast Evolutionary Programming. In: Fogel, L.J., Angeline, P.J., Bäck, T. (eds.) Evolutionary Programming V: Proc. of the Fifth Annual Conference on Evolutionary Programming, Cambridge, MA, pp. 257–266 (1996)
7. Bäck, T., Schwefel, H.-P.: An overview of evolutionary algorithms for parameter optimization. Evolutionary Computation 1(1), 1–23 (1993)

8. Fogel, D.B.: An Introduction to Simulated Evolutionary Optimization. IEEE Trans. on Neural Networks 5(1), 3–4 (1994)
9. Fogel, D.B.: Evolutionary computation: Towards a new philosophy of machine intelligence. IEEE Press, New York (1995)
10. Chellapilla, K.: Combining mutation operators in evolutionary programming. IEEE Trans. on Evolutionary Computation 2(3), 91–96 (1996)
11. Bäck, T., Fogel, D.B., Michalewicz, Z.: Handbook of Evolutionary Computation. IOP Publishing, Oxford University Press (1997)
12. Schwefel, H.-P.: Evolution and Optimum Seeking. John Wiley & Sons, New York (1995)
13. Törn, A., Zilinskas, A.: Global Optimization. LNCS, vol. 350. Springer, Heidelberg (1989)
14. Wolpert, D.H., Macready, W.G.: No free lunch theorems for search. IEEE Transcation on Evolutionary Computation 1(1), 67–82 (1997)
15. Omran, M.G.H., Salman, A., Engelbrecht, A.P.: Self-adaptive Differential Evolution. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-m., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005, Part I. LNCS (LNAI), vol. 3801, pp. 192–199. Springer, Heidelberg (2005)
16. Lam, T., Soliman, O., Abbass, H.A.: A Modified Strategy for the Construction Factor in Particle Swarm Optimization. In: Randall, M., Abbass, H.A., Wiles, J. (eds.) ACAL 2007. LNCS (LNAI), vol. 4828, pp. 333–344. Springer, Heidelberg (2007)
17. Yao, X., Liu, Y.: Fast Evolution Strategies. Control and Cybernetics 26(3), 467–496 (1997)
18. Duan, M., Povinelli, R.: Nonlinear Modeling: Genetic Programming vs. Fast Evolutionary Programming. In: Intelligent Engineering Systems Through Artificial Neural Networks (ANNIE 2001), pp. 171–176 (2001)
19. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Trans. Evolutionary Computation 3(2), 82–102 (1999)
20. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proc. IEEE International Conference on Neural Networks, vol. IV, pp. 942–948. IEEE Service Center, Piscataway (1995)
21. Clerk, M., Kennedy, J.: The particle swarm explosion, stability and convergence in a multidimensional complex space. IEEE Trans. Evol. 6(1), 58–63 (2002)
22. Angeline, P.J.: Evolutionary optimization versus particle swarm optimization: philosophy and performance difference. Evolutionary programming, 601 (1998)

Hybridized Optimization Genetic Algorithm for QOS-Based Multicast Routing Problem

Yunliang Chen[1,2], Jianzhong Huang[1,*], and Changsheng Xie[1]

[1] School of Computer Science, Huazhong University of Science and Technology,
Wuhan, Hubei 430074, China
[2] School of Computer Science, China University of Geosciences,
Wuhan, Hubei 430074, China
husthjz@126.com

Abstract. This paper analyzes the mathematic model of QOS(quality-of-service) based multicast routing with nodes delay. Based on the original, an optimization model called GP-GA is proposed by hybridizing Gene-Pool (GP) with traditional Genetic Algorithm (GA). The model uses the gene-pool to save the better individual during the process of each generation. In the mean time, the crossover and mutation operator are improved to accelerate the convergence. As the problem may be easily trapped by local optimization, the evolution strategy based on 'reserved and non-reserved' is also constructed to enhance the ability of finding optimal solution and decrease the probability of 'premature' phenomena commendably. The emulation experiments demonstrate that the probability of GP-GA is higher than the general GA in converging optimal solutions, and the algorithm is also effective for the adjustment to the dynamic multicast routing.

Keywords: routing optimization, Steiner tree, dynamic multicast routing, gene-pool, genetic algorithm.

1 Introduction

Multicast service is becoming a key requirement of computer networks supporting multimedia applications. To carry large numbers of multicast sessions, a network must minimize the sessions' resource consumption, while meeting their quality of service (QOS) requirements.

Multicast routing is to find a multicast tree with least-cost sources and a set of destination nodes, which is concluded as the Steiner Tree problem in mathematics. Meanwhile, considering the nodes adding or decreasing dynamically in reality, constructing the dynamic multicast routing is involved in the research.

To resolve the NP-complete problem a lot of achievements have been done. In the past, heuristic models were mostly adopted. However, original heuristic models are limited in the complexity of problems.

[*] Corresponding author.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 211–219, 2010.

Using Intelligence computing is another choice. Sun using Hopfield neural networks introduced the steiner tree to meet the demand of nodes delay and delay variation constraints [1]. Then Song proposed the model using Ant Colony algorithm to parallel the multicast routing [2]. Liu constructed the immunity operator by operating inoculation immunity and immunity selection, as well as the immunity algorithm in the multicast routing choice [3]. Zhang used the memory function and the rule of tabu search to construct the delay constraints least-cost multicast tree[4]. In Ref.[5], the optimal solution was achieved by constructing annealing schedule to simulate the network delay and cost of the corresponding routing tree with the thought of simulated annealing algorithm . Esbensen used GA to search the possible Steiner tree nodes and code in the traditional binary system [6]. In Ref.[7-11], the code scheme of the whole chromosome was made by regarding the possible path(s,t), between the given multicast tree source ‘s’ and the destination node ‘t’ , as the chromosome gene.

This paper presents an optimization GA based on GP to solve the Steiner tree problem. During exploring the solutions, GP can acquire and save the better gene individual, improve the traditional GA way of using the crossover and mutation operator, modify the strategy of control and choice, and decrease the probability of ‘premature’ phenomena commendably. And the experiments show that comparing with the original GA, the rate of GP-GA is greatly improved in finding the optimal solution.

The remainder of this paper is organized as follows. Section 2 gives shot overviews of multicast routing, Steiner tree and GA. Section 3 gives the details of GP-GA to solve the multicast routing problem: initialization of chromosome, crossover operator, evolution strategy and fitness function. Section 4 introduces the flow of GP-GA in dynamic network environments. Then Section 5 evaluates the performance of GP-GA for multicast routing problem in different kinds of network environments. Finally, we summarize the paper in section 6.

2 Preliminary

2.1 Multicast Routing and Steiner Tree

A multicast group is a set of nodes in a network that need to share the same piece of information. A multicast group can have one or more source nodes, and more than one destination. Note that even when there is more than one source, the same information is shared between all nodes in the group.

Steiner tree problems are very useful in representing solutions to multicast routing problems. They are employed mostly when there is just one active multicast group and the minimum cost tree is wanted. In the Steiner tree problem, given a graph G (V; E), and a set R $\subseteq$ V of required nodes, we want to find a minimum cost tree connecting all nodes in R. The nodes in V can be used if needed, and are called “Steiner” points. This is a classical NP-hard problem [12], and has a vast literature on its own [13].

The Steiner tree technique for multicast routing consists of using the Steiner problem as a model for the construction of a multicast tree. In general, it is considered that there is just one source node for the multicast group. The set of required nodes is defined as the union of source and destinations. This technique is one of the most studied for multicast tree construction, with many algorithms available [14].

2.2 Genetic Algorithm

The Genetic Algorithms (GAs) are used to solve an optimization problem based on the principle of evolution. A population of candidate solutions, called chromosomes, is maintained at each iteration of the evolution. Each chromosome consists of linearly arranged genes which are represented by binary strings.

Three basic operations, namely, reproduction, crossover, and mutation, are adopted in the evolution to generate new offspring. Reproduction is based on the Darwinian survival of the fittest among strings generated. Samples (represented as bit strings) with larger fitness function values are selected to generate new offspring bit strings by means of crossover operations, and the offspring are converted into new parameter solutions. Intuitively, a bit string with a larger fitness function value should have a higher probability of contributing one or more offspring bit strings in the next generation and vice versa. Crossover is used to cut individually two parent bit strings into two or more segments and to then combine the segments undergoing crossover to generate two offspring bit strings. Crossover can produce offspring that are radically different from their parents.

Mutation is to perform random alternation on bit strings by means of some operations, such as bit shifting, inversion, rotation, etc. A mutation operation will create new offspring bit strings different from those generated by the reproduction and crossover operations. Mutation can extend the scope of the solution space and reduce the possibility of falling into local extremes.

3 GP-GA for Steiner Tree Problem

3.1 Initialization of Chromosome

In this paper, chromosome is composed of two paths, source 's' to midst 'k' and 'k' to destination node 't'. In the process of initialization a strategy is as followed. Random route, from source s to midst k, or the shortest route from s to k, is set as the former part of the chromosome, while the latter part is the neighbor node t linking to the randomly chosen midst k. GP in this algorithm is designed to adopt cross chain with array point, in which array shows each midst node, and the corresponding cross chain table saves the route from s to k in the evolution as the optimal or worst solution.

3.2 Crossover Operator

Traditional genetic crossover operator includes one-point crossover, multi-point crossover, and linear recombination. In the experiments, Disjoin crossover operator is adopted because the parental route is saved and it can accelerate the convergence speed [15]. Comparing with the traditional One-point crossover operator, Disjoin crossover operator can hold faster convergence speed [7,8,15-18].

From Fig1, We can conclude that compared with traditional one-point crossover operator, Disjoin operator obviously has higher evolution speed. When the network nodes are less than 50, Disjoin operator is less advantaged than one-point crossover operator. But with the increasing nodes, more and more midst nodes and sides are added in the chromosome. Once there are crossings, offspring of one-point operator

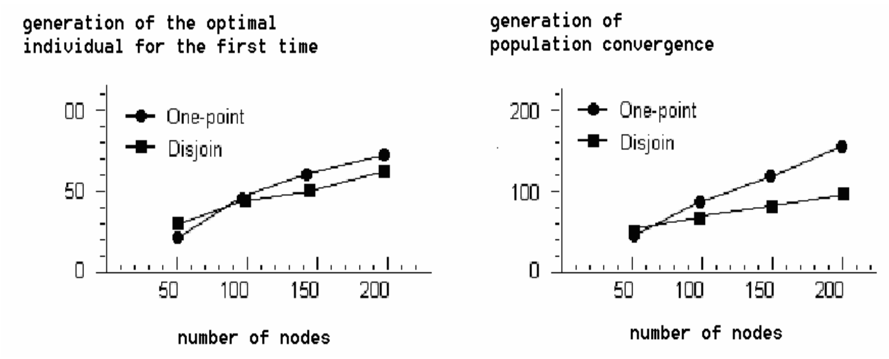


Fig. 1. Compare with one-point and Disjoin

need fixing（adding new sides）and deleting(avoiding circle), as the priority of Disjoin operator is shown. We also notice that supernatural individual would usually control the following evolutionary process for they are more competitive or have better fitness function. Fig 1 shows that for the 50th-60th generation of the optimal individual, the algorithm is inclined to converge, which would influence the whole optimization and only acquire local optimal solution. To solve this typical evolution feint, we improve the evolution strategy, the details can be seen in subsection 3.4.

3.3 Mutation Operator

Besides the traditional mutation operator, randomly chosen route from source to destination replacing the route with same destinations in chromosome, there are two other types of mutation operators.

Mutation 1: according to the mutation probability pm1, the mutation procedure randomly selects route from gene-pool replacing the route with same destinations.

Mutation 2: according to the mutation probability pm2, randomly selects a destination node t, breaks down the side with middle node ki, then connects to the other nearby middle node kj.

Obviously Mutation 1 takes the reserved route in GP into consideration, while Mutation 2 is applied to search for the local optimal individual.

3.4 Evolution Strategy

In many references ,we find that the traditional strategy ,roulette method, also called Monte Carlo choosing method, is adopted to choose the chromosome, but it would be inclined to converge too early, search inefficiently in the latter part and the rate to search for the whole optimal solution is lower. In [18], it is more effective to adopt simulated annealing method to improve the choosing on the base of the double populations, so we are enlightened and take “Reserved and Non-reserved”/RAN method.

1） Reserved phase
   a. When the fitness of the offspring individual is greater than the parent’s, the offspring will replace the parent.

b. Oppositely, the probability exp -Δf/Ep will replace the parent. 。 Δf is the fitness balance between the parent and offspring, Ep is the variance of the chromosome fitness, and the calculating formula is $E_p = \frac{1}{M}\sum_{i=1}^{M}(f(t)-\overline{f})^2$ (f is the fitness function). During the initial stage of implementing GA, the variety of chromosome is considered more in group updating, for the individual difference, *Ep , and* the probability exp (*-Δf /Ep)* are larger. In the latter stage, *Ep* and probability exp (*-Δf /Ep)* keep smaller and the optimal parent is reserved.

2) Non-reserved phase

The optimal 40% individuals are deleted from the Oldpop, new constructed route chromosomes are filled and a Newpop is set. While the mutation probability is set as 1and crossover probability is 0, the whole observing ability of Newpop is strengthened.

The above design is coming from the consideration of that the chromosome will usually converge and get some optimal solution in the latter part of the traditional GA. It is too difficult to improve, but oscillating is needed to develop the solution space to search the optimal solution instead of being trapped in the local optimal solution, in which the times of oscillating is controlled by oscillator.

Generally the algorithm steps into the reserved phase first in the replacement of population. In this phase, the current 40% optimal individual mean will be compared with the former generation's after each generation evolution. If the fitness mean never increases after the set successive generation evolution, the Non-reserved phase will be applied forcedly. Till the Newpop converges to a proper fitness value, the Oldpop and Newpop will be united. The probability exp (*-Δf /Ep)* decides which new individuals to be reserved in the newpop to replace Oldpop to finish the population evolution, in which N is the number of the chromosomes in the population.

When setting oscillator, we have done a lot experiments. In the experiments, with the increasing of oscillator, the success ratio of RAN to find the solution increases greatly, especially when oscillator is 5, RAN, as well as the SA method [19], can find the optimal solution and have high stability. So we can conclude that to increase the oscillator value can improve the probability of converging to the optimal solution. But we also point out that we do not work on the maximum of oscillator, for the usability instead of the best executive final is focused. Considering the complexity the increasing of oscillator value brings to the average algorithm convergence time, we set oscillator as 3 in the following experiments to achieve the effective balance between the convergence time and the success solution ratio.

3.5 Fitness Function

This paper defines the fitness function of Steiner tree problem as:

$F(T(s,M))=\sum\{ u_1C(e)+u_2\Psi(\Delta-D(e))+u_3\Psi(\delta-W(T)),\ e\in T$; Network delay and its variation maximum are Δ andδ. ui > 0 is weight (i = 1, 2, 3) ; single charge function is C (e) ; network delay function is D (e) ;delay variation function is W (T) ;Ψ (z) is the fine function, which can be set as:

$\psi(Z) = \begin{cases} \varepsilon, Z \le 0 \\ 1, Z \succ 0 \end{cases}$ ε is set between[0, 0. 5], and min (F (T)) is the solution.

3.6 GP Updating

After each generation evolution, if the quality of solution is improved, it is necessary to use the effective information to update GP to make it more directional. Updating strategy is as follows:

1) if there is a new route in the chromosome, it will be added to the corresponding destination crossing chain of GP.
2) The chosen priority of the route in GP will be changed, as the route weight of the current optimal solution or sub-optimal solution will be increased. GP can not only be used in GP-GA, but also used in the following dynamic GP-GA.

4 GP-GA in Dynamic Network Environments

Actually multicast members are always entering and exiting. To reconstruct the multicast tree will cost too much time and space. So it is better to use the acquired optimal chromosome and the reserved the better route to construct the initial population to speed up the convergence. The following is the example to show the JOIN(t) and REMOVE(t), t is the dynamic destination node.

```
   1) Procedure JOIN (t):
   Begin
         Add point t to the gene-pool;
         Construct the routs from s to t randomly and rebuild the chromosome;
         Initialize population P with N individuals based-on the previous optimization;
         GP-GA (N+t);
   End;
   M=BestOf(P);
   Output M;

   2) Procedure REMOVE (t):
   Begin
         Delete the t route from the previous chromosome;
         Initialize population P with N individuals from the previous optimization;
         GP-GA (N-t);
   End;
   M=Best Of (P);
Output M;
```

5 Evaluation

Network model with random graph was presented in Ref.[19], for the random graph is most close to the real network. The same model is also introduced in Ref.[7,8]. It can also help to delete the experimental error environmental elements and make an objective comparison of the algorithm performance. The experiments aim to prove that GP-GA is more vigorous and feasible to solve the Steiner problem from the following two aspects.

5.1 The Average Accurate Rate

Here are the parameters of our model: Pc is the crossover ratio in the range of 0. 2 ～ 0. 9; Pm is the mutation ratio in the range of 0. 01 ～ 0. 1; Popsize is the number of the populations in [20,100]; Max is the maximum generations in [2000, 4000]; oscillator r = 3。 The condition to end GP is that all the populations converge to the same solution or the generation reaches the set maximum algebra. In the network topology, |N |is the total number of nodes, | E | is the number of sides, εis the multicast density *(%)* , the delay function and cost function is [0, 50]ms and [0, 200] respectively at normal school. The delay maximumΔis 25, the delay variation maximum δ is 6; Each topological group will have 100 experiments and the final results are shown in Tab. 1.

Table 1. Algorithm efficiency between A-GA,B-GA and GP-GA

Average efficiency topology	200th			500th		
	A-GA[5]	B-GA[10]	GP-GA	A-GA[5]	B-GA[10]	GP-GA
|N|=50, |E|=150, (ε=20%)	76.8%	75.3%	74.2%	88.7%	90.5%	90.4%
|N|=100, |E|=300, (ε=15%)	54.3%	53.6%	64.8%	78.4%	75.88%	87.6%
|N|=150, |E|=500, (ε=18%)	48.65%	45.3%	62.7%	64.5%	68.7%	84.5%
|N|=200, |E|=725, (ε=13%)	45.8%	47.2%	59.8%	69.1%	48.3%	80.1%

Tab. 1 exhibits that GP-GA adopting the RAN evolution strategy shows slower speed in evolution than A[7]、B[8]], but the convergence speed is in an acceptable scale because of GP and special crossover operator. The advantage of GP-GA is the average success ratio of solution keeps at more than 80%, but the ratio of A, B drops fast with the development of network scale. It can be concluded that GP-GA is quite fit for the large and medium-sized network.

5.2 GP-GA in Dynamic Network

To algorithm A, T_A shows the generated multicast tree of algorithm A, *Cost (T_A)* is the cost of all the routes in the generated tree, the definition of the comparative cost rate between algorithm A and GA-Online is : $\bar{E}_A$= [Cost(T_A)-Cost（T GA-Online）]/ Cost (T GA-Online). The definition of MPH can be found in [7]. The experimental results are shown in tab. 2.

Tab. 2 depicts that there is no big difference between dynamic GA-Online and GA-Off at the cost of acquiring the smallest Steiner tree and both of them can converge to reach an optimal solution, as MPH is inferior in stability and cost. So we conclude

Table 2. The efficiency of dynamic GA-Online

Net topology	$\bar{E}$MPH	$\bar{E}$GA-Off	$\bar{E}$GA-Online
$\lvert N\rvert=50$, $\lvert E\rvert=150$, (ε=20%)	0%	0%	0%
$\lvert N\rvert=50$, $\lvert E\rvert=170$, (ε=20%)	2.78%	0.02%	0%
$\lvert N\rvert=50$, $\lvert E\rvert=185$, (ε=20%)	1.32%	0%	0%
$\lvert N\rvert=100$, $\lvert E\rvert=300$, (ε=15%)	3.62%	0%	0%
$\lvert N\rvert=100$, $\lvert E\rvert=350$, (ε=15%)	2.29%	0.02%	0%
$\lvert N\rvert=100$, $\lvert E\rvert=400$, (ε=15%)	0%	0.01%	0%
$\lvert N\rvert=100$, $\lvert E\rvert=470$, (ε=15%)	1.69%	-0.08%	0%
$\lvert N\rvert=100$, $\lvert E\rvert=500$, (ε=15%)	3.32%	0.07%	0%
$\lvert N\rvert=150$, $\lvert E\rvert=575$, (ε=18%)	4.10%	-0.10%	0%
$\lvert N\rvert=150$, $\lvert E\rvert=700$, (ε=18%)	0.72%	0.01%	0%
$\lvert N\rvert=150$, $\lvert E\rvert=780$, (ε=18%)	1.84%	0%	0%
$\lvert N\rvert=150$, $\lvert E\rvert=835$, (ε=18%)	2.55%	0%	0%
$\lvert N\rvert=150$, $\lvert E\rvert=900$, (ε=18%)	3.62%	0%	0%
$\lvert N\rvert=200$, $\lvert E\rvert=975$, (ε=13%)	0.06%	0.1%	0%
$\lvert N\rvert=200$, $\lvert E\rvert=1250$, (ε=13%)	1.15%	0%	0%

that GA-Online can effectively shield the network inaccuracy and more adaptable the dynamic network environment.

6 Conclusion

This paper puts forward the GA model based on Gene-Pool to solve the Steiner tree problem in network. To solve the low accurate ratio of the traditional solution, the paper hybridizes the GP-GA model with RAN evolutionary method. To solve the slow convergence speed, the paper improves the crossover mutation operators. Evaluational experiments show that GP-GA algorithm can achieve higher success solution ratio in topological and dynamic network, as well as being more practical.

Acknowledgment

This paper is supported by the National Natural Science Foundation of China (key program) under the Grant NO. 60933002, National "863" Project under Grant NO. 2009AA01A402, and the Special Fund for Basic Scientific Research of Central Colleges, China University of Geosciences (Wuhan) NO. CUGW090230.

References

1. Sun, W.S., Liu, Z.: Multicast Routing Based Neural Networks. J. Journal of China Institute of Communications 19(11), 126–130 (1998)
2. Song, X.J., Liu, W.: Dynamic Multi-constrained Distributed Multicast Routing Algorithm Based on Multi-point Parallel Ant Search. J. Journal of Circuits and Systems 9(1), 243–247 (2004)

3. Liu, F., Feng, X.J.: Immune Algorithm for Multicast Routing. J. Chinese Journal of Computers 26(6), 66–681 (2003)
4. Zhang, K., Wang, H., Liu, F.Y., Cao, H.X.: A Multicast Routing Algorithm with Delay-constrained Based on Tabu Search. J. Computer Engineering 31(11) (2005)
5. Wang, X.W., Cheng, H., Cao, J.N., et al.: A Simulated-annealing based QoS multicasting algorithm. In: Proceedings of International Conference on Communication Technology, pp. 469–473. IEEE Press, Beijing (2003)
6. Esbensen, H.: Computing Near -Optimal Solutions to the Steiner Problem in a Graph Using a Genetic Algorithm. J. Networks 26, 173–185 (1995)
7. Chen, L., Yang, Z.Y.: A Degree-Delay-Constrained Genetic Algorithm for Multicast Routing Tree. In: Proceedings of the Fourth International Conference on Computer and Information Technology (CIT 2004), pp. 1033–1038. IEEE Press, Los Alamitos (2004)
8. Hagh, A.T., Faez, K., Dehghan, M., et al.: A Genetic Algorithm for Steiner Tree Optimization with Multiple Constraints Using Prüfer Number. In: Shafazand, H., Tjoa, A.M. (eds.) EurAsia-ICT 2002. LNCS, vol. 2510, pp. 272–280. Springer, Heidelberg (2002)
9. Hwang, R.H., Do, W.Y., Yang, S.C.: Multicast routing based on genetic algorithms. J. Journal of Information Science and Engineering 16(6), 885–901 (2000)
10. Shi, J., Zou, L., Dong, T.L., Zhao, E.D.: The Application of Genetic Algorithm in Multicast Routing. J. ACTA Electronica Sinica 5(28), 668–671 (2000)
11. Chen, C.J., Kang, L.S., Chen, Y.P.: A Genetic Algorithm for the Multicast Routing Optimization Problem with Delay and Delay-Variation Constraints. J. Computer Engineering and Applications 21, 147–149 (2003)
12. Garey, M.R., Johnson, D.S.: Computers and intractability. In: A guide to the theory of NP-completeness. A Series of Books in the Mathematical Sciences (1979)
13. Du, D.Z., Lu, B., Ngo, H., Pardalos, P.M.: Steiner tree problems. In: Encyclopedia of optimization, vol. 5, pp. 227–290. Kluwer Academic Publishers, Dordrecht (2001)
14. Kompella, V., Pasquale, J., Polyzos, G.: Optimal multicast routing with quality of service constraints. J. Journal of Network and Systems Management 4(2), 107–131 (1996)
15. Karab, I.M., Fathym Dehghan, M.: QoS Based Multicast Routing On a Heuristic Genetic Algorithm. In: 17th Canadian Conference on Electrical and Computer Engineering, Niagara Falls, Canada, pp. 1727–1730 (2004)
16. Esbensen, H.: A Genetic Algorithm for the Steiner Problem in a Graph. J. IEEE Journal on Selected Areas in Communications, 402–406 (1994)
17. Zhou, W., Chen, C.J.: A Genetic Algorithm for Multicasting Routing Problem. In: International Conference Communication Technology Proceedings, WCC-ICCT, pp: 1248–1253 (2000)
18. Chen, P., Liu, S.Y.: The Multicast Routing Algorithm Based on Genetic Strategy With Two Populations. J. Journal of Electronics and Information Technology 12(24), 442–446 (2002)
19. Salama, H.F., Reeves, D.S., Viniotis, Y.: Evaluation of multicast routing algorithms for real time communication on high speed networks. IEEE Journal on Selected Areas in Communications 15(3), 332–345 (1997)

Retroviral Iterative Genetic Algorithm for Real Parameter Function Optimization Problems

Renato Simões Moreira[1,3], Glauber Duarte Monteiro[2,3], Otávio Noura Teixeira[1,3],
Átila Siqueira Soares[1], and Roberto Célio Limão de Oliveira[1]

[1] Instituto de Tecnologia, Universidade Federal do Pará (UFPA), Caixa-Postal 479,
66075-110, Belém, Pará, Brazil
[2] Instituto de Ciências Exatas e Naturais, Universidade Federal do Pará (UFPA),
Caixa-Postal 479, 66075-110, Belém, Pará, Brazil
[3] Laboratório de Computação Natural, Centro Universitário do Estado do Pará (CESUPA),
Av. Gov. José Malcher, 1963, 66060-230, Belém, Pará, Brazil
{renatosm,glauberbcc,onoura,atilass}@gmail.com, limao@ufpa.br

Abstract. This paper describes the development of a new hybrid meta-heuristic of optimization based on a viral lifecycle, specifically the retroviruses (the nature's swiftest evolvers'), called Retroviral Iterative Genetic Algorithm (RIGA). This algorithm uses Genetics Algorithms (GA) structures with features of retroviral replication, providing a great genetic diversity, confirmed by better results achieved by RIGA comparing with GA applied to some Real-Valued Benchmarking Functions.

Keywords: Evolutionary Computation, Genetic Algorithm, Viruses, Retroviruses, Hybrid Metaheuristic.

1 Introduction

Over the years the virus has been treated as villains in the destruction of organic structures, resulting from the disappearance of entire species and even long and fatal epidemics (such as HIV). However, its effectiveness to perpetuate themselves is really impressive, although being not living forms. Retroviruses are the nature's swiftest forms [1] and this retroviral feature could not be discarded in the application of some computational structure (specifically evolutionary computation) that takes them as inspiration.

The proposed work will describe the development of a new metaheuristic structure inspired on viral structures of family Retroviridae (retroviruses), this algorithm is called as Retroviral Iterative Genetic Algorithm (RIGA). The source of the name comes from the junction of its features: *Genetic Algorithm* for having all the behavior of a GA, *Retroviral* for having retroviruses structures and *Iterative* because occurs every single generation.

2 Biological Basement: From Viruses to Retroviruses

Viruses are compulsory intracellular parasites with a very simple structure. Their acceptance as life forms is very controversial, since they are very different from the

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 220–228, 2010.

most simple bacteria and they have unique features, like the absence cell membrane, they don't have any known organelles and their size is several smaller, thus, the only possible way to see them is by electronic microscopy. They are also metabolically inert unless they are inside a host cell. It is important to notice also that they cannot contain simultaneously DNA and RNA molecules [3].

The viruses are formed basically by two components: the capsid, consisting of viral proteins, and the core, which contains their genetic information; the combination of these two structures is known as nucleocapsid. The main objective of viruses is to replicate themselves. To achieve this, they need to penetrate a host cell, make copies of themselves and put those copies out of the host cell.

2.1 Retroviruses

Retroviruses are the only known entities that are able to convert RNA into DNA under normal circumstances. After the adsorption and the injection of their genetic material into the host cell, the process of retro transcription takes place in the cytoplasm of the infected cell, using the viral reverse transcriptase enzyme. This process will convert a single stranded molecule of RNA (ssRNA) into a double-stranded DNA (dsDNA) molecule that is larger than the original RNA and has a high error rate, creating DNAs sequences different of which should be [4].

The retroviruses replication process can be described basically in follow steps [1]:

1. Viral recognition by the receptors present in the host cell surface
2. Penetration into the host cell
3. Reverse transcription (RNA to DNA)
4. Viral integration to the host's genome, where it will replicate
5. Viral DNA translation (produces viral mRNA that will translated in viral proteins)
6. Viral assembling
7. Viral shedding, when the new viruses leave the host cell

One of the proteins of the virus is the integrase, which is still associated with provirus. This enzyme cuts the chromosomal DNA of the host cell and inserts the viral converted DNA, integrating the provirus into the host cell chromosome. (Fig. 1). The next time this infected cell divides, the provirus will be replicated to the daughter cells [1].

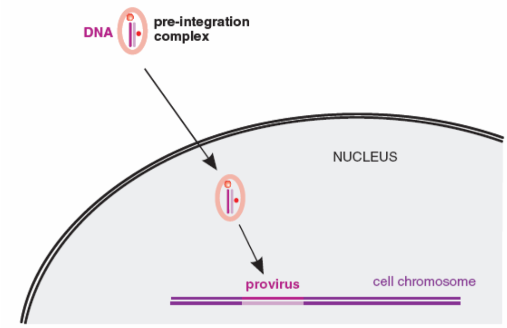


Fig. 1. Provirus creation and reverse transcription

After the viral genome is integrated in the host cell genome, the virus will be totally dependent of the cellular metabolism to continue its process of transcription, translation, genome replication, viral assembling and shedding.

The concepts of Charles Darwin about reproduction and Natural Selection applied to organics forms are also applied to viruses. Even though their acceptance as an organic form is very questionable, the viruses have genes that are striving to perpetuate the species. The main mechanisms used for viral evolution are mutation, recombination, reassortment and acquisition of cellular genes [1].

3 Genetic Algorithms

Genetic algorithms are part of probabilistic techniques that try to find different solutions in different executions with the same parameters, and sometimes with the same population [5]. Some of the main advantages of GA are [2]:

1. Optimization of discrete and continues values
2. Simultaneous search
3. Possibility to work with many variables
4. Provide a set of optimum variables, instead of only specific solution

The fundamental principle is to explore a population of chromosomes inside a search space, whose evolution depends on mutation and crossover operations, as in the natural evolutionary process [5].

The basic GA steps are:

1. Initialize random population
2. Evaluate chromosomes and check solution
3. Selection to crossover
4. Perform crossover
5. Mutation
6. Evaluate the chromosomes and check solution, if the solution wasn't found then back to step 2

3.1 Genetic Algorithm with Viral Infection (Related Works)

The viral infection is not an innovation in GA. It was discussed other times like in VEGA [10] and GAVI [6]. In both methods is used another population composed by virus, called viral population and infection of chromosomes called transcription [6][10].

In VEGA the viral population is a subset of chromosomes, created from initial hosts [10].

During the process of infection, a virus is selected by the same process of rank. However, as the virus has no fitness value, because it doesn't have a complete solution to be evaluated, it is used a parameter of infection, called *fitvirus* [10]. This is an indicator of how well the virus has acted. After selecting the virus that will perform the infection (in a particular chromosome), the transcription is made in this moment, which consists in the modification of infected chromosome by the viral information that contains a section similar to the one represented by the infecting agent [6][10].

When a virus is created or modified, its infectivity level is set at a fixed initial value. If the virus infects a chromosome (increase chromosome's fitness) their level of infectivity is increased by 1 (one), otherwise (if turn down the fitness) this value is reduced by 1 (one). If the virus infectivity value reaches 0 (zero) it discards its own parts and copies a part of chromosome for itself [6].

The main difference between VEGA and GAVI is because GAVI uses viral infection as operator, ignoring the operator of mutation, and VEGA is a complete GA [6][10].

4 Retroviral Iterative Genetic Algorithm

The main reason for the use of viral structure in the algorithm is the fact that these viruses are associated with a source of genetic innovation, which is influenced by the rapid rate of replication and changing [9].

For biological inspiration of RIGA, the family of retroviruses was chosen. These viruses do not possess correction mechanisms to undo possible genetic mutations that occur naturally during viral multiplication, which causes a high mutation rate, arising genetically modified individuals at each generation, what is considered an important characteristic during the processing time of GA [2][8]. The acquisition of cellular genes was chosen as a method to viral evolution, since it is quite common in retroviruses [8].

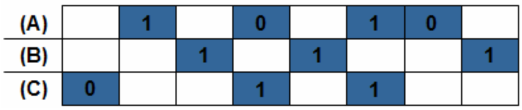
There are many differences between RIGA, GAVI and VEGA, some of them:

1. RIGA doesn't change any GA component, GAVI remove the mutation operation
2. In the GAVI the worst viruses has them genetic material changed, in the RIGA they are completely changed, thereby, the viral population is constantly remade, increase the possibility of infection in chromosome population
3. The biological basement of RIGA is very specific for the use of retroviral structure
4. VEGA creates virus only from host chromosomes, RIGA creates virus from host chromosomes too, but, uses the main concept of a retroviruses: high mutation rate
5. VEGA and GAVI handle a virus as a sequenced subset of chromosome, in the other hand, RIGA handle virus with dispersed information
6. The virus lifecycle in RIGA is well-defined

In the next session are presented the components of RIGA.

4.1 Viruses

Viruses in RIGA are structures that have the same size of a chromosome, however, with random empty spaces, because the idea is to share genetic material and avoid other population of chromosomes working in parallel. The amount of empty spaces and its positions are determined randomly. Thus for a problem that requires a binary representation of eight positions, some viruses have information as can seen in Fig 2.

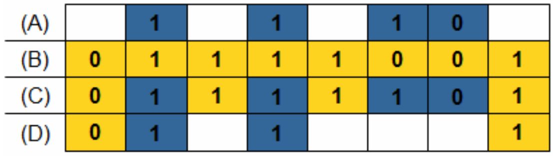


(A)		1		0		1	0	
(B)			1		1			1
(C)	0			1		1		

Fig. 2. Possible viruses for a binary chromosome with eight positions

4.2 Viral Population and Creation of New Viruses

For creation of new viruses that will compose the viral population, RIGA was inspired by the natural process common in retroviruses called reverse transcription. The process consists basically of the steps showed in Fig 3.

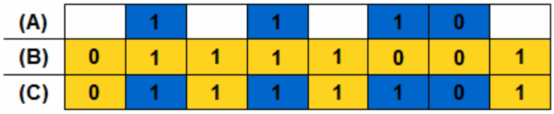


(A)		1		1		1	0	
(B)	0	1	1	1	1	0	0	1
(C)	0	1	1	1	1	1	0	1
(D)	0	1		1				1

Fig. 3. Creating a new virus process(A) Random virus (B) Chromosome from population (C) Auxiliary chromosome contained the mix of genetic material (D) New virus contained genetic material from virus and host.

4.3 Infection

Infection is the process of inclusion of the viral genetic material into the host chromosome, which is required a virus and a chromosome. The target chromosome will have changed their genetic material in the same positions where the genes are arranged on the virus, so all the viral genetic information, will be copied to the target chromosome, excepting the empty spaces, which will be filled by the host chromosome. The RIGA infection is represented in Fig 4.

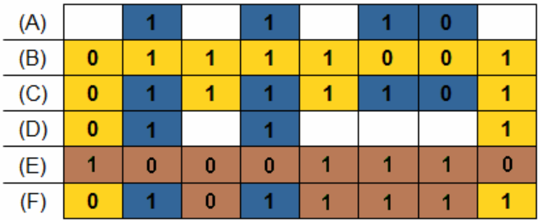


(A)		1		1		1	0	
(B)	0	1	1	1	1	0	0	1
(C)	0	1	1	1	1	1	0	1

Fig. 4. Chromosome infection (A) Virus (B) Chromosome (C) Infected Chromosome

The general view about the creation and infection process is represented in Fig 5. In the same figure is possible verify the exchange between genetic material of all structures involved. It's possible verify Fig 5 as well, that the new virus (D) is made from an chromosome (B) infected by a virus (A) of viral population. The auxiliary chromosome (C) is made to be template to virus (D).

The infection process depends exclusively on one single factor: increasing chromosomal fitness. The infection is successful when an infected chromosome has an increase on its fitness and unsuccessful when it has a decrease on its fitness. This is an



(A)		1		1		1	0	
(B)	0	1	1	1	1	0	0	1
(C)	0	1	1	1	1	1	0	1
(D)	0	1		1				1
(E)	1	0	0	0	1	1	1	0
(F)	0	1	0	1	1	1	1	1

Fig 5. Creating a new virus and infection process (A) Random Virus (B) Chromosome from population (C) Auxiliary chromosome contained the mix of genetic material (D) New virus contained genetic material from virus and host (E) Target chromosome (F) Infected chromosome.

important factor because it determines which viruses will infect the next generation. The viruses with less infection rates will be extinct. For RIGA it is important to restrict the infection only when there is an increase in the chromosomal fitness, because, if any infection was considered, good chromosomes could turn into bad ones. Thus, only the successful infections are important to RIGA.

4.4 Parameters

The RIGA uses the same parameters from classic GA (number of individuals, rate of mutation, crossover and elitism and the type of selection and crossover). However, to apply the concepts of viral infection by retroviruses, the defined parameters are:

1. Infection Population rate: the rate of chromosomes that will be infected
2. Viral Elitism rate: the rate of viruses that will be kept in the next viral population
3. Number of Viruses: the number of viruses of viral population
4. Weakest Infection: this parameter forces the infection of weakest chromosome
5. Single Infection: this parameter forces a unique infection per chromosome
6. Internal Infection rate: this parameter indicates the maximum percentage of genetic material from any chromosome that will form a new virus

4.5 The Algorithm

RIGA has an additional step in the traditional algorithm (step 6), above:

1. Initialize random population
2. Evaluate chromosomes and check solution
3. Evaluate selection to crossover
4. Effects crossover
5. Mutation
6. Viral application
    a. If it is the first time, generate a random viral population
    b. Generate new virus based in chromosome population
    c. Infect chromosomes
        i. Infect each chosen chromosome with each existent virus

ii. For each successful infection, increment 1 to viral fitness, however, decrement 1

iii. Check the virus with the highest level infection rate and keep according with viral elitism rate

7. Evaluate the chromosomes and check solution, if the solution wasn't found then back to step 2

5 Results

To the tests of RIGA, the implementations were based to support the functions: F1 (Shifted Sphere Function), F2 (Shifted Schwefel's Problem), F3 (Shifted Rotated High Conditioned Elliptic Function) and F5 (Schwefel's Problem 2.6 with Global Optimum on Bounds) [7].

Considering all performed experiments (a total of 80 executions for 40 to RIGA and 40 to GA with same configurations and initial population), the RIGA was superior to GA in all the experiments, getting closer to the result in all of them. In the figures Fig. 6, Fig. 7, Fig. 8 and Fig. 9 are the results of the best fitness of each execution.

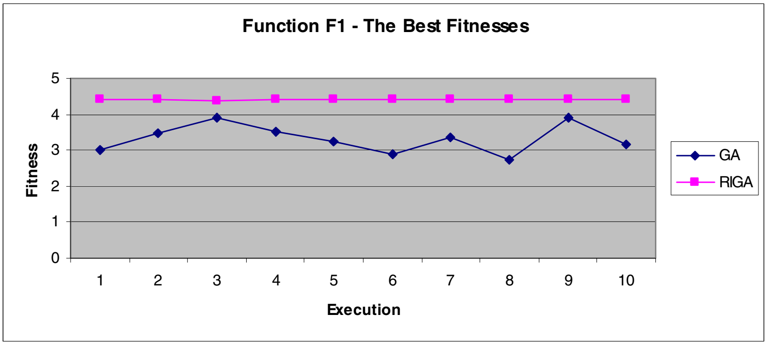


Fig. 6. The best fitnesses for function F1

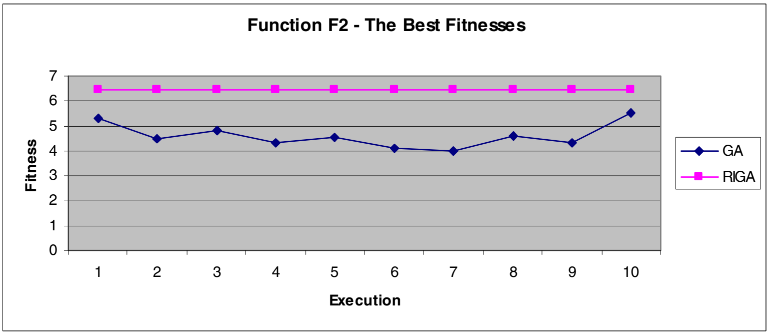


Fig. 7. The best fitnesses for function F2

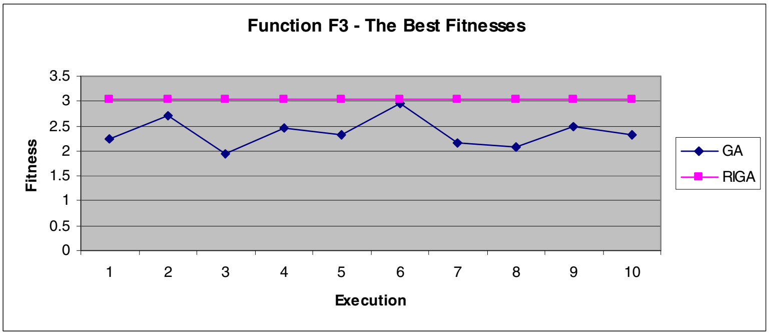


Fig. 8. The best fitnesses for function F3

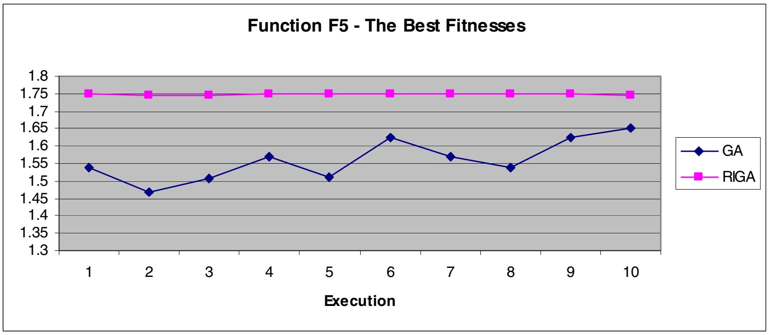


Fig. 9. The best fitnesses for function F5

According to Table 1, it is possible verify that the average fitness of the RIGA was even higher (F1, F2, F3 and F5) than the best fitnesses found in some executions by the GA. Thus, it can affirm that the RIGA, with the same population, elevates the fitness of individuals considered bad, getting closer to an optimal result in only 100 generations, which may be verified by average of the standard deviations generated by RIGA which were lower in all cases.

Table 1. The best executions for functions F1, F2, F3 and F5. (BF) Best Fitness (AF) Average Fitness

CEC-LIB		GA		RIGA	
Function	Max(f(x))	BF	AF	BF	AF
F1	4,4 x 10^4	**3,9214**	1,0015	**4,3893**	3,9341
F2	6,5 x 10^4	**5,4998**	1,4292	**6,4784**	5,6315
F3	3,8 x 10^{10}	**2.9530**	0.5893	**3.0421**	2.7696
F5	1,8 x 10^4	**1.6513**	0.6740	**1.7486**	1.6474

Thereby, it is correct affirm that the method (RIGA) was more efficient for mathematical problems exposed. However, it's not possible assure if the method exposed is effective for all GA's situations.

6 Conclusions

This work presented the creation of a genetic algorithm based on the structure of the retroviruses family. The biological application in conjunction with genetic algorithm took this algorithm to be called Retroviral Iterative Genetic Algorithm.

After the present work, it was found that the retroviral biological inspiration had a beneficial effect in the application of RIGA compared to the GA, because all the RIGA results were significantly better when compared to GA results.

Thus, with positive results obtained by RIGA, it is conclude that RIGA proves be efficient when compared to GA for the functions exposed at work. What should not be taken as a rule for any problem involving resolutions from GA's problem.

References

1. Carter, J., Saunders, V.: Virology Principles and Applications. John Wiley & Sons Ltd., England (2007)
2. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. John Wiley & Sons Ltd., England (1998)
3. Hogg, S.: Essential Microbiology. John Wiley & Sons Ltd., Chichester (2005)
4. Agut, A., Um Sistema Estratégico De Reprodução.: Scientific American Brasil. Edição Especial 28, 14–19 (2009)
5. Linden, R.: Algoritmos Genéticos, 1st edn. Brasport, Rio De Janeiro (2006)
6. Guedes, A., Leite, J., Aloise, D.: Um Algoritmo Genético Com Infecção Viral Para O Problema Do Caixeiro Viajante (2005)
7. Suganthan, P., Hansen, N., Liang, J., Deb, K., Chen, Y., Auger, A., Tiwari, S.: Problem Definitions And Evaluation Criteria For The Cec 2005 Special Session On Real-Parameter Optimization. Technical Report, Nanyang Technological University, Singapore and Kangal Report Number 2005005 (Kanpur Genetic Algorithms Laboratory, Iit Kanpur) (2005)
8. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1999)
9. Villarreal, L., Virus São Seres Vivos.: Scientific American Brasil. Edição Especial 28, 21–24 (2009)
10. Kubota, N., Fukuda, T., Shimojima, K.: Virus-evolutionary genetic algorithm for a self-organizing manufacturing system. Computers & Industrial Engineering 30, 1015–1026 (1996), doi:10.1016/0360-8352(96)00049-6

The RM-MEDA Based on Elitist Strategy

Li Mo, Guangming Dai, and JianKai Zhu

Scholl of Computer, China University of Geosciences, Wuhan 430074, China
moliwh@gmail.com

Abstract. The Estimation of Distribution Algorithms(EDAs) is a new paradigm for Evolutionary Computation. This new class of algorithms generalizes Genetic Algorithms(GAs) by replacing the crossover and mutation operators by learning and sampling the probability distribution of the best individuals of the population at each iteration of the algorithm. In this paper, we review the EDAs for the solution of combinatorial optimization problems and optimization in continuous domains. The paper gives a brief overview of the multiobjective problems(MOP) and estimation of distribution algorithms(EDAs). We introduce a representative algorithm called RMMEDA (Regularity Model Based Multi-objective Estimation of Distribution Algorithm). In order to improve the convergence performance of the algorithm, we improve the traditional RM-MEDA. The improvement we make is using part of the parent population with better performance instead of the entire parent population to establish a more accurate manifold model, and the RM-MEDA based on elitist strategy theory is proposed. Experimental results show that the improved RM-MEDA performs better on the convergence metric and the algorithm runtime than the original one.

Keywords: EDA, RMMEDA, Elitist Strategy.

1 Introduction

Many optimization problems, in scientific research and engineering areas, belong to multi-objective optimization problems. Multi-objective problems are different with single objective problems, the optimal solutions of multi-objective problem is a set of solutions. As multi-objective evolutionary algorithms themselves are based on the population and the solutions from them are non-dominated solutions set, they are becoming the most effective approach to solve multi-objective problems. In recent decades, multi-objective evolutionary algorithms became hot spots and achieved many good results in multi-objective area.

Since the publication of Schaffer's seminal work [1], a number of evolutionary algorithms (EDAs) have been developed for multi-objective optimization problems [2]-[5]. The major advantage of these multi-objective evolutionary algorithms (MOEAs) over other methods are that they work with a population of candidate solutions and thus can produce a set of Pareto optimal solutions to approximate the Pareto front and Pareto set in a single run.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 229–239, 2010.

Estimation of distribution algorithms (EDAs) are a new computing paradigm in evolutionary computation [6]. There is no crossover or mutation in EDAs. Instead, they explicitly extract globally statistical information from the selected solutions and build a posterior probability distribution model of promising solutions, based on the extracted information. New Solutions are sampled from the model thus built and, fully or in part, replace the old population. Several EDAs have been developed for continuous MOPs [7]-[9]. However, these EDAs do not take the regularity into consideration in building probability models. Since probability modeling techniques under regularity have been widely investigated in the area of statistical learning [10], [11], it is suitable to take advantage of the regularity in the design of EDAs for a continuous MOP. In 2007, Qingfu Zhang etc. developed a regularity model-based multi-objective estimation of distribution (RM-MEDA) [12]. RM-MEDA captures and utilizes the regularity of the Pareto set in the decision space. Systematic experiments have show that, overall RM-MEDA outperforms GDE3 [13], PCX-NSGA-II [14] and MIDEA [8], on a set of test instances with variable linkages.

To obtain more accurate Pareto front, we improved the RM-MEDA. When establishing the manifold model of the population with N solutions, we didn't take the entire one but only half of it, which is a new population with just $N/2$ solutions that perform better, that is they are closer to the real PS manifold. Experiments have been conducted to compare the proposed algorithm with the traditional algorithm on a set of biobjective or triobjective test instances with linear or nonlinear variable linkages. Experiments results show that the improved RM-MEDA performs better than the original RM-MEDA on the convergence metric and the CPU-time costs.

Section 2 introduces continuous multi-objective optimization problems, Pareto optimality. Section 3 shows the ideas of the original RM-MEDA. Section 4 presents the motivation and the details of our improved RM-MEDA. Section 5 presents introduces the convergence and diversity metrics and shows the experiment results.

2 Problems Definition

In this paper, we consider the following continuous multi-objective optimization problem (continuous MOP):

$$\begin{aligned} &\min \vec{F}(x) = (f_1(x), f_2(x), \cdots, f_m(x))^T \\ &s.t. \quad x \in X \end{aligned} \tag{1}$$

where $X \subset R^n$ is the decision space and $x = (x_1, \cdots, x_n)^T \in R^n$ is the decision variable vector. $\vec{F}: X \rightarrow R^m$ consists of m real-valued continuous objective functions $f_i(x)$ $(i = 1\ ,2, \cdots, n)$. R^m is the objective space.

Let $a = (a_1, \cdots, a_m)^T, b = (b_1, \cdots, b_m)^T \in R^m$ be two vectors, a is said to dominate b, denoted by $a \prec b$, if $a_i \le b_i$ for all $i = 1, \cdots, n$, and $a \ne b$. A point

$x^* \in X$ is called (globally) Pareto optimal if there is no $x \in X$ such that $\vec{F}(x) \prec \vec{F}(x^*)$. The set of all the Pareto optimal points, denoted by PS, is called the Pareto set. The set of all the Pareto objective vectors, $PF = \{y \in R^m \mid y = \vec{F}(x), x \in PS\}$, is called the Pareto front [2], [15].

3 Traditional RM-MEDA

3.1 Theoretical Foundation

Under certain smoothness assumptions, it can be induced from the Karush-Kuhn-Tucker condition that the PS of a continuous MOP defines a piecewise continuous (m-1)-dimensional manifold in the decision space [16], [17]. Therefore, the PS of a continuous biobjective optimization problem is a piecewise continuous curve in R^n, while the PS of a continuous MOP with three objectives is a piecewise continuous surface.

3.2 Basic Idea

EDAs build a probability model for characterizing promising solutions in the search space based on statistical information extracted from the previous search and then sample new trial solutions from the model thus built.

The population in the decision space in an EA for (1) will hopefully approximate the PS and be uniformly scattered around the PS as the search goes on. Therefore, the algorithm envisage the points in the population as independent observations of a random vector $\xi \in R^n$ whose centroid is the PS of (1). Since the PS is an (m-1)-dimensional piecewise continuous manifold, ξ can be naturally described by

$$\xi = \zeta + \varepsilon \tag{2}$$

where ζ is uniformly distributed over a piecewise continuous (m-1)-dimensional manifold, and ε is an n-dimensional zeromean noise vector.

3.3 Algorithm Framework

At each generation t, the RM-MEDA maintains:

- a population of N solutions (i. e. points in X)

$$Pop(t) = \{x^1, x^2, \cdots, x^N\}$$

- their $\vec{F}$ values: $\vec{F}(x^1), \vec{F}(x^2), \cdots, \vec{F}(x^N)$

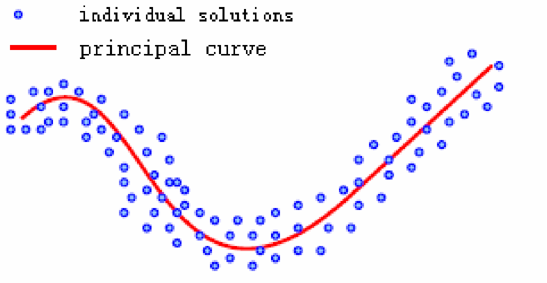


Fig. 1. Illustration of the basic idea. Individual solutions should be scattered around the PS in the decision space in a successful MOEA.

The algorithm works as follows.

RM-MEDA

Step 0) Initialization: Set $t := 0$. Generate an initial population $Pop(0)$ and compute the $\vec{F}$ values of each individual solution in $Pop(0)$.

Step 1) Stopping Condition: If stopping condition is met, stop and return the nondominated solutions in Pop(t) and their corresponding $\vec{F}$ vectors. All these $\vec{F}$ vectors constitute an approximation to the PF.

Step 2) Modeling: Build the probability model (2) for modeling the distribution of the solutions in $Pop(t)$.

Step 3) Reproduction: Generate a new solution set Q from the model (2). Evaluate the $\vec{F}$ value of each solution in Q.

Step 4) Selection: Select N individuals from $Q \bigcup Pop(t)$ to create $Pop(t+1)$.

Step 5) Set $t := t+1$ and go to **Step 1.**

4 RM-MEDA Based on Elitist Strategy

4.1 Basic Idea

Traditional RM-MEDA build probability model (2) from the $Pop(t)$ with N solutions. Firstly, we considered that the N solutions in the $Pop(t)$ also have "good" or "bad", and the "good" ones are the non-dominated solutions which we would like to find in the end. Then we can just don't use the entire $Pop(t)$ but only the "good" ones or better ones to build more accurate manifold model, which could better characterize promising solutions in the search space, and approximate the real PS more closely! So it can enhance the convergence of the algorithm. Secondly, when running the process, each generation should establishment model (2), which leads to a high degree of time complexity, then if we reduce the number of the solutions to build probability model (2), it would simply reduce the algorithm's complexity and improve the algorithm speed.

4.2 Algorithm Framework

Based on the basic idea above, we take some improvements during each generation on building the manifold model (2).

The improved algorithm works as follows:

RM-MEDA base on elitist strategy

Step 0) Initialization: Set $t := 0$. Generate an initial population $Pop(0)$ and compute the $\vec{F}$ values of each individual solution in $Pop(0)$.

Step 1) Stopping Condition: If stopping condition is met, stop and return the nondominated solutions in Pop(t) and their corresponding $\vec{F}$ vectors. All these $\vec{F}$ vectors constitute an approximation to the PF.

Setp 2) Selection: Sort the N solutions in $Pop(t)$ based on the nondominated sorting of NSGA-II [14]. Then make a new population $Pop'(t)$ with $N/2$ better solutions in $Pop(t)$.

Step 3) Modeling: Build the probability model (2) for modeling the distribution of the solutions in $Pop'(t)$.

Step 4) Reproduction: Generate a new solution set Q from the model (2). Evaluate the $\vec{F}$ value of each solution in Q.

Step 5) Selection: Select N individuals from $Q \bigcup Pop(t)$ to create $Pop(t+1)$.

Step 6) Set $t := t+1$ and go to Step1.

5 Comparison Studies

5.1 Performance Metric

In order to measure the algorithm numerically, we used two metrics γ and Δ to evaluate the algorithm [17].

$$\gamma = \sum_{i=0}^{H} \min d(f, f_{i,front}) / n \tag{3}$$

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} | d_i - \overline{d} |}{d_f + d_l + (N-1)\overline{d}} \tag{4}$$

The first metric γ measures the extent of convergence to a known set of Pareto optimal solutions. The smaller γ is, the better convergence toward the Pareto optimal front. The second metric Δ measures the extent of spread achieved among the obtained solutions. The smaller Δ is, the better diversity of the optimal solutions.

5.2 General Experimental Setting

All algorithms is implemented in C. The machine used in our experiments is Core 2 Duo T6570 (2.1GHz, 2.00GB RAM). In this section, the experimental setting is as follows.

- The number of new trial solutions generated at each generation: The number of new solutions generated at each generation in the two algorithms is set to be 200 for all the test instances with two and three objectives.
- The number of decision variables: It is set to be 30 for all the test instances.
- In Local PCA algorithm, K , the number of clusters, is set to be 5.
- Number of runs and stopping condition: We run each algorithm independently 10 times for each test instance.
- Initialization: Initial populations in all the algorithms are randomly generated.

5.3 Test Instances

The test instances we used is the same as in the traditional RM-MEDA[12]. They are ten instances named ZZJ07_ F1-9[17]. ZZJ07_F1-4 are problems with linear PS shapes, and ZZJ05-9 with Quadratic PC shapes.

F1-F4 are variants of ZDT1, ZDT2, ZDT6[17], and DTLZ2[18], respectively. Due to $g(x)$ used in these instances, F1-F3 have the same PS . Their PS is a line segment

$$x_1 = \cdots = x_n,\ 0 \le x_i \le 1,\ 1 \le i \le n.$$

The PS of F4 is a 2-D rectangle

$$x_1 = x_3 = \cdots = x_n,\ 0 \le x_i \le 1,\ 1 \le i \le n.$$

There are linear variable linkages in these test instances. The variable linkages in these instances [14] are obtained by performing the following linear mapping on the variables in the original ZDT and DTLZ instances:

$$x_1 \rightarrow x_1,\ \ x_i \rightarrow x_i - x_1,\ i = 2, \cdots, n.$$

F5-F8 are test instances with nonlinear variable linkages. The PS of F5-F8 is a bounded continuous curve defined by

$$x_1 = x_i^2,\ \ i = 2, \cdots, n.\ \ 0 \le x_1 \le 1.$$

The PS of F8 is a 2-D bounded continuous surface defined by

$$x_1 = x_i^2,\ \ i = 3, \cdots, n.\ \ 0 \le x_1, x_2 \le 1.$$

The variable linkages in these instances [19] are obtained by performing the following nonlinear mapping on the variables in the original ZDT and DTLZ instances:

$$x_1 \to x_1,\ \ x_i \to x_i^2 - x_1,\ i = 2, \cdots, n.$$

F9 are nonlinear variable linkages. Further more, this instance have many local Pareto fronts since their $g(x)$ has many locally minimal points.

5.4 Test Instances with Linear Variable Linkages

We first compare improved RM-MEDA with traditional RM-MEDA on four continuous MOPs with linear variable linkages. The test instances F1-F4 are used for this purpose.

Table 1-4 show the evolution of the average and standard deviation γ and Δ metric of the non-dominated solutions in the current populations among 10 independent runs with the number of function evaluations in four algorithms. It is clear from the following results that improved RM-MEDA performs better than traditional RM-MEDA on convergence metric γ on all these four instances.

We notice that the improved RM-MEDA performs much better on F3. As we know, F3 is the hardest among these four instances. The distribution of the Pareto optimal solutions in the object space in this instance is very different from that in the other three. If we uniformly sample a number of points in the PS of F3 in the decision space, most of the corresponding Pareto optimal vectors in the objective space will be more likely to be in the left apart of the PF. This makes it very difficult for an algorithm to approximate the whole PF. Also traditional RM-MEDA performs better on F3, but our improved RM-MEDA performs much better than it! All this could be attributed to the fact that improved RM-MEDA does extend along the principal directions more closely so that it has a good chance of approximation the whole PF, and the elitist strategy makes the manifold model more closely to the real PS.

Table 1. The statistic results on F1 on 10 runs

Algorithm	*Convergence* γ	*Diversity* Δ
traditional RM-MEDA	0.001638 ± 0.000945397	0.5384 ± 0.035587428
improved RM-MEDA	0.001284 ± 0.000120017	0.54896 ± 0.02698283

Table 2. The statistic results on F2 on 10 runs

Algorithm	*Convergence* γ	*Diversity* Δ
traditional RM-MEDA	0.000806 ± 5.004E-05	0.533682 ± 0.036616846
improved RM-MEDA	0.000788 ± 5.17301E-05	0.524034 ± 0.03505074

Table 3. The statistic results on F3 on 10 runs

Algorithm	*Convergence*	*Diversity*
	γ	Δ
traditional RM-MEDA	0.45525 ± 0.293599102	0.716244 ± 0.086163036
improved RM-MEDA	0.338337 ± 0.121726532	0.799542 ± 0.04512084

Table 4. The statistic results on F4 on 10 runs

Algorithm	*Convergence*	*Diversity*
	γ	Δ
traditional RM-MEDA	0.128035 ± 0.022854906	0.701646 ± 0.062609226
improved RM-MEDA	0.109137 ± 0.008761657	0.673993 ± 0.039923816

5.5 Test Instances with Nonlinear Variables Linkages

We compare improved RM-MEDA with traditional RM-MEDA on four continuous MOPs with nonlinear variable linkages. F5-F8 are used for this purpose.

Table 5-8 show the results with γ and Δ metric.

Table 5. The statistic results on F5 on 10 runs

Algorithm	*Convergence*	*Diversity*
	γ	Δ
traditional RM-MEDA	0.001706 ± 0.000161753	0.552248 ± 0.024664749
improved RM-MEDA	0.001655 ± 0.000140801	0.52892 ± 0.03440603

Table 6. The statistic results on F6 on 10 runs

Algorithm	*Convergence*	*Diversity*
	γ	Δ
traditional RM-MEDA	0.001712 ± 0.000157594	0.533772 ± 0.025690472
improved RM-MEDA	0.00186 ± 9.95992E-05	0.551718 ± 0.02569711

Table 7. The statistic results on F7 on 10 runs

Algorithm	*Convergence*	*Diversity*
	γ	Δ
traditional RM-MEDA	0.541667 ± 0.046831853	0.999692 ± 4.93559E-05
improved RM-MEDA	0.11812 ± 0.065838184	0.999806 ± 5.9867E-05

Table 8. The statistic results on F8 on 10 runs

Algorithm	***Convergence*** γ	***Diversity*** Δ
traditional RM-MEDA	0.128035 ± 0.022854906	0.701646 ± 0.062609226
improved RM-MEDA	0.109137 ± 0.008761657	0.673993 ± 0.039923816

The experimental results show that improved RM-MEDA outperforms the traditional RM-MEDA on all these instances on convergence γ except for the instance F6. However, the difference is very small. And what encourages us is the result on F7. F7 is the hardest instance for the traditional RM-MEDA due to the fact that the Pareto optimal solutions of F7 are not uniformly distributed in its linkage counterpart. The RM-MEDA based on elitist strategy takes the better half solutions of the whole populations to create the manifold model may be the key factor for this.

5.6 Test Instances with Many Local Pareto Fronts

Here we compare improved RM-MEDA with traditional RM-MEDA on a continuous MOPs with nonlinear variable linkages. Furthermore, this instance has many local Pareto fronts. Table 9 shows the results with γ metric and Δ metric.

Table 9. The statistic results on F9 on 10 runs

Algorithm	***Convergence*** γ	***Diversity*** Δ
traditional RM-MEDA	0.018754 ± 0.005322969	0.588791 ± 0.059738113
improved RM-MEDA	0.021819 ± 0.010345351	0.603867 ± 0.08466081

However, improved RM-MEDA falls back with the traditional RM-MEDA over F9. This is due to the fact that this instance has many local Pareto fronts. Since improved RM-MEDA choice the elite individuals and abandoned the other individuals, this may loses diversity metric of populations to some extent, so the algorithm is more easily trapped into local optimum. This may be the shortcoming of the RM-MEDA based on elitist strategy.

5.7 CPU-Time Cost

The CPU times used by traditional RM-MEDA and improved RM-MEDA with the same experimental settings are given in Table 10.

Table 10. The CPU-Time Cost on one runs

Instance	*Traditional RM-MEDA run time(s)*	*improved RM-MEDA run time(s)*
F1	2041.343	1112.938
F2	1890.234	989.25
F3	896.813	331.719
F4	920.172	587.844
F5	1168.172	686.61
F6	1148.079	494.172
F7	880.391	205.391
F8	950.25	165.672
F9	1203.532	450.36

It's clear that traditional RM-MEDA needs extra CPU time compared with the improved RM-MEDA. This is all because that traditional RM-MEDA needs to run Local PCA on a whole population at each generation while improved RM-MEDA just run on half elitist population, then at least half the time can be reduced.

6 Conclusion

In this paper, the traditional RM-MEDA has been improved. Experimental results show that the RM-MEDA based on elitist strategy performs better to the Traditional RM-MEDA on the convergence metric. This is mainly due to the strategy that we use half better solutions of parent population instead of the whole one to create the manifold model, which would produce a more imminent manifold closely to the real PS manifold. Also the algorithm's time complexity will be reduced. However, since we abandoned for half of the population to model, the diversity of the final solutions along the PS has a little decline, therefore the decrease is not significantly obvious. On some instances the improved RM-MEDA also performs even better. Overall, the RM-MEDA based on elitist strategy can really enhance its convergence capabilities, and reduce the algorithm's time complexity. In our following studies, we would study how to improve the convergence of the solutions while maintaining the diversity of them. Furthermore, the percentage of the parent population we chose to create the probability model will be studied to get the best performance.

Acknowledgment

This paper is supported by the Natural Science Foundation No.60873107, the National High Technology Research and Development Program No.2008AA12A201, and Natural Science Foundation of Hubei Province No.2008CDB348.

References

[1] Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: Proc. 1st Int. Conf. Genetic Algorithms, Pittsburgh, PA, pp. 93–100 (1985)

[2] Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Baffins Lane (2001)

[3] Coello Coello, C.A., van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for solving Multi-Objective Problems. Kluwer, Norwell (2002)

[4] Tan, K.C., Khor, E.F., Lee, T.H.: Multiobjective Evolutionary Algorithms and Applications. Springer, Heidelberg (2005)

[5] Knowles, J., Corne, D.: Memetic algorithms for multiobjective optimization: Issues, methods and prospects. In: Recent Advances in Memetic Algorithms. Studies in Fuzziness and Soft Computing, vol. 166, pp. 313–352. Springer, New York (2005)

[6] Larranaga, P., Lozano, J.A. (eds.): Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Norwell (2001)

[7] Okabe, T., Jin, Y., Sendhoff, B., Olhofer, M.: Voronoi-based estimation of distribution algorithm for multi-objecbive optimization. In: Proc. Congr. Evol. Comput (CEC 2004), Portland, OR, pp. 1594–1601 (2004)

[8] Bosman, P.A.N., Thierens, D.: The naive MIDEA: A baseline multi-objective EA. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 428–442. Springer, Heidelberg (2005)

[9] Pelikan, M., Sastry, K., Goldberg, D.: Multiobjective HBOA, clustering, and scalability. Illinois Genetic Algorithms Laboratory (IlliGAL), Tech. Rep. 2005005 (2005)

[10] Cherkassky, V., Mulier, F.: Learning from Data: Concepts. Theory. and Methods. Wiley, New York (1998)

[11] Hasite, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediciton. Springer, Berlin (2001)

[12] Zhang, Q., Zhou, A., Jin, Y.: RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm. IEEE Transactions on Evolutionary Computation 12(1), 41–63 (2008)

[13] Kukkonen, S., Lampinen, J.: GDE3: The third evolution step of generalized differential evolution. In: Proc. Congr. Evol. Comput (CEC 2005), Edinburgh, U.K., pp. 443–450 (2005)

[14] Deb, K., Pratap, A., Agarwal, S., Meyaarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6, 182–197 (2002)

[15] Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer's International Series in Operations Research & Management Science, vol. 12. Kluwer, Norwell (1999)

[16] Schutze, O., Mostaghim, S., Dellnitz, M., Teich, J.: Covering Pareto sets by multilevel evolutionary subdivision techniques. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 118–132. Springer, Heidelberg (2003)

[17] Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genitic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Trans. Evol. Comput. 7, 204–223 (2003)

[18] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary Multiobjective Optimization, Theoretical Advances and Applications, pp. 105–145. Springer, New York (2005)

[19] Li, H., Zhang, Q.: A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 583–592. Springer, Heidelberg (2006)

A Proactive Perception Model in Agent-Based Software Integration and Evolution

Qingshan Li, Chengguang Zhao, Haishun Yun, and Lili Guo

Software Engineering Institute, Xidian Univ., Xi'an, China, 710071
qshli@mail.xidian.edu.cn

Abstract. Introducing agent techniques into system integration and software evolution can solve the flexible and dynamic problems in the process of system integration. A Capability-Centered Proactive Perception Agent Model (CCPPA) is introduced in this paper. At first agent capability is defined, based on which two proactive perception behaviors are designed in CCPPA. One is active perceiving capability behavior and the other is active perceiving tender behavior. These two behaviors give a good support to agent collaboration under acquaintance protocol or contract net protocol. An improved Contract Net Protocol based on Agent Active Perception tender (CNPAAP) introduced offers a good support of dynamical agent cooperation. Moreover, an Acquaintance Protocol based on Agent Active Perception capability (APAAP) is proposed for the stable cooperation among agents. Finally, CCPPA is applied in the Agent-based System Integration Tool, and has been tested in the Border and Coast Defense Simulation System. The result proves that CNPAAP not only solves the dynamic problem in system integration, but also improves the task completion efficiency when task number increases. As APAAP is comparatively more efficient than CNPAAP, it can be used where stable cooperation is needed.

Keywords: Agent; System integration; Muti-Agent System; Proactive Perception; Software Evolution.

1 Introduction

The Military command and control system (MCCS) requires an integration framework and a series of related supporting tools to provide effective and practical support of MCCS development and integration. Traditional MCCS integration focuses on utilizing component and plug-in techniques. It meets the system integration requirement at a certain degree. However, for the ever-changing needs, multiple departments oriented and dynamically integration requirements, there are still some problems.

An agent is a computer system that is situated in some environment and capable of acting autonomously on the environment in order to meet its design objectives [1, 2]. It is also used to denote a human or computer system software with the following properties: bounded autonomy, rationality, social ability, reactivity, cooperation and responsibility [3]. The above agent's characteristics enable agent to describe the information systems with the properties of entity autonomy, geographical dispersion,

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 240–251, 2010.

and interaction among members demand flexibility in a changing environment. Based on this, the agent technology is applied into the system integration.

Traditional agent models are divided into three types: reactive agent model, rational agent model and a hybrid of the above. Brooks first developed the idea of reactive agent model [4, 5] and proved its practical utility with robots experiments. The task strategy of reactive agent model is hierarchy. The low level tasks are atom behaviors which are similar to the human stress response. The high level tasks are composed task and can be divided into low level tasks. The reactive agent model is quite efficient in task execution. Although the reactive model embodies the process of reaction to its surrounding, it is often passive and is low in adaptability and portability. The rational agent model, originated from artificial intelligence, focuses on formal logic to describe agent [6]. The most influenced is Anand S. Rao and Michiael P. Georgeff's BDI model [7]. BDI model has a sound theoretical foundation, and many researchers and scholars, such as Shi Chunyi [8, 9] and Shi Zhongzhi [10] in China, have made expansion and application on it. Core idea of BDI is its internal goal oriented reasoning and planning process which is expressed in Belief, Desire and Intention. BDI model performs well in agent intelligence, but it is difficult to implement and has a certain distance for practical application.

In recent years, Michael Wooldridge put forward the VSK agent model [11]. It describes agent in the concepts of Visible, Perceive and Know. The VSK model also belongs to the rational agent model. However, it does not specify its internal structure, while it places emphasis on agent semantics under the MAS environment.

MCCS is a kind of complex information system. The real time information acquisition requires the agent model to be reactive, and for the dynamic integration requires the agent to be initiative to perceive and adapt join of new agent. This Paper, grounded on the integration of MCCS, provides a Capability-Centered Proactive-Perception Agent Model (CCPPA).

2 The Agent-Based Integration Framework

Many researchers in agent field have developed the multi-agent system development frameworks, such as ZEUS [12] and JADE [13], which are FIPA compliant and ABFSC [14] in china.

Referring to FIPA [15] platform specifications, we have developed our agent-based system integration tool (ASIT). The integrated framework of ASIT is illustrated in Figure 1. In the function of this framework, a number of executable programs (EXE) can be integrated with dynamic link libraries (DLL) from different systems. For a specific domain, the integrated development staff in the first place will perform analysis on it, extract its business process and abstract rules then add them to the integrated rules library.

The framework is built on a distributed system in which there is only one computer functioning as the control platform and the other computers as non-control platforms. The Capability Register Center (CRC), the Agent Management Service (AMS) and the Public Information Blackboard (PIB) just exist on the control platform. CRC stores the capabilities of all the agents in the system. AMS performs real time monitoring on each state of the agents (idle, running, waiting and resume). PIB is an intermediary serving for agents who initiate or respond to a negotiation. There

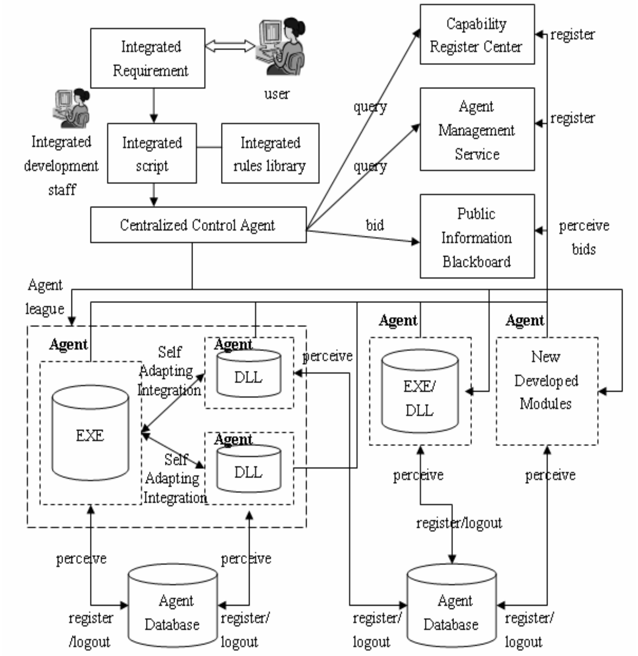


Fig 1. Agent-based integration framework

are two queues in PIB: the valid tender queue and the invalid tender queue. When a contract is signed by a manager and a contractor, the information of the signed contracts is recorded in the former. When the task announced in some tender has been finished by agents, the tender is put into the latter. There is an Agent Database on the control platform and each non-control platform which is used to perceive the agents on the platform. Once one agent is perceived, it is added in the local address book that exists in the Agent Database and starts running. Then the agent registers capabilities to CRC and address and state to AMS.

After users input their requirement in human-computer interface, the integrated development staffs will call the integrated rules from corresponding field and map to integrated script based on the users' demands. The centralized control agent as a special agent calls the script interpreter to explain the integrated script. When running into a simple capability of the script, the centralized control agent will send a message to CRC looking for agents with the capability. After receiving the searching result from CRC, the centralized control agent will send messages to them to request executing the task. If the agent holding the capability cannot complete the task by itself, it will request collaboration with other agents through the contract net protocol based on agent active perception. After the agent completes the task, it will inform the centralized control agent of the result for continuous execution of the next task. In this framework, the centralized control agent, CRC, AMS, PIB as well as other agents communicate with each other through information mechanism, which can support synchronous and asynchronous communication, and can bring about convenience to build distributed systems.

3 Architecture of CCPPA

3.1 Definition of CCPPA

Definition 3.1. Resource (referred to R), is defined as R = <Type, Name, Value>: Type denotes kind of resources; Name denotes the name of resources; Value denotes quantity of resources.

Definition 3.2. Agent Capability (short as C), is defined as C = <Precon, PostEffect, Desc, Const>, where, Precon refers to the input parameters and the precondition of the capability; PostEffect refers to the output result and effect after completing the capability; Prerequisite and effect are both resources; Desc gives some brief description information of the functionality of the capability; Cons refers to constraints of the capability, mainly refers to time and conditions constraints.

Definition 3.3. Agent Task (short as T), is defined as T = <c, f>, which $c\in$ C and f $\in$ Function, Function is the set of functionality procedures for completing the tasks. Task expresses the dynamic running process of ability, while ability gives a description about the functionality and constrains of task.

Definition 3.4. Composite Capability (short as CC), is defined as CC=<Input, Output, Desc, Cons, SubCapas, Plan>. CC needs reasonably scheduling a set of sub-capabilities to complete. Besides the general prerequisites and effect of general capability, CC has a set of sub-capabilities SubCapas. SubCapas={c_i| $c_i\in$ C ^ SC needs c_i to be completed }. Plan refers to sequence order of SubCapas. Only if it is properly planed in a certain order, SC can be completed.

Definition 3.5. Sub-capability is an Atomic Capability (AC for short). AC has the same definition is the same as C, i.e., AC = <Precon, PostEffect, Desc, Const>, but AC is not dividable. Through assembling atomic abilities, an SC can be composed.

Definition 3.6. CCPPA Agent is defined as CCPPA=<Aid, Mq, Mh, Pe, Co, Di, Ab, Cb>. It indicates the eight component elements,

- *Aid = <IP, Port, Name>* is defined as the agent identifier, including the agent host machine IP and communications port, as well as agent's name;
- *Mq* – represents message queue, caches received messages.
- *Mh* – represents message processor, analysis and encapsulates message content.
- *Pe* – represents agent sensor, sense related ability and new tenders on PIB.
- *Co* – represents agent collaboration engine, does collaboration between the Agent;
- *Di* – represents agent's task scheduler, plans sub-abilities according to request task.
- *Cb={c|c∈ C have(Agenti, c)}* is defined as the ability base, stores all ability information;
- *Ab={<aid, cb>|Acq(Agent$_i$, Agent$_{aid}$) ^ have(Agent$_{aid}$, cb)}* is defined as the agent$_i$'s acquaintances, each of the acquaintances, including its aid, and ability information, that is, Cb of the acquaintance.

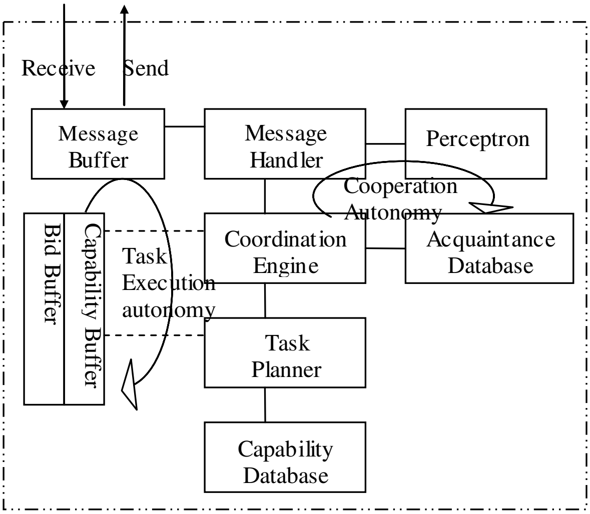


Fig. 2. Architecture of CCPPA

Architecture of CCPPA

As is shown in figure 2, besides the seven components indicated in the formal definition, (*Aid* is not a module), CCPPA has an internal buffer, to coordinate the planner and collaboration engine. The intermediate value is stored in the internal buffer, to suspend the coordination engine to wait other agent to complete a sub-task.

Each agent has a corresponding Agent Definition File (ADF). ADF is an XML document. It defines the agent's *Aid*, capabilities, key capabilities to perceive, acquaintances and task decomposition strategies. While an agent instance is generated, its ADF will be resolve d to initialize.

There are two autonomous running process cycles when agent is running dynamically: Perceive-Capability-Autonomy and Task-Process-Autonomy.

Sensor of the agent can perceive the capabilities in CRC that are related to the current task planning, if any is found, it will add the agent that owns the ability as its acquaintance. By the way, agent can also add the agents that interact frequently as its acquaintance. The sensor can also perceive the new published tenders on PIB according to its busy level, so as to negotiate with the contract manager.

In the task scheduling process, agent follows the steps provided in section 3.2. For each subtask, an agent tries to find if it could be handled itself. If not, it will request out for acquaintances. If all acquaintances answer no, a tender for the subtask is made, and the subtask is tried to be solve under contract net.

All successfully scheduled subtasks are put into *Task_table*, which is a task queue. The task handling thread processes each task in it. Dynamic Programming algorithm is used in the scheduling process to decide whether a subtask can be scheduled into the current *Task_table*. If yes, the task is committed to insert into *Task_table*.In task execution process, agent will perform its task according to the plan it formerly scheduled, invoking its sub-capabilities according the sub-task, interacting with acquaintances, or collaborating with the tender winners.

4 Strategy for Proactive Perception and Collaboration

Initiative perception is one of the most important characteristics of CCPPA. In CCPPA, the sensor has two kinds of perception actions: Active to perceive key capabilities it is in need of and active to perceive tender.

With regard to the former, agent can send a message to CRC, and request to access the relevant capability information of other agent, including mailing address, agent abilities, etc. Then the agents can be added into acquaintance. Whenever agents need help of other agent, it first requests its acquaintance agent, and interacts directly.

For the latter, agent perceive tenders published on PIB according to its busy level. Agent receives a tender, and decides whether to bid or not. We have improved the traditional contract net to a Contract Net Model based on Agent Active Perception (CNMAAP). This kind of contract net collaborative approach, increases agents' activeness in negotiation and collaboration processes with regard to its current state, rather than blindly accept the tender whenever it is busy or not.

4.1 Active Perception Tender and Contract Net

A widely used coordination mechanism in a multi-agent system is the well-known Contract Net Protocol (CNP) [16]. CNP is one of classical negotiation protocols in distributed artificial intelligence (DAI) area. There are two roles in CNP: the manager, who initiates the negotiation process, monitors the task's execution and processes the results of its execution, and the contractor, who is responsible for the actual execution of the task. After the original CNP was proposed in [17], many researchers had made much effort to refine and extend it with more flexibility, adaptability and better performance. This paper designs an improved contract net model based on agent active perception (CNMAAP).

Formal Definition about CNMAAP

Definition 4.1. The perception degree of agent to external environment is called the perception coefficient. It is relevant to the available resource of the agent and the number of tasks in the task queue at the present time.

Definition 4.2. The degree of credibility is defined as the degree to which the tasks are executed of the ones obtained through its bidding since the start of the agent, and the ones requested by other agents, with its range set (0, 1).

Definition 4.3. The tender is represented as:

Bid_i=<Id, Owner, Prohi, Abstr, Sca, Speci, Expir>;
Where:

- *Id* is the identifier of a tender. It is assigned by PIB and can not be changed in the tender's life.
- *Owner* is the identifier of the agent who initiates the negotiation.
- *Prohi* is a set of agents who are not allowed to bid.
- Abstr is the brief description of the task.
- *Sca* is the task level.

- *Speci* is the information that must be provided by contractors when they are bidding for the task. It is a triple: *Speci=<pera, rela, time>*. Here, *pera* and *rela* are the perception coefficient and the degree of credibility for agent a respectively, and *time* is the duration agent a needs to complete the task.
- *Expir* is the bidding deadline.

The Negotiation Process

In the process of negotiation, each party involved in the contract evaluates information from its own perspective and compromises according to its strategy.

We refer to the agent that publishes the tender as contract manager, and the agent who bids for the tender as contract tender.

The first step, contract manager publishes a tender on PIB, and other agents in the system is always active to perceive the tender according to its current busy level. As shown in section 3.3, the agent main monitoring thread is always observing the message numbers it has received and the task number it is handling, according to which agent calculate its perception coefficient Per_i. If $Per_i > Per_o$, where Per_o is the system threshold for each agent, agent will send message to PIB to perceive new published tenders.

PIB will inform the agents who are not in *Prohi* of the tender and be active to perceive new tenders published on PIB.

After the contract tenders receives the tender, it has the authority to bid or not depending on its capability and other factors. If it bids for it, it will send bidding information to the contract manager with the additional information of its perception coefficient and credibility.

The contract manager will sort the contract tenders according to their perception coefficient and the degree of credibility, and the tender will be awarded to the top one ranking agent.

After agent *a* finishes the task *t*, it will be given an evaluation by the contract manager to show its satisfaction to the actual performance of the task. Then the contract manager informs PIB the task level and the evaluation of the task in the form of information. PIB updates the degree of credibility of agent *a* and promptly notify it. The use of this method can prevent the fraud that some agents give false information in order to be the successful bidder.

Strategy of the Contract Manger and the Contract Tender

In the process of negotiation, each party involved in the contract evaluates information from its own perspective and compromises according to its strategy. In other words, we suppose that each agent in the system is selfish, and maximizes its own profit, thus increases efficiency of each agent while assuring the system efficiency. For the contract tender, he will evaluate its perception coefficient (Per_i) on the basis of the tender task's importance and its current task load. If Per_i is higher than the threshold (Per_o), it will bid for it. Otherwise, it will throw the tender away. On the other side, the contract manager will rank the contract tenders on the basis of their perception coefficient and credibility. He has the authority to choose the most favorable contract tenders, i.e. either select the one whose coefficient is the highest if he wants to finish the task as soon as possible, or select the one whose credibility is the highest if he wants high quality task execution.

Compared with traditional contract net model, CNMAAP considers the current load of agent and the history record of task completions. It can effectively reduce the negotiation cost with assured quality while improving the efficiency of negotiation as well.

4.2 Proactive Perception Capability and Acquaintance Cooperation

Comparatively speaking, acquaintance interaction is a kind of more effective collaboration protocol. Acquaintance agents could communicate directly, thus greatly reduce the communication overhead. Many researchers have studied acquaintance collaboration in MAS. Acquaintance collaboration based on agent actively perceiving (ACAAP) is introduced into our system. Agents can perceive the capabilities which are related to it, and add the agents who have the capability as his acquaintances. When planning a task, agents will ask for its acquaintance, ranking them by creditability. Generally speaking, acquaintances are relatively stable and suitable for the interaction with requirement of high quality. Therefore, in our system, we use acquaintance collaboration when any key capabilities are needed.

Formal Definition about ACAAP

Definition 4.4. *Intera(a, b)* denotes that agent *a* has interacted with *b* and cooperated to complete one task.

Definition 4.5. *Acq(a, b)* denotes that agent *a* is acquainted with agent *b*. When it exits *Acq(a, b)*, it must have *Intera(a, b)* and the frequency of *Intera(a, b)* is greater than γ.

As described in section 3.2, acquaintance set of agent *a* is defined as a set of agents who have successfully cooperated with agent *a*, and the frequency coefficient runs over γ. Here, γ is the thresholds of an agent who gets to be acquaintances of another. γ is in proportion to agent credibility rel_i.

Definition 4.6. *RA(a, b) = Acq(a, b) ^ Acq(b, a);* Acquaintance-ship is defined as the fact that agent *a* is acquainted with agent *b* and agent *b* is aquatinted with agent *a*.
RA(a, b) has the following properties:

Property 1. Reflexive: Agent *a* must know himself, i.e. *Acq(a, a)* holds, therefore, it holds that *RA(a, a) = Acq(a, a) ^ Acq(a, a);*

Property 2. Symmetric: *RA(a, b) = Acq(a, b) ^ Acq(b, a) = Acq(b, a) ^ Acq(a, b) = RA(b, a).*

Property 3. Non-transitive: If *RA(a, b) ^ RA(b, c)* ≠ > *RA(a, c).*

The Negotiation Process

In our test case, two kinds of situation are supposed. One is that capability *c* is a key sub-capability and definitely it is not held by agent *a*, then *c* is inserted into the capability list for agent *a* to be perceived. The other is that when agent *a* fails to plan a task through the contract net, i.e., no agents bid for a task which needs capability *c* until the deadline, or when the tender winner failes to finish the task because of some uncertain reason. Both of the above situations will cause agent *a* make the intension to perceive capability *c*.

The above situations are defined in definition 4.8.

Definition 4.7. *PerceiveCap (a, c)* denotes that agent *a* perceives the key sub-capability *c*.

Definition 4.8. $((c \in PL_a) \vee (Bidding(a,\ c) \rightarrow Timeout(c))) \rightarrow Intention\ (a,\ PerceiveCap\ (a,\ c));$

The following process is agent *a* actively perceiving a key sub-capability *c* and get to know agent *b* who hold capability *c* immediately after agent *b* dynamically join MAS.

(1) Agent *a* sends the key capability *c* to be perceived to CRC;
(2) CRC store the capability *c* in the list $CL_{per;}$
(3) CRC first search *c* in its whole capability list, if any found, send the owner agents' identifier to agent *a*;
(4) If any new agents join MAS, it will first register its capabilities to CRC;
(5) CRC monitor the new joined agents, and search if its capabilities conform to any capabilites in CL_{per}. If any is found, it will send the new joined agents identifier to agent *a*.
(6) Agent *a* adds the agent as its acquaintance.
Then agent *a* and agent *b* collaborate to complete a certain task.

When more than one acquaintance have the same capability, the task initiator will rank the acquaintance agent according to their credibility and perception coefficient.

The Strategy of Active Perceiving Capability

There are many ways in which agent actively perceive certain capabilities. The simplest way is to send message to CRC at intervals, while, we choose another way. Agent only need to send the capabilities to be perceived to CRC, and let CRC handle them. Just as each agent put a trigger in CRC, whenever respected capabilities increased in CRC, CRC will inform the trigger agents. This way reduces the number of threads in agents and decreases the risk of the message blocked on the system communication.

In CRC, when agents register their capabilities, CRC will check whether these capabilities include the abilities that other agents perceive.

5 Simulation Verification and Result Analysis

Based on the agent model and MAS frame model, we developed the Agent-based System Integration Tools (ASIT). An application of MAS model in the Border and Coast Defense Simulation System (BCDSS) was used as a test case to confirm the rationality and feasibility of this framework and the agent model. BCDSS is composed of 8 subsystems, each of which contains 1 to 3 agents.

BCDSS is a typical kind of information system in military command and control simulation system (MCCS). The data acquisition subsystems such as radar, infrared, sonar gather the original data in real time, and transmit the data to access aggregation subsystem on another machine, record the original information, and convert them to a unified data format, in order to do situation assessment and to be displayed on different client from different view. The officer at the client end analysis the comprehensive situation, and make appropriate decision.

5.1 Task Completion Efficiency in CNMAAP

The efficiency of tasks completion is inversely proportional to time. In BCDSS, one task can be completed on time or not. In this case, we just calculate the efficiency of tasks finished successfully.

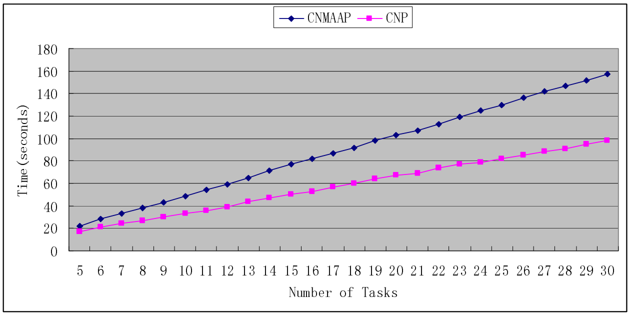


Fig. 3. Comparison of task completion efficiency in CNMAAP and CNP

The above result shows that, with the increasing number of tasks, the time required to complete the task in traditional CNP is more than that in CNMAAP. The reason is that with the increase of task number, the lower task load agents with higher perception coefficient will often win the bidding in CNMAAP. While in the traditional CNP, the winner may be always the same agent and thus causes its task load increase greatly. Then, the later tasks must wait in the long task queue. Besides, it also indirectly proves that the task load of each agent in CNMAAP is relatively balanced.

5.2 Task Completion Efficiency between CNMAAP and ACAAP

In this section, we will discuss task completion efficiency between CNMAAP and ACAAP. CNMAAP and ACAAP share both factors of agent perception coefficient and credibility. The more times agents collaborate under CNMAAP, the greater the credibility is. When the cooperation frequent coefficient exceeds γ, and the contract manager requires quality first, i.e. when he receives more than one contract tenders, he ranks them by credibility first. The contract manager and the contract tender winner will get to be acquaintances. Then the collaboration transmits from CNMAAP to ACAAP. For some task, collaboration between agents may transit from CNMAAP to ACAAP.

The task completion time it takes in CNMAAP is:

$$T_{cnmaap} = T_{perceiveBid} + 6{*}T_{trans} + T_{execute}\ , \quad \text{Where } \sum_{i=1}^{n} T_{perceiveBid_i} = T/2$$

$T_{perceiveBid}$ is the time that agents manage to perceive the new published tender on PIB. The average of $T_{perceiveBid}$ is T/2. Here, T is the interval at which each agent send perception message to PIB. T_{trans} is the message transmission time on networks. It needs 6 negotiation messages in the process of CNMAAP. $T_{execute}$ is the task execution time of the bidding winner.

The task completion time it takes in ACAAP is:

$$T_{acaap} = 4{*}T_{trans} + T_{execute}$$

Comparing the above two formulas, we can see that the cooperation under ACAAP is more efficient than that under CNMAAP, while CNMAAP is more flexible.

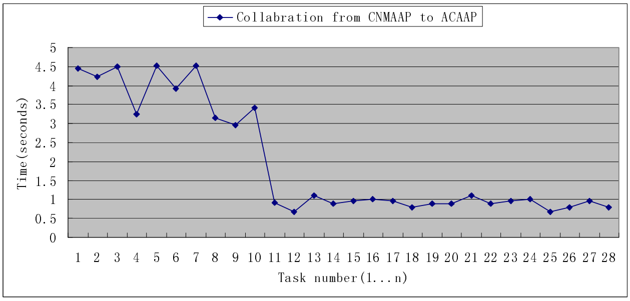


Fig. 4. Task completion efficiency when collaboration transmits form CNMAAP to ACAAP

Figure 4 is the collaboration efficiency as the system dynamically runs. It can be seen that the collaboration way changed at time 11, which is just the situation described in section 4.3. It can be seen that the task completion time it needs under acquaintance protocol is much less than that under contract net protocol. These two collaboration protocols are both introduced in the system test case. CNMAAP is used when flexible and dynamical collaboration is needed, and ACAAP is used when the collaboration is stable and high efficiency is required.

6 Conclusion

Based on the definition of agent capability, an initiative perception agent model is designed in this paper. According to proactive perceiving capabilities and initiative perceiving tenders, agents achieve collaboration under ACAAP and CNMAAP. For the key sub-capabilities, agents get acquainted with the other as soon as possible, and cooperate directly. The collaboration under CNMAAP is dynamical and flexible. It improves the tradition contract net in system efficiency based on agent current task load and history task completion. Both of the two ways are introduced in the test case, and it proves the reasonable and feasible of the agent model.

Acknowledgments. This work is supported by the Defense Pre-Research Project of the ‘Eleventh Five-Year-Plan’ of China under contract number 513060401 and the Fund of C4ISR National Defense Science and Technology Key LAB under contract number 9140C8301011001, In addition, it is supported by the ‘Fundamental Research Funds for the Central Universities’.

References

[1] Wooldridge, M., Jennings, N.R.: Intelligent Agent: Theory and Practice. Knowledge Engineering 10(2) (1995)

[2] Jennings, N.R., Sycara, K., Wooldridge, M.: A Roadmap of Agent research and development. In: Autonomous Agents and Multi-Agent Systems, pp. 1275–306 (1998)

[3] Jennings, N.R., Wooldridge, M.: Agent-oriented Software Engineering. In: Handbook of Agent Technology. AAAI/MIT Press (2001)

[4] Brooks, R.: Intelligence without Reason. In: Proc. 12th Int. Joint Conf. on AI, pp. 569–595 (1991)

[5] Brooks, R.: Intelligence without Representation. Artificial Intelligence 47(1-3), 139–159 (1991)

[6] Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. Artificial Intelligence 42(3) (1990)

[7] Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, pp. 473–484 (1991)

[8] Zhang, S.-M., Shi, C.-Y.: Modeling Agents Combined with the State of Environment. Computer Research and Development 39(12), 1587–1591 (2002)

[9] Wang, Y.-C., Shi, C.-Y.: A Concurrent BDI-Agent Model. Jounal of Software 14(3), 422–428 (2003)

[10] Dong, M.-K., Zhang, H.-j., Shi, Z.-Z.: An Agent Model Based on Dynamic Description Logic. Journal of Computer Research and Development 41(5), 780–786 (2004)

[11] Wooldridge, M., Lomuscio, A.: Multi-Agent VSK logic. In: Brewka, G., Moniz Pereira, L., Ojeda-Aciego, M., de Guzmán, I.P. (eds.) JELIA 2000. LNCS (LNAI), vol. 1919, pp. 300–312. Springer, Heidelberg (2000)

[12] Nwana, H.S., Ndumu, D.T., Lee, L.C., Collis, J.C.: ZEUS: a toolkit and approach for building distributed multi-agent systems. Applied Artificial Intelligence

[13] Bellifemine, F., Poggi, A., Rimassa, G.: JADE–A FIPA-compliant agent framework. In: Proceedings of PAAM (1999)

[14] Gao, J., Wang, J.: ABFSC: An Agents-Based Framwork for Software Composition 22(10), 1050–1058 (October 1999)

[15] Foundation for Intelligent Physical Agents. FIPA Abstract Architecture Specification, SC00001L (2002)

[16] Smith, R.G., Davis, R.: The contract net protocol: High level communication and control in a distributed problem solver. IEEE Tran. Computers 29(12), 1104–1113 (1980)

Algorithmic Trading Strategy Optimization Based on Mutual Information Entropy Based Clustering

Feng Wang[1,2], Keren Dong[2], and Xiaotie Deng[2]

[1] State Key Lab. of Software Engineering, Wuhan University, Wuhan, 430072, China
[2] Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
fengwang@whu.edu.cn

Abstract. Algorithmic trading strategies are automated defining a sequence of instructions executed by a computer. A good strategy should be profitable which includes identification of what to trade and how to trade. In this paper, we focus on the study of algorithmic trading strategy optimization and propose a strategy optimization model based on an initialized strategy pool. In order to get a better strategy, a mutual information entropy based clustering algorithm is employed to analyze the correlations among the stocks and a reward and punishment scheme is also set up for updating the latest transaction data in the strategy optimization process. Experimental results on several different groups of stocks showed that in most cases, this optimization model can find a profitable strategy swiftly.

1 Introduction

Algorithmic trading is a new trading style which use the computer software programmes to initiate trades in electronic financial markets based on sets of algorithms. There has been a vast array of algorithms developed across financial markets in order to achieve different trading strategies.

Trading strategy is a very important research issue in algorithmic trading [1][2]. One such strategy is basic arbitrage. This kind of strategy employs the market efficiency theory and testifies arbitrage by searching for the disparity of the prices in the foreign exchange market. Currently, there are many more complicated strategies implemented by using different kinds of market information. 'Benchmarking' attempts to mimic indices' returns, 'sniffers' detects volatile and unstable markets, while 'icebergs' trades to reduce the transaction cost [3]. Although these kind of trading strategies have been used in the real market, mostly, they are decided by the traders' experiences and not so accurate to be executed efficiently. On the other hand, the market information is so comprehensive that the traders can not handle it well to make a good trading strategy. Furthermore, the prices of the stock often have some potential correlations with some other securities in market. Although some related work have been done on market information (such as tick-by-tick time series data) analysis by using some data mining techniques, such as k-means, support vector machine, neural network, genetic programming and reinforcement learning [4][5][6][7][8] [2][9], these methodologies have not paid much attention on the stocks' correlation analysis of the stocks' movement. Correlation analysis is very important for the trend prediction, especially in the complex electronic market.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 252–260, 2010.

In this paper, we propose a new data analysis approach by employing a correlation measurement algorithm based on mutual information entropy. We suppose each stock in the market as a variable (vector) and propose a trading strategy optimization model. The trading strategy making problem can be seen as a multi-variate correlated optimization problem since the price of one stock can be affected by some other stocks's prices. The mutual information entropy based clustering algorithm which explores the mutual information entropy to measure the relationships of the variables has proceeded successfully in many multi-variate related optimizations. In the algorithmic trading strategy optimization problem, we firstly initialize a strategy pool. Each strategy is represented as a $0 - 1$ sequence, where for each bit i, 1 means buy stock i and 0 means hold. In the strategy evolution process, since the latest market data might affect the optimization orientation, we set up a frequency matrix to store the correlations of the stocks and develop an updating scheme named reward and punishment scheme to update the strategies in the strategy pool. The strategies are evaluated by the strategy optimization function. This approach can detect the relationships of the trends among different stocks with different attributes, and after some iterations it can get a good trading strategy as solution.

The rest of this paper is organized as follows. Section 2 introduces the correlation measurement among variables based on mutual information entropy. Section 3 goes into details of describing the trading strategy optimization approach by using the correlation measurement algorithm. Experimental results and their analyses are given in Section 4. Section 5 concludes the paper.

2 Mutual Information Entropy Based Clustering

In information theory, entropy which represents the energy of the variables is usually used to evaluate the state of an evolving system [10]. The most important issue for the optimization problem is to identify the probability distribution of the solutions. In the current literature, most work use the conditional probability to represent the correlations among the variables.

$$p(x) = \sum_{i=1}^{n} p(x_i|x_{\pi_i})$$

where x_{π_i} is the variable set which has correlations with variable x_i.

Suppose the distance of the two variable distribution is $D(p||\hat{p})$, according to the definition of mutual information, we have

$$\begin{aligned} I(x_1;x_2) &= \sum_{x_1 \in \mathrm{X}} \sum_{x_2 \in \Upsilon} p(x_1,x_2) \log \frac{p(x_1,x_2)}{p(x_1)p(x_2)} \\ &= D(p(x_1,x_2)||p(x_1)p(x_2)) \end{aligned}$$

Theorem 1 (Optimal probability distribution model based on mutual information entropy). *In an optimization problem, the probability distribution $\hat{p}(x)$ is the distribution of the optimal solution of $p(x)$ if and only if the conditional entropy of the probability distribution $H(x|x_\pi)$ reaches its minimum.*

Proof. In information theory, the similarities among the variables are often measured by distances. Here we use Kullback-Leiber distance as the distance measurement among the variables, then we have

$$\begin{aligned} D(p||\hat{p}) &= \sum_x p(x)\log p(x) - \sum_x p(x)\log \hat{p}(x) \\ &= \sum_x p(x)\log p(x) - \sum_x p(x) \sum_{i=1}^{n} \log p(x_i|x_{\pi_i}) \\ &= H(x) - \sum_x p(x) \sum_{\substack{i=1 \\ \pi_i \neq i}}^{n} \frac{p(x_i, x_{\pi_i})}{p(x_i)p(x_{\pi_i})} - \sum_x p(x) \sum_i^{n} p(x_i) \end{aligned}$$

Since $p(x_i)$ and $p(x_i, x_{\pi_i})$ follow the same distribution of $p(x)$we have,

$$-\sum_x p(x)\log p(x) = H(x_i)$$

and

$$\begin{aligned} \sum_x p(x)\log \tfrac{p(x_i,x_{\pi_i})}{p(x_i)p(x_{\pi_i})} &= \sum_{x_i,x_{\pi_i}} p(x_i, x_{\pi_i})\log \tfrac{p(x_i,x_{\pi_i})}{p(x_i)p(x_{\pi_i})} \\ &= I(x_i; x_{\pi_i}) \end{aligned}$$

hence,

$$\begin{aligned} D(p||\hat{p}) &= H(x) + \sum_{i=1}^{n} H(x_i) - \sum_{i=1}^{n} I(x_i, x_{\pi_i}) \\ &= H(x) + \sum_{i=1}^{n} H(x_i|x_{\pi_i}) \end{aligned}$$

since $H(x_i|x_{\pi_i}) > 0$, the optimization problem equals to minimize $D(p||\hat{p})$, that is,

$$\min \sum_{i=1}^{n} H(x_i|x_{\pi_i})$$

After optimal probability distribution model being established, we made a normalization of $H(x_i|x_{\pi_i})$, and set $\rho_i = 1 - \frac{H(x_i|x_{\pi_i})}{H(x_i)}$ as the correlation measurement parameter.

For each variable x_i, we compared the correlation parameter ρ_i between it and other variables and choose the one which has maximum correlations with x_i to construct a connection between this variable and x_i. If the variable has little correlation with the current variables in the clusterings, we set this variable as the root for a new cluster. After some iterations, we can get the mutual information entropy based clusterings of the variables.

3 Trading Strategy Optimization Based on Correlation Measurement

3.1 Trading Strategy Optimization Model

Trend analysis and prediction play a vital role in the practical electronic trading. Experienced traders can often predict the future trend of a stock's price based on their observations of the performance of the stock in the past. An early sign of a familiar pattern may alert a domain expert to what is likely to happen in the coming future. On the other hand, it is well known that different clustering methods may discover different patterns in a given set of data. This is because each clustering algorithm has its own bias resulting by the optimization of different criteria. Here we follow this idea and try to find some good trading patterns for the trading strategy making by exploring our mutual information entropy based clustering on the market data.

Since there are various complex correlations among the stocks in the market, the future price of a stock can also be affected by some other stocks which have close correlations with it. As a result of this, the most important problem is how to evaluate the relationships of the stocks. In the exchange market, different stocks can be regarded as different variables which have some correlations with other variables. Then the trading strategy optimization problem can be defined as,

$$\max \sum_{i=1}^{n} \omega_i p_i$$
$$s.t. \begin{cases} \sum_{i=1}^{n} \omega_i = 1 \\ \omega_i \geq 0 \\ \omega_i \leq v_i / \sum_{i=1}^{n} v_i \end{cases}$$

where n is the number of the stocks, p_i stands for the trading behavior of stock i, and v_i is the volatility for each stock i as the standard deviation. Since the trend of stock i might has correlations with the trend of stock j, it is very necessary to exploit the correlations among the stocks.

3.2 Correlation Measurement of the Stock Prices

Frequency Matrix Construction. We use a frequency matrix to represent the possible strategies in the financial market. Firstly, we initialize a virtual strategy pool S with m individuals. $S_t = [p_1, p_2, ...p_n]$ $(0 \leq t \leq m)$ is a 0-1 sequence with length n, and $p_i = 1$ means to buy the stock i, otherwise, $p_i = 0$. Then we calculate the frequencies of $00, 01, 10, 11$ of the strategies in the strategy pool and store them in a frequency matrix F.

$$F = \begin{array}{c|cccc} & p_1 & p_2 & \cdots & p_n \\ \hline p_1 & 10\varepsilon & 9\varepsilon & \ldots & 0 \\ p_2 & \begin{pmatrix} a & b \\ c & d \end{pmatrix} & 0 & \ldots & 6\varepsilon \\ \vdots & \cdots & \cdots & \cdots & \cdots \\ p_n & 0 & 0 & \ldots & 8\varepsilon \end{array} \dashrightarrow p_{p_1,p_2} = \begin{pmatrix} f_{p_1=0,p_2=0} = a & f_{p_1=0,p_2=1} = b \\ f_{p_1=1,p_2=0} = c & f_{p_1=1,p_2=1} = d \end{pmatrix}$$

The frequency matrix F collects the historical strategy information which is updated with the new generated transaction data in the market. We set an incremental learning factor as α and the frequency matrix is updated by $F_{t+1} = \alpha F_t + (1 - \alpha) F_{t+1}$. Then we calculate the statistical mutual information by following the following formulas.

$$P(p_1, p_2, ...p_N) = \prod_{i=1}^{n} P\left(p_i | p_j, j \neq i\right)$$

$$I\left(p_i, p_j\right) = \sum_{a,b} P\left(p_i = a, p_j = b\right) \cdot \log \frac{P\left(p_i = a, p_j = b\right)}{P\left(p_i = a\right) \cdot P\left(p_j = b\right)}$$

Then we calculate the correlation parament ρ and construct the mutual information entropy based clusters.

Reward and Punishment Scheme. In each iteration, the strategies can be evaluated by the trading strategy optimization model. The frequency matrix is updated after each iteration of the strategy evolution. In each iteration, we calculate the statistics of the price trend for each stock. A simple trading strategy is to buy or hold a share if the price is higher than previous one, and sell if the value received exceeds the value at which we bought previously. For example, if the price of stock i goes up, we will hold $p_i = 1$, otherwise $p_i = 0$. Each strategy is evaluated by calculating the rate of return. Suppose the best strategy in the pool is G_{best} and the worst is G_{worst}.

Algorithm 1. Reward and punishment scheme

```
reward($g_{best}$)
for $i = 1 \cdots m$ do
    for $j = i \cdots m$ do
        update frequency matrix:
        $freq(g_{best}[i], g_{best}[j]) \leftarrow freq(g_{best}[i], g_{best}[j]) + \varepsilon$
    end for
end for
penalize($g_{worst}$)
for $i = 1 \cdots m$ do
    for $j = i \cdots m$ do
        $freq(g_{worst}[i], g_{worst}[j]) \leftarrow freq(g_{worst}[i], g_{worst}[j]) - \varepsilon$
    end for
end for
```

In this algorithm, the reward and punishment parameter ε is decided by the number of the individuals in the current strategy pool m and the number of the stocks taken into account n. It is set as

$$\varepsilon = \frac{2m}{n(n-1)}$$

4 Experimental Results and Discussions

To evaluate the performance of the algorithm, we test 1 year daily data of 6 groups of American stocks. Each group contains 20 different stocks, each stock has historic price data from January 1, 2009 to December 31, 2010. The price data of these stocks are shown at figure 1-6. The horizontal axis represents trading day of year 2009, while vertical axis represents prices of corresponding stock.

Parameter settings in our experiments are given as follows, the number of samples per iteration is set to 100, the population of genes is set to 50, the incremental learning factor $\alpha = 0.1$ and the value of ϵ in Reward and Punishment Scheme sets to default value. All parameters are held constant through all the runs. We perform the algorithm to find best allocations in each trading day. In the end of each trading day we get the corresponding price data of each stock and calculate ratio of prices changes.

The results are shown at figure 1-6. The horizontal axis represents trading day, while the vertical axis represents profit of the corresponding trading day. In each trading day we use the algorithm described above to calculate the best portfolio of stocks, and calculate the allocation of asset by linear programming.

To determine the allocation of asset, we first calculate volatility for each stock as the standard deviation of the daily return ratio, then use the linear programming determine the parameter ω in the strategy optimization model.

In these experiments, the profit of our strategies are shown to be positive. The return of best stock can not be matched in all time because we need to consider risk factor,

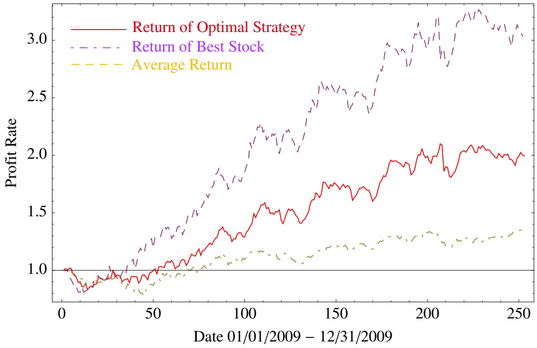


Fig. 1. Performance of First Group of Stock

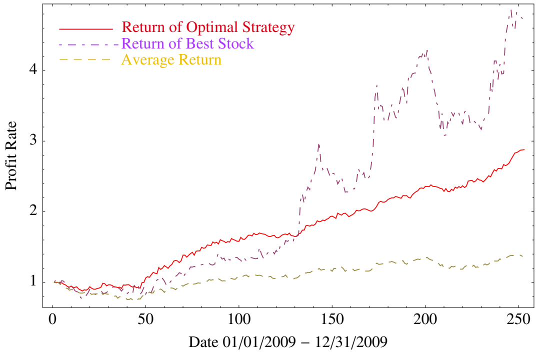


Fig. 2. Performance of Second Group of Stock

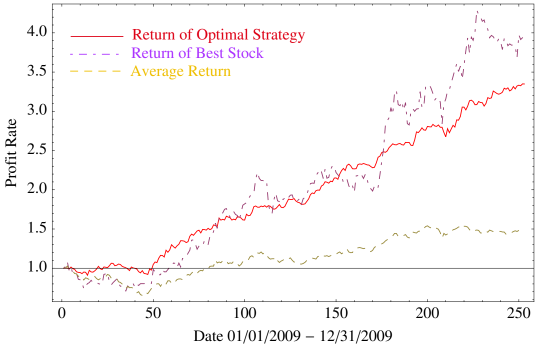


Fig. 3. Performance of Third Group of Stock

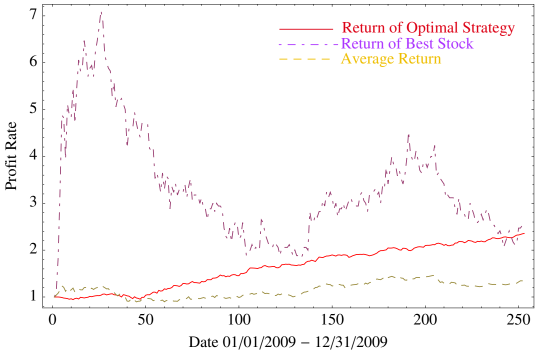


Fig. 4. Performance of Fourth Group of Stock

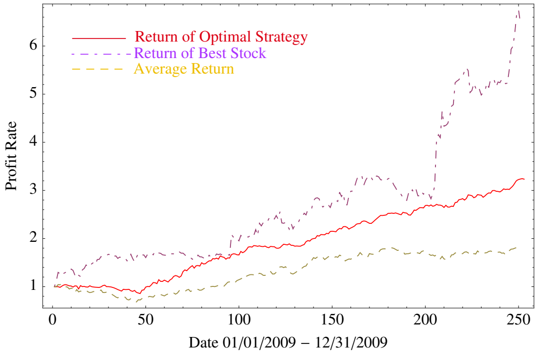


Fig. 5. Performance of Fifth Group of Stock

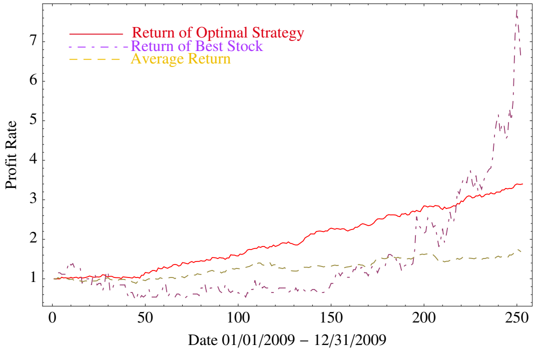


Fig. 6. Performance of Sixth Group of Stock

Table 1. Statistical Turnover Rate

Test Group	Total turnover times	Average Turnover Rate (per day)
1	1050	4.2
2	1039	4.156
3	1056	4.224
4	1086	4.334
5	1038	4.152
6	1018	4.072

this is more obvious in figure 4, where the best stock increases rapidly in a short time but drops in the following time; our strategy is shown to avoid this case and keep increasing.

In future we will take account of transaction cost in experiments. As shown in table 1, we have a high trading frequency in current algorithm. This would bring relatively high transaction cost which may reduce final return of the strategy.

We also gather statistics of trading frequency represented as the turnover rate. Due to the turnover rate is so high and the transaction cost is very complicated to be testified, here we did not taken it into account for the return computation in the current experiments. But in the real transaction process, the transaction cost does exist and should not be ignored. In our future work, we are going to do some improvement on the current algorithm and the transaction cost will also be considered in the return evaluation process.

5 Conclusion

In this paper, we proposed an algorithmic trading strategy optimization model. In this model, each stock in the real market can be regarded as a variable and different variables have complex correlations in prices. We then proposed a mutual information entropy based clustering algorithm and applied it into mining the stock price correlation. We also explore a reward and punishment scheme to update the strategy pool with the latest transaction data in the strategy optimization process. We use 1 year daily data in the American stock exchange market in 2009 as our test data and experimental results on some different groups of stocks showed that in most cases, this optimization model can find a profitable strategy efficiently.

In the following step, we would explore this methodology to analyze the tick data in the real trading market. The convergence of this mutual information entropy based clustering algorithm would also be discussed in our future work.

Acknowledgement

This work was supported by the Fundamental Research Funds for the Central Universities (Wuhan University) under Grant NO. 6082018.

References

1. van Bunningen, A.H.: Augmented Trading: from news articls to stock price prediction using syntactic analysis. Master's thesis, University of Twente (2004)
2. Chen, S., Navet, N.: Pretests for genetic-programming evolved trading programs: "zero-intelligence" strategies and lottery trading. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4234, pp. 450–460. Springer, Heidelberg (2006)
3. Choice, A.S.: Market risk and algorithmic trading. Tech. rep., AMD White Paper (2008)
4. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 2nd edn. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, Hoboken (2006)
5. Crammer, K., Kearns, M., Wortman, J.: Learning from multiple sources. Journal of Machine Learning Research 9, 1757–1774 (2008)
6. He, H., Chen, J., Jin, H., Chen, S.: Stock trend analysis and trading strategy. In: JCIS. Atlantis Press (2006)
7. Kakade, S., Kearns, M.J., Mansour, Y., Ortiz, L.E., Competitive, L.E.: algorithms for vwap and limit order trading. In: ACM Conference on Electronic Commerce, pp. 189–198 (2004)
8. Nevmyvaka, Y., Feng, Y., Kearns, M.: Reinforcement learning for optimized trade execution. In: ICML 2006, pp. 673–680 (2006)
9. Pranav, P., Eamonn, K., Jessica, L., Stefano, L.: Mining motifs in massive time series databases. In: Proceedings of IEEE International conference on data mining, pp. 370–377 (2002)
10. Rakesh, A., Christos, F., Arun, S.: Efficient similarity serach in sequence databases. In: Proceedings of 4th International Conference on Foundations of Data Organization and Algorithms, pp. 13–15 (1993)

An Improved Similarity Algorithm Based on Hesitation Degree for User-Based Collaborative Filtering*

Xiangwei Mu[1], Yan Chen[1], Jian Yang[1], and Jingjing Jiang[2]

[1] Transportation Management College, Dalian Maritime University,
Dalian, Liaoning, P.R. China
[2] Department of Information Technology and Business Administration,
Dalian Neusoft Institute of Information
Liaoning, P.R. China
xiangwei.mu@gmail.com

Abstract. With the fast development of World Wide Wed, web-based applications and services should allow users to get the right personalized information quickly and effectively. Collaborative Filtering plays a very important role in web service personalization and Recommender System. In this paper, Hesitation Degree was proposed to improve the accuracy of user based collaboration filtering and three kinds of Hesitation Degree were introduced into similarity computation. The results show that the prediction accuracy can be improved by 11 percents, and *Mean Absolute Error* can be reduced faster than classic method.

Keywords: stability degree, similarity computation, collaborative filtering, personalized recommendation, recommend system.

1 Introduction

With the development of the Internet, the Web is becoming the largest data repository. Facing the explosion of information ever available in the history of humankind, web applications and services are lack of personalized information service, and people are difficult to get the information that they are interesting. Selecting relevant content and constructing user profile are still considered as the most challenging tasks.

Web mining and Personalized Recommendation System is a way to meet the needs of individual users. They try to analyze usage pattern from web log, build user profiles and generate the recommendation pages to meet the user's requirements of personalized. The most famous personalization recommendation systems include: the Fab of Stanford University Digital Library Program, the GroupLen of the University of Minnesota, the SIFTER of Indiana State University and so on.

One of the most promising recommendation technologies is collaborative filtering (CF) [1,2,3,4]. Collaborative filtering works by finding user neighbors [6] and similar

* This work is supported by Research Fund for the Doctoral Program of Higher Education (No. 200801510001), the National Natural Science Foundation of China (70801007).

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 261–271, 2010.

items[5], and then building preferences prediction for users. Collaborative filtering has been very successful in both research and practice, and in both information filtering applications and E-commerce applications, such as Tapestry [7], Usenet news [8] , Web Watcher[9], Let's Browse[10], Firefly[11], SELECT[12], SiteSeer[13]. However, there remain some important challenges for collaborative filtering recommender systems, such as cold star, scalability and recommendation quality [5].

In order to improve the recommendation quality and prediction accuracy of CF, this paper proposes three kinds of stability: Hesitation Degree based on Standard Deviation (*HDS*), Hesitation Degree based on Mean (*HDM*) and Hesitation Degree based on Distance (*HDD*). They are introduced to influence the result of user neighbor computation. The remainder of this paper is organized as follows: Theoretical background is presented in section 2, three kinds of Hesitation Degree are proposed in section 3. The empirical analyses of three types *HD* are discussed in section 4. Section 5 followed by results of some experiments. Conclusions are given in Section 6.

2 User-Based Collaborative Filtering Algorithms

CF approaches assume that those who agreed in the past tend to agree again in the future. For example, a collaborative filtering or recommendation system for music tastes could make predictions about which music a user should like given a partial list of that user's tastes (likes or dislikes) [5]. CF methods have two important steps, firstly, CF collects taste information from many users, and this is collaborating phase. In the second step, using information gleaned from many users, predictions and recommendation of users' interest were automatically generated, and this is filtering phase.

Researchers have devised a number of collaborative filtering algorithms that can be divided into two main categories, User-based and Item-based algorithms [6].

User-based CF is also called nearest-neighbor based Collaborative Filtering, it utilize the entire user-item data base to generate a prediction. These systems employ statistical techniques to find users' nearest-neighbors, who have the similar preference with the target user. Once the nearest-neighborhood of users are found, these systems use different algorithms to combine the preferences of neighbors to produce a prediction or top-N recommendation for the target user. The techniques are popular and widely used in practice.

User-based algorithms operate over the entire user database to make predictions [6], which predict the votes of the active user based on some partial information regarding the active user and a set of weights calculated from the user database. Assume that the predicted vote of the active user for item j, $p_{a,j}$ is a weighted sum of the votes of the other users:

$$P_{a,j} = \overline{R_a} + k \sum\nolimits_{i=1}^{n} w(a,i) \left(R_{i,j} - \overline{R_i}\right) \tag{1}$$

Or

$$P_{a,j} = \frac{\sum_{all\ the\,neighbor,N}(w(a,N) * R_{N,j})}{\sum_{all\ the\,neighbor,N}(|w(a,N)|)} \tag{2}$$

Where n is the number of users in the collaborative filtering database with nonzero weights. k is a normalizing factor such that the absolute values of the weights sum to unity. $\overline{R_a}$ is the mean of the rating for active user a. The weights $w(a,i)$ can reflect distance, correlation, or similarity between each user i and the active user.

2.1 Correlation Similarity

This general formulation of statistical collaborative filtering (as opposed to verbal or qualitative annotations) first appeared in the published literature in the context of the GroupLens project, where the Pearson correlation coefficient was defined as the basis for the weights. The correlation between users a and i is:

$$w(a,i)=\frac{\sum_{u\in U}(R_{a,i}-\bar{R}_a)(R_{i,j}-\bar{R}_i)}{\sqrt{\sum_{u\in U}(R_{a,i}-\bar{R}_a)^2}\sqrt{\sum_{u\in U}(R_{i,j}-\bar{R}_i)^2}} \tag{3}$$

Where the summations over j are over the items for which both users a and i have recorded votes.

2.2 Vector Similarity

In the field of information retrieval, the similarity between two documents is often measured by treating each document as a vector of word frequencies and computing the cosine of the angle formed by the two frequency vectors. We can adopt this formalism to collaborative filtering, where users take the role of documents, titles take the role of words, and votes take the role of word frequencies. Note that under this algorithm, observed votes indicate a positive preference, there is no role for negative votes, and unobserved items receive a zero vote. The relevant weights are now

$$w(a,i)=\sum_{\mathrm{j}}\frac{R_{a,j}R_{i,j}}{\sqrt{\sum_{k\in I_a}R_{a,k}^2}\sqrt{\sum_{k\in I_i}R_{i,k}^2}} \tag{4}$$

where the squared terms in the denominator serve to normalize votes so that users that vote on more titles will not a priori be more similar to other users. Other normalization schemes, including absolute sum and number of votes, are possible.

User-based CF algorithms have been popular and successful in past, but the widespread use has revealed some potential challenges. In practice, users only have purchased few percents of all the items, maybe 1% of 2 million books, so that recommender system based on nearest neighbor algorithms may be unable to make any item recommendations for a particular user, and the accuracy of recommendations may be poor. Nearest neighbor algorithms require computation that grows with both the number of users and the number of items. With millions of users and items, a typical web-based recommender system running existing algorithms will suffer serious scalability problems.

The weakness of nearest neighbor algorithm for large, sparse database led us to explore alternative recommender system algorithms. Our first approach attempted to bridge the sparsity by incorporating semi-intelligent filtering agents into the system [14, 15].

3 Hesitation Degrees

In this paper, hesitation is considered as a kind of uncertainty and temporary psychological activity, when one is faced with a variety of choices to make decisions. At a particular time and context, people can make different choices. People with different background always have different ways of describing the same item and different values they place upon it. Comprehending hesitation can provide a significant universal insight into human awareness and behavior; whatever we do or observe others doing occurs in a temporal frame of reference and hence requires some degree of hesitation [16]. In collaboration filtering, the performance of the user's hesitation is the instability of a user’s score for an item or similarity item set. The instability of an item’s score from all users also can describe the average degree of hesitation.

There is another definition of Hesitation in psychology. Hesitation shall refer to the time elapsing between the external or internal stimulation of an organism and his, her, or its internal or external response [16]. But in collaboration filtering, we can’t get data about time elapsing or process of decision from user-item matrix. In some other recommendation methods, which can gather information from user behaviors, hesitation degree can be simulated by time elapsing and other user behaviors [17].

We assume that, according to users’ psychological or emotional changes, users’ opinion on the same or similar project is unstable. This unstable evaluation affects the selection of nearest neighbor and most similar item. In this section we give the Hesitation Degree definition based on Item.

In this paper, we assume that, according to users’ psychological or emotional changes, users’ opinion on the same or similar project is unstable. This unstable evaluation affects the selection of nearest neighbor and most similar item. In this section we give the Hesitation Degree definition based on user neighbor.

Definition 1 (Hesitation Degree) –A Hesitation Degree is

$$HD_I^a(b) \in [0,1] \tag{5}$$

This definition use target user’s neighbors as the reference object to compute stability degree. *HD*=0 means that user b is completely stability relative to user a. *HD*=1 means user b is completely unstable relative to user a.

Furthermore, we assume that, firstly, *HD* is inherent nature of user, does not infulence by other user or neighbors, however it can be known by the exist user’s information. Secondly, The similarity of two users is inversely with the stability, and HD can affect similarity computation between target user and neighbors, thus *HD* can affect the accuracy of prediction.

HD is measured by computing the relationship or statistical property between target user and reference users. Three methods were proposed to calculate the Stability Degree, including Hesitation Degree based on Standard Deviation (*HDS*), Hesitation Degree based on Mean (*HDM*) and Hesitation Degree based on Distance (*HDD*).

3.1 Hesitation Degree Based on Standard Deviation

HD based on Standard Deviation (*HDS*) uses standard deviation of absolute value of difference between target user a and neighbor user b to measure the stability degree. Let the set of items who both rated by user *a* and user *b* are denoted by *I*.

$$HDS_I^a(b) = \frac{std(|R_{a,I} - R_{b,I}|)}{maxRating - minRating} \tag{6}$$

maxRating is the maximum of rating value. minRating is the minimum of rating value. std is the function to compute standard deviation between two vectors.

3.2 Hesitation Degree Based on Mean

HD based on Mean (*HDM*) uses mean of absolute value of difference between target user a and neighbor user b to display the stability degree. Let the set of item who both rated by user *a* and user *b* are denoted by *I*.

$$HDM_I^a(b) = \frac{mean(|R_{a,I} - R_{b,I}|)}{maxRating - minRating}\ , \tag{7}$$

mean is the function to compute mean of vector.

3.3 Hesitation Degree Based on Distance

HD based on Distance (*HDD*) is proposed to calculate the *HD* by the distance between target user a and neighbor user b. Let the set of items who both rated *a* and *b* are denoted by *I*.

$$HDD_I^a(b) = \sqrt{\frac{\sum_{i\in I}(R_{a,i} - R_{b,i})^2}{size(I) * (maxRating - minRating)}} \tag{8}$$

Because of the different method, these three kind *HD* can't compare directly. In the follow, we randomly selected an item, three fig.s were presented to show the trends of these *HDs*, when user size, item size and similar item set size growing.

4 Empirical Analysis of Stability Degree

We used data from MovieLens recommender system. MovieLens is a web-based research recommender system that debuted in Fall 1997. Each week hundreds of users visit MovieLens to rate and receive recommendations for movies. The site now has over 43000 users who have expressed opinions on 3500+ different movies. We selected 943 users and 1682 items from the database. A user was chosen as the target.

Let item size is 1682, and similarity item size is 5, with the number of users increasing from 1 to 943, the mean value of *SD*s' (*MSD*) trends were shown in fig. 1.

$$MSD = \frac{\sum_{j \in neighbor} SD_U^i(j)}{neighborSize} \tag{9}$$

neighbor is the neighbor user set, *neighborSize* is the size of set *neighborSize* for target user a. *MSD* of SDS' value is around (0.3, 0.4), *MSD* of *SDM* and *SDD* is around (0.6, 08).

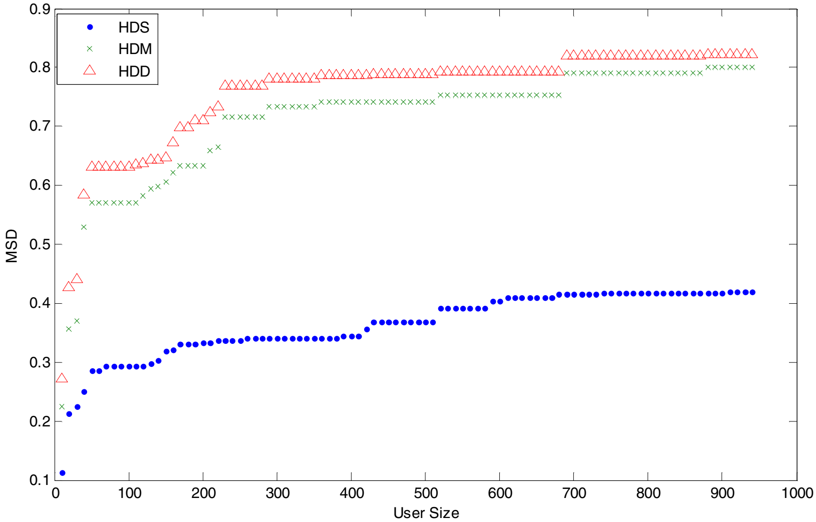


Fig. 1. Impact of user size on three types of *SD*

Let item size is 1682, and user size is 943, with the neighbor size increasing from 1 to 500, three kinds of *SD*s' trends were shown in fig. 2. All of them are decreasing slowly when neighbor size is from 100 to 500.

Let similarity item size is 10, and user size is 943, with the item size increasing from 1 to 1000, three kinds of *SDs'* trends were shown in fig. 3. All of them is unchanging after item size is 300.

As is shown in the figures above, although *SDs* change with the size of user, item and similar item set, in a short term view, *SD* value is stable, size of user, item and similar set do not have a great influence on it.

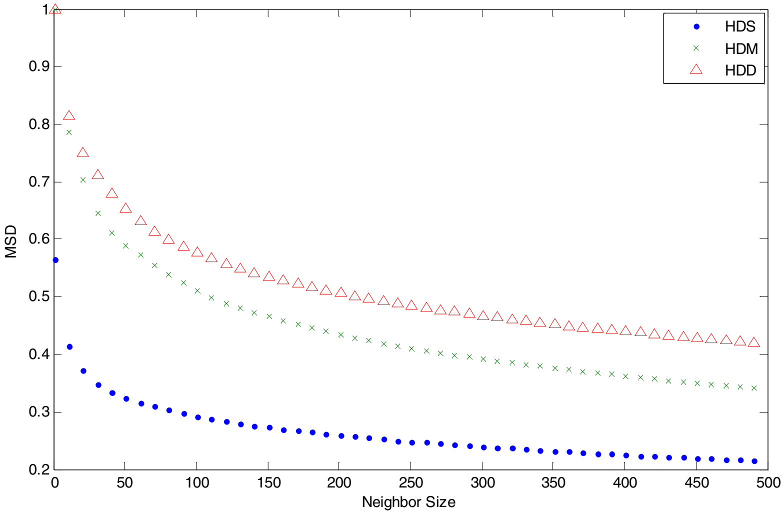


Fig. 2. Impact of neighbor size on three types of SD

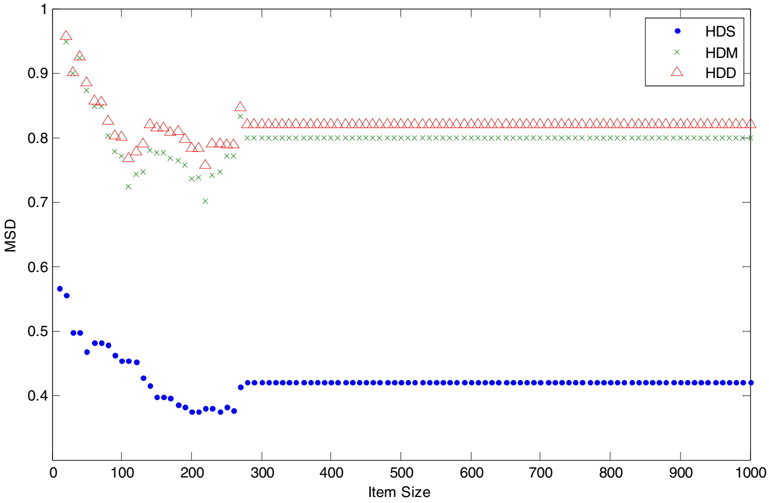


Fig. 3. Impact of item size on three types of SD

5 Experiment

In the follow experiment, all of three similarity measure was chosen as the method to select similar items, Weighted Sum was chosen as prediction method. Absolute Error (*AE*) and Mean Absolute Error (*MAE*) were used to show the deviation of recommendations from their true user-specified values. For each ratings-prediction pair $< p_i, q_i >$ this metric treats the absolute error between them.

$$MAE = \frac{AE}{N} = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{10}$$

In this paper, neighborhood between two user: active user a and user b is measured by two methods: the Correlation and Vector Similarity, mentioned in section 2; the Top-N most similar neighbor set was used to generate the prediction. HD was added as factor to affect the ranking of the neighbor set.

$$w(a,b)^{SDS} = sim(a,b) * (1 - SDS_I^a(b)) \tag{11}$$

$$w(a,b)^{SDM} = sim(a,b) * (1 - SDM_I^a(b)) \tag{12}$$

$$w(a,b)^{SDD} = sim(a,b) * (1 - SDD_I^a(b)) \tag{13}$$

In fig. 4, predictions was generated based on three the similarity measures. “No HD” bar shows the MAE that was calculated without HD factor. "HDS","HDM" and "HDD" bar are calculated by $\mathrm{w(a,b)^{SDS}}$ as in (16), $\mathrm{w(a,b)^{SDM}}$ as in (17) and $\mathrm{w(a,b)^{SDD}}$ as in (18), $\mathrm{w(a,b)}$can be Correlation Similarity as in (3) or Vector Similarity as in (4). As is shown in Fig. 4, the first group used Correlation Similarity and weight sum as in (1), the second group used Vector Similarity and weight sum as in (1), the third group used Correlation Similarity and Weight Sum as in (2), the forth group used Vector Similarity and Weight Sum as in (2), the forth group show the best results. We use Vector Similarity and Weight Sum as in (2) in next experiments.

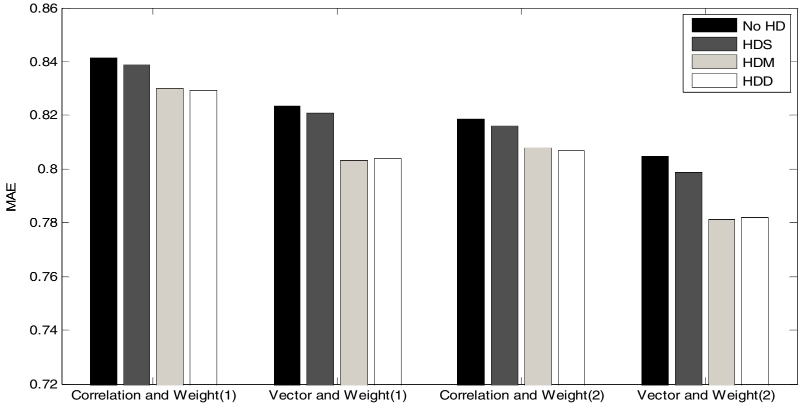


Fig. 4. Comparison of prediction quality of different similarity and prediction measures

In the case of size of users and items are unchanged, with the increasing of similar item size, using the HDM and HDD as the impact factor, using vector similarity and weight sum (2), the MAE can be reduced faster, as shown in the fig. 5.

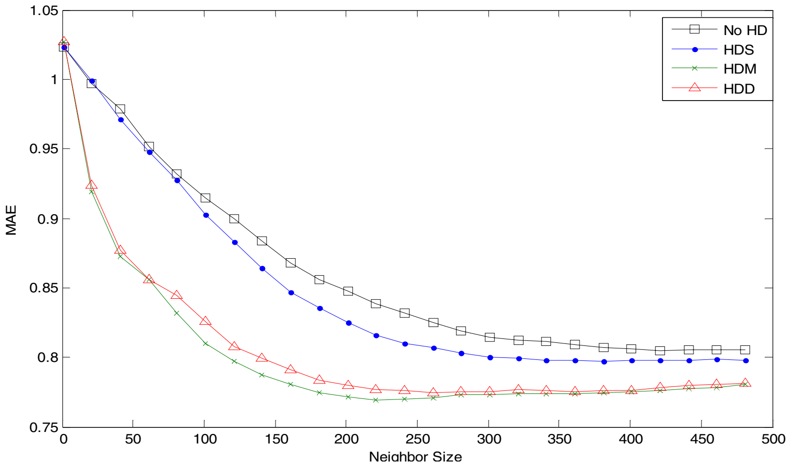


Fig. 5. Comparison of prediction quality based on Vector similarity and Weight Sum (2)

when similar set size is in the interval of [0,500], the accuracy was significantly improved by using the *HDM* and *HDD* as the factor, *MAE* decreased up to 11 percents (*HDM*) and 10 percents (*HDD*) at most comparing with no using any *HD*.

The data sets u1.base and u1.test through u5.base and u5.test are 80%/20% splits of the u data into training and test data. Each of u1… u5 has disjoint test sets. We used five data sets to compare the prediction quality, Weight Sum as in (2) was used to generate prediction, Vector Similarity was used as neighbor computing measure and take three types HDs as factor.

Table 1. Comparison of prediction quality of different similarity measures

Measures		**u1**	**u2**	**u3**	**u4**	**u5**
Vector Similarity and Weight Sum (2)	*No HD*	0.81	0.80	0.81	0.81	0.81
	HDS	0.80	0.79	0.79	0.79	0.79
	HDM	0.77	0.76	0.76	0.76	0.77
	HDD	0.77	0.76	0.76	0.76	0.77

As shown as the Table 1, all of three predictions, whose similarity measures is added HD as factor, have lower MAE than predictions whose similarity measures without HD factor. Using HDM as the factor has the lowest MAE in five tests.

6 Conclusion

In this paper we have proposed the Hesitation Degree to improve user-based CF. There are three kinds of Stability Degree, including *HDV*, *HDM* and *HDD*. Empirical analysis show that size of user, item and similar set do not have a great influence on the stability degree. In the experiment three types of Hesitation Degree were introduced into neighbor similarity computation, and the results show that the prediction accuracy can be improved by 11% at most, *MAE* of Prediction, which use *HDD* and *HDM* as the factor, reduce faster than using cosine-based similarity without HD factor.

Our further researches lay on developing more methods to get Stability Degree, Hesitation Degree in user Analysis and *HD* application in content-based recommendation system and recommendation system modeling [18].

References

1. Hill, W., Stead, L., Rosenstein, M., Furnas, G.: Recommending and Evaluating Choices in a Virtual Community of Use. In: Proceedings of CHI 1995 (1995)
2. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM 40(3), 77–87 (1997)
3. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: Proceedings of CSCW 1994, Chapel Hill, NC (1994)
4. Shardanand, U., Maes, P.: Social Information Filtering: Algorithms for Automating 'Word of Mouth'. In: Proceedings of CHI 1995, Denver, CO (1995)
5. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-Based Collaborative Filtering Recommendation Algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295 (2001)
6. Breese, J.S., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52 (1998)
7. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM 35(12), 61–70 (1992)
8. Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., Riedl, J.: GroupLens: An open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 Computer Supported Cooperative Work Conference, pp. 175–186. ACM, New York (1994)
9. Joachims, T., Freitag, D., Mitchell, T.: WebWatcher: a tour guide for the World Wide Web. In: Georgeff, M.P., Pollack, E.M. (eds.) Proceedings of the International Joint Conference on Artificial Intelligence, pp. 770–777. Morgan Kaufmann Publishers, San Francisco (1997)
10. Lieberman, H., Dyke, N.V., Vivacqua, A.: Let's browse: a collaborative web browsing agent. In: Maybury, M., Szekely, P., Thomas, C.G. (eds.) Proceedings of the International Conference on Intelligent User Interfaces, pp. 65–68. ACM Press, Los Angeles (1999)
11. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating word of mouth. In: Roberts, T., Robertson, S. (eds.) Proceedings of the ACM CHI 1995 Conference on Human Factors in Computing Systems, pp. 210–217. ACM Press, New York (1995)
12. Alton-Scheidl, R., Ekhall, J., Geloven, O.V., et al.: SELECT: social and collaborative filtering of web documents and news. In: Kobsa, A., Stephanidis, C. (eds.) Proceedings of the 5th ERCIM Workshop on User Interfaces for All: User-Tailored Information Environments, pp. 23–37 (1999)
13. Rucker, J., Polanco, M.J.: Siteseer: personalized navigation for the web. Communications of the ACM 40(3), 73–75 (1997)

14. Good, N., Schafer, B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J.: Combining Collaborative Filtering With Personal Agents for Better Recommendations. In: The AAAI 1999 Conference, pp. 439–446 (1999)
15. Sarwar, B.M., Konstan, J.A., Borchers, A., Herlocker, J., Miller, B., Riedl, J.: Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. In: Proceedings of CSCW 1998, Seattle,WA (1998)
16. Doob, L.W.: Hesitation: Impulsivity and Reflection, pp. 1–17. Greenwood Press, New York (1990)
17. Xiang-Wei, M., Yan, C., Li-Li, Q., Tao-Ying, L.: A new user profile model based on intuitionistic fuzzy set for personalized information analysis and sharing. In: 2009 International Conference on Management Science and Engineering - 16th Annual Conference Proceedings, ICMSE 2009, pp. 64–69 (2009)
18. Mu, X., Chen, Y., Li, N., Jiang, J.: Modeling of personalized recommendation system based on ontology. In: International Conference on Management and Service Science, MASS 2009, September 20 (2009)

Intrusion Detection System Based on Support Vector Machine Active Learning and Data Fusion

Man Zhao[1,2], Jing Zhai[2], and Zhouqian He[3]

[1] State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China
[2] School of Computer, China University of Geosciences, Wuhan 430074, China
[3] School of Foreign Language, China University of Geosciences, Wuhan 430074, China
zhaoman@cug.edu.cn

Abstract. As the viruses and Trojans become more and more rampant and ingenious, the Intrusion Detection technology is a new security technology which is considered to be the second safe gate after the fire wall. This thesis brings forth new ideas of Intrusion Detection System based on support vector machine active learning and data fusion which is completely different from traditional IDSs. This IDS model has an improved algorithm in its incident analysor part that presents some advantages of finding details of concrete attack detecting efficiency and being convenient to update because of the dependence of each classifiers.

Keywords: Intrusion Detection System, SVM, Active Learning, Data Fusion.

1 Introduction

Intrusion Detection Systems (IDS) have become an important part of operational computer security. They are the last line of defense against malicious hackers and help detecting attacks as well as mitigate their damage. However, intrusion detection systems are heavily dependent on expensive and scarce security experts for successful operation. By emphasizing self-learning algorithms, this model can reduce dependence on the domain expert but instead require massive amounts of labeled training data, another scarce resource in intrusion detection.

Traditional intrusion detection systems mainly are misuse detection system of centralization controlling or based on pattern matching. Pattern matching technology have high rate of attack detection to the type-known attack while low rate of attack detection to the type-unknown attack. In addition, the structures of centralization systems are easily to be attacked because of its weak adaptive ability. Therefore, it becomes a hotspot to use anomaly intrusion detection systems based on data mining or data fusion in this field. In the anomaly intrusion detection based on data mining technology, SVM which is discussed in literature [1] and [2] takes unique advantages in small samples, nonlinearity and high-dimensional domains. It is fitness to deal with imbalanced data sets of high-dimensional domains and isomerism[3]. On the other side, intrusion detection systems with data fusion center could solve the problems of weak adaptive ability and weak expansibility in traditional intrusion detection systems.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 272–279, 2010.

In this thesis we investigate whether an active learning with data fusion algorithm can perform on a par with a traditional self-learning algorithm in terms of detection accuracy but using significantly less labeled data.

2 Basic Theory Description

2.1 IDS

Intrusion Detection System (IDS for short) means the combination of software and hardware that complete the function of intrusion detection. IDS includes three function parts: the information source that offer time flowing record; the detection analysis engine that find signs of intrusion; the engine based on analysis of the results of the response part.

The following figure is to show a system structure of IDS. It includes the incident generator, incident analysor, incident database, response these four parts functions[4]. There are certain connections among these four parts.

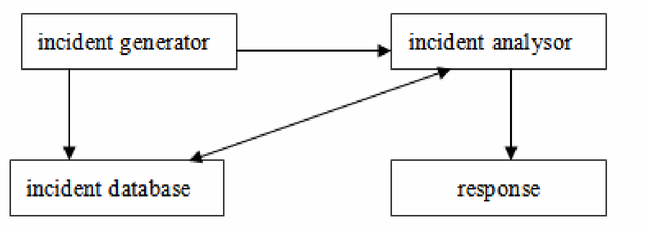


Fig. 1. IDS structure

(1) Incident generator

Responsible for the part of primitive data collection, track to the dataflow, daily record file, etc., then change initial data collected into the incident, and offer this incident to the other parts of the system. Data collection is to intrude the first step measured; its content includes systematic daily record, network data wrapping up, state and behavior of the user activities, etc.

(2) Incident analysor

Responsible for receiving the incident message, then analyze to them, judge that intrudes the behavior or anomaly phenomenon, change the result judged into the warning message finally, thus achieve the goal of measuring and is intruded. The commonly used information analytical method is the mode matches, statistical analysis and integrality analysis[5]. Among them first two kinds of methods are used in real-time intrusion to measure, but the integrality is analyzed and used in analyzing afterwards more. The research of a lot of IDS concentrates on how to improve the ability of information analysis, including improving and discerning known intruding accuracy and improving and finding unknown intruding probability etc..

(3) Incident database

It is the places of preserving the middle various and final data. In order to make it convenient for the system manager to look over and analyze the information of attacking, it receives the data from the incident generator or incident analysor, the information that will intrusion detection system and collect is kept, the information stored kept the digital evidence in order to attack at the same time.

(4) Response

React according to the warning message, it can cut off connecting, change strong responses such as file attribute, etc. It can be only a simple alarm too. It is the active weapon in the intrusion detection system. After analyzing and confirming the type attacked to the information of attacking, and then carry on corresponding treatment to the attack measuring.

From above we can know that, intrusion detection technology is a security technology to avoid a hacker attacking voluntarily. And the most important part is the incident analysor. So it is very essential for us to do the module of data analysis. If we choose better algorithms, the system will have stronger defense.

2.2 Active Learning

In traditional supervised learning problems, the learning algorithm use the labeled sample sets given by the outside world as training sets, then generalize the model. However, in a lot of reality applications, labeling a sample set is costly, difficult or boring, while getting unlabeled samples is relatively easy for us. Faced with this situation, the traditional method of supervised learning (i.e. passive learning) to meet the requirements of building a correct rate classification will be very difficult. Therefore, the active learning method is proposed to deal with these problems effectively.

In the active learning, learner can select unlabeled samples including a large amount of information actively, and send them to the experts to label, then place them in a training set for training[6]. Thus we can get higher classification accuracy in the case of smaller training set. So it can effectively reduce the cost of building high-performance classifier.

In general, active learning can be divided into two parts: learning engine and selecting engine. Learning engine is responsible for the maintenance of a standard classifier, and makes the labeled sample sets learning or training by the supervised learning algorithms so that the performance of the classifier can be improved. On the other side, the selecting engine is responsible for running the sample selection algorithm to choose an unlabeled sample and sends them to human experts to label, and then add the labeled sample to the labeled sample set. After that this updated labeled sample set will be used as a training sample set for training. Learning engine and selecting engine work in turn. After several cycles, the performance of the standard classifier is gradually improved. The whole process will be terminated until the classification accuracy rate of the learning algorithm achieved the default threshold. Active learning has more advantages than the traditional passive learning in the reducing samples' complexity. It has great development these years, but when the present situation is concerned, there are still a lot of deep researches for us to do.

2.3 SVM

SVM, short for Support Vector Machine (Vapnik, 1982) have strong theoretical foundations and excellent empirical successes. This method is a machine learning

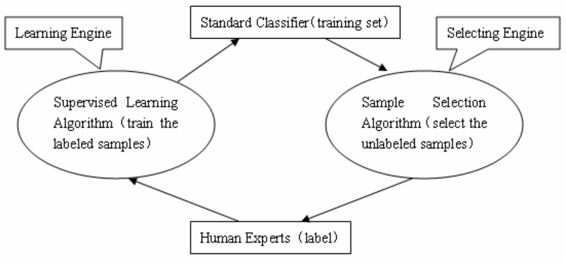

Learning Engine
Standard Classifier(training set)
Selecting Engine
Supervised Learning Algorithm (train the labeled samples)
Sample Selection Algorithm (select the unlabeled samples)
Human Experts (label)

Fig. 2. Active learning structure

method based on statistical learning theory. SVM can automatically find out the support vectors that have the good discriminant ability for classification. Thus the classifier constructed could maximize the interval between classes as a result of having better adaptability and higher accuracy rate. This method only requires the final classification results determined by the classes of samples from various types of domain margins.

The target of the SVM algorithm is looking for a hyperplane H(d) which can divides the data from training set into two parts[7], and they have the longest distance to the class domain margin which is orthogonal to the hyperplane. Thus SVM algorithm is also called maximum margin algorithm. The most samples in the sample set waiting for classification are not support vectors, so it nearly has no effects on the classification results if we move or reduce these samples. SVM algorithm can get a good classification result of classifying small sample sets automatically.

SVM is implemented by a pre-selection nonlinear mapping (kernel function) to make the input vector map to a higher dimensional feature space. The optimal classification hyperplane is structured at this dimension[8]. The process of using SVM for data sets classification is mapping the input space to the higher dimension al feature space, through some certain pre-selected nonlinear mappings.

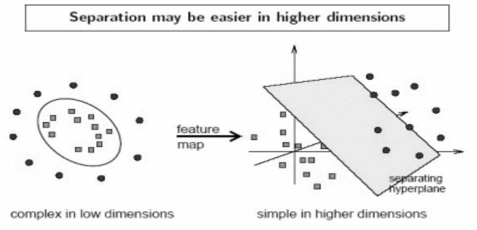

Separation may be easier in higher dimensions
feature map
separating hyperplane
complex in low dimensions
simple in higher dimensions

Fig. 3. Example of separation

Given a domain X, a linear support vector machine (Scholkopf et al., 1999) is defined in terms of the hyperplane

$$w \cdot x + b = 0 \tag{1}$$

corresponding to the decision function

$$f(x) = \mathrm{sgn}(w \cdot x + b) \tag{2}$$

for $w\in R^N$ and $b\in R$, Given a set of labeled data D = $\{(x_1,y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, where $x_i\in R^N$ and $y_i\in\{+1,-1\}$, the optimal hyperplane is the unique hyperplane that separates positive and negative examples for which the margin is maximized:

$$\max_{w,b}\left\{\min_{x_i}\left\{\|x-x_i\| : x\in R^N, w\cdot x+b=0\right\}\right\} \tag{3}$$

When the data are not separable, a soft margin classifier is used. This introduces a misclassification cost C, which is assigned to each misclassified training example.

Equation 3 is usually optimized by introducing Lagrange multipliers α_i and recasting the problem in terms of its Wolfe dual:

$$\text{maximize}: \; L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i x_j \tag{4}$$

$$\text{subject to}: \; 0\le\alpha_i\le C, \quad \sum_{i=0}\alpha_i y_i = 0 \tag{5}$$

The x_i for which are non-zero have a special meaning. They are the training examples which fall on the margin, and thus limit the position of the optimal hyperplane. These x_i are the support vectors. The x_i for which α_i=C also have special meaning — these are bound examples which are incorrectly classified or are within the margin of the hyperplane.

2.4 Data Fusion

Data fusion is generally defined as the use of techniques that combine data from multiple sources and gather that information in order to achieve inferences, which will be more efficient and potentially more accurate than if they were achieved by means of a single source.

Data fusion may be useful for several objectives such as detection, recognition, identification, tracking, change detection, decision making, etc.[9]. These objectives may be encountered in many application domains as Defense, Robotics, Medicine, Space, etc.

Using an efficient fusion scheme, one may expect significant advantages as:

(1) Improved confidence in decisions due to the use of complementary information. (e.g. silhouette of objects from visible image, active/non-active status from Infra-Red image, speed and range from radar, etc.)

(2) Improved performance to countermeasures. (it is very hard to camouflage an object in all possible wave-bands)

(3) Improved performance in adverse environmental conditions. Typically smoke or fog cause bad visible contrast and some weather conditions (rain) cause low thermal contrast (Infra Red imaging), combining both types of sensors should give better overall performance.

Fusion processes are often categorized as low, intermediate or high level fusion depending on the processing stage at which fusion takes place[10].

(1) Low level fusion, also called data fusion, combines several sources of raw data to produce new raw data that is expected to be more informative and synthetic than the inputs.

(2) Intermediate level fusion, also called feature level fusion, combines various features. Those features may come from several raw data sources (several sensors, different moments, etc.) or from the same raw data. In the latter case, the objective is to find relevant features among available features that might come from several feature extraction methods. The objective is to obtain a limited number of relevant features. Methods of feature fusion include Principal Component Analysis (PCA) , Diabolo shaped Multi-Layer Perceptrons (MLP) for the non-linear counterpart, etc.

(3) High level, also called decision fusion, combines decisions coming from several experts. By extension, one speaks of decision fusion even if the experts return a confidence (score) and not a decision. To distinguish both cases, one speaks of hard and soft fusion. Methods of decision fusion include voting methods, statistical methods, fuzzy logic based methods, etc.

Here is a simple data fusion model:

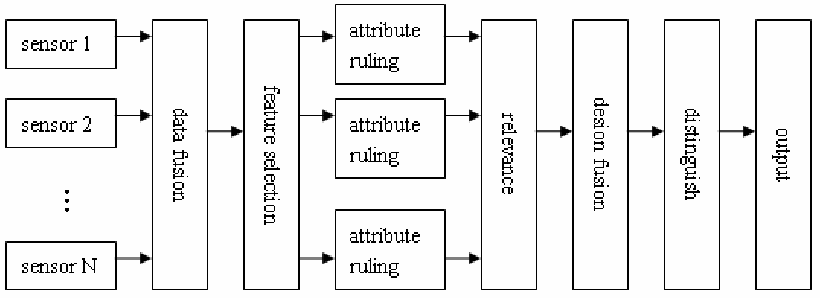


Fig. 4. A simple data fusion model

3 Algorithm Implementation

3.1 Algorithm Description

In this thesis we will combine data fusion, active learning and SVM to design an algorithm of IDS based on SVM active learning and data fusion which can obtain a better classification performance with less label cost. The algorithm is as follows:

Input: Candidate sample set U composed of the unlabeled candidate samples chosen from the data source set.

Output: Classifier f, pre-label samples.

(1) Preprocess samples in the candidate sample set U through N different preprocessors, then get N different candidate sample sets U_1, U_2,…, U_N . For each candidate sample set Un (n=1,2,…,N), do operation as follows:

(2) Select i samples from the candidate sample set U_n and label them correctly to construct an initial training sample set T_n which includes a sample’s output y is 1 and a sample’s output y is -1 at least.

(3) Construct a SVM classifier f_n according to the training set T_n.

(4) Apply f_n to all samples in U_n, label them as (x, $y^{\wedge}$) (thereamong $y^{\wedge}$ is the label which classifier fn pre-label the vector x).

(5) Choose an unlabeled sample (x_t, $y_t^{\wedge}$) which is the closest one near the class margin from the sample set U_n.

(6) Label the sample (x_t, $y_t^{\wedge}$) correctly to (x_t , y_t) and add it into the training set T_n (thereamong y_t is the correct label of x_t).

(7) If the test accuracy achieves at a certain value which we set before, turn to (8); or else turn to (3) and repeat steps.

(8) Send the last results f_1, f_2,…, f_N to the data fusion center, use relevant fusion algorithms, and output f.

3.2 Algorithm Flow

(1) Data Processing

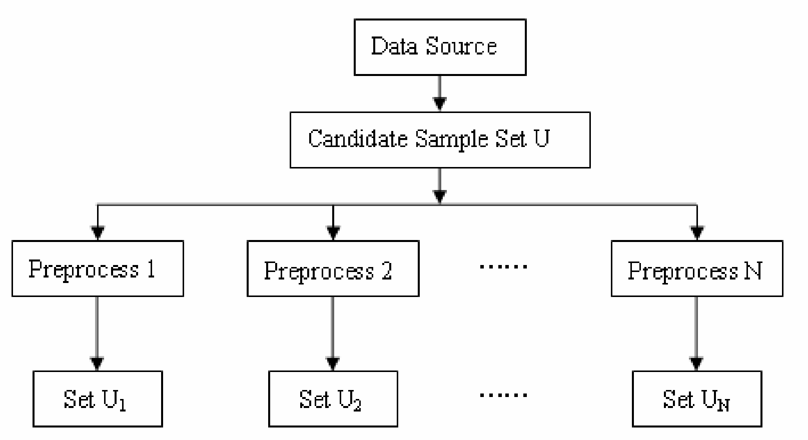


Fig. 5. Data processing model

(2) SVM Active Learning

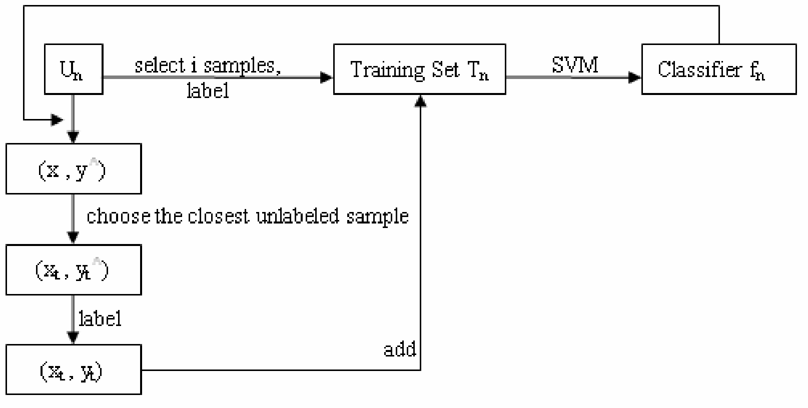


Fig. 6. SVM active learning model

(3) Data Fusion

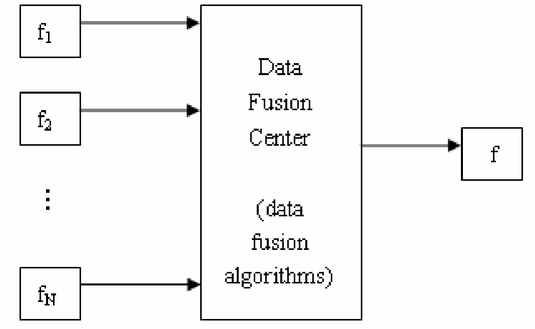


Fig. 7. Data Fusion model

4 Conclusions

This thesis brings forth new ideas of Intrusion Detection System based on support vector machine active learning and data fusion which is completely different from traditional IDSs. This IDS model has some more advantages than traditional IDS: Firstly, it could find details of concrete attack detecting efficiency. Secondly, it is convenient to update because of the dependence of each classifiers. It is believed that using this method could improve the performance of attack detecting effects to a certain extent.

References

1. Almgren, M., Jonsson, E.: Using Active Learning in Intrusion Detection, Chalmers University of Technology, Computer Engineering. In: Proc. IEEE Computer Security Foundations Workshop, pp. 88–98 (2004)
2. Schohn, G., Cohn, D.: Less is more: Active learning with support vector machines. In: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 839–846 (2000)
3. Xizhe, Z., Jie, Z., Cher, W.: Study on a SVM-based Data Fusion Method. In: Proc. of 2004 IEEE Conference on Robotics, Automation and Mechatronics, pp. 413–415. [Sn], Singapore (2004)
4. Mizutani, M.: The Design and Implementation of Session-Based IDS. Electronics & Communications, Part I: Communications 89(3) (2006) (8756–6621)
5. Lee, W.: Recent Advances in Intrusion Detection, 1st edn. Springer, Heidelberg (2001)
6. Stokes, J.W., Platt, J.C., Kravis, J., Shilman, M.: ALADIN: Active Learning for Statistical Intrusion Detection (2008)
7. Onoda, T., Murata, H.: SVM-based Interactive Document Retrieval with Active Learning. New Generation Computing 26, 49–61 (2008)
8. Tong, S., Koller, D.: Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research, 45–66 (2001)
9. Basst: Intrusion detection systems and multi-sensor data fusion. Communications of the ACM 43(4), 99–105 (2000)
10. Wennung, L., Chenkang, S.: News Video Classification Based on Multi-modal Information Fusion Image Processing. In: Proc. of 2005 IEEE International Conference on ICIE [S.1.]. IEEE Press, Los Alamitos (2005)

Long-Distant Pipeline Emergency Command Decision-Making System Based on Agent

Huagang He[1], Man Yang[2], and Ling Qian[3]

[1] Faculty of Engineering, China University of Geosciences,
Wuhan, Hubei 430074, China
[2] School of Journalism and Communication, Wuhan University,
Wuhan, Hubei 430072, China
[3] Qinghe Oil Extraction Factory,
Shengli Oilfield Subsidiary Company of SINOPEC

Abstract. According to full analysis of Long-distance Pipeline Decision-making System, integrating with the timeliness character of pipeline accident emergency, emergency command decision-making system based on Agent technology is proposed in this paper. Information Fusion Agent is applied to associate and analyze the data collected from different information resources. In terms of Assessment Agent, it includes Risk Damage Assessment, Operation Performance Evaluation and Coordination Assessment, which evaluate the personnel and property damage by Situation Assessment Agent after the pipeline accident. Resource Programming Assessment Agent is proposed, which employs CLIPS as the core platform, constructed principal modules as Structure Management Module, Regulation Management Module, Variable Analyzing Module and Compiling Module etc. The paper systematically analyzed the operational mechanism of emergency command system, and the association between Emergency Command Decision-making System and Emergency Command Agencies.

Keywords: Long-distance Pipeline, Emergency Command System, Agent, Expert System.

1 Introduction

Energy is the motivation of national economic development. In the energy sector, it was predicted in the 1990s of the last century that, 21st century will turn into the "natural gas era". The pipeline transmission sprung up 1970 and turn into the 5th way of the transportation following the railway, road, water and airline. In contrast with the other measures, pipeline is advantageous of high efficient and low cost in transporting gas; accordingly, it has been widely spread. The gas long-distant pipeline featured the high pressure, combustion, explosion, and complex environment; therefore it is more dangerous than the other chemical transportation. A 720mm diameter gas pipeline leaked out and exploded due to serious corrosion, causing intense fire and 10 deaths, leaving the ground a crater of 25m length, 6m depth at Southeastern

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 280–288, 2010.

New Mexico, August 2000. The huge fire ball rushed into the sky, which can be seen beyond 30km.

Emergency Command Decision-making System (ECDS) is an integrated system based on command and control (C2) process framework, including decision support model. The research of ECDS involved several decision-making command fields. The implementation decision-making process can be abstractly realized by SHORE (Stimuli, Hypothesis, Option, Response, and Environment) model. The process of C3I command controlling system was corresponded with the SHORE model, and SHORE C2 model was proposed (Whol et al.,1981). SHORE C2 model is a typical knowledge-based command controlling system, which is divided into four main basic processes: information acquisition, information fusion, resource allocation and planning, and program implementation.

Purposing to reduce the lost of people's lives and property, and minimize the consequences of pipeline damage caused by the accidents, an effective pipeline ECDS must be established.

2 System Infrastructures

In this paper, long-distance pipeline ECDS is based on a three-dimensional spatial decision support skill (SDSS), SDSS can help decision makers analyze spatial characteristics, logical relations and hierarchies from a large number of information, clear decision-making tasks and objectives, effectively generate a variety of solutions to the problem, study and compare their advantages and disadvantages and contradictions, then identify practical solutions and operations to take corresponding measures and proceedings. SDSS structures generally include three levels: system support layer, control layer and application layer. System support layer realize the system functions mainly by providing data support, model support, measurement support and knowledge support; main function application layer of system consists of disaster consultation, disaster stimulation, emergency decision-making, emergency command and feedback subsystem; system control layer set core control subsystem (including information upload, download and publication), primarily responsible for the coordination and control function of each part of the system, synthetically processing various data of system support layer and system application layer (Wang et al. 2002).

Through the analysis of SDSS system structure, combining with the emergency features of long-distance pipeline accidents, modifying the contents of control layer, long-distance pipeline ECDS can be divided into three sections, which are application layer, service layer and data layer (demonstrated in Fig. 1) Data layer serves mostly for system functional implementation by offering data support, model support, measurement support and knowledge support. It is mainly constituted by monitor, disaster and department information, geographic database, model base, decision-making knowledge base managing system. System application layer consists with Disaster Consultation System (DCS), Emergency Decision-making System (EDS), Emergency Command System (ECS) and Emergency Rehearsal System (ERS). Modules of service layer will provide specific functional implementation to application layer, such as Disaster Simulation Module (DSM), Emergency Assessment Module (EAM) Assessment Information Fusion Module (IFM), System Decision-making Module (SDM) etc. Data layer is the data foundation of ECDS, which serves for storing information associated to emergency process, such as geographic information, meteorological information, pipeline information, accident disaster information, expert knowledge and disaster model etc. Besides of data, data layer contents with database management system and videlicet database server. Existing data server consists with database management software as SQL SERVER based on Windows, INFORMIX based on LINUX, and ORACLE etc. The software listed above can provide efficient data search, data access, system maintenance, network transmission and other functions.

System service layer consists of basic functionality modules, which mutually cooperate proposing to complete the object of application layer. Service layer includes information integration, disaster assessment, and system decision-making, three-dimensional information platform service. System services layer is the main implementer of emergency system functions, it coordinate the data layer with application layer by processing and exchanging information. Application layer is the direct interface users interacting by with system, according to the different demands of emergency situation; it provides related functional support, including disaster consultation, emergency decision-making, emergency rehearsal, resource management and other subsystems.

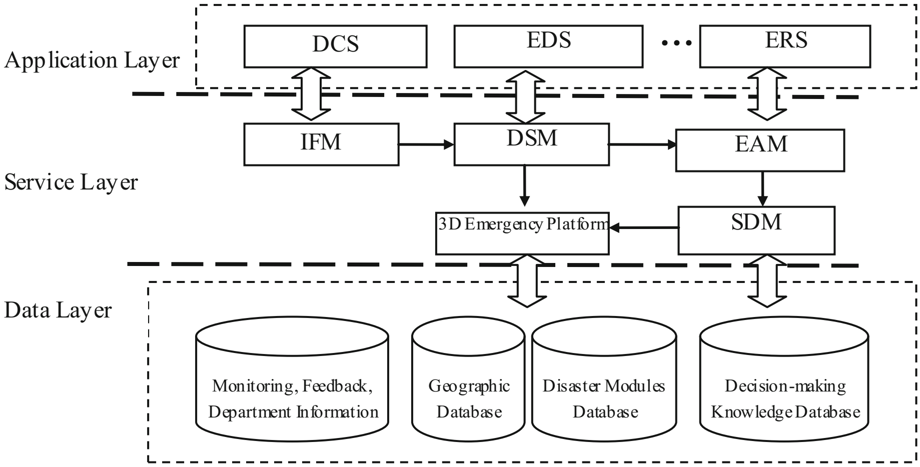


Fig. 1. System Infrastructures

3 Agent-Based Emergency Command Process Model

3.1 Information Fusion Agent (IFA)

According to the emergency features of natural gas pipeline accidents, IFA collect, organize, and process information of different sources, most possibly collect comprehensive relevant information to ensure a reliable and scientific command process.

As shown in Fig. 2, Fusion process of information data consists of data collection and data integration. Data involved in fusion process may be cited from: pipeline SCADA monitoring information, observed information in accident scene, communication information etc. Pipeline monitoring information includes pipeline operating conditions, which is an important basis of the initial ECDS. Observed information in scene is the main information source of ECDS, including current situation of the pipeline, climate, regional staff distribution, accident status and other information. Communications information is the relevant data system exchanges with exterior, the content may include the superior order or information, supporting information from government emergency agencies or organizations, information provided by meteorological department.

Information Collection primarily analysis data from different information sources, and classify the same types of data. Such as reporting information of on-site situation, computer simulation and prediction information of disaster, aerial inspection reports, reports of emergency response agencies. Foregoing information may contain content related accident injuries (injury to person, property damage), so they should be attributed to accident information. During Information Collection process, the reliability of the information (the reliability of sources, authenticity of information) should be evaluated.

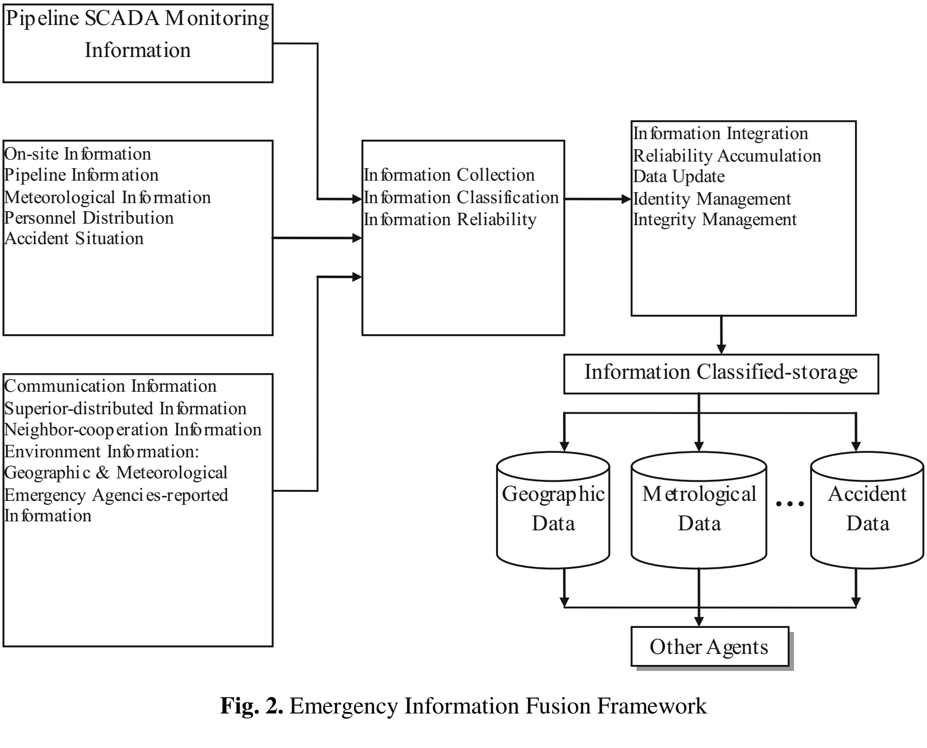


Fig. 2. Emergency Information Fusion Framework

3.2 Situation Assessment Agent (SAA)

Situation Assessment of long-distance pipeline disaster mainly aims to evaluate and analysis personal and property damage after pipeline accident, including Risk Damage Assessment (RDA), Operation Performance Evaluation (OPE) and Coordination Assessment (CA). Situation Assessment is essential judging basis of Emergency Command. Pipeline Emergency in Command Center (IC) regularly receives situation analysis report of scene, purposing to set goals and adjust resources for the next stage. If Situation Assessment result shows that, the scope of person injury impacted by the weather conditions on-site. IC must re-adjusted fire control, medical and transportation operations on-site.

RDA evaluates the scope and extent of personal injury and property damage caused by inflammable, explosive, or toxic gas around accident area, after pipeline leak accidents. OPE assess the operation performance of the fire control, medical, first aid, personnel transfer, pipeline repair, communications, transportation, logistical support etc during accident emergency process. CA aims to ensure the effect of communication among operational units, agents and personnel involved in pipeline accident ER, and it is an evaluation of the ability system coordinating various operational units.

3.3 Resource Programming Agent (RPA)

Resource Programming Expert System (RPES) is based on CLIPS core platform. CLIPS (C Language Integrated Production System) is an expert system tool developed by NASA's Johnson Space Center. The infrastructure of CLIPS is production system, adopting with forward inference mechanism. Contrasting with other general production system, CLIPS distinguish itself by unique RETE pattern matching algorithm utilized in inference process, which significantly improved the responding speed of system. Pipeline ER RPES utilizes the core-RETE matching pattern of CLIPS 6.24(latest version 6.30), as well as its syntax. The inference structure is exhibited in Fig. 3.

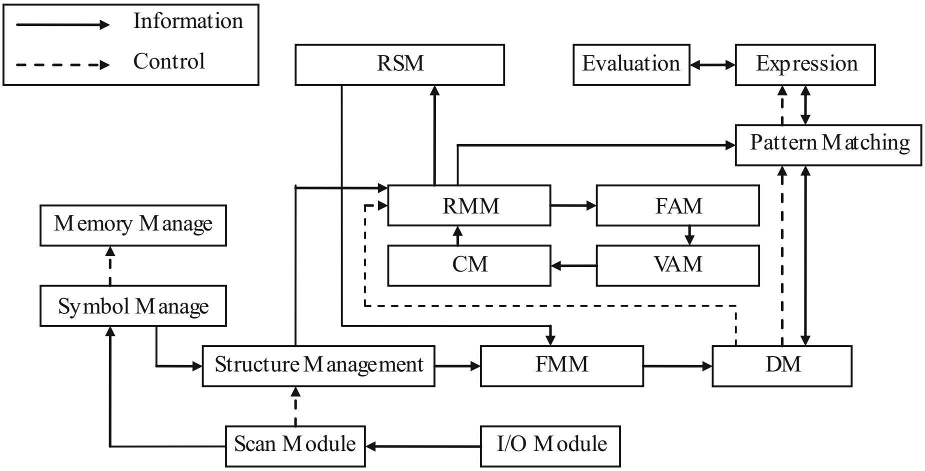


Fig. 3. Inference Structure of Expert System

During the establishment of expert system knowledge base, inputted information was analyzed by structure module, being identified as whether the Fact data or the Regulation expression. Regulation Management Module (RMM) coordinates the element format in both sides of the regulation, to form the correct expression. Variable Analyzing Module (VAM) evaluates the variable of regulation. Regarding to the variable appeared in multiple elements, VAM should reveal the correct variable expression in pattern network and connection network.

Compiling Module (CM) integrates the information and expression derived from VAM with pattern network and connect network. Pattern network is the most essential core of RETE algorithm; it prevented the information overload in the process of module matching. Pattern network and connection network were established simultaneous with Regulation input; different regulations could share the same pattern network and connection network, which declined the calculation amount of in sequence matching between Regulation and fact. Driving Module (DM) is the engine of expert system, while a new Fact enters the knowledge base, DM will guide the Fact to implement partial pattern matching (final module matching won't finish until other Fact have been inputted), and update the pattern and connection network of Regulation.

If several inputted Fact meet the matching requirement of Regulation, the Regulation will be activated and stored in Regulation Schedule Module (RSM). RSM is a manager of Regulation array following the first thing first principles to execute the operation of Regulation, and transfer the new Fact generated by Regulation execution into Fact Management Module to activate other Regulation. Circulating as mentioned above, when activated Regulation are eliminated in the Regulation Schedule Module, the inference process of knowledge will be finished and wait for the input of new Fact.

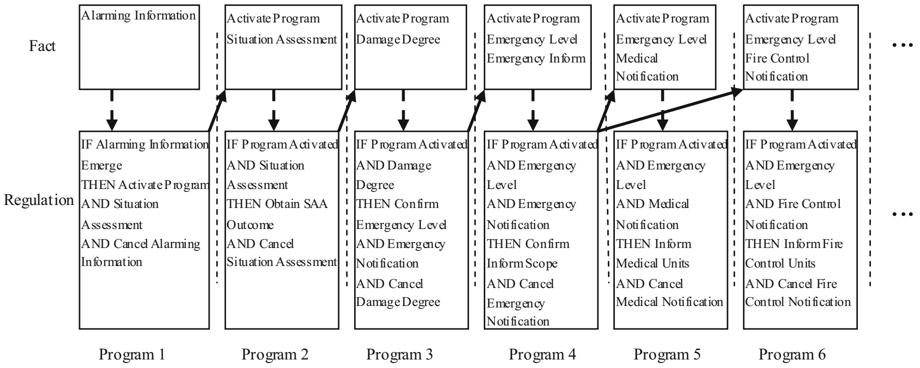


Fig. 4. Emergency Programming Process of Pipeline Accident

The partial of proposed resource programming processes is shown in Figure 4 in the initial part of long-distance pipeline accident emergency reaction. Upper part of the figure is the Fact information generated or collected during the emergency command process, for instance, Alarming Information inputted from external. Situation Assessment is generated from the programming Regulation itself. The inferior part of the figure is the Regulation knowledge of programming expert system, representing the considering procedure of programming expert. Dashed line separates the programming Regulation activated in the initial part of emergency; they are all attributed to the programming process.

When the ECDS receives alarm information (probably from SCADA, on-site staff or external alarm), the information access into the system through Data Fusion Agent.

Firstly, the expert system of proposed RPA will be activated to start programming process. Following the requirement of concept plan of emergency process, expert system should calculate the damage degree based on the Situation Assessment Agent to classify emergency level. According to different emergency level, Fact information will inform to local government, emergency agencies, rescue units and fire control departments. The constant Fact information will activate more Regulation, implement more operation, and generate more Fact information. Ultimately the operation chain and information chain will be constituted to finish the program of proposed resource.

4 Operational Mechanism of Emergency Command System

Pipeline ECDS is a supporting system serves to emergency manager; it can't replace the IC to directly make final operation decisions or to command the rescue operation. This system only utilizes pre-knowledge expert experience to collect the dynamic information in the process of accident emergency, and advance appropriate responding measurement to the corresponding accident as the reference for IC personnel. Both of Emergency Command Agency (ECA) and ECDS belong to pipeline accident emergency system. As the active side of the system, ECA presents demanding request to the decision-making system based on the necessity of current accident decision-making. As the passive side of the system, ECDS waits for the request from command agencies assisting them to analysis and process data, provide suggestions and con proposals. The two sides cooperate mutually completing the emergency rescue assignment fast and efficiently.

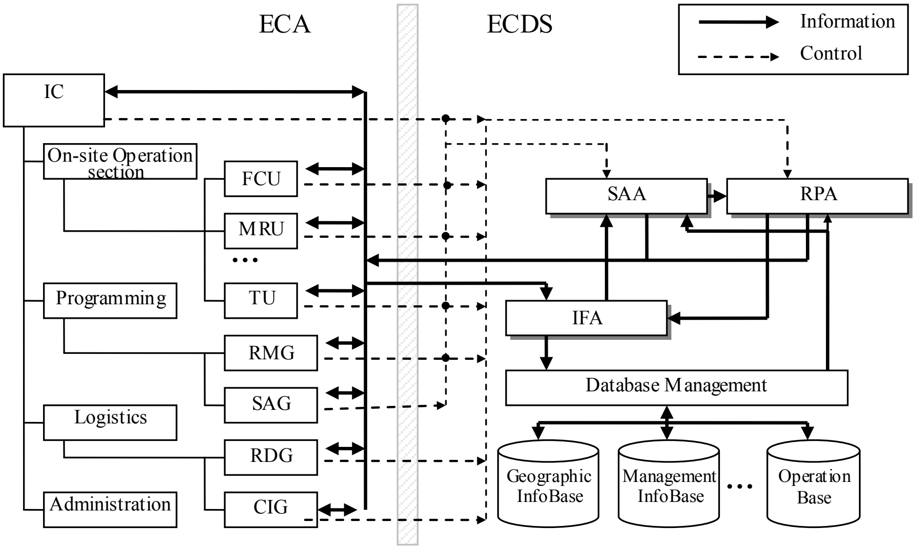


Fig. 5. Operational Mechanism of Emergency Command System

It's shown about the operation mechanism and process of ERCS in Fig. 5, which demonstrates the data communication and control request between ECA and ECDS. The continuous line represents the data flow between ECA and ECDS among internal agents of ECDS, and dashed line is the control command request from ECA sent to decision-making agents. The data exchange between ECA and ECDS mainly consists of the fusion and input of ER information, as well as result data output of agents in ECDS. Some sections such as IC, on-site operation section (including Fire Control Unit (FCU), Medical Rescue Unit (MEU) and Traffic Unit (TU) etc.), programming section (including Resource Programming Group (RPG) and Situation Analysis Group (SAG)), and logistics section (including Resource Distribution Group (RDG) and Communication Insurance Group (CIG)) will input the current information into ECDS, which comprises meteorological information, pipeline leaking circumstance and traffic condition etc. After the fusion, the input data will be analyzed by SAA and PSPA, and the results (risk damage degree, scope, impact area, operation plans, necessary resources, etc.) will be submitted to IC or assignment requesting units. For instance, SAG need to calculate the situation within the next two hours, and sent the prediction request to the SAA of ECDS. SAA calculated the concentration distribution, hazardous area, explosion and toxic extent of the next two hours by means of mathematical models such as Monte Carlo and based on the RT data in the database (wind field, leak rate, etc.), and then put back the data to SAG as the evidence of situation analysis.

5 Conclusions

ECDS is another Expert member of ECA ,which processes numerous of emergency information rapidly and efficiently and is capable of propose a reasonable operation plan instantly. ECDS has significantly expended the decision-making ability of IC, ensures the rapid implementation of emergency rescue operation, and minimizes the damage caused by the risks.

References

1. Wohl, J.G.: Force Management Decision Requirements for Air Force Tactical Command and Control. IEEE Transactions on Systems, Man, and Cybernetics 11 (1981)
2. Wang, S., Feng, Q.: Spatial decision support technologies apply in urban earthquake emergency command software system. World Earthquake Engineering 22 (2006)
3. Uhr, C., Johansson, H., Fredholm, L.: Analysing Emergency Response Systems. Journal of Contingencies and Crisis Management 16 (2008)
4. Alexander, D.E.: Principles of emergency planning and management. Oxford University Press, US (2002)
5. Dove, K.: Emergency management information system. A dissertation of Pepperdine University (2007)
6. Sugiyama, G., Chan, S.T.: A New Meteorological Data Assimilation Model for Real-Time Emergency Response. In: 10th Joint Conference on the applications of air pollution meteorology with the air and waste management association (1998)

7. Liang, T.-P.: Model Management for Group Decision Support. MIS Quarterly 12(4), 667–680 (1988)
8. Kivijärvi, H.: A Substance-Theory-Oriented Approach to the implementation of Organizational DSS. Decision Support Systems 20, 215–241
9. Cosgrave, J.: Decision making in emergencies. Disaster Prevention and Management 5 (1996)
10. Yoon, S.W., Velasquez, J.D., Partridge, B.K., Nof, S.Y.: Transportation security decision support system for emergency response: A training prototype. Decision Support Systems 46, 139–148 (2008)

Sales Forecasting Based on ERP System through BP Neural Networks

Min Zhang, Haifeng Zhang, and Yujuan Huang

College of Information Engineering, Dalian University,
Dalian, China
zhangmin_dlu@163.com, zhanghaifeng0616@163.com,
huangyujuan0112@126.com

Abstract. For the characteristics of manufacturing industry, with the application of business process analysis and optimization methods, the enterprise resources are optimized and a ERP system is formed. For the deficiencies of ERP in decision-making, BP neural network is used to mine the rules. To carry out the nonlinear prediction, the BP neural network defines the inherent relation between the inputs and outputs and the solution to the involved problem by adjusting the weight of the neural network. By extracting the data of the ERP and constructing the neural network prediction model, sales management functions of decision supporting module are to expanded so the sales data of ERP is transformed into the information of decision. In the end, the model is validated by the application.

Keywords: ERP, decision supporting, neural network, prediction.

1 Introduction

In recent years, with the development of information technology, in order to optimize the allocation of resources, improve the productivity of the corporation, many manufacturing companies have to optimize the producing process. But, many companies think that if the ERP management system has been introduced, all the problems from the enterprise can be solved, however traditional ERP can not respond to customer needs by the existing resources timely in the planning function, it is difficult to support real-world supply chain and to arrange production according to the forecasting. So how to utilize the information resources of the enterprise ERP and how to exert the effectiveness of information resources are currently pressing issues. In this context, data mining techniques is applied in the ERP system, making full use of rich data, processing the data, to mine potential valuable information from the data, in order to improve the depth of development and utilization of enterprise information resources, assisting the activities of enterprise management and decision-making.

As the artificial neural network can solve the problem of nonlinear prediction, in this paper, the ERP project is introduced in a manufacturing enterprise, improving and expanding the original sales management system, using the BP algorithm to achieve decision support on the sales forecasting, to meet the needs of business decision makers.

2 Enterprise Resource Planning

The ERP System is based on information technology, making use of advanced modern enterprise management ideas, providing a series of information services for procurement,

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 289–296, 2010.

sales, inventory, integrates the logistics, the people, the finance, the information into a management software. The following is function module of manufacturing enterprise ERP.

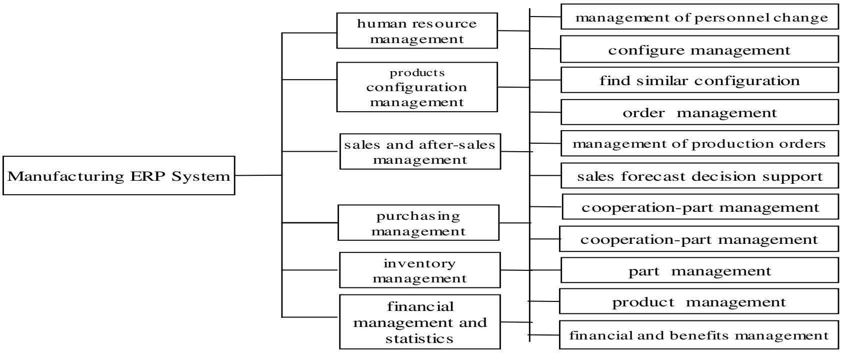

Manufacturing ERP System
human resource management
products configuration management
sales and after-sales management
purchasing management
inventory management
financial management and statistics
management of personnel change
configure management
find similar configuration
order management
management of production orders
sales forecast decision support
cooperation-part management
cooperation-part management
part management
product management
financial and benefits management

Fig. 1. ERP functional block diagram

The traditional ERP is evolved based on MRP Ⅱ, Its basic feature is the online transaction processing, generally with the main line of the enterprise's back management and the purpose of demanding forecasting and decision making[1], The following is the main business process analysis of a manufacturing enterprise:

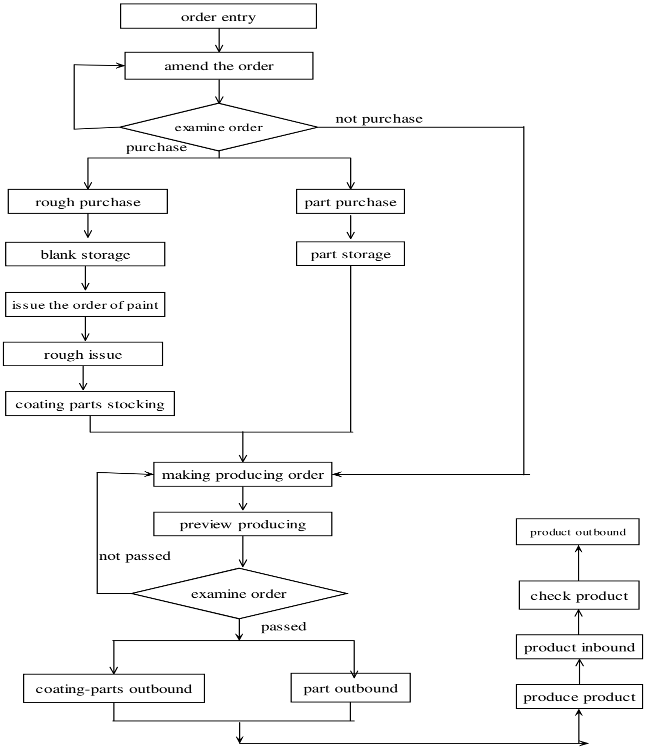


Fig. 2. ERP business flow chart

3 Sales Forecasting

Among many decision factors of enterprise marketing management, sales forecasting is one of important factors, largely determining the company's long-term planning and short-term planning. Sales forecast can be divided into sales forecasting with parameters and sales forecasting without parameters.

3.1 Sales Forecasting with Parameters

Mainly depending on price, customer income levels and other objective factors to forecast future sales. The difficulty is to find the sales contacts with various objective factors. Generally through the establishment of data warehouse system, establish the data source to analysis, identify links between quantity of sales and objective factors by data mining[2]. Then to predict the specific circumstances in the future by the rules finded out,implementation model is shown in figure:

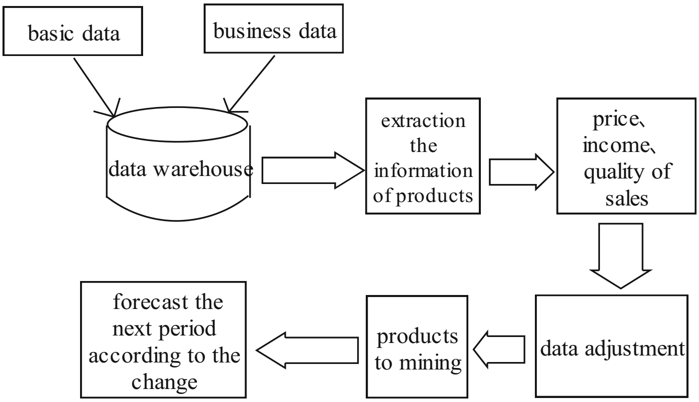

basic data
business data
data warehouse
extraction the information of products
price、income、quality of sales
data adjustment
products to mining
forecast the next period according to the change

Fig. 3. Prediction model with parameters

3.2 Sales Forecasting without Parameters

Based on sales of each past period, dynamic forecast sales situation of the next time period, provide the basis for the company's production plan. The data refer to the past sales of all time. This article focuses on non-parameter model and its implementation.

4 BP Artificial Neural Network Model Algorithm

Back-propagation algorithm (BP) is a multilayer feed forward neural networks of one-way transmission. Its structure: input nodes, output nodes, one or more hidden layers nodes. No any connection between the nodes of same layer, no coupling between the nodes of the same layer, therefore, the output on each layer nodes only influence output of the next layer nodes. BP algorithm automatically discovers class rules from the training sample under the guidance of the instructor, according to its network architecture, neuron state and the result is the value of the classification. The learning of BP neural network consist of transmission forwards of the input signal, error back propagation, iterative training, the convergence[3]. BP neural network structure is shown in figure 4:

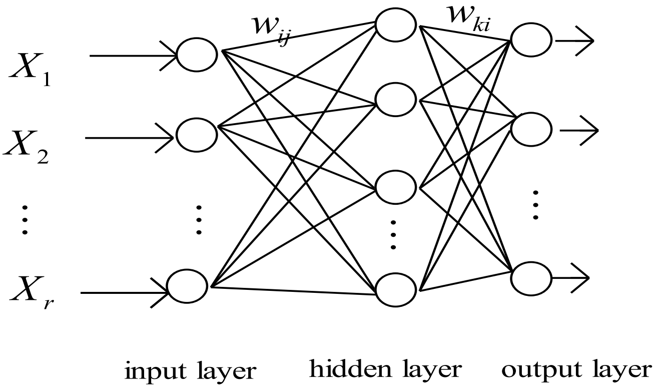


Fig. 4. Artificial neural network

If the input of neural network is matrix P,the network includes r input neurons, s_1 hidden neurons, transfer function is f_1,output layer has s_2 neurons, transfer function is f_2,real output is matrix A, desired output is T. Before starting the learning weights of neural network is initialized.

4.1 The Learning Steps of BP Neural Network

(1) extract one training sample from training set, put the sample into the network.
(2) calculate the ouput matrix of the network according to the formula.
the output of the ith hidden layer neuron is:

$$v_i = f_1(\sum_{j=1}^{r} w_{ij} p_j + b_j), i = 1, 2, \cdots, s_1 \tag{1}$$

the output of the kth output layer neurons is:

$$u_k = f_2(\sum_{i=1}^{s_1} w_{ki} v_i + b_k), k = 1, 2, \cdots, s_2 \tag{2}$$

(3) calculate the error between desired output and real output.

$$E = \frac{1}{2} \sum_{k=1}^{s_2} (t_k - u_k)^2 \tag{3}$$

(4) adjust the weights according to certain methods (gradient descent), so that decrease the error.

① weights changes in the output layer

for weights from ith input to kth output:

$\Delta w_{ki} = \eta \delta_{ki} v_i$, η is called as the "learning rate" ,control the rate of the change of weight.

Among the formula,

$$\delta_{ki} = (t_k - u_k) f_2' = e_k f_2' \quad (4)$$

$$e_k = t_k - u_k \quad (5)$$

②weights changes in the hidden layer:

$$\Delta w_{ij} = \eta \delta_{ij} p_j \quad (6)$$

$$\delta_{ij} = e_i f_1' \quad (7)$$

$$e_i = \sum_{k=1}^{s_2} \delta_{ki} w_{ki} \quad (8)$$

(5) repeat the steps(1)---(4), repeat the training on the training set, until the total error of the entire set get to the satisfactory value.

4.2 BP Network Defects

As the BP algorithm uses gradient descent to converge error between actual output and ideal output. It has its own limitations and constraints, mainly in the uncertainty of the training process, the detects can be summarized in the following:

(1) Local minimum
BP algorithm can make the network weights converge to a solution, but it can not guarantee that the solution is minimum on the error hyperplane. This is because the BP algorithm uses the gradient descent method, training is from a starting point along the error function slope gradually to the minimum error, for complex networks, The error function is multidimensional surfaces, like a bowl, the bottom is the minimum point, but the bowl's surface is uneven, therefore their training process may fall into the local minimum, the changing towards the each direction from this point will increase the error, so that training can not escape the local minimum.

(2) Longer training time
BP algorithm achieves object recognition by training the error back propagation and modifying the weight of network. For identification of a nonlinear equation, training usually takes a few thousand times, however, to realize identification of the

complicated nonlinear relationship or fuzzy uncertainty relation, need to train tens of thousands of times. This is mainly due to too small rate, so adopting changed learning rate or adaptive rate to train the network is necessary.

4.3 Improvement of Network

For the above shortcomings of BP network, make the following improvements, to accelerate the training speed and avoid local minima.

(1) Additional momentum method

It adds the proportional value to the variation of the previous value on the changes of weight based on the back-propagation generate the new change of weight according to Back-propagation algorithm, adjusting weight equation with additional momentum factor is:

$$\Delta w_{ij}(t+1) = (1-mc)\eta\delta_j p_j + mc\Delta w_{ij}(t) \tag{9}$$

$$\Delta b(t+1) = (1-mc)\eta\delta + mc\Delta b(t) \tag{10}$$

In the fomula, η is the step length of learning, t is the training times; mc is momentum factor, generally, the value is about 0.95.

After adding the momentum, make adjustment of the weight change towards average direction on the bottom of error surface, momentum has played the role of buffering and smoothing, it is useful to improve the stability of the process of network convergence, to adjust convergence speed, and contribute to the network steping out of the error surface from the local minimum[4].

(2) Adaptive learning rate method

Adaptive learning rate method can ensure the network to train with the largest acceptable learning rate. When a larger learning rate can make the network stably learn, make the errors continue to decline, to increases the learning rate, the network will learn with greater learning rate. Once with the excessive learning rate, it can not guarantee that the error continues to decrease, learning rate is reduced until the network learning process is stable.

The adjustment formula of adaptive learning rate:

$$\eta(t+1) = \begin{cases} 1.05\eta(t) & sse(t+1) < sse(t) \\ 0.7\eta(t) & sse(t+1) > 1.04sse(t) \\ \eta(t) & others \end{cases} \tag{11}$$

5 Forecasting Examples

In this paper, using the improved BP network algorithm to predict sales, based on the good producing process, combined with a manufacturer's sales records over the years, extracting the data from the background database, create a BP network, the network includes a input layer, a hidden layer and a output layer, using historical sales data from 1999 to 2005 as the training data, the data of 2006 as testing data, according to the principle of minimum error, to set the number of hidden layer nodes of the network, adjusting the parameters of the network to train the network, on the basis of the trained network, input the testing data to forecast the sales data of next year.

The following table includes historical sales data extracted from the ERP background database, from the year 1999 to the year 2006, and the next table is the comparison of the real sales data of 2006 and the forecasting sales data of 2006, the last figure is the simulation curve of the actual sales and forecast sales in the matlab environment.

Table 1. Monthly sales of the products sheet

month	1999	2000	2001	2002	2003	2004	2005	2006
1	112	115	145	171	196	204	242	284
2	118	126	150	180	196	204	242	284
3	132	141	178	193	236	235	267	317
4	129	135	163	181	235	227	269	313
5	121	125	172	183	229	234	270	318
6	135	149	178	218	243	264	315	374
7	148	170	199	230	264	302	364	413
8	148	170	199	242	272	293	347	405
9	136	158	184	209	237	259	312	355
10	119	133	162	191	211	229	274	306
11	104	114	146	172	180	203	237	271
12	118	140	166	194	201	229	278	306

Table 2. BP neural network prediction accuracy

2006	real value	forecasting value	error
1	284	287	1%
2	284	282	0.7%
3	317	319	0.6%
4	313	323	3.1%
5	318	321	0.3%
6	374	351	6.1%
7	413	417	0.9%
8	405	417	1.4%
9	355	358	0.8%
10	306	309	0.9%
11	271	283	4.4%
12	306	308	0.6%

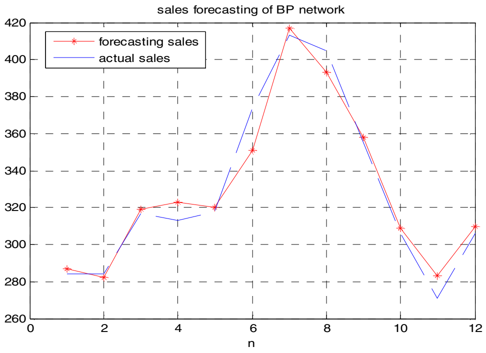


Fig. 5. BP network prediction curve

6 The Summary

The ERP system mentioned in the article runs very well in a manufacturing company, it played a role in resource allocation optimization and standarding business process.In the sales management module the prediction function provides a theoretical reference for management decision-making, thus grasp the market direction more effectively and timely, create production plans rationally. However, due to the inherent defects of the traditional BP network, there is a large forecast errors and other shortcomings of slow convergence, using the improved BP algorithm, such as dynamically adjusing learning rate, additional momentum and so on, prediction error, and slow convergence problems have been improved greatly.

References

1. Wang, P., Jianxin, D.: The research and application of marketing sales decision support about the ERP. Machine Design and Manufacturing 5, 175–219 (2008)
2. Chen, Y.: Sales forecasting system based on BP neural network. The Application of Microcomputer 11, 144–150 (2008)
3. Hong, L.: Nonlinear Combination Forecasting Model and Application Based on Neural Networks. In: International Conference on Control, Automation and Systems Engineering (2009)
4. Liu, Y.: Sales forecasting through fuzzy Neural networks. In: International Conference on Electronic Computer Technology (2009)

Capacity Allocation Policy of Third Party Warehousing with Dynamic Optimization

Chang Lin

School of Transportation Engineering, Tongji University, 201804, Shanghai, China
Linch@tongji.edu.cn

Abstract. Dynamic booking control is a hard problem in warehousing capacity allocation. With the similar method to solve stochastic knapsack problems, a dynamic and stochastic programming model is established in this paper. The dynamic booking control policies are put forward based on threshold value and the analysis of the characteristic of the model. Finally, optimization warehousing allocation policy is achieved by digital simulation. This shows that expecting revenue is concave function of surplus capacity, and also concave function of booking leading time, while the opportunity cost is a non-increase function of surplus capacity, and also a non- increase function of booking leading time. These policies provide scientific foundation for real-time decision-makers of 3PW companies.

Keywords: third party warehousing; revenue management; booking control; dynamic and stochastic programming.

1 Introduction

With the fast development of warehousing industry and the increasing intensity of competitive market, the revenue problem has become a hotspot that arouses great concern from the third party warehousing (3PW) companies [1].

According to the data released by Chinese Federation of Logistics and Purchasing in 2005, the internal rate of return (IRR) of the majority of logistics enterprises in China was only 1%, while U.S was 8.3% [2]. With the Government's close attention and policy support, logistics enterprises increased rapidly, furthermore, warehousing companies grew significantly faster than the sole transport companies and integrated logistics companies [3]; and revenue created grows much faster at 107.3% [3], investigated by National Development and Reform Commission of China in 2006.

However, costs have been increased 23.8% [3]. Under such circumstance, it’s not always feasible to keep the growth of profit only by reducing costs [4][5]. Revenue Management (RM) is such a highly effective tool [6], which can help companies sell their products or services to RIGHT customers at RIGHT price and RIGHT time, and make greatest revenue [7].

Only a few literatures focused on profits of third party warehousing. Analysis of third party warehousing contracts with commitments [1] was studied without revenue consideration with capacity allocation. Capacity allocation model for container

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 297–303, 2010.

logistics Revenue Management with empty container transportation [8], extended RM application, and guided many literatures focus on this topic. However, there are apparent difference between ocean shipping industry, hotel and human resource industry. Directly 3PW revenue optimization [4] [5] considered such difference and applied RM in warehousing industry successfully, and the fundamental part of third party warehousing Revenue Management was get focused without considering the complexity of multi-classed demand [9]. However, dynamic characteristic under time-varying demand was not included.

2 Problem Description and Hypotheses

Consider such a 3PW company, which starting to provide warehousing service for customers T days later, with maximum capacity C. Each customer arrives stochastically, and books a certain amount of stock place. Different physical characteristics of goods and different customer classes lead to different stock charge and different revenue contribution to the 3PW company. But according to experience, customers of low revenue contribution come first ordinarily; the higher class comes at the last.

Facing such orders, this company has to make the decision whether take this order, or not. If takes, opportunity cost maybe occurs; however, if not, satisfaction degree of customer will decrease, and give orders to other competitors, which also make opportunity cost happen. In order to full use the storage facilities and maximize revenue, this stochastic and dynamic capacity allocation problem is: how will this 3PW company allocate its limited warehousing capacity, which order to take and which is not.

Hypotheses are made as:

This company has customers from m classes, the probability of ith class customer arrives and give orders is $\lambda_t(t)$. So, $\lambda_0(t)$=1-$\lambda_i(t)$ means the probability of no customer arrival in t period of time. The revenue of each warehousing capacity is r_i, and without lose of generality, $r_1 > r_2 > \ldots > r_\mathrm{m}$ is assumed. The influence on company operation by goods stock time is not considered for the simplicity of this problem, which means all goods are stocked from the first day to the last. $V(t, x)$ is the expected revenue when company has x capacity to sell at time period t.

Similar way to the stochastic knapsack problem, dynamic warehousing capacity allocation model is obtained as formula (1).

$$\begin{aligned} V(t,x) &= \sum_{i=1}^{m} \lambda_i(t) \cdot \max\left\{\left(r_i + V(t+1,x-1)\right), V(t+1,x)\right\} + \lambda_0(t) \cdot V(t+1,x) \\ &= V(t+1,x) + \sum_{i=1}^{m} \lambda_i(t) \cdot \left[r_i - \left(V(t+1,x) - V(t+1,x-1)\right)\right]^+ \end{aligned} \tag{1}$$

In which, $[x]^+=\max\{0,x\}$;

Boundary conditions are:

$$V(T, x)= \textstyle\sum_i \lambda_i(T)\, r_i,\ x \in [0,C] \tag{2}$$

Formula (2) means if capacity is surplus by operating period, the best strategy is to take all orders which are less than the surplus capacity, and the expected revenue is the summation of all orders.

$$V(t,\ 0)=0, \mathrm{t} \in [0,T] \tag{3}$$

Formula (3) means the system revenue at the time period t is zero, when all capacity is pre-sold.

3 Elementary Model Properties Analysis

Formula (1) shows, the expected revenue function is monotonically, and it changes to the same direction with the remaining reservation time. The less remaining reservation time is, the less expected revenue is; the more surplus capacity is, the more expected revenue will be.

Proposition 1. $V(t,\ x)$ decreases with the increase of t; $V(t,\ x)$ increases with the increase of surplus capacity.

Proof: It is easy to be derived from formula (1), $V(t,\ x) \geq V(t+1,x)$,when x is under some condition. $V(t,\ x)$ decreases with the increase of t. The first part of proposition 1 is proved.

Mathematical induction can be used to get the proof of second part of proposition 1. When $t= T$, derived from boundary conditions:

$V(t,\ x) = \sum\lambda_i(T)r_i \geq 0, \mathrm{x} \geq 1$, $V(t,\ x) = 0$, $\mathrm{x}=0$, The first part of proposition 1 is proved. Finally, formula (1) can be derived as ：

$$\begin{aligned} V(T-1,x) &= V(T,x)+\sum_{i=1}^{m}\lambda_i(t)\cdot\left[r_i - V(T,x)+V(T,x-1)\right]^{+} \\ &\geq V(T,x-1)+\sum_{i=1}^{m}\lambda_i(t)\cdot\left[r_i - V(T,x)+V(T,x-1)\right]^{+} \end{aligned}$$

Which means, $V(T\text{-}1,\ x) \geq V(T\text{-}1,x)$, and so forth, when $t=T\text{-}1, T\text{-}2, \ldots, 1$, $V(t,\ x) \geq V(t,\ x\text{-}1)$, proposition 1 is proved.

Definition. ♉ $(t,\ x)$ is the opportunity cost for sell one capacity, and ♉ $(t,\ x) = V(t+1,\ x)\ \text{-}V(t+1,\ x\text{-}1)$, so proposition 2 can be obtained.

Proposition 2. When capacity surplus x at time period t, optimal decision rule $u_i^*(t,\ x)$ can be expressed as below:

$$u_i^*(t,\ x)=1, \text{ when } r_i \geq ♉\,(t,\ x);\ u_i^*(t,\ x)=0, \text{ when } r_i \leq ♉\,(t,\ x).$$

Proof: Derived from formula (1), $r_i+\ V(t+1,\ x\text{-}1)\text{-}\ V(t+1,\ x)$ can be looked as the pure revenue of accept the order from customers of ith class. For the risk neutral, the object of decision is maximize the expect revenue. Obviously, when opportunity cost is less than zero, company should accept this order, which means $r_i \geq ♉\,(t,\ x)$; Otherwise, order should be rejected.

Proposition 2 is proved.

The economic significance of proposition 2 is: when the charge of *i*th-classed goods is more than its opportunity cost, order should be accepted; otherwise, rejected. In consequence, the $u_i^*(t, x)$ is decision rule based on a kind of **threshold curve**, and expected revenue and opportunity cost function have such properties:

Proposition 3. $V(t, x)$ is the concave function of x and t; ♉ (t, x) is the non-increase function of x and t.

Proposition 3 has been proved by lots of literatures [10], [11], [12], and illustrative examples below will also prove this proposition.

Inference can be obtained from proposition 3:

(1) There is a KEY order time period t^*(also known as time threshold), when x and i are determined. Orders expressed before t^* will be rejected, and after t^*, accepted.

(2) There is a KEY surplus capacity x^*(also known as capacity threshold), when x and i are determined. Orders expressed will be accepted when surplus capacity is more than x^*; otherwise, rejected.

Proposition 3 and its inference show that: the value of each capacity increase gradually with the decrease of surplus capacity (equivalent to the opportunity cost is increasing); equally, the more the time period t is close to the starting operation time, the more the influence to the revenue is when possess more extra time period (equivalent to the opportunity cost is decreasing).

4 Illustrative Examples and Analysis

Giving time period T is 30, total capacity is 10, and each customer can and only can order one capacity, the arrival probability of customers are shown as table 1.

Table 1. The arrival probability of customers

	1-4	5-11	12-18	19-25	26-30
P_1	0.16	0.13	0.10	0.06	0.08
P_2	0.10	0.14	0.10	0.06	0.10
P_3	0.01	0.16	0.10	0.14	0.14

Using MATLAB 2006a, dynamic program is coded, and the result is give as figures. Main steps of coding are shown below:

(1) Initialize each variable. Time period is 30, total capacity is 10, and demand is 1, the value of P_1 P_2 P_3 are shown as TABLE I, expected revenue is 0, intermediate variable is 0;

(2) Time starting from 1 to 30 and surplus capacity is from 10 to 1. Calculating the recursion revenue and establishing the dynamic program model;

(3) Calculating sum expected revenue and probabilities of non-revenue, using total probability formula to calculate the summation revenue;

(4) Using MATLAB function PLOT to description figures, relationship of expected revenue and surplus capacity (Fig.1), relationship of expected revenue and

starting order time period (Fig.2), relationship of opportunity cost and surplus capacity (Fig.3), relationship of opportunity cost and starting order time period (Fig.4).

According to such steps, result can be got as below:

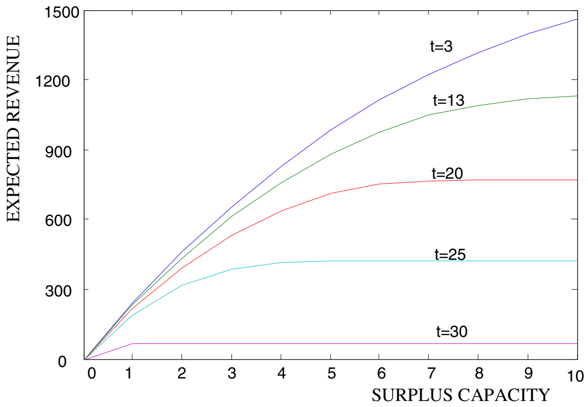


Fig. 1. Relationship of expected revenue and surplus capacity

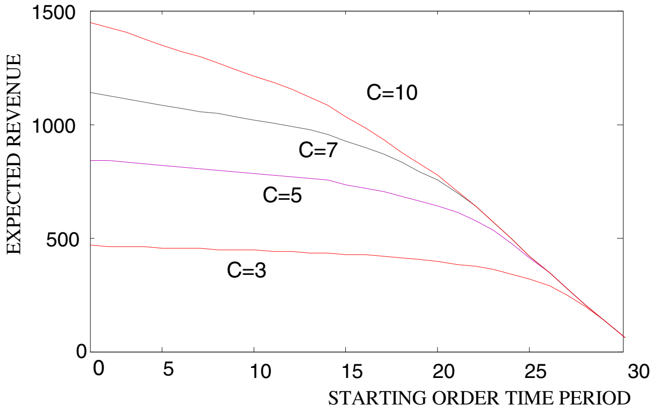


Fig. 2. Relationship of expected revenue and starting order time period

From Fig.1 and Fig.2 illuminate that: when starting time period t is determined, expect revenue is concave function of surplus capacity x; and when surplus capacity x is determined, expected revenue is concave function of starting time period t.

From Fig.3 and Fig.4 illuminate that: when starting time period t is determined, opportunity cost of accepting an order is non-increasing function of surplus capacity x; and when surplus capacity x is determined, opportunity cost of accepting an order is non-increasing function of starting time period t.

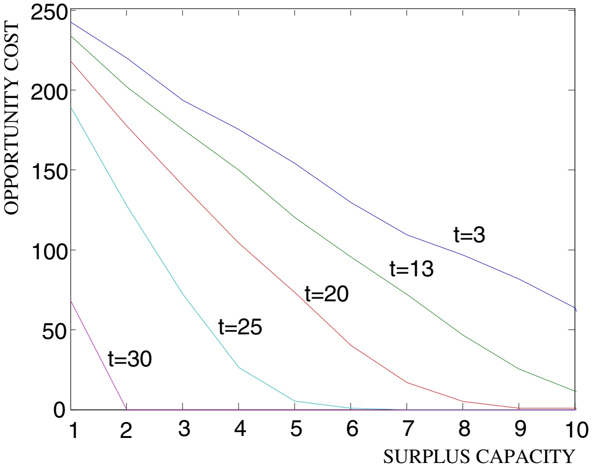


Fig. 3. Relationship of opportunity cost and surplus capacity

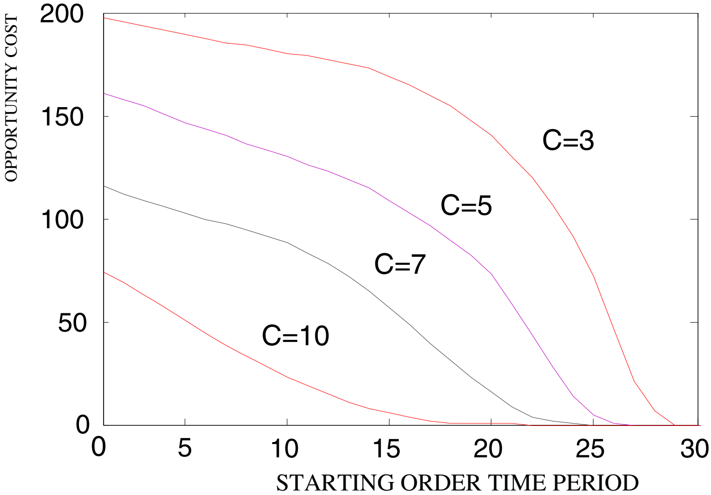


Fig. 4. Relationship of opportunity cost and starting order time period

5 Conclusions and Future Direction

In this paper, dynamic booking control problem in warehousing capacity allocation is studied. A dynamic and stochastic programming model is established with the similar method to solve stochastic knapsack problems. Based on the analysis of the characteristic of the model, the dynamic booking control policies are put forward based on threshold value. Finally, the optimization warehousing allocation policy is achieved by digital simulation, which shows that expecting revenue is concave function of surplus capacity, and is also concave function of booking leading time, the opportunity costs is a non-increase function of surplus capacity, and is also is a non- increase function of booking leading time. The dynamic booking control policies provide scientific foundation for real-time decision-making of 3PW companies.

However, physical characteristics of goods (such as stock time and goods volume) are not considered in this paper. Actually, such parameters can be integrated in one model, using warehousing platform as an underside, stock time as the height of cube, and the object of the problem is maximizing the space utilization of this cube, which is alike suspended stochastic encasing problem. However, such problems are NP-hard ones. The future direction is how to model this problem and find a good algorithm to solve it.

Acknowledgment

This research is supported by the outstanding researcher grants program of the Tongji University. ISICA 2010 Conference is gratefully acknowledged.

References

1. Chen, F.Y., Hum, S.H., et al.: Analysis of third party warehousing contracts with commitments. European Journal of Operation Research 131, 603–610 (2001)
2. CLPP China.: National logistics companies' survey (2005)
3. NDRC China.: National key logistics companies' survey(2006)
4. Lin, C., Shuai, B., et al.: Capacity Allocation Optimization of Third Party Warehousing with Demand Updating. Systems Engineering 25(3), 73–77 (2007)
5. Lin, C., Huang, Q., Bu, X.Z.: Revenue Optimization of Capacity Scheme and Allocation of Third Party Warehousing. Journal of Southwest Jiaotong University 42(3), 320–325 (2007)
6. McGill, J.I., van Ryzin, G.J.: Revenue management: research overview and prospects. Transportation Science 33, 233–256 (1999)
7. Boyd, E.A., And Bilegan, I.C.: Revenue Management and E-Commerce. Management Science 49(10), 1363–1386 (2003)
8. Bu, X.Z., Zhao, W.Q., et al.: Optimal Capacity Allocation Model of Ocean Shipping Container Revenue Management Considering Empty Container Transportation. Chinese Journal of Management Science 13(1), 71–75 (2005)
9. Lin, C.: Distribution Processing Capacity Allocation Optimization for Third Party Warehousing Based on Revenue Management. In: Proceedings of ICTE 2009, Chengdu, China, pp. 3514–3519 (2009)
10. Lee, T.C., Hersh, M.: A model for dynamic airline seat inventory control with multiple seat bookings. Transport Science 27(3), 252–265 (1993)
11. Lin, G.Y., Lu, Y., Yao, D.D.: The Stochastic Knapsack Revisited: Structure, Switch-Over Policies, and Dynamic Pricing. Working paper, IBM T.J. Watson Research Center (2002)
12. Subramanian, J., Stidham, S., Lautenbacher, C.J.: Airline yield management problem with over-booking, cancellations, and no shows. Transport Science 33(2), 147–167 (1999)

Dynamical Multi-objective Optimization Using Evolutionary Algorithm for Engineering

Lingling Wang and Yuanxiang Li

State Key Laboratory of Software Engineering, Wuhan University,
430072 Wuhan, P.R. China
{llwang,yxli}@whu.edu.cn

Abstract. This paper deals with multi-attribute classification problem based on dynamical multi-objective optimization approaches. The matching of attribute is seen as objective of the problem and user preferences are uncertain and changeable. Traditional sum weighted method and simple evolutionary algorithm are employed for experimental study over practical industry product classification problems. A integrate system framework is proposed to realize the dynamical model for multi-objective optimization. The experimental results show that classification performance system can be improved under the dynamical system framework according to user preference.

Keywords: multi-objective optimization, multi-attribute classification, sum weighted, evolutionary algorithm.

1 Introduction

Optimization refers to choosing the best from available alternatives, which can be considered as look for the maximum or minimum of one or more desired objectives. Optimization has many applications in engineering, industry, science, business, economics, etc. It has attracted the interest and attention from many engineers in various areas for several decades. For some engineering applications, such as optimization of simple chemical processes, we can transform them into single objective optimization problems, however, for most practical applications, the problems cannot be considered as simple optimization of one objective and several objectives should be considered simultaneously. The appropriate objectives for a particular application are often conflicting, which means that a trade-off is needed for achieving a satisfying result for a practical application.

Multi-objective optimization, also known as multi-criteria or multi-attribute optimization, is the process of simultaneously optimizing two or more conflicting objectives subject to certain constraints. Unlike in single objective optimization which gives a unique optimal solution, there will be many optimal solutions for a multi-objective problem. Marler and Arora [2] divided the multi-objective optimization methods into three major categories: methods with, methods with a posteriori articulation of preferences, and methods with no articulation of preferences. If the user has an articulate preference and can express an explicit approximation of the preference

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 304–311, 2010.

function. They can specify preferences with the relative importance of different objectives. However, in some cases, it is difficult for a decision-maker to give an explicit preference. Therefore, the methods in the second category can be effective to allow the decision-maker to choose from a palette of solutions by determining a representation of the Pareto optimal set [1, 7]. Allowing the decision-makers choose to choose gives them more freedom on problem solution, but too many candidates may also cause confusion and fatigue [6].

In this paper we proposed a dynamical multi-objective optimization method to deal with multi-attribute classification problem in practical engineering application. In this problem the attributes may not be all exactly satisfied when a sample matches with a certain type in all categories. Thus the trade-off among these attribute is based on the preference of users. In practice, different users may have different preferences, the preferences may change with the condition, and users are not like to choose from too many candidates. Considering all these things, we introduced EA into sum weighted method, the simplest multi-objective optimization method, to realize adaptive classification according to user's variational preferences. The rest of this paper is arranged as follows:

In section 2, we introduce the problem of dynamical multi-objective optimization and make a specific description for our problem. In section 3, we show how to make use of evolutionary algorithm to deal with the multi-attribute classification problem and give a framework of the whole dynamical system. We present experimental results in section 4 and the whole work is concluded in section 5.

2 Dynamical Multi-objective Optimization Problem

Multi-objective optimization problems are ubiquitous in engineering applications, such as operation decision based on environmental conditions in procedural control process, etc. In all these applications, multi-attribute classification problem is a kind of representative problem, which classifies targets into different classes according to the matching degree with a group of attributes related to these different classes. When there is a class that matches all attributes more than other classes, the problem degrades to simple template matching problem. While there are several classes that match better than others on different attributes, the problem becomes a kind of multi-objective optimization problem. What's more, if the optimal classification decision should be changed with different decision-makers and conditions, the problem can be considered as a kind of dynamical multi-objective optimization problem. In this paper, the multi-attribute classification problem is studied as the representative problem for dynamical multi-objective optimization problems.

In general, a multi-objective optimization problem can be described as (1), where $\mathbf{f}(\mathbf{x})$ represents the set of objectives and $\mathbf{x}$ represents the vector of decision variables [1].

$$\begin{aligned}
&\min \mathbf{f}(\mathbf{x}) = \min\{f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots f_m(\mathbf{x})\} \\
&\text{with respect to } \mathbf{x} \\
&\text{subject to} \quad \mathbf{x}^L \le \mathbf{x} \le \mathbf{x}^U \\
&\qquad\qquad\quad \mathbf{h}(\mathbf{x}) = 0 \\
&\qquad\qquad\quad \mathbf{g}(\mathbf{x}) \le 0
\end{aligned} \tag{1}$$

For the multi-attribute classification problem, the different objectives should be considered as finding the best matching with different classes on different attributes. The definition of multi-attribute classification problem is as follows:

$$\min_{\mathbf{y}} \mathbf{f}(\mathbf{x}^*, \mathbf{y}) \qquad \mathbf{y} \in C = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_m\}$$

where

$$\begin{aligned} &\mathbf{y}_j = (A_1^j, A_2^j, \cdots A_n^j), \quad A_i^j \subset R_i \\ &\mathbf{x}^* = (a_1^*, a_2^*, \cdots a_n^*), \quad a_i^* \in R_i \\ &\mathbf{f}(\mathbf{x}^*, \mathbf{y}_j) = D(\mathbf{x}^*, \mathbf{y}_j) = \left\langle d_1(a_1^*, A_1^j), d_2(a_2^*, A_2^j), \cdots, d_i(a_i^*, A_i^j) \right\rangle \end{aligned} \tag{2}$$

Distance on attribute A_i between $\mathbf{x}^*$ and $\mathbf{y}_j$ is denoted as $d_i(a_i^*, A_i^j)$ and there can be different means for evaluation of distance on different attributes.

According to the previous researches, there are many different methods to determine the optimal solution for multi-objective problems. For simple illustration, traditional weighted sum method is investigated here to present our method for dynamical optimization. The distance calculations for different attributes are considered to be totally same. Then the formula of $\mathbf{f}(\mathbf{x}^*, \mathbf{y}_j)$ in (2) can be changed into:

$$\mathbf{f}(\mathbf{x}^*, \mathbf{y}_j) = \sum_{i=1}^{n} w_i(t) d_i(a_i^*, A_i^j) \tag{3}$$

where $w_i(t)$ is the weight for attribute A_i at time t, and it can be changed dynamically and determined by both the time and user feedback.

3 Evolutionary Algorithm for Dynamical Weights

For multi-attribute classification problem described above, if good weights can be defined in advance, it is just a simple weighted sum problem. However, satisfying weights are not available in most practical applications. A great amount of experiments are needed before the parameters are tuned well, and sometimes, useful data can only be obtained after the optimization system is applied in real environment. Thus, dynamic weights, which can be adjusted automatically during the use of the model, are very important for engineering applications.

Generating the new weights according to user feedback can be computationally expensive and no general mathematical model can be employed directly. For this reason, stochastic search strategies, such as evolutionary algorithms, simulated annealing, and ant colony optimization, can be taken into consideration. They usually do not guarantee to find optimal one, but try to find a good approximation. Here, we take the evolutionary algorithm as the means to realize the dynamical weights.

3.1 System Model for Dynamical Multi-attribute Classification

Before evolutionary algorithm for dynamical weights is presented, the optimization system model for multi-attribute classification is detailed described here (see Fig.1.).

The model $\mathbf{f}(\mathbf{x},\mathbf{y})$ should be build according to the optimized $(w_1, w_2, \cdots, w_n)$ based on the sample set and target classes. When a user employs the model to classify a new sample, he/she can get an answer from the system. If the answer is just the one he/she wants, then no user feedback is needed. Or else, the user can give a feedback of 'right' class to the system, and then the feedback process is motivated. To respond the user feedback, the weights $(w_1, w_2, \cdots, w_n)$ will be optimized again based on both exist sample set and new sample using evolutionary algorithm.

The detailed algorithm for optimizing weights is presented in next subsection.

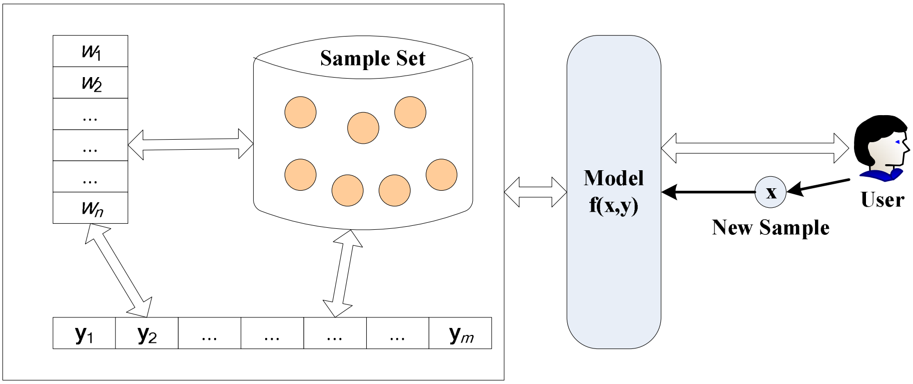


Fig. 1. Optimization system model for multi-attribute classification

3.2 Evolutionary Algorithm for Dynamical Multi-attribute Classification

Evolutionary algorithms (EAs) are search methods that come from Darwin's theory of natural selection and survival of the fittest in the biological world. Different from other traditional optimization techniques, EAs start a search from a "population" of solutions, not from a single point. Each iteration, also called "generation", of an EA involves a competitive selection that wipes off poor solutions. The solutions with good "fitness" do "crossover" with other solutions by swapping parts of a solution with another. Solutions are also "mutated" by making a small change to a single element of the solution. Crossover and mutation are used to generate new solutions, in which only those with good fitness can survive. This process pushes the search towards regions of the optimal solution. Pseudo-code for a general genetic algorithm is as follows:

```
Initialize the population
Evaluate initial population
Repeat
     Perform competitive selection
     Apply genetic operators to generate new solutions
     Evaluate solutions in the population
Until some stop criteria is satisfied
```

EAs have been successfully applied to a variety of optimization problems and a variety of methods have been proposed to improve the traditional EA. Although there may be better choices of improved EAs [3, 4], here we just take the standard EA with elitism strategy as the basic optimization approach to evaluate the dynamical model, in order to validate the optimization process. Some important settings for the EA is presented as follows.

Population. The individuals in population is coded with real number vector $(w_1, w_2, \cdots, w_n)$, $0 < w_i < 1$. The population size should not be too large considering time complexity of the training process. A population size between 10 to 20 is recommended.

Genetic Operators. Crossover operator and mutation operator are obvious in this problem. By swapping part of vector $(w_1, w_2, \cdots, w_n)$ of two individuals, crossover is done. When mutation is needed, simply replace the element w_i with a real number between 0 and 1.

Fitness Evaluation. Fitness evaluation may be the most complex and important part in this problem. The rate of successful matching for the sample set under the model $\mathbf{f}(\mathbf{x}, \mathbf{y})$ based on $(w_1, w_2, \cdots, w_n)$ is employed to evaluate the fitness. Pseudo-code for fitness evaluation is as follows:

```
Initialize Matching Value: MV•0;
For each element s in sample set do
        Determine the class of s using model f(x,y);
        If y(s) is equal to the real class of s
            MV • MV + con(s);
Return MR as fitness of model f(x,y).
```

The matching rate represents the accuracy of the decision made by the model. In practical, 100% accuracy cannot be assured, so samples with more importance should be assigned with larger contribution value $con(\mathbf{s})$ to make sure that they will be satisfied in priority. Elements in original sample set, which may not be arranged carefully and is just a group of sample we can obtain conveniently, could be set the basic contribution value. The first model of the whole optimization system can be trained with the original sample set.

After the first model has been built up, users can make use of the model to classify their samples. The system will collect the user feedback on the effect of classification. For those user samples that obtain the satisfying results, the corresponding samples with the same class mark in sample set will increase their value of $con(\mathbf{s})$. Once user finds that the classification result of a certain sample is “wrong”, the sample should be added into the sample set and evaluated a relative large value. Then the training progress should be motivated to renew the model according to the new sample set.

Competitive Selection. The selection of individuals is based on elitism strategy. The best individual should be chosen to survive in population and has the largest probability to do crossover. Other individuals should be chosen to dead according to their fitness, larger matching rate smaller chance to be wiped off.

Stop Criteria. The evolutionary process should be stopped when the best fitness have not changed for a certain period and the difference between the best and the worst is small enough.

4 Experimental Results

In order to evaluate the performance of our optimization system and algorithm, experimental tests are performed based on a set of data that come from a practical application on industry product classification. The aim is to classify products into 70 different categories and 16 attributes are used to distinguish them. The problem of this application lies in: (1) lack of expertise knowledge for defining the weights of attributes in advance; (2) lack of uniform preference for different users; (3) lack of the product samples of some categories, although the standard features of categories can be obtained. These characters of the application acquire the classification system to deal with multi-attribute simultaneously and dynamically evolve based on user feedback. Our dynamical multi-attribute classification system model fits well to this kind of problem.

There are 1300 samples related to the 70 categories respectively. The experiment is divided into three phases. At the first phases, about 600 samples with 60 of the 70 categories are chosen to form the original sample set and the first model is built up using the EA proposed in section 3. The second phase is incremental learning phase, and other 500 samples in the 60 categories are used to test the first model and dynamically renew it. Finally, the other 200 samples, which relate to the rest 10 categories, are employed to test the extensibility of the system.

The parameters of the EA are set as follows: population size is 15; crossover and mutation rate are 0.8 and 0.05; contribution values of all original samples are set to be 1 and the increment value of them is 0.1. The evolutionary loop should be stopped when the best fitness have not changed for 20 generations and the average difference between the best fitness and the worst one is smaller than 0.01. What's more, the size of the sample set should be limited to 15 times to the number of categories. If a new sample should be added to the sample set that is full, a sample, which is in a categories with more than 10 samples and have smallest contribution value, should be wipe off from the sample set.

In this application, EA has been executed for 12 times to build or renew a model, and the best one is chosen. We also build 5 different "first models" based on the same samples to find that 12 times training should bring us a relative stable accuracy of model and the results are shown in Table 1.

Table 1. Accuracy and average generations of EA execution for 5 models

	Model 1	Model 2	Model 3	Model 4	Model 5
Accuracy (%)	98.6%	97.7%	98.6%	99.0%	98.7%
Average Generations	223	196	262	155	291

After the first models are built up, 500 samples more in the 60 categories are used to test the ability of incremental learning of this system. Here, we define "user satisfaction" to evaluate the system. Classification accuracy of new samples using the first model is denoted as "pre-satisfaction" and that using the renewed model is denoted as "post-satisfaction". The same experimental method is adopt to evaluate the next 2 phase and the results are shown in Table 2 and Table 3.

Table 2. Pre-satisfaction and Post-satisfaction with renewed models based on 500 samples in the 60 categories

	Model 1	Model 2	Model 3	Model 4	Model 5
Pre-satisfaction (%)	90.2%	89.5%	90.6%	89.7%	90.5%
Post-satisfaction (%)	98.3%	98.0%	97.8%	98.5%	98.9%

Table 3. Pre-satisfaction and Post-satisfaction with renewed models based on 200 samples in the rest 10 categories

	Model 1	Model 2	Model 3	Model 4	Model 5
Pre-satisfaction (%)	80.6%	85.3%	82.5%	87.2%	81.9%
Post-satisfaction (%)	97.9%	98.3%	98.8%	98.1%	99.2%

The experimental results show that the performance of the classification system can be improved according to user preference by increasing the classification accuracy of product samples that have more similarity with the samples in the sample set. Thus, the sample set can be used to represent the user preference.

5 Conclusions

In this paper, we consider the multi-attribute classification problem as a kind of multi-objective optimization problem and the relations between them are discussed. Traditional weighted sum method for multi-objective optimization is adopt for general studying of building a dynamical system model for practical engineering applications. According to the personalization and dynamical character of practical problems, we divided the process of building classification model into three phases. EA has been introduced into the whole process to ensure the dynamical and personalized improvement of the model based on user feedback. The preference of user plays an important role to renew the model dynamically. EA, as a kind of stochastic algorithm, can effectively fit the personalization and dynamical character based on the change of the training sample set. The users can unconsciously express their preference by inquiring the model for sample classification. The system shows powerful in satisfying the variation of user preference.

Acknowledgments. The authors are very grateful for the financial support by the Fundamental Research Funds for the Central Universities.

References

1. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester (2001)
2. Marler, R.T., Arora, J.S.: Survey of Multi-Objective Optimization Methods for Engineering. Struct. Multidisc Optim. 26, 369–395 (2004)
3. Amanifard, N., Nariman-Zadeh, N., Borji, M., Khalkhali, A., Habibdoust, A.: Modelling and Pareto Optimization of Heat Transfer and Flow Coefficients in Microchannels using GMDH type neural networks and genetic algorithms. Energy Conversion and Management 49(2), 311–325 (2008)
4. Amodeo, L., Chen, H., Hadji, A.E.: Multi-objective Supply Chain Optimization: An Industrial Case Study. In: Giacobini, M. (ed.) EvoWorkshops 2007. LNCS, vol. 4448, pp. 732–741. Springer, Heidelberg (2007)
5. Aranha, C., Iba, H.: Modelling Cost into a Genetic Algorithm-Based Portfolio Optimization System by Seeding an Objective Sharing. In: 2007 IEEE Congress on Evolutionary Computation, pp. 196–203. IEEE Press, Singapore (2007)
6. Askar, S.S., Tiwari, A.: Finding Exact Solutions for Multi-Objective Optimisation Problems using a Symbolic Algorithm. In: 2009 IEEE Congress on Evolutionary Computation, pp. 24–30. IEEE Press, Trondheim (2009)
7. Avigad, G., Moshaiov, A., Brauner, N.: MOEA-Based Approach to Delayed Decisions for Robust Conceptual Design. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 584–589. Springer, Heidelberg (2005)
8. Bader, J., Brockhoff, D., Welten, S., Zitzler, E.: On Using Populations of Sets in Multiobjective Optimization. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 140–154. Springer, Heidelberg (2009)

Faster Convergence and Higher Hypervolume for Multi-objective Evolutionary Algorithms by Orthogonal and Uniform Design

Siwei Jiang and Zhihua Cai*

School of Computer Science, China University of Geosciences
Wuhan, P.R.China 430074
amosonic@gmail.com, zhcai@cug.edu.cn

Abstract. Multi-Objective Evolutionary Algorithms (MOEAs) are powerful and efficient tools to deal with multi-objective problems. In the framework of MOEAs, initialization is an important for the decision of the amount of space filling design information in first population. To fasten the convergence and heighten the hypervolume for MOEAs, in this paper, we adopt experimental methods to generate the first population including Orthogonal Design and Uniform Design. Compared with the traditional Random Design, the experimental methods can get well scattered solutions in feasible searching space and provide guiding information for the next offspring. In the experiment on bio-objective and tri-objective problems by jMetal 2.0, we tested four state-of-art algorithms: NSGA-II, SPEA2, GDE3 and 2-MOEA. The results show that the orthogonal and uniform design outperforms the random design as it can significantly quicken the convergence and enhance the hypervolume. In addition, MOEAs with statistical initialization can obtain higher quality Pareto-optimal solutions in fewer numbers of fitness function evolutions.

Keywords: Multi-Objective Optimization, Random Design, Orthogonal Design, Uniform Design, Convergence, Hypervolume, jMetal.

1 Introduction

In real world, there are many problems include several criteria or objectives, and these objectives always stay in conflict with each other, it is impossible to simultaneously optimize all the objectives at the same time. In past few years, Multi-Objective Evolutionary Algorithms(MOEAs) are demonstrated as useful and powerful tools to deal with complexity Multi-objective Problems (MOPs)[1]. The main advantage of MOEAs is that it provides an amount of non-dominated solutions in a single run, and such solutions are lying or near to the Pareto-optimal front.

The main challenge for MOEAs is to be satisfied with three goals at the same time: (1) the Pareto-optimal solutions are as near to true Pareto-front,

* The Project was supported by the Research Foundation for Outstanding Young Teachers, China University of Geosciences(Wuhan)(No:CUGQNL0911).

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 312–328, 2010.

which means the convergence of MOEAs, (2) the non-dominated solutions are evenly scattered along the Pareto-front, which means the diversity of MOEAs, (3)MOEAs obtain Pareto-optimal solutions in limit evolution times.

The frame of MOEAs is constituted by three important issues: population initialization, regeneration method and acceptance rule. Many work of MOEAs focuses on regeneration method and accept rule. Unfortunately, population initialization has long been ignored, but a good quality first population will fast the convergence and provide good final solutions. Fang suggests that statistical methods seek design points that are well scattered on the experimental domain, and orthogonal design, uniform design can provide more information in limit experiment times[7]. Recently, some works have been proposed which combine the statistical methods with Evolution Algorithms(EAs)[8,9,10,11].

Leung uses orthogonal designs and quantization to enhance genetic algorithm, and it designs a new orthogonal crossover operator to generate the next offspring, the new algorithm is more robust to solve single global numerical optimization with continuous variables[8].

To solve the Multi-Objective Problems(MOPs), Zeng employs orthogonal method and statistical optimal method to fast the convergence and yield evenly distributed solutions with high precision[9]. Cai proposes the ϵ-ODEMO, which combines the orthogonal, differential evolution and ϵ-dominate, and it is very efficient in terms of convergence, diversity, and computational time[10]. Leung utilizes the uniform design to initial population and designs a new crossover operator, which can find the Pareto-optimal solutions scattered uniformly[11].

Interesting in this topic, we adopt statistical methods to generate the first population include *Orthogonal Design* and *Uniform Design*. Experiment on bio-objective and tri-objective problems by jMetal[14], results show that orthogonal and uniform design can significantly faster the convergence and higher the Hypervolume than random design. In addition, MOEAs with statistical initialization can obtain high quality Pareto-optimal solutions in less numbers of fitness function evolutions.

The paper is organized as follows. In section 2, we briefly describe the orthogonal design and uniform design. In section 3, we scale the four state-of-art algorithms NSGA-II, SPEA2, GDE3 and ϵ-MOEA to four groups, each group has same algorithms only different in population initialization with random, orthogonal and uniform design. In section 4, Experiment on bio-objective and tri-objective by jMetal 2.0, the results show that orthogonal and uniform design successful enhance the performance of MOEAs. In section 5, we make conclusions and discuss the future research on MOEAs.

2 Experimental Design Method

Orthogonal design and uniform design are experimental design method, which belongs to a sophisticated branch of statistics. Both of them can get better scattered solutions in feasible searching space than random design.

2.1 Orthogonal Design

In this section, we briefly describe the *Orthogonal Design*. Define the orthogonal array as $O_R(Q^C)$, where Q is the level and it's odd, R, C represent the row and column of orthogonal array, they must be satisfied with formulation 1 as follows:

$$\begin{cases} R = Q^J \geq NP \\ C = \dfrac{Q^J - 1}{Q - 1} \geq n \end{cases} \tag{1}$$

where J is a positive integer, NP is the number of population size, n is the number of variables. When select a proper parameters of Q, J, the orthogonal array can be created by Algorithm 1, the detail can get from references[7,8,9,10].

Algorithm 1. Construction of Orthogonal Array

```
/* Construct the basic columns */
for $k = 1$ to $J$ do
    $j = \frac{Q^{k-1}-1}{Q-1} + 1$
    for $i = 1$ to $R$ do
        $a_{i,j} = \lfloor \frac{i-1}{Q^{J-k}} \rfloor \bmod Q$
    end for
end for
/* Construct the nonbasic columns */
for $k = 2$ to $J$ do
    $j = \frac{Q^{k-1}-1}{Q-1} + 1$
    for $s = 1$ to $j - 1$ do
        for $t = 1$ to $Q - 1$ do
            for $i = 1$ to $R$ do
                $a_{i,(j+(s-1)(Q-1)+t)} = (a_{i,s} \times t + a_{i,j}) \bmod Q$
            end for
        end for
    end for
end for
Increment $a_{i,j}$ by one for all $i \in [1, R]$ and $j \in [1, C]$
```

The orthogonal array has a special property: If some columns are taken away from an orthogonal array, the resulting array is still an orthogonal array with a smaller number of factors. When $C > n$, we just need to cut off the $C - n$ column, when $R > NP$, we choose the best NP solutions, the remain population retains the orthogonal property.

2.2 Uniform Design

In this section, we briefly describe the *Uniform Design*. Define the uniform array as $U_R(C)$, where Q is the level and it's primer, R, C represent the row and column of uniform array, they must be satisfied with formulation 2 as follows:

$$\begin{cases} R = Q > n \\ C = n \end{cases} \tag{2}$$

where n is the number of variables. When select a proper parameters of Q, σ form table 1, uniform array can be created by Equation 3

$$U_{i,j} = (i * \sigma^{j-1} \bmod Q) + 1 \tag{3}$$

For example, we can construct a uniform array with five factors and seven levels, the parameter σ is choose to $\sigma = 3$ from table 1, then the uniform array $U_7(5)$ is constructed by Equation 3 as follows:

$$U_7(5) = \begin{bmatrix} 2 & 4 & 3 & 7 & 5 \\ 3 & 7 & 5 & 6 & 2 \\ 4 & 3 & 7 & 5 & 6 \\ 5 & 6 & 2 & 4 & 3 \\ 6 & 2 & 4 & 3 & 7 \\ 7 & 5 & 6 & 2 & 4 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{4}$$

Both orthogonal design and uniform design can get well distributed initialization population, the main difference of them is that uniform design needs less experiment times than orthogonal design. For example, if the optimization problem has 30 variables, when the two design methods have the same level $Q = 31$, J is a integer, we can set $J = 2$, the orthogonal array is $O(961 * 32)$, so the orthogonal population choose the best 100 solutions from the 961 rows of orthogonal array; but the uniform array is only $U(31 * 30)$, then uniform population add the $100 - 31 = 69$ solutions by random design.

Table 1. Values of the parameter σ for different number of variables and levels

number of levels of per variable	number of variables	σ
5	2-4	2
7	2-6	3
11	2-10	7
13	2	5
	3	4
	4-12	6
17	2-16	10
19	2-3	8
	4-18	14
23	2,13-14,20-22	7
	8-12	15
	3-7,15-19	17
29	2	12
	3	9
	4-7	16
	8-12,16-24	8
	13-15	14
	25-28	18
31	2,5-12,20-30	12
	3-4,13-19	22

3 NSGA-II, SPEA2, GDE3 and ϵ-MOEA with Random/Orthogonal/Uniform Design

The frame of MOEAs is constituted by three important components: (1) population initialization, (2) regeneration method, (3) acceptance rule. The non-dominated solutions are equal good in the population, so the number of them is increasing quickly in the running process of MOEAs. Acceptance rule is designed to decide which solution should be retained or cut off when the no-dominated solutions are full to the maximum population size.

In long term, many researchers focus on regeneration methods and accept rules, while initialization population has long been ignored, but it's also an important part of MOEAs.

Experimental design methods are space filling designs and widely used in mathematics and industrial experiments, it mainly includes orthogonal design and uniform design. The main advantage of experimental design methods is that it only needs limit experimental times to obtain space filling points, and such points are scattered in design space, and these points include large amount of experimental design information. In this paper we use experimental design methods to initialize the first population in MOEAs, it has three benefits: (1) the points in first population are evenly scattered in feasible searching space, (2) the first population is full of design information, it guides the search direction and enhance the searching ability to generate the next offspring, (3) it needs small number of evolution times to get population with high hypervolume.

Four state-of-art algorithms have been tested in our experiment include NSGA-II, SPEA2, GDE3 and ϵ-MOEA[2,3,4,5,6].

1. NSGA-II adopts a fast non-dominated sorting approach to reduce computer burden, NSGA-II design an acceptance rule as combining the *Ranking* and *Crowding Distance* to choose the candidate solutions[2].
2. SPEA2 proposals a fitness assignment which is the sum of its strength raw fitness and *Neighbor Density Estimation*, SPEA2 uses cluster technique to evaluate fitness by calculating the distances based on the k-th nearest neighbor[3].
3. GDE3 is a developed version of Differential Evolution, which is suited for global optimization with an arbitrary number of objectives and constraints. Similar to NSGA-II, GDE3 uses *Crowding Distance* to accept candidate solutions[4].
4. ϵ-MOEA designs a box dominance concept to combine the convergence and diversity, it can get desired quality Pareto-optimal solutions by ϵ parameter setting, ϵ-MOEA divides the searching space into several equal size of hyper boxes, and only one solution can survive in one hyper box[5,6].

We choose the four state-of-art algorithms in our experiment, because they are gained popularity in MOEAs, these algorithms have different regeneration methods and acceptance rules. In this paper, each algorithm is scaled to three algorithms with different initialization methods: *Random/Orthogonal/Uniform Design*, while the three algorithms in same group has the same regeneration method and acceptance rule.

4 Experiment Results

The levels for orthogonal and uniform design is setting to $Q = 31$. For orthogonal initialization, it needs to cut off some columns and rows to get the the first 100 solutions; for uniform initialization, it needs to full the first 100 solutions, so the other $100 - Q = 69$ solutions are created by random design.

Experiment is based on jMetal 2.0[14], which is a Java-based framework aimed at facilitating the development of metaheuristics for solving MOPs, it provides large block reusing code and fair comparison for different MOEAs. The parameter setting for algorithms is adopted the default setting in MOEAs java platform jMetal 2.0[14]. The ϵ setting about ϵ-MOEA is described in table 2.

The test problems are five bio-objective problems: ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 [12]and one tri-objective problem DTLZ6[13]. Each algorithm independent runs for 50 times and maximum evolution times is $25,000$.

Table 2. The ϵ setting for testing problems in ϵ-MOEA

problems	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ6
ϵ	$[0.0075, 0.0075]$	$[0.0075, 0.0075]$	$[0.0025, 0.0035]$	$[0.0075, 0.0075]$	$[0.0075, 0.0075]$	$[0.05, 0.05, 0.05]$

The performance metrics have four categories:

Hypervolume. This quality indicator calculates the volume (in the objective space) covered by members of a non-dominated set of solutions Q (the region enclosed into the discontinuous line in the figure 1 below), for problems where all objectives are to be minimized. Mathematically, for each solution , a hypercube v_i is constructed with a reference point W and the solution i as the diagonal corners of the hypercube. The reference point can simply be found by constructing a vector of worst objective function values. Thereafter, a union of all hypercubes is found and its hypervolume (HV) is calculated:

$$HV = volume(\bigcup_{i=1}^{|Q|} v_i) \tag{5}$$

Generational Distance. The metric is to measure how far the elements are in the set of non-dominated vectors found from those in the true Pareto-optimal set. It is defined as:

$$GD = \sqrt{\frac{\sum_{i=1}^{|n|} d_i^2}{n}} \tag{6}$$

Spread. The Spread indicator is a diversity metric that measures the extent of spread achieved among the obtained solutions. This metric is defined as:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{n-1} |d_i - \overline{d}|}{d_f + d_l + (n-1)\overline{d}} \tag{7}$$

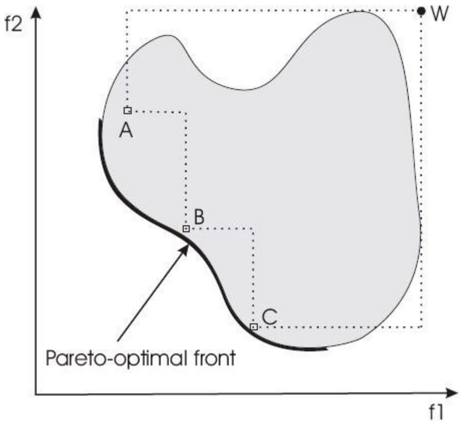


Fig. 1. The hypervolume of non-dominated solutions A, B, C for worst reference point W is dashline region $ABCW$. The closer to Pareto-optimal front (convergence) and more evenly spread along Pareto-optimal front (diversity), the hypervolme is higher.

Evolution Counts. This metric is to measure how many numbers of fitness function evolutions(*NFFEs*) is needed, when the Hypervolume of non-dominated set is larger than the 98% of Hypervolume of the true Pareto-front set(Success Rate means the percentage of getting $HV_{paretoSet} \geq 0.98 * HV_{truePF}$ in 50 independent run, NA means cannot get 100% success).

$$Evolution\ Counts = \{NFFEs | HV_{paretoSet} \geq 0.98 * HV_{truePF}\} \qquad (8)$$

The higher Hypervolume, and the lower Genetic Distance, Spread, Evolution Counts mean the algorithm is better. The results are compared by mean and standard variance for 50 independent runs, the better result is bold font and basic compared object is random design for first population.

Table 3 is the results of NSGA-II group with different initialization methods. Comparison with NSGA-II, in term of Evolution Counts, NSGA-II^{OD} needs less *NFFEs* to get higher hypervolume for all problems except ZDT3, NSGA-II^{OD} only has 42% Success Rate in ZDT3, which means that the number of being satisfied with $HV_{paretoSet} \geq 0.98 * HV_{truePF}$ is 21 times in 50 independent runs; In term of Hypervolume, NSGA-II^{OD} is better in ZDT4, ZDT6, DTLZ6; In term of Genetic Distance, NSGA-II^{OD} is better for all problems; In term of Spread, NSGA-II^{OD} is only better for DTLZ6.

Comparison with NSGA-II, in term of Evolution Counts, Hypervolume and Genetic Distance, NSGA-II^{UD} is better for all problems; In term of Spread, NSGA-II^{UD} is only better only for ZDT3 and DTLZ6.

From the number of items for the six problems in four performance metrics, orthogonal design gets better results(bold font) for 15 items than NSGA-II, uniform design is better for 20 items than NSGA-II.

Table 3. The mean and variance of NSGA-II with random/orthogonal/uniform design

problems	NSGA-II		NSGA-II^{OD}		NSGA-II^{UD}	
	mean	std	mean	std	mean	std
Evaluation Counts: $HV_{paretoSet} \geq 0.98 * HV_{truePF}$ or **Success Rate**						
ZDT1	14216.0	630.99	**9445.0**	1807.1	**2682.0**	675.5
ZDT2	NA(78%)	NA(78%)	**10883.0**	1749.4	**6314.0**	2270.4
ZDT3	12670.0	595.40	NA(42%)	NA(42%)	**3944.0**	2590.1
ZDT4	NA(82%)	NA(82%)	**11355.0**	2284.9	**NA(92%)**	NA(92%)
ZDT6	NA(0%)	NA(0%)	**5985.0**	870.5	**5064.0**	971.1
DTLZ6	NA(0%)	NA(0%)	**6539.0**	914.0	**6170.0**	1011.4
Hypervolume						
ZDT1	0.65940	3.2565E-4	0.65899	3.1174E-4	**0.66020**	2.6842E-4
ZDT2	0.32611	2.5101E-4	0.32590	3.2438E-4	**0.32709**	2.3729E-4
ZDT3	0.51478	1.4460E-4	0.47582	4.1312E-2	**0.51541**	9.9030E-5
ZDT4	0.65530	2.8482E-3	**0.65884**	3.5600E-4	**0.65581**	3.7967E-3
ZDT6	0.38864	1.1875E-3	**0.39742**	3.7391E-4	**0.39858**	3.8104E-4
DTLZ6	0.00000	0.00000	**0.09363**	2.7595E-4	**0.09395**	1.6982E-4
Generation Distance						
ZDT1	2.2119E-4	3.3383E-5	**1.7865E-4**	3.8761E-5	**1.8699E-4**	3.4291E-5
ZDT2	1.7890E-4	5.2907E-5	**1.3389E-4**	3.8521E-5	**1.3725E-4**	3.1703E-5
ZDT3	2.1267E-4	1.4648E-5	**1.6501E-4**	2.1531E-5	**2.0650E-4**	1.0524E-5
ZDT4	4.4133E-4	1.9450E-4	**1.0575E-4**	3.5305E-4	**4.0039E-4**	2.2048E-4
ZDT6	1.0110E-3	6.9295E-5	**5.4113E-4**	4.8461E-5	**5.9068E-4**	3.1724E-4
DTLZ6	0.17157	0.019483	**6.5825E-3**	9.3785E-3	**5.6462E-4**	3.6673E-5
Spread						
ZDT1	**0.36734**	0.03332	0.57069	0.03545	0.37757	0.02870
ZDT2	**0.37335**	0.02975	0.57350	0.03619	0.39624	0.02788
ZDT3	0.74850	0.01395	0.81906	0.03084	**0.74612**	0.01760
ZDT4	**0.39476**	0.03646	0.69061	0.04457	0.40959	0.06509
ZDT6	**0.36745**	0.03078	0.79051	0.03314	0.65127	0.04357
DTLZ6	0.81599	0.05941	**0.76888**	0.15296	**0.56057**	0.04107

Figure 2 is the hypervolume varies in different *NFFEs* for ZDT1-6,DTLZ6 problems by NSGA-II with Random/Orthogonal/Uniform Design, in the plot we only give a random test result for the problems. The black horizon line is 98% hypervolume of true Pareto-front, orthogonal and uniform design reach the black line more quickly than random design. In addition, uniform design needs less experiment times than orthogonal design for all problems except ZDT4.

Table 4 is the results of SPEA2 group with different initialization methods. Comparison with SPEA2, in term of Evolution Counts, SPEA2^{OD} needs less *NFFEs* to get higher hypervolume for ZDT1, ZDT2, ZDT4 and DTLZ6, SPEA2^{OD} get higher Success Rate for ZDT4, Success Rate means that the number of SPEA2^{OD} gets $HV_{paretoSet} \geq 0.98 * HV_{truePF}$ is 48 times in 50 runs, while SPEA2 is 30 times in 50 runs; In term of Hypervolume, SPEA2^{OD} is better for all the problems except ZDT3; In term of Generation Distance, SPEA2^{OD} is better on all problems; In term of Spread, SPEA2^{OD} is better for ZDT2, ZDT4, ZDT6 and DTLZ6.

Comparison with SPEA2, in term of Evolution Counts, SPEA2^{UD} is better for all problems except ZDT3, and it has the same Success Rate with SPEA2 for ZDT4; In term of Hypervolume, Genetic Distance and Spread metric, SPEA2^{UD} is better for all problems.

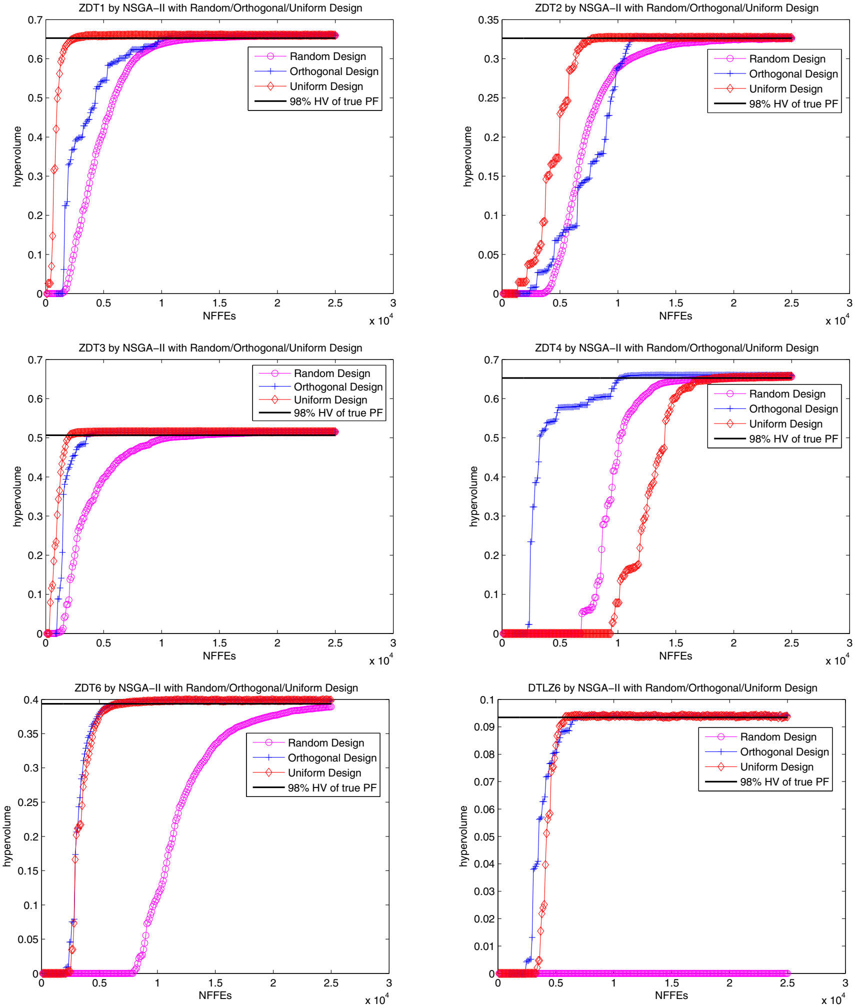


Fig. 2. The hypervolume varies in different *NFFEs* for ZDT1-6,DTLZ6 problems by NSGA-II with Random/Orthogonal/Uniform Design

Table 4. The mean and variance of SPEA2 with random/orthogonal/uniform design

problems	SPEA2		SPEA2^{OD}		SPEA2^{UD}	
	mean	std	mean	std	mean	std
Evaluation Counts: $HV_{paretoSet} \geq 0.98 * HV_{truePF}$ or **Success Rate**						
ZDT1	16042.0	664.6	**14151.0**	3311.4	**3848.0**	1121.6
ZDT2	NA(56%)	NA(56%)	**16883.0**	2929.2	**12170.0**	5011.9
ZDT3	**15384. 0**	773.4	NA(36%)	NA(36%)	NA(98%)	NA(98%)
ZDT4	NA(60%)	NA(60%)	**NA(96%)**	NA(96%)	NA(60%)	NA(60%)
ZDT6	NA(0%)	NA(0%)	**6081.0**	1487.1	**5110.0**	1449.7
DTLZ6	NA(0%)	NA(0%)	**10971.0**	2517.4	**9956.0**	1578.9
Hypervolume						
ZDT1	0.65995	2.5898E-4	**0.66159**	5.0639E-4	**0.66167**	9.4033E-5
ZDT2	0.32633	5.9330E-4	**0.32846**	8.3786E-5	**0.32846**	8.6549E-5
ZDT3	0.51404	4.9932E-4	0.45907	5.4958E-2	**0.51445**	7.2847E-3
ZDT4	0.65126	7.8660E-3	**0.66045**	5.3945E-3	**0.65161**	7.8519E-3
ZDT6	0.37784	2.9404E-3	**0.40139**	2.8387E-5	**0.40137**	1.5328E-4
DTLZ6	0.00000	0.00000	**0.09462**	6.4983E-5	**0.09464**	4.4337E-5
Generation Distance						
ZDT1	2.3104E-4	5.9382E-5	**1.6971E-4**	2.1395E-5	**1.7175E-4**	1.8347E-4
ZDT2	1.8923E-4	5.2818E-5	**7.2927E-5**	2.0327E-5	**7.3366E-5**	1.8107E-5
ZDT3	2.3165E-4	1.6998E-5	**1.7334E-4**	3.3275E-5	**2.0989E-4**	1.6797E-5
ZDT4	6.3027E-4	4.5520E-4	**1.5879E-4**	1.4632E-5	**5.8481E-4**	3.9782E-4
ZDT6	1.8019E-3	2.2101E-4	**5.3152E-4**	1.7950E-5	**5.3390E-4**	1.8563E-5
DTLZ6	0.16277	1.2668E-2	**2.6211E-3**	4.6225E-3	**5.8135E-4**	2.2234E-5
Spread						
ZDT1	0.14906	0.01580	0.15101	0.02046	**0.14586**	0.01355
ZDT2	0.15561	0.02405	**0.14442**	0.01685	**0.14119**	0.01422
ZDT3	0.71095	0.00579	0.75267	0.05990	**0.70678**	0.00565
ZDT4	0.25487	0.09969	**0.16305**	0.05627	**0.24867**	0.08661
ZDT6	0.23106	0.01875	**0.13845**	0.01161	**0.13909**	0.01267
DTLZ6	0.58590	0.02901	**0.30771**	0.23416	**0.17234**	0.01408

From the number of items for the six problems in four metrics, orthogonal design can get better results(bold font) in 20 items than SPEA2, uniform design better in 22 items than SPEA2.

Figure 3 is the hypervolume varies in different *NFFEs* for ZDT1-6,DTLZ6 problems by SPEA2 with Random/Orthogonal/Uniform Design, in the plot we only give a random test result for the problems. The first population initialized by experimental design methods get the higher hypervolume more quickly than random design. In addition, uniform design reach the black horizon line faster than orthogonal design for all problems except ZDT4.

Table 5 is the results of GDE3 group with different initialization methods. Comparison with GDE3, in term of Evolution Counts, GDE3^{OD} needs less *NFFEs* to get higher hypervolume for ZDT1, ZDT4 and ZDT6, but GDE3^{OD} need more evolution times in ZDT2 and DTLZ6, GDE3^{OD} has zero Success Rate in ZDT3; In term of Hypervolume, GDE3^{OD} is only better for ZDT1, ZDT6; In term of Genetic Distance, GDE3^{OD} is only better for ZDT2,ZDT3; In term of Spread, GDE3^{OD} is better for ZDT1, ZDT6, DTLZ6.

Comparison with GDE3, in term of Evolution Counts, GDE3^{UD} is better for all test problems; In term of Hypervolume, GDE3^{UD} is better for all except ZDT6; In term of Genetic Distance, GDE3^{UD} is only better for ZDT2,ZDT4 and ZDT6; In term of Spread, GDE3^{UD} is better for ZDT1, ZDT2, ZDT3, ZDT4.

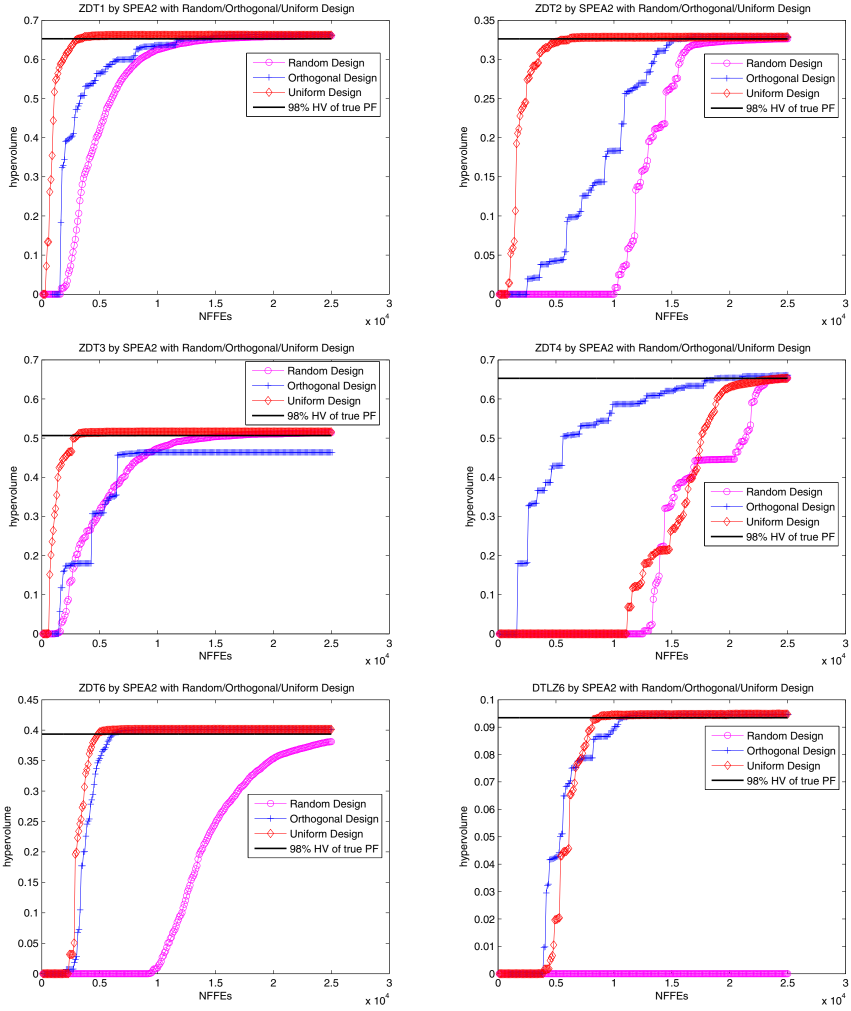


Fig. 3. The hypervolume varies in different *NFFEs* for ZDT1-6,DTLZ6 problems by SPEA2 with Random/Orthogonal/Uniform Design

Table 5. The mean and variance of GDE3 with random/orthogonal/uniform design

problems	GDE3		$GDE3^{OD}$		$GDE3^{UD}$	
	mean	std	mean	std	mean	std
Evaluation Counts: $HV_{paretoSet} \geq 0.98 * HV_{truePF}$ or **Success Rate**						
ZDT1	9730.0	384.8	**9039.0**	1078.2	**5566.0**	692.7
ZDT2	11268.0	407.6	15399.8	3168.1	**7072.0**	1060.6
ZDT3	10400.0	453.4	NA(0%)	NA(0%)	**6256.0**	979.2
ZDT4	16238.0	700.5	**10811.0**	1765.4	**15994.0**	629.1
ZDT6	4750.0	393.6	**3851.0**	306.1	**3952.0**	472.1
DTLZ6	3976.0	261.2	6269.0	615.4	**3098.0**	371.7
Hypervolume						
ZDT1	0.661926	2.2943E-5	**0.661927**	2.5305E-5	**0.661932**	2.0780E-5
ZDT2	0.328631	2.2298E-5	0.326910	1.1891E-2	**0.328634**	2.0277E-5
ZDT3	0.515932	3.0409E-5	0.227329	7.9089E-2	**0.515956**	1.8267E-5
ZDT4	0.661993	2.7272E-5	0.661910	2.0303E-5	**0.662007**	2.1120E-5
ZDT6	0.401340	1.9004E-5	**0.401341**	1.6326E-5	0.401336	2.2257E-5
DTLZ6	0.094896	2.7697E-5	0.094893	2.3137E-5	**0.094908**	2.5947E-5
Generation Distance						
ZDT1	**9.4072E-5**	3.1386E-5	9.9534E-5	3.6550E-5	9.9956E-5	3.3886E-5
ZDT2	4.6173E-5	2.7640E-6	**4.5587E-5**	3.0930E-6	**4.5523E-5**	2.4570E-6
ZDT3	1.7422E-4	9.5430E-6	**1.3660E-4**	9.2070E-6	1.7645E-4	1.1651E-5
ZDT4	1.0210E-4	3.9050E-5	1.1706E-4	3.5137E-5	**8.8560E-5**	3.1906E-5
ZDT6	5.3160E-4	1.7476E-5	5.3414E-4	1.5216E-5	**5.2523E-4**	1.5188E-5
DTLZ6	**5.7138E-4**	2.2753E-5	5.7885E-4	2.6026E-5	5.7275E-4	2.6387E-5
Spread						
ZDT1	0.14852	0.01319	**0.14234**	0.01129	**0.14246**	0.01241
ZDT2	0.13766	0.01104	0.15492	0.09266	**0.13656**	0.00867
ZDT3	0.71019	0.00466	0.84599	0.05039	**0.70828**	0.00379
ZDT4	0.14154	0.01240	0.14537	0.00923	**0.13917**	0.01423
ZDT6	0.12907	0.01066	**0.12816**	0.01005	0.13025	0.01031
DTLZ6	0.16188	0.01104	**0.16157**	0.01463	0.16361	0.01125

From the number of items for the six problems in four metrics, orthogonal design can get better results(bold font) in 10 items than GDE3, uniform design better in 18 items than GDE3.

Figure 4 is the hypervolume varies in *NFFEs* by GDE3 with different initialization. Uniform design need less *NFFEs* to get higher hypervolume than random desin for all test problems, orthogonal design need less evolution time than random design for ZDT1, ZDT4, ZDT6.

Table 6 is the results of ϵ-MOEA^{UD} group with different initialization methods. Comparison with ϵ-MOEA, in term of Evolution Counts, ϵ-MOEA^{OD} needs less *NEEFs* to get higher hypervolume for all except ZDT3; In term of Hypervolume, ϵ-MOEA^{OD} is better for all except ZDT3; In term of Genetic Distance, ϵ-MOEA^{OD} is better for all except ZDT3; In term of Spread, ϵ-MOEA^{OD} is better for all except ZDT1.

Comparison with ϵ-MOEA, in term of Evolution Counts, ϵ-MOEA^{UD} is better for ZDT1,ZDT2 and ZDT6; In term of Hypervolume, ϵ-MOEA^{UD} is better for all except ZDT3, ZDT4; In term of Genetic Distance, ϵ-MOEA^{UD} is better for all; In term of Spread, GDE3^{UD} is better for all except ZDT1, ZDT3.

From the number of items for the six problems in four metrics, orthogonal design can get better results(bold font) in 19 items than ϵ-MOEA, uniform design better in 17 items than ϵ-MOEA.

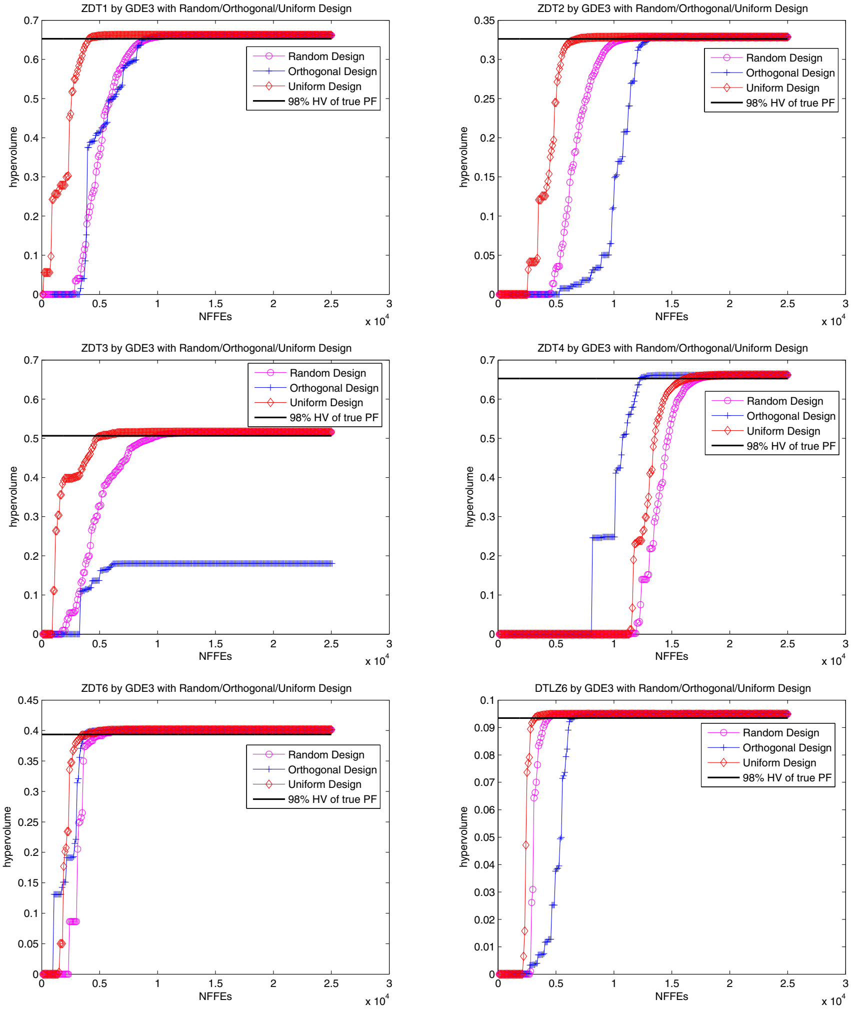


Fig. 4. The hypervolume varies in different *NFFEs* for ZDT1-6,DTLZ6 problems by GDE3 with Random/Orthogonal/Uniform Design

Table 6. The mean and variance of ϵ-MOEA with random/orthogonal/uniform design

problems	ϵ-MOEA		ϵ-MOEA^{OD}		ϵ-MOEA^{UD}	
	mean	std	mean	std	mean	std
Evaluation Counts: $HV_{paretoSet} \geq 0.98 * HV_{truePF}$ or **Success Rate**						
ZDT1	10656.0	1115.4	**4529.0**	1826.8	**3126.0**	1441.5
ZDT2	14844.0	1931.4	**7865.0**	2882.9	**7340.0**	2959.5
ZDT3	NA(98%)	NA(98%)	NA(92%)	NA(92%)	NA(98%)	NA(98%)
ZDT4	NA(94%)	NA(94%)	**12351.0**	4208.7	NA(94%)	NA(94%)
ZDT6	19394.0	959.0	**2807.0**	554.8	**2092.0**	546.2
DTLZ6	NA(0%)	NA(0%)	NA(0%)	NA(0%)	NA(0%)	NA(0%)
Hypervolume						
ZDT1	0.66128	9.6772E-5	**0.66179**	4.9530E-6	**0.66179**	6.2520E-6
ZDT2	0.32832	2.0725E-4	**0.32870**	3.0570E-6	**0.32870**	3.3320E-6
ZDT3	**0.51345**	5.7765E-3	0.51070	1.3349E-2	0.51302	9.2142E-3
ZDT4	0.65762	2.7614E-3	**0.66169**	4.3038E-4	0.65747	3.3522E-3
ZDT6	0.39763	4.7900E-4	**0.40086**	3.5350E-6	**0.40086**	3.5090E-6
DTLZ6	0.00000	0.0000E0	**0.07509**	1.0000E-6	**0.07509**	1.0000E-6
Generation Distance						
ZDT1	6.0696E-5	5.5050E-6	**5.5100E-5**	2.1440E-6	**5.5028E-5**	2.4550E-6
ZDT2	5.5052E-5	6.5200E-6	**4.8109E-5**	1.3660E-5	**4.8041E-5**	1.4280E-6
ZDT3	1.8964E-4	3.3048E-4	1.9793E-4	4.3128E-4	**1.8166E-4**	3.2511E-4
ZDT4	2.7957E-4	1.4941E-4	**5.5487E-5**	2.1490E-6	**2.4489E-4**	1.5379E-4
ZDT6	6.3748E-4	2.2151E-5	**6.2082E-4**	2.4330E-6	**6.2078E-4**	2.7900E-6
DTLZ6	1.1733E-1	1.3611E-2	**1.8195E-3**	4.5930E-6	**1.8194E-3**	3.8038E-6
Spread						
ZDT1	**0.27802**	0.00721	0.28887	0.00240	0.28849	0.00253
ZDT2	0.25922	0.00605	**0.25671**	0.00207	**0.25618**	0.00166
ZDT3	0.82957	0.01540	**0.81962**	0.01390	**0.82039**	0.01364
ZDT4	0.30121	0.03235	**0.29383**	0.01481	0.31446	0.04307
ZDT6	0.19783	0.01406	**0.18125**	0.00187	**0.18157**	0.00200
DTLZ6	0.57290	0.046913	**0.36276**	0.00008	**0.36279**	0.00020

Figure 5 is the hypervolume varies in *NFFEs* by ϵ-MOEA with different initialization. Hypervolume is an important performance metric for MOEAs. The non-dominated solutions founded by MOEAs are closer to the true Pareto-front, the hypervolume is higher; in other hand, the solutions is evenly scattered along the Pareto-optimal front, the hypervolume is larger.

In the plot of figure 5 we only give a random test result for the test problems. The black line is 98% HV of true Pareto-front, uniform design need less *NFFEs* to reach the black line for all test problems except DTLZ6 than random design, orthogonal design get the similar result except it can not reach the black line for ZDT3. Uniform, orthogonal, random design can not reach the black line in DTLZ6, but uniform, orthogonal design has larger hypervolume than random design. In generally, uniform design needs less evolution times to get high hypervolume, the second is orthogonal design, the following is random design.

Orthogonal and uniform design expand the scope of population initialization for MOEAs. Comparison of algorithms group in four categories: Evolution Count, Hypervolume, Generation Distance and Spread, results show that orthogonal and uniform design successfully enhance the performance of MOEAs. Now, we summarize the highlight as follows:

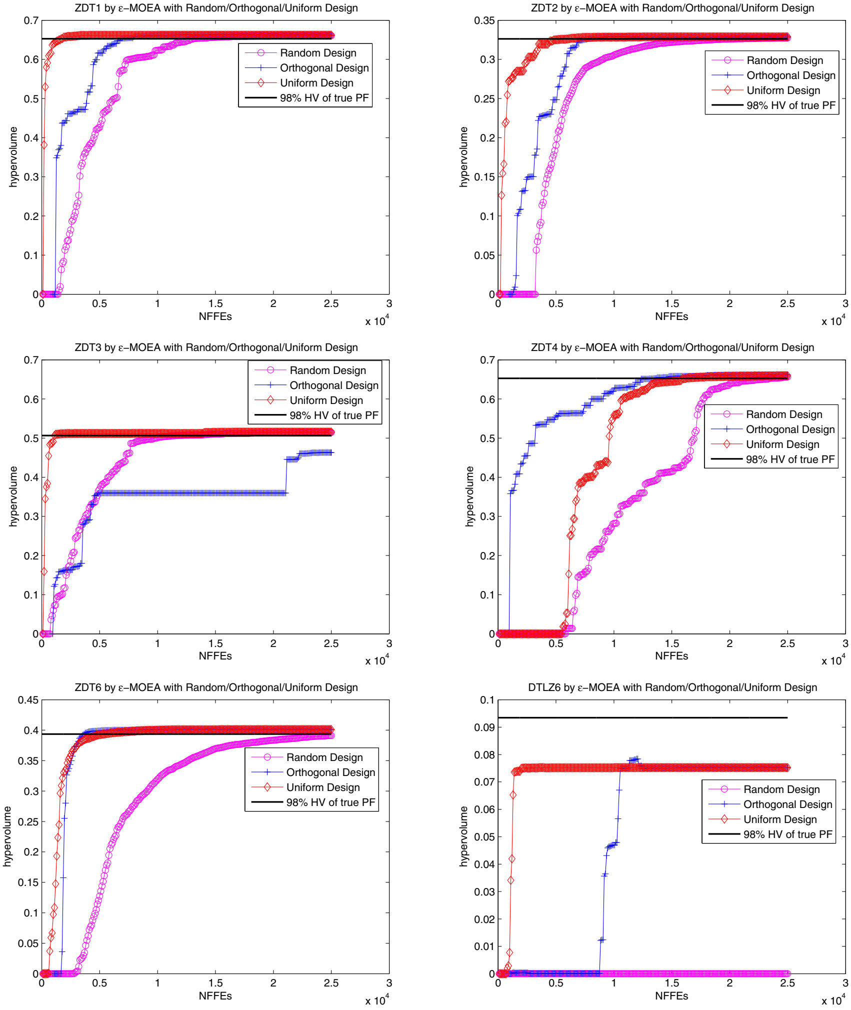


Fig. 5. The hypervolume varies in different *NFFEs* for ZDT1-6,DTLZ6 problems by ϵ-MOEA with Random/Orthogonal/Uniform Design

1. Testing the MOEAs algorithms include NSGA-II, SPEA2, GDE3 and ϵ-MOEA, when first population is created by orthogonal and uniform design, it close to the true Pareto-front faster than random design.
2. Orthogonal design and uniform design methods get higher hypervolume than random design.
3. To get Pareto-optimal solutions with 98% hypervolume of true Pareto-optimal front, the relationship of the number of fitness function evolutions is that *Uniform Design < Orthogonal Design < Random Design*.

5 Conclusion and Further Research

In this paper, we discuss the influence of experimental design methods for Multi-objectives optimization. Four algorithms have been tested include NSGA-II, SPEA2, GDE3 and ϵ-MOEA with *Random/Orthogonal/Uniform Design*. Experiment on bio-objectives and tri-objectives problems by jMetal 2.0, the results show that orthogonal and uniform design can faster the convergence and higher the Hypervolume than random design. In addition, MOEAs with statistical initialization can obtain high quality Pareto-optimal solutions in less numbers of fitness function evolutions.

In the future, more experimental design methods can be introduce into MOEAs; We can use the statistical methods to reproduce the next generation, such as the mixture of statistical methods and common regeneration methods, which may enhance the search ability of MOEAs.

References

1. Coello, C.A.C.: Evolutionary multi-objective optimization: A historical view of the Field. IEEE Computational Intelligence Magazine 1(1), 28–36 (2006)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA - II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
3. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm, Technical Report 103, Computer Engineering and Networks Laboratory (2001)
4. Kukkonen, S., Lampinen, J.: GDE3: The third evolution step of generalized differential evolution. In: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, vol. (1), pp. 443–450 (2005)
5. Deb, K., Mohan, M., Mishra, S.: Evaluating the epsilon-domination based multiobjective evolutionary algorithm for a quick computation of Paretooptimal solutions. Evolutionary Computation 13(4), 501–525 (2005)
6. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multi-objective optimization. Evolutionary Computation 10(3), 263–282 (2002)
7. Fang, K.T., Ma, C.X.: Orthogonal and uniform design. Science Press, Beijing (2001) (in Chinese)

8. Leung, Y.W., Wang, Y.: An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Transactions on Evolutionary Computation 5(1), 41–53 (2001)
9. Zeng, S.Y., Kang, L.S., Ding, L.X.: An orthogonal multiobjective evolutionary algorithm for multi-objective optimization problems with constraints. Evolutionary Computation 12, 77–98 (2004)
10. Cai, Z.H., Gong, W.Y., Huang, Y.Q.: A novel differential evolution algorithm based on epsilon-domination and orthogonal design method for multiobjective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 286–301. Springer, Heidelberg (2007)
11. Leung, Y.-W., Wang, Y.: Multiobjective programming using uniform design and genetic algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part C 30(3), 293 (2000)
12. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8, 173–195 (2000)
13. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multi-Objective Optimization. Zurich, Switzerland, Tech. Rep. 112 (2001)
14. Durillo, J.J., Nebro, A.J., Luna, F., Dorronsoro, B., Alba, E.: jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics, Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, E.T.S.I. Informática, Campus de Teatinos,ITI-2006-10, December (2006), http://jmetal.sourceforge.net

Multi-objective Fuzzy Clustering Method for Image Segmentation Based on Variable-Length Intelligent Optimization Algorithm

Yuankang Fang[1,2], Ziyang Zhen[3], Zhiqiu Huang[1], and Chao Zhang[3]

[1] College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, 210016, Nanjing, P.R. China
[2] Department of Mathematics and Computer Science, Chi Zhou University, 247000, Chizhou, P.R. China
[3] College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, 210016, Nanjing, P.R. China
Fyk80@163.com

Abstract. Fuzzy clustering method for image segmentation usually needs the determination of the cluster number in advance. Therefore, an adaptive fuzzy clustering image segmentation algorithm based on jumping gene genetic algorithm (JGGA) is investigated in this paper. A new weighted multi-objective evaluation function considering the cluster number, the inner-class distance and the inter-class distance is proposed. Because the cluster number is uncertain during the optimization process, a variable-length JGGA (VJGGA) is introduced. The cluster number and the cluster centers of the image gray values are determined by the minimization of evaluation function. Simulation results of the segmentation for a real image indicate that VJGGA algorithm is characterized by strong global capability of searching the optimal segmentation number and cluster centers, compared with variable-length GA (VGA).

Keywords: image segmentation, fuzzy clustering, jumping gene genetic algorithm.

1 Introduction

Image segmentation technique, which is an essential process in image analysis and an important task in computer vision, tries to separate objects from image background. Generally, image segmentation routines are divided into edge detection approaches, region based approaches, histogram based approaches, clustering approaches and combination of some of the presented approaches.

The color and texture features in natural images are complex so that the fully automatic segmentation of the object from the background is very hard. Therefore, semi-automatic segmentation methods incorporating user interactions have been proposed [1, 2] and are becoming more and more popular. Ref. [3] uses a self-organizing map (SOM) and a hybrid genetic algorithm (HGA) to segment four different types of multicomponent images, verification of the results is performed using two different techniques: field verification and functional model. These verification

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 329–337, 2010.

techniques show that the HGA is more accurate than the SOM. Since fully automatic image segmentation is usually very hard for natural images, interactive schemes with a few simple user inputs are good solutions. Ref. [4] presents a new region merging based interactive image segmentation method. A novel maximal-similarity based region merging mechanism is proposed to guide the merging process with the help of markers. Rough sets have been applied in the domain of image segmentation routines in the solutions proposed in [5] designed for MRI imagery segmentation. The contribution of [6] is the extension of the rough entropy framework into distinctive set of adaptive data analysis measures that are useful and potentially advantageous during specialized data analysis. The adaptiveness of rough entropy measures depends upon the possibility of the simultaneous capturing of the crisp, fuzzy and rough nature of the analyzed data. The experimental results have proved high level of correlation between standard and rough entropy based measures giving the potential way into further insight into this matter. Rough entropy measures are capable of simultaneous capturing various data structures. In order to make the evaluation of the rough entropy measures multi-criterions performance relative to standard validity indices more robust, rank based segmentation quality indices have been designed and tested.

Thresholding methods concentrate on the selection of the reliable thresholds and are widely used. Ref. [7-10] propose several fuzzy entropy thresholding methods based on ant colony optimization and grey relational analysis. Clustering techniques are high quality procedures when applied in the domain of image segmentation and can be roughly divided into two categories of hierarchical clustering and partitioning clustering [11]. Ref. [12] presents an adaptive image segmentation algorithm based on variable-length particle swarm optimization optimized fuzzy clustering method.

Evolutionary multi-objective optimization algorithms have attracted significant attention from researchers in various fields, due to their effectiveness and robustness in searching a set of trade-off solutions. In addition, their real-world applications have become increasingly popular in the last few years. Jumping gene genetic algorithm (JGGA) incorporates the concept of jumping gene phenomenon discovered by Nobel Laureate McClintock in her work on the corn plants [13-16]. The main feature of JGGA is that it consists of a simple operation in which a transposition of gene(s) is induced within the same or another chromosome under the GA framework. JGGA is an adaptation of NSGAII [17] – an elitist multiobjective genetic algorithm. NSGAII uses the concept of elitism, which unfortunately reduces the diversity of the population to some extent. In the 1940s, McClintock stated in her Nobel-prize winning work that DNAs, called jumping genes or transposons, could jump in and out of chromosomes.

A real-coded JGGA is applied to solve the optimization problem of an evaluation function to obtain the cluster number and cluster centers for image segmentation. The paper is organized as follows. In section 2 theoretical notions and computation of fuzzy clustering method for image segmentation are presented. The detailed description of JGGA is described in section 3, and the steps for solving the optimization problem are given in section 4. Simulation results and conclusions are exhibited in section 5 and section 6, respectively.

2 Object Function Definition for Fuzzy Clustering Image Segmentation

Let $X=\{x_1, x_2, \ldots, x_n\}$ be a data set, let n be the data number, and let c be a positive integer greater than one. A partition of X into c clusters is represented by mutually disjoint sets $X_1, X_2, \ldots, X_c$. Let $\{u_1,\ldots u_c\}$ be a fuzzy c-partition of X which takes values in the interval [0,1] such that $\sum_{i=1}^{c} u_i(x)=1$ for all x in X. Thus, the membership degree of the x_i belonging to the k-th class is calculated by

$$u_{ik}=\frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}}, \tag{1}$$

where $u=\{u_1,\ldots u_c\}$is a fuzzy c-partition with $u_{ik}=u_i(x_k)$, the weighted exponent m is a fixed number greater than one establishing the degree of fuzziness, and $d_{ik}{}^2=\left\|x_i-v_k\right\|^2$ represents the Euclidean distance or its generalization such as the Mahalanobis distance. Here, $v=\{v_1, \ldots, v_c\}$ is the c cluster centers vector.

Hence, the fuzzy weighted error square sum function is given by

$$E=\sum_{k=1}^{c}\sum_{i=1}^{n}u_{ik}\left\|x_i-v_k\right\|, \tag{2}$$

In fact, E reflects the tightness of the inner-class data.

The inter-class distance function is given by

$$D=\max_{j,k=1}^{c}\left\|v_j-v_k\right\|. \tag{3}$$

The evaluation function of the fuzzy clustering image segmentation is defined as

$$f=\ln E+k_D\frac{1}{D}+k_c c\ . \tag{4}$$

where k_D and k_c are the constants, “ln” makes the inner-class distance smaller.

The evaluation function is composed of three components, including the cluster number, inner-class distance and inter-class distance. Smaller the distance between the gray value of inner-class pixels and the cluster center is, and father the distance between the centers are, more accurate the cluster centers are, and the smaller the evaluation function will be. Therefore, the determination problem of the cluster centers can be transformed into the minimum problem of the evaluation function, which is equal to the optimization problem that can be solved by the intelligent algorithms.

The selection of the cluster number and the cluster centers is equal to the parameters optimization problem of the evaluation function, by minimizing the function (4) using the intelligent algorithms.

3 Jumping Gene Genetic Algorithm

The JGGA was originally proposed to solve the multiobjective optimization problem. JGGA emulates a jumping gene discovered from a geneticist's work on corn plant that the corn chromosome had two parts dissociated from each other and the breaking place is always the same from generation to generation. It was later explained that the dissociation elements were found to transpose from one position to another within the same chromosome or to another chromosome under the presence of activator elements.

The main feature of JGGA is that it comprises an additional operation called transposition which is induced within the same chromosome or to another chromosome. The only difference between JGGA and GA is the addition of jumping-gene transposition. There are two jumping ways for the transposons, including the cut-and-paste and the copy-and-paste. The former one means a piece of DNA is cut and pasted somewhere else, while the latter one means that the genes remain at the same location whereas the message in the DNA is copied into RNA and then copied back into DNA at another place in the genome.

For the JGGA, each chromosome includes some transposons, the length of each transposon can be more than one unit, and the locations of the transposons are randomly assigned. The actual manipulation of the cut-and-paste operation is that the element is cut from an original position and pasted into a new position. As for the copy-and-paste operation, the copy of the element from an original position is inserted into a new location.

The procedures for the transposition operation are implemented between the selection process and the crossover mechanism. The transposition operation is similar to other genetic operations. Furthermore, in the transposition operation, the transpositions made within the same chromosome or to a different chromosome are also selected randomly and there is no restriction to the chromosome choice.

The jumping operation can be described by the following pseudo code.

```
For each chromosome in population
  For each transposon in chromosome
    If a random number <jumping rate
      If a random number <jumping mode switching rate
          Copy-and-paste operation;
       Else
            Cut-and-paste operation;
```

The detail flowchart of the transposition of JGGA is shown in Fig.1. Because the original position and the destination position of the jumping gene are randomly determined, in a sense, the jumping operation has similar function with the mutation operation, both of which can improve the diversity of the chromosome complex. However, for JGGA, some genes are changed from other chromosomes, while in GA some genes are changed randomly. In allusion to the randomness of the jumping operation of JGGA, Ref. [16] presents several modifications of JGGA to improve the guiding opinions and decrease the blindness of the jumping operation.

A lot of simulations of benchmark optimization functions or real engineering optimization problems are carried out to compare the JGGA with the basic GA, the results of which show that the JGGA has faster convergence speed and higher global

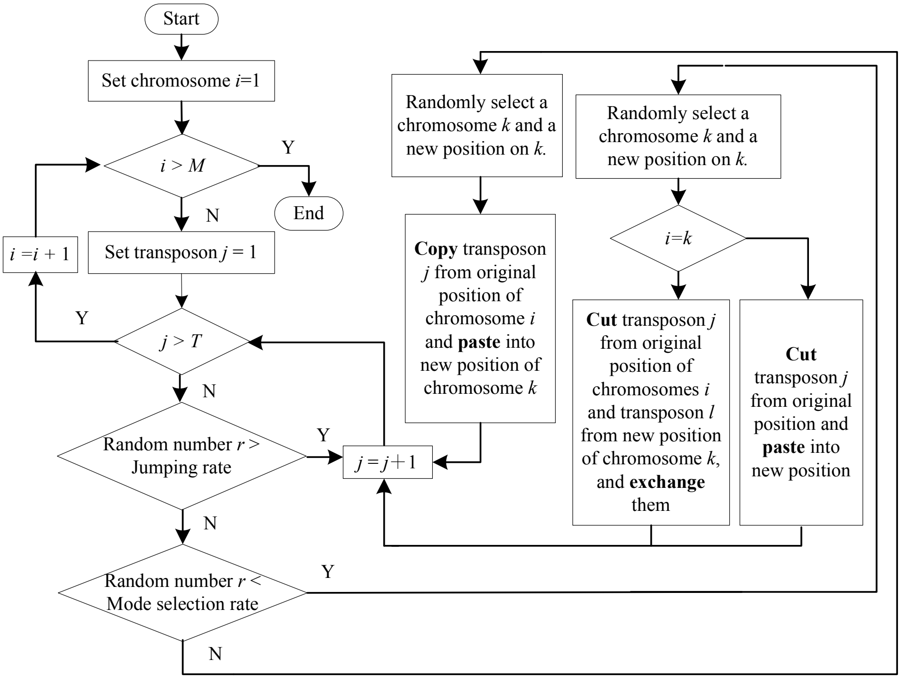

Start
Set chromosome i=1
i > M
Y
End
N
i =i + 1
Set transposon j = 1
Y
j > T
N
Random number r > Jumping rate
Y
j = j+1
N
Random number r < Mode selection rate
Y
N
Randomly select a chromosome k and a new position on k.
Copy transposon j from original position of chromosome i and paste into new position of chromosome k
Randomly select a chromosome k and a new position on k.
i=k
Cut transposon j from original position of chromosomes i and transposon l from new position of chromosome k, and exchange them
Cut transposon j from original position and paste into new position

Fig. 1. Flowchart of transposition of JGGA

searching precision [13-16]. Hence, the JGGA is more suitable to solve the multi-peak and multi-objective optimization problems.

4 Variable-Length JGGA Based Fuzzy Clustering for Image Segmentation

Let $X = \{x_{ij}, i = 1, 2, \cdots M, j = 1, 2, \cdots N\}$ be the data matrix of the image, and $x_{ij} \in [1, G]$, G is the grey level. The histogram of the image is $h_g = n_g$, n_g is the number of the pixel which grey level is g.

Because the image data matrix is large, the traditional data clustering method will make the computing time too long. Therefore, according to the features of the image data, a fast clustering segmentation algorithm is applied.

Because the cluster number is changed with time in the optimization process, the code length of the chromosome is changed with it. Therefore, a variable-length JGGA (VJGGA) is presented in this fuzzy clustering algorithm. The range of cluster number is [2, Ceiling($\sqrt{G}$)], thus the length of chromosome is Ceiling($\sqrt{G}$)$+1$, the first gene in chromosome represents the cluster number, and the others represent the cluster centers.

The main steps of the VJGGA for fuzzy clustering image segmentation are shown in following:

Step 1: Generate the initial population;

Step 2: Set generation number $k = 0$;

Step 3: For the image data are the repeated integers between [0, G]. Hence, Let data vector is [1, 2, …, G]. The fuzzy membership of each data is calculated by (1), to confirm which class it belongs to;

Step 4: Calculate the inner-class fuzzy clustering function of the image data by (2), and calculate the inter-class distance by (3). Finally, calculate the evaluation function by (4);

Step 5: Judge whether the terminal condition of the evaluation function index is satisfied, if yes then finish the generation process and record the results, else continue the generation process;

Step 6: Execute selection operation for the chromosome population;

Step 7: Execute gene jumping operation for the chromosomes with transposons;

Step 8: Execute crossover operation for the chromosome population;

Step 9: Execute mutation operation for the chromosome population;

Step 10: Judge if k is bigger than the maximum generation number, if yes then finish the generation process and record the results, else $k = k+1$ and return to step 4.

5 Simulation Study

In this section, a real image named Zhen is used to verify the effectiveness of the VJGGA based fuzzy clustering image segmentation method, comparing with a variable-length GA (VGA). The parameters of VJGGA are set as follows: population size is 50, maximum generation number 500, test time is 10. The best solutions obtained by VGA and VJGGA are shown in table 1. The VGA and the VJGGA obtain the same cluster number but different cluster centers, and the evaluation function values calculated by (4) show that the cluster centers of the VJGGA are better than that of the VGA.

Table 1. Best fuzzy clustering segmentation results of Zhen by VGA and VJGGA

Algorithm	Cluster number	Cluster centers	Evaluation function value
VGA	7	[9.9836, 118.0061, 192.2687, 64.9628, 31.8785, 219.2436, 13.7254]	15.6799
VJGGA	7	[119.3019, 13.7985, 36.0166, 87.3766, 202.7854, 225.0616, 161.9126]	15.6657

Fig.2 shows the mean convergence process of the optimization function solved by the VGA and the VJGGA under all the tests. Fig.3 shows the histogram and the cluster centers distribution of the real image under all the tests. The VGA has higher convergence speed during the initial stage, while the VJGGA has higher precision in the end. Viewing from Fig.3, we get that the cluster centers are usually close to the peak values of gray histogram, and the cluster centers obtained by VJGGA are closer to the peak values of gray histogram. The segmented results obtained by the intelligent algorithms are satisfactory, as shown in Fig.4-Fig.6. The VJGGA based fuzzy clustering method has better segmentation quality, due to the stronger capability of overcoming the uneven light and the noise, when comparing with VGA.

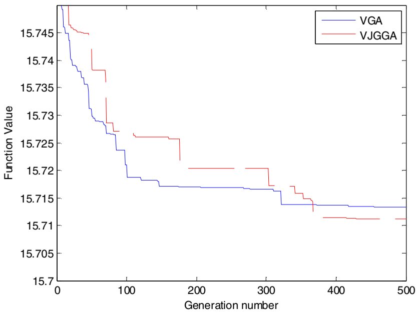


Fig. 2. Mean convergence process of evaluation function

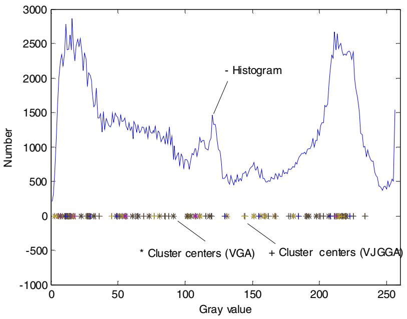


Fig. 3. Histogram and cluster centers distribution of the real image

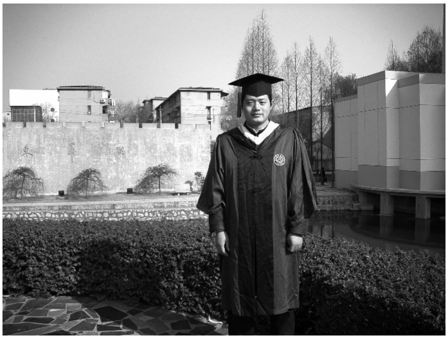


Fig. 4. Original image of Zhen

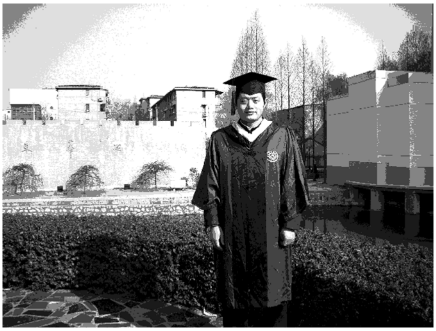


Fig. 5. Segmented image of Zhen by VGA

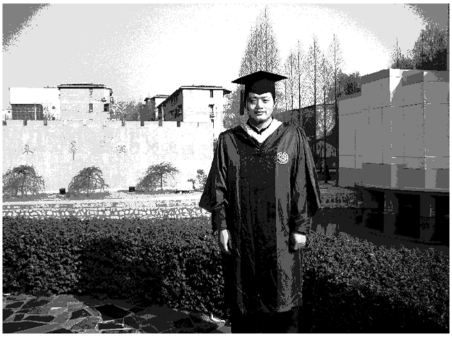


Fig. 6. Segmented image of Zhen by VJGGA

6 Conclusion

For the traditional clustering method, the cluster number is usually set beforehand, according to the peak value number of the image gray histogram. However, the cluster numbers of the complex images are always difficult to be determined. Considering this problem, this paper proposes a new fuzzy clustering evaluation function of image segmentation which is the sum of the cluster number component, inner-class distance and inter-class distance. By minimizing the evaluation function, the cluster number and the cluster centers of the image gray values are obtained. For improving the adaptability of the fuzzy clustering method for image segmentation, a VJGGA is designed to optimize the fuzzy clustering function. Simulation results of a real image show that the VJGGA has stronger global searching ability than the VGA and both of them can obtain the good segmentation effect.

Comparing with GA, JGGA owns more adjustable parameters, such as jumping rate, transposon number, transposon length and jumping mode selection rate. Moreover, the gene jumping operation adds the randomness of chromosome genes, without any guidance and learning. Therefore, the parameters selection problem and the improvement of gene jumping operation are the main work in future.

Acknowledgments

The authors would like to thank the anonymous referees for the useful suggestions for improving this paper. This project is supported by the National High-Tech Research and Development Plan of China under Grant No. 2009AA010307.

References

1. Felzenszwalb, P., Huttenlocher, D.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision 59(2), 167–181 (2004)
2. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive Image Segmentation Using an Adaptive GMMRF Model. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 428–441. Springer, Heidelberg (2004)
3. Awad, M.M., Chehdi, K., Nasri, A.: Multicomponent Image Segmentation: A Comparative Analysis Between a Hybrid Genetic Algorithm and Self-Organizing Maps. International Journal of Remote Sensing 30(3), 595–610 (2009)
4. Ning, J.F., Zhang, L., Zhang, D., Wu, C.K.: Interactive Image Segmentation by Maximal Similarity Based Regionmerging. Pattern Recognition 43, 445–456 (2010)
5. Widz, S., Slezak, D.: Approximation Degrees in Decision Reduct-Based MRI Segmentation. In: Frontiers in the Convergence of Bioscience and Information Technologies, Korea, October 11-13, pp. 431–436. IEEE Computer Society, Los Alamitos (2007)
6. Malyszko, D., Stepaniuk, J.: Adaptive Rough Entropy Clustering Algorithms in Image Segmentation. Fundamenta Informaticae 98, 199–231 (2010)
7. Zhen, Z.Y., Wang, Z.S., Liu, Y.Y.: Image Thresholding Method by Minimizing Fuzzy Entropy Function Based on Ant Colony Algorithm, Progress in Intelligence Computation and Applications. In: Second International Symposium on Intelligence Computation and Applications, September 21-23, pp. 308–311. China University of Geosciences Press, Wuhan (2007)
8. Zhen, Z.Y., Gu, Z., Liu, Y.Y.: A Novel Fuzzy Entropy Image Segmentation Approach Based on Grey Relational Analysis. In: Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services, Nanjing, China, November 18-20, pp. 1019–1022 (2007)

9. Liu, Y.Y., Liu, W.B., Zhen, Z.Y., Zhang, G.: Image Segmentation Based on Fuzzy Entropy and Grey Relational Analysis. In: Proceedings of the 4th International Conference on Image and Graphics, August 22-24, pp. 372–376. IEEE Computer Society Press, Los Alamitos (2007)
10. Liu, Y.Y., Liu, W.B., Zhen, Z.Y., Zhang, G.: Image Segmentation Method Based on Fuzzy Entropy and Grey Relational Analysis. Journal of Optoelectronics. Laser 19(9), 1250–1253 (2008)
11. Forouzanfar, M., Forghani, N., Teshnehlab, M.: Parameter Optimization of Improved Fuzzy C-Means Clustering Algorithm for Brain MR Image Segmentation. Engineering Applications of Artificial Intelligence 23, 160–168 (2010)
12. Zhen, Z.Y., Wang, D.B., Liu, W.B., Liu, Y.Y.: Variable-Length PSO Optimized Fuzzy Clustering for Self-Adaptive Image Segmentation. Journal of Optoelectronics. Laser 20(1), 99–102 (2009)
13. Chan, T.M., Man, K.F., Tang, K.S., Kwong, S.: A Jumping Gene Algorithm for Multi-Objective Resource Management in Wideband CDMA Systems. Computer Journal 48(6), 749–768 (2005)
14. Chan, T.M., Man, K.F., Tang, K.S., et al.: Multi-Objective Optimization of Radio-to-Fiber Repeater Placement Using a Jumping Gene Algorithm. In: IEEE International Conference on Industrial Technology, pp. 291–296 (2005)
15. Ripon, K.S.N., Kwong, S., Man, K.F.: A Real-Coding Jumping Gene Genetic Algorithm (RJGGA) for Multi-Objective Optimization. Information Science 177(2), 632–654 (2007)
16. Pu, H.Z., Zhen, Z.Y., Wang, D.B., Liu, Y.Y.: Improved Jumping Gene Genetic Algorithm for Multi-Peak Function Optimization. Nanjing University of Aeronautics and Astronautics 39(6), 829–832 (2007)
17. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transaction on Evolutionary Computation 6(2), 182–197 (2002)

Perspectives in Dynamic Optimization Evolutionary Algorithm

Zhiqiong Bu[1] and Bojin Zheng[2,*]

[1] Dept. of Information Engineering, Guangdong Polytechnic Normal University,
Guangzhou 510665, China

[2] College of Computer Science, South-Central University For Nationalities,
Wuhan 430074, China
dr.zhengbj@gmail.com

Abstract. Dynamic Optimization Evolutionary Algorithm(DOEA) is an intrinsic development of traditional Evolutionary Algorithm. Different to the traditional Evolutionary Algorithm which is designed for stationary or static optimization functions, it can be used to solve some dynamic optimization problems. The traditional Evolutionary Algorithm is hard to escape from the old optimum after the convergence when dealing with dynamic optimization problems, therefore, it is necessary to develop new algorithms. After reviewing the relative works, three directions are proposed: first,by treating the time variable as a common variable, DOPs can be extended as a kind of special Multi-objective Optimization Problems, therefore, Multi-objective Optimization Evolutionary Algorithm would be useful to develop DOEAs; second, it would be very important to theoretically analyze Dynamic Optimization Evolutionary Algorithm; finally, DOEA can be applied into more fields, such as industrial control etc..

Keywords: Evolutionary Algorithm, Dynamic Optimization, Multi-objective Optimization.

1 Introduction

Since 1960s, the researches on Evolutionary Algorithms(EAs) have been greatly promoted. The most important application of EAs certainly is optimization. In many circumstances, the optimization problems can be represented as the minimization of functions, and the minimum values is irrelative to time, that is, they would not vary with the elapsing time variable. But with the intrinsic development of EAs, the researchers are paying more and more attention to Dynamic Optimization Problem(DOP), or say, Non-stationary Optimization Problem, and develop Dynamic Optimization Evolutionary Algorithms (DOEAs) to deal with this kind of problems.

Dynamic Optimization Problems come from the industry, actually, many control problems can be transferred into DOPs. But in general, the researchers did not achieve a consensus on the definition and the taxonomy of DOPs, since "Dynamic" is too complex and complicated.

[*] Corresponding author.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 338–348, 2010.

Formally, DOP can be depicted as equation 1.

$$\begin{aligned} &\min f(\boldsymbol{x},t) = \begin{cases} \min f_1(\boldsymbol{x},t) & \mathrm{t_1}< \mathrm{t} \leq \mathrm{t_2} \\ \min f_2(\boldsymbol{x},t) & \mathrm{t_2}< \mathrm{t} \leq \mathrm{t_3} \\ \ldots & \\ \min f_n(\boldsymbol{x},t) & \mathrm{t_n}< \mathrm{t} \leq \mathrm{t_{n+1}} \end{cases} \\ &s.t. g(\boldsymbol{x},t) \leq 0 \\ &g(\boldsymbol{x},t) = \begin{cases} g_1(\boldsymbol{x},t) & \mathrm{t_1}< \mathrm{t} \leq \mathrm{t_2} \\ g_2(\boldsymbol{x},t) & \mathrm{t_2}< \mathrm{t} \leq \mathrm{t_3} \\ \ldots & \\ g_n(\boldsymbol{x},t) & \mathrm{t_n}< \mathrm{t} \leq \mathrm{t_{n+1}} \end{cases} \end{aligned} \tag{1}$$

When $n \to \infty$, the solution of a DOP can be expressed as a sequence of the solutions of a series of optimization problems. When $n = 1$ and for constant t, DOP degenerates as a single-objective optimization problem.

Notice that DOP defined here is quite different to the current researches in some details, although this equation seemly expressed similar ideas. First, the definition here can be explained as that EAs can have prior knowledge on the problems, EAs can know the period, the value of n, the functions themselves, so EAs can take appropriate actions according to the knowledge. Second, the definition here somehow ignores "the dynamic environment"."The dynamic environment" may imply that the environment is not predictable, this implication would confuse the researchers and make the communications between the scientists inefficient. Actually, we agree on that the researches on DOP, or say, DOEA, would be built on a solid foundation, that is, based on clear and specific problems, not an algorithms for universal aims.

Although the researchers have put various implicit preconditions on DOPs, some basic principles are regarded reasonable. For examples, the dynamic environment should change continuously, the dynamic environment after a change should have exploitable similarities to the environment before the change and so on. Actually, these principles demand that the environment should not be random, such that there are some interests to research these problems.

This paper is organized as follows. First, this paper introduces the current advances, include the current algorithms, the performance evaluation indictors, the test-bed functions and the theoretical researches in section 2. Second, according to the-state-of-the-art researches, this paper discusses the open issues. Finally, this paper makes conclusions.

2 Current Works

Although the researchers did not reach consensus on DOPs, the researchers proposed several DOEAs to deal with this kind of problems. They also proposed several evaluation indicators to measure the performance of DOEAs, and several test-bed functions to standardize the comparisons. Moreover, they made some progresses on the theoretical analysis.

2.1 Current Algorithms

Since 1960s, a couple of DOEAs had been proposed, but these algorithms mainly focused on keeping the population's diversity or storing the historical best solutions based on memory.

Generally speaking, these algorithms can be categorized by two means, one is based upon the employed techniques, the other is based upon the characters of problems.

The taxonomy by the techniques. J. Branke categorized DOEAs into four classes[3,6,29].

1. Increasing the diversity after the environmental change.
   The simplest method is the re-starting method. The idea of this method is that when the environment changes, every individual will be randomly generated again. Since this method did not save the information in the evolution, many problems will take a long time to restart the process of evolution and cannot get good results.

   Hyper-mutation is the representative of this type of method[20], and the basic idea of this method is that population's mutation ratio should be increased sharply in some generations after environment changes. The improvement of this method includes [25] and [24], they gradually increase the mutation ratio and adaptively increase the mutation ratio.
2. Maintaining the diversity.
   Evolutionary algorithm tends to converge to one solution, which is inappropriate for the dynamic optimization problems. If all the individuals converge to the only one point, the population is hard to come out from this point to search better solutions when the environment changes. Therefore, some methods are proposed to prevent the algorithm to converge. An approach is that a number of randomly generated individuals will be inserted into population in every generation[11], which can be thought as the compromise version of the re-starting method.
3. The method based on memory.
   There are two ways to reflect the idea of memory, one of them is the implicit memory method, and the other is the explicit memory method. The implicit memory method was proposed earlier, and redundancy can be regarded as the original version. The explicit memory method is attracting current researchers now, such as [4][2].

   By employing the memory, DOEAs can keep diversity and record the history, and the history may be useful of predicting the hopeful spaces when the environment changes.
4. Multi-population method and migration method.
   The basic idea of this kind of methods is to divide the whole population into some sub-populations to trace multiple peaks in the fitness landscape [5,33,22,27]. Every sub-population is responsible of several hopeful areas, so this method can be regarded as an improvement of the memory method.

Besides the method proposed above, there are some algorithm can not be categorized, such as the futurist approach[12].

The taxonomy by the characters of DOPs. Different to the taxonomy of J. Branke, Yamasiki[26,15] points out that when the optimal solutions jump from here to there at every time, adaption of evolutionary algorithms would not perform well, moreover, periodical environments are not the environments of all the real-world DOPs. He also proposed that the aim of adaption is not for the optimal solutions, but for how the environment changes, therefore, the adaption in current algorithms is used for the characters of changing environments.

Yamasiki categorized current DOEAs into four classes, that is, the algorithms utilized the reappearance,the algorithms utilized the continuity,the algorithms utilized the sparsity, the algorithms utilized the predictable property.

1. The reappearance.
   The reappearance means that the best solution will appear after n times changes of the environment. Obviously, the best method to deal with the problems with such a character would be the methods based on memory.
2. The continuity. That is, the next best solution will appear nearby the historical best solutions. For this kind of problem, the best method is to use Neighborhood Search Operators.
3. The sparsity. That is, few environmental changes would happen. If the environment changes greatly, then the frequency of the changes of environment would be small. To deal with this kind of problems, the best algorithms would be the algorithms based on keeping diversity, include the algorithms which keep the diversity for the while evolving process and increase the diversity drastically in a certain time.
4. The predictable property. That is , the environment is predictable. Here, the futurist approach[12] would be a good method. Moreover,Literature [1] proposed two variants of ecGA to deal with dynamic environment, called dcGA(1) and dcGA(2) respectively. In these algorithm, implicit predication approach is employed. Literature[16] emphasized the predication method, because the predication method is also a kind of adaption. But since now, a few works utilized the prediction approach. Some work, such as literature[21] explored how to avoid the collision of boats by the predication approach, literature[17] discussed DOPs in automobile industry.

Moreover, because DOPs have tight relationships with MOPs, some researchers have proposed some DOEAs based on MOEAs. Of course, people can also use DOEAs to develop MOEAs.

2.2 The Evaluations on Performance

Single-objective optimization evolutionary algorithms aim to the capability of “finding” the optimal solution. But DOEAs aim to the capability of “tracking” the sequence of optimal solutions for various time variable. So there is a trade-off in the evaluation of the performance, that is , on one hand, it is very important to obtain the best solution at certain time; on the other hand, it is also very important to obtain good solutions at all the time. A saying goes that the worst clock would have two times to signal absolutely right, but the best clock would never run right. Based on different viewpoint, the evaluation would be very different.

Currently, some performance indicators have been proposed. For examples, online performance, offline performance, adaptation, accuracy, etc.

Online performance. In stationary single-objective optimization, online performance is the average value of the fitness of all historical individuals. It can be used to measure the performance of DOEAs.

Here Let f_{ti} is the fitness of $i-th$ evaluation at the $t-th$ environment, T is the total number of environmental change, I_t is the total number of all evaluations in the $t-th$ environment, thus, online performance can be depicted as equation 2.

$$ONP = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{I_t} f_{ti} \tag{2}$$

Offline performance. Offline performance means the average value of the fitness of all the historical best solution. Here Let$f_{best(t)}$ is the best fitness at the $t-th$ environment, offline performance can be defined as equation 3.

$$OFP = \frac{1}{T}\sum_{t}^{1} f_{best(t)} \tag{3}$$

Adaptation. Mori et al. proposed *adaption* as the measure of performance[18] as equation 4.

$$Ada = \frac{1}{T}\sum_{t=1}^{T} \frac{f_{best(t)}}{f_{opt(t)}} \tag{4}$$

here, $f_{opt(t)}$ represents the best fitness at time t.

Accuracy. Mori[19] et al. and Trojanowski et al.[23] proposed *accuracy* as the measure of performance as equation 5.

$$Acc = \frac{1}{T}\sum_{t=1}^{T} (f_{best(t)} - f_{opt(t)}) \tag{5}$$

Besides, some researchers have proposed some other evaluation indicators.

In general, for any indicator, there must exist a basic hypothesis that the algorithms should converge, that is, when $t \to \infty$ the algorithm would certainly track the optimal solutions. However, current designs on the evaluation indicators did not emphasize, even mention this precondition. Actually, if the algorithms do not converge, all the indicators above can not deterministically measure the performances, since the future is not predictable for the algorithms and therefore the deviation is also unpredictable. Thus, the values of the indicators would depend on the choice of time variable t. It would be possible that if $t = 100$, algorithm A is better than B; but if $t = 200$, B is better than A.

Current DOEAs did not pay enough attention to the convergence, so the evaluation indicators should be improved. Only if the compared algorithms are convergent, the comparisons are trustable.

2.3 The Test-Bed Functions

As to the constructed test-bed functions, they would have some features. Commonly, they should be efficiently calculated to save the computation cost; they would be easy to implement various dynamic features; they would be easy to control the complexity of fitness landscape; they would also be easy to obtain the trajectory of the optimal solutions by catalytical means.

Currently, the popular test-bed functions include Dynamic Bit-Matching, moving parabola, Moving Peaks Function, Dynamic TSPs etc.

Dynamic Bit-Matching. Dynamic Bit-Matching Problem [8] is the simplest dynamic problem. This problem is constructed to solve a bit sequence which change randomly with time, that is, the bit sequence would change with time in the dynamic environment, and the objective function si to summarize the same bits between the dynamic environment and the chromosomes of the individuals.

Moving Parabola. Moving Parabola is a popular test-bed problem[7], described as equation 6.

$$Min\ f(x,t) = \sum_{i=1}^{n} (x_i(t) + \delta_i(t))^2 \tag{6}$$

Here, t is the time variable, $x_i(t)$ is $i - th$ decision variable, and $\delta_i(t)$ can be depicted as equation 7.

$$\begin{array}{l} \delta_i(0) = 0, \forall i \in \{1, 2, \cdots, N\} \\ \delta_i(t+1) = \delta_i(t) + S, \forall i \in \{1, 2, \cdots, N\} \end{array} \tag{7}$$

Here, S commonly is 1.

Dynamic TSPs. Lishan Kang et al.[14] proposed a series of test-bed functions called CHN144+M based on CHN144 problem, that is, based on the original CHN144 problem and add M satellites. They also used a real-time evolutionary algorithm[32] to solve these problems. Among this series of problems, CHN144+1 is the simplest. This problem can be depicted as follows,

Based on the original CHN144 problem, add a satellite, this satellite circles with the center (2531,1906) and a radius of 2905, the time variable is the real time, the algorithms are asked to obtain the shortest TSP distance of 145 nodes.

Moving Peaks. J. Branke proposed a test-bed function with multi-modal and slight move[4] as equation 8.

$$f(x(t)) = \max_{i=1,..m} \left[H_i(t) - R_i(t) \times \sqrt{\sum_{j=1}^{n} (x_j - X_{ij}(t))^2} \right] \tag{8}$$

Here, $H_i(t) = H_i(t-1) + k_H\sigma$,$W_i(t) = W_i(t-1) + k_W\sigma$,$\sigma \in N(0,1)$, $X(t) = X(t-1) + \omega(t)$,$\omega(t) = \frac{s}{|r+\omega(t-1)|}((1-\lambda)r + \lambda\omega(t-1))$.

The other test-bed functions. Yang and Yao et al. proposed that the test-bed instances can be generated by a XOR generator[28], Farina [9,10] and Jin [13] et al. also proposed dynamic multi-objective optimization problems which integrate the multi-objective optimization problems.

2.4 Theoretical Analysis

Currently, few works focused the theoretical analysis of DOESs. Zheng et al. [30,31] defined the convergence of DOEAs and designed an algorithm to deal with predictable DOPs, and proved that this algorithm can converge under certain conditions. However, this problem can not deal with the DOPs with randomness and noise, also can not deal with the DOPs with implicit functions.

The theoretical analysis is necessary to be promoted.

3 Perspectives

In contrast to hundreds of Multi-objective Optimization Evolutionary Algorithms, current researchers only proposed tens of DOEAs. There are great room for the development of DOEAs. For examples, the new branches such as Particle Swarm Optimization, Culture Algorithm and Quantum Inspired Algorithm etc. could be applied into this field, thus to improve DOEAs. Moreover, there are still many works to do in the metrics of performance and the test-bed functions.

But the most important directions would be: integration with MOEAs, theoretical analysis and application.

3.1 Integration with MOEAs

DOPs are very similar to MOPs. Actually, the nature of DOPs is to solve a sequence of minimum values. This sequence can be regarded as a sampling on the dimension of time for dynamic functions. Similarly, the nature of MOPs is a sampling on the true Pareto fronts.

In MOEAs, histogram method is a feasible method to obtain the approximated Pareto front. That is, only considering to the bi-objective optimization problem, this method divides the functional value region of the first objective function into many parts, and for each parts, to obtain the minimum (or maximum) of the second objective function, finally, calculates the non-dominated set of all the minimum values.

We can reasonably extend DOPs to MOPs by adding an additional objective functions. Assume that a DOP is defined as $F(x,t)$, we can rewrite it as $F_1(x,t)$, and add another objective function as $F_2(x,t) = t$. Hence, DOP is transferred as MOP, and "tracking the optimal solutions" means a sampling on the second objective function. As to the solutions of the extended DOP and MOP, the only difference is that DOP would not need a calculation of "non-dominated solution set".

Although DOP is quite similar to MOP, there are still many difference.

First, the time variable. The time variable is independent variable, and its value is predefined and belongs to $[0, +\infty]$.

Second, the aims are different. MOEAs hope to obtain a spatial sampling, and then the results of sampling is refined to suit for the definition of Pareto front. But DOEAs just want to obtain the minimum value sequence, not to be Pareto, and the sampling should base on the time variable.

When we regard the time variable as the real time, DOEA would have a concept of "trade-off solutions".

In the stationary single-objective optimization problems, the solutions are stable and invariant, if the computing time is infinite, the well-designed algorithm would certainly obtain the best solutions. But DOPs are different. Assume that the time variable is the real time, the computing time would be limited, so the algorithms should trade off between the better solutions and the time.

3.2 Theoretical Analysis

Theoretical analysis on DOEAs is very difficult, but the difficulties partly owe to the unclear definition on DOPs. If treating the time variable in DOP as a variable, not the real time, it would be simple to reduce them as special MOEAs. If not, theoretical analysis would be possible only in some special circumstances.

If the time variable is just a variable, the most important issue would be the convergence of DOEAs. Different to MOEAs, there are at least two definition on it[30]. First, DOEAs may converge only when $t \to \infty$, we can call it "asymptotic convergence". Second, DOEAs may converge for each t, here, the time variable t is fully treated as a variable, we can call it "full convergence". Different definitions show different viewpoints on time variable. Another important issue would be the convergent rate. These issues would be very important directions.

If the time variable is not a common variable, the convergence of DOEAs would base on a hypothesis that the environment would be predictable. Otherwise, even the asymptotic convergence is not obtainable. Considering that some MOEAs are capable of convergence, so we can surely find the convergent DOEAs.

3.3 Applications

Currently, the fields of application of DOEAs still are a bit narrow, mainly in sea transportation, telecommunication etc.. Actually, if we treat the time variable as a common variable, most control problems can be transferred into DOPs, and then DOEAs can be used to solve them. This issue would be another important direction.

4 Conclusions

This paper reviewed the development of dynamic optimization evolutionary algorithms. According to the-state-of-the-art researches, this paper proposed that: since DOP can be extended as MOP, DOEAs are very similar to MOEAs, the experience of MOEAs can be applied into DOEAs; moreover, theoretical analysis is very important to the furthermore development of DOEAs.

Acknowledgements. The authors are very grateful for the financial support by the National Key Basic Research Program (973 Program) under Grant No.: 2007CB310804 and the National Natural Science Foundation of China under Grant No.: 60803095 and SCUEC Foundation under Grant No. YZZ 06025.

References

1. Abbass, H.A., Sastry, K., Goldberg, D.E.: Oiling the wheels of change: The role of adaptive automatic problem decomposition in nonstationary environments. Tech. rep., Illinois Genetic Algorithms Laboratory (IlliGAL), Department of General Engineering, University of Illinois at Urbana-hampaign (May 2004)
2. Bendtsen, C.N., Krink, T.: Dynamic memory model for non-stationary optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002, vol. 1, pp. 145–150 (2002)
3. Branke, J.: Evolutionary approaches to dynamic optimization problems: A survey. In: Proceeding of GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems, pp. 133–137 (1999)
4. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999, vol. 3, pp. 1875–1882 (1999)
5. Branke, J.: A multi-population approach to dynamic optimization problems. In: Parmee, I. (ed.) Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM 2000), pp. 299–308. Springer, Plymouth (2000)
6. Branke, J.: Evolutionary approaches to dynamic optimization problems: Updated survey. In: Proceeding of GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems, pp. 27–30 (2001)
7. Branke, J.: Evolutionary Optimization in Dynamic Environments. Kluwer Academic Publishers, Dordrecht (2002)
8. Droste, S.: Analysis of the (1 + 1) EA for a dynamically bitwise changing OneMax. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 909–921. Springer, Heidelberg (2003)
9. Farina, M., Deb, K., Amato, P.: Dynamic multiobjective optimization problems: Test cases, approximation, and applications. In: Carlos, M.F., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 311–326. Springer, Heidelberg (2003)
10. Farina, M., Deb, K., Amato, P.: Dynamic multiobjective optimization problems: test cases, approximations, and applications. IEEE Transactions on Evolutionary Computation 8(5), 425–442 (2004)
11. Grefenstette, J.J.: Genetic algorithms for changing environments. In: Maenner, R., Manderick, B. (eds.) Parallel Problem Solving from Nature, pp. 137–144. North-Holland, Amsterdam (1992)
12. Hemert, J., Hoyweghen, C., Lukshandl, E., Verbeeck, K.: A futurist approach to dynamic environments. In: Bc̆k, J.u.B., Thomas (eds.) Evolutionary Algorithms for Dynamic Optimization Problems, San Francisco, California, USA, pp. 35–38 (2001)
13. Jin, Y.C., Sendhoff, B.: Constructing dynamic optimization test problems using the multi-objective optimization concept. Applications of Evolutionary Computing 3005, 525–536 (2004)

14. Kang, L.S., Zhou, A.M., McKay, B., Li, Y., Kang, Z.: Benchmarking algorithms for dynamic travelling salesman problems. In: Proceedings of the 2004 Congress on Evolutionary Computation, New York, vol. 1, pp. 1286–1292 (2004)
15. Kazuo, Y.: Dynamic pareto optimum ga against the changing environments. In: yamasaki::DPOGACE, J.u.B., Dynamic Optimization Problems, San Francisco, California, USA, pp. 47–50 (2001)
16. Michalewicz, Z., Schmidt, M., Michalewicz, M., Chiriac, C.: Case study: an intelligent decision support system. Intelligent Systems IEEE [see also IEEE Intelligent Systems and Their Applications] 20(4), 44–49 (2005)
17. Michalewicz, Z., Schmidt, M., Michalewicz, M., Chiriac, C.: Prediction, distribution, and transportation: A case study. In: Lishan, K., Zhihua, C., Xuesong, Y. (eds.) Progress in Intelligence Computation & Applications, Wuhan, China, vol. 1, pp. 545–557 (2005)
18. Mori, N., Imanishi, S., Kita, H., Nishikawa, Y.: Adaptation to changing environments by means of the memory based thermodynamical genetic algorithm. In: Bäck, T. (ed.) International Conference on Genetic Algorithms, pp. 299–306. Morgan Kaufmann, San Francisco (1997)
19. Mori, N., Kita, H., Nishikawa, Y.: Adapation to a changing environment by means of the thermodynamical genetic algorithm. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 513–522. Springer, Heidelberg (1996)
20. Morrison, R.: Design evolutionary algorithms for dynamic environments. Doctoral thesis, George Mason University, Fairfax, Virgina (2002)
21. Smierzchalski, R.: An intelligent method of ship's trajectory planning at sea. In: Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, 1999, pp. 907–912 (1999)
22. Tins, R., Yang, S.: A self-organizing random immigrants genetic algorithm for dynamic optimization problems. Genetic Programming and Evolvable Machines 8(3), 255–286 (2007)
23. Trojanowski, K., Michalewicz, Z.: Searching for optima in non-stationary environments. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999, vol. 3, p. 1850 (1999)
24. Vavak, F., Jukes, K., Fogarty, T.C.: Learning the local search range for genetic optimisation in nonstationary environments. In: IEEE International Conference on Evolutionary Computation, 1997, pp. 355–360 (1997)
25. Vavak, F., Jukes, K., Fogarty, T.C.: Adaptive combustion balancing in multiple burner boiler using a genetic algorithm with variable range of local search. In: Bäck, T. (ed.) Proceedings of the 7th International Conference on Genetic Algorithms, pp. 719–726. Morgan Kaufmann, East Lansing (1997)
26. Yamasaki, K., Kitakaze, K., Sekiguchi, M.: Dynamic optimization by evolutionary algorithms applied to financial time series. In: Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002, vol. 2, pp. 2017–2022 (2002)
27. Yang, S.: Genetic algorithms with elitism-based immigrants for changing optimization problems. In: Giacobini, M. (ed.) EvoWorkshops 2007. LNCS, vol. 4448, pp. 627–636. Springer, Heidelberg (2007)
28. Yang, S., Yao, X.: Experimental study on population-based incremental learning algorithms for dynamic optimization problems. Soft Comput 9(11), 815–834 (2005)
29. Yaochu, J., Branke, J.: Evolutionary optimization in uncertain environments-a survey. IEEE Transactions on Evolutionary Computation 9(3), 303–317 (2005)
30. Zheng, B.: Researches on Evolutionary Optimization. Ph.D. thesis, Wuhan University (2006)

31. Zheng, B., Li, Y., Hu, T.: Vector prediction approach to handle dynamical optimization problems. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G.-L., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 353–360. Springer, Heidelberg (2006)
32. Zhou, A.M., Kang, L.S., Yan, Z.Y.: Solving dynamic tsp with evolutionary approach in real time. In: 2003 Congress on Evolutionary Computation, New York, vol. 1, pp. 951–957 (2003)
33. Zou, X., Wang, M., Zhou, A., McKay, B.: Evolutionary optimization based on chaotic sequence in dynamic environments. In: IEEE International Conference on Networking, Sensing and Control, 2004, vol. 2, pp. 1364–1369 (2004)

A Real Time Vision-Based Hand Gestures Recognition System

Lei Shi, Yangsheng Wang, and Jituo Li

Institute of Automation, Chinese Academy of Sciences,
100190, Beijing, China
shilei_inneu@163.com, {yangsheng.wang,jituo.li}@ia.ac.cn

Abstract. Hand gesture recognition is an important aspect in Human-Computer interaction, and can be used in various applications, such as virtual reality and computer games. In this paper, we propose a real time hand gesture recognition system. It includes three major procedures: detection, tracking and recognition. In hand detection stage, an open hand is detected by the histograms of oriented gradient and AdaBoost method. The hand detector is trained by the AdaBoost algorithm with HOG features. A contour based tracker is applied in combining condensation and partitioned sampling. After a hand is detected in the image, the tracker can track the hand contour in real time. During the tracking, the trajectory is saved to perform hand gesture recognition in the last stage. Recognition of the hand moving trajectory is implemented by hidden Markov models. Several HMMs are trained in advance, and the results from the tracking stage are then recognized using the trained HMMs. Experiments have been conducted to validate the performance of the proposed system. Under normal webcam it can recognize the predefined gestures quickly and precisely. As it is easy to develop other hand gestures, the proposed system has good potential in many applications.

Keywords: hand detection; hand tracking; hand gesture recognition.

1 Introduction

With the development of the advanced human-computer interaction technology, hand gesture recognition has been becoming one of the significant research areas. It has many applications in virtual reality, computer games, intelligent home appliances and other HCI fields. However, the high degree of freedom of human hand, the changing conditions of the environment, the distinction of one gesture between different people, or even one person at different time, makes visual hand tracking and recognition a challenging problem.

In science and technique world, the trend is to facilitate people to use complicated tools. Many attempts have been developed in hand detection from images or videos. There are some kinds of methods. Skin color is a strong clue in a video to distinguish hands from other objects, so color based hand detection is possible [1]. However, there may be some other skin-colored objects, such as the face, the arm, so only using skin color is not enough. Hand shape is another strong clue to perform hand detection [2],

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 349–358, 2010.

but as a concave object in cluttered backgrounds, it is not easy to describe it. Viola and Jones [3] applied integral images and AdaBoost to face detection, which brought a breakthrough to that field. Similarly, a lot work has been done using Harr-like features and AdaBoost algorithm [4].

Motion, color, contour, boundary, shape and view are features used for hand tracking. So there are some classical methods, such as mean shift [5], CAM-Shift [6], Kalman filter [7] and its extensions. Condensation [8] is also called Particle Filter, and it is the most popular method to solve tracking problems. And also in the resent few years, there are many researchers using Condensation combining other methods [9]. There are also some other methods. Haris proposed a tracking method using multiple clue fusion [10]. Based on their prior work, it can distinguish hand and face, but when arms expose in the images, it cannot distinguish the hand and arm. Yuan proposed a tracking method based on color classifier technique [11]. It applies Lab color space to train a skin classifier and searches hand bounding box in the regions of the hand location of last few frames.

As for dynamic hand gesture recognition, there are several approaches. Recent years, DTW and HMM are introduced into hand recognition. Hand gesture recognition is achieved by matching input motion trajectories and model trajectories using Dynamic Time Warping (DTW) [12]. And also HMM is widely used because of its good performance [13]. Huang present a method for hand gesture recognition based on Gabor filters and support vector machine [14].

The proposed work applies AdaBoost algorithm with HOG features (Histogram of Oriented Gradient) to perform the hand detection section. As for an open hand, the contour of the hand is almost the same, and HOG features have good descriptions for this kind of objects. More importantly, HOG features are robust to illumination variation. At the tracking stage, we apply the condensation and partitioned sampling algorithm to track the hand contour. We use the B-Spline to depict the hand contour. Condensation is one of the most efficient methods to solve tracking problems. HMM is a classical gesture recognition method which is capable of modeling spatio-temporal series where the same kind of gestures may differ in shapes and durations.

Fig.1 shows the flow chart of the system. Images input can be got from a live web camera or from a video file. From beginning, the hand detector detects hand in all scales of every input image. If there is no hand in the image, the detector continues detecting hand. If a hand is detected successfully, then the hand tracker begins to initialize. After initializing, the hand tracker continues to track the hand contour in real time, and the moving trajectory is recorded for the recognition stage. Meanwhile, the tracker is checking if there is a finish gesture, if a finish gesture is detected at some time, the recorded trajectory is sent to the recognition section to decide which type of gesture is. After the recognition, the system then runs state over process to recover into the detection stage, in order to recognize another hand gesture. If there is no finish gesture detected all the time, there is a time limit, when it comes to the time limit, the recorded trajectory is also submitted to the recognition processor to recognize the gesture type.

The structure of the paper is as follows. Section 2 describes the idea of hand detector. Section 3 describes the details of the contour based hand tracker. Section 4 describes the HMM used in the hand gesture recognition stage. Experiment results are presented in section 5 to validate the performance of the system.

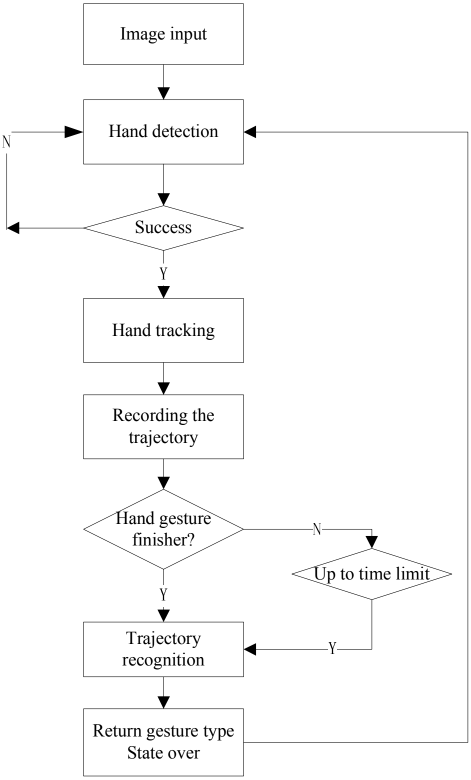


Fig. 1. System flow chart

2 Hand Detection

AdaBoost algorithm is widely used in face detection, while in this paper, it is used in hand detection. Haar features are not sufficient to describe the hand, because there are not enough and clear texture information. HOG (Histogram of Oriented Gradient) features are put forward by N.Dalal and B.Triggs [15], and HOG features are used in human detection by them, which validates a good result.

In the system, we use cascade structure based AdaBoost algorithm with HOG features to detect the user's hand. HOG features have powerful ability to describe the open hand shape information. And more importantly, HOG features are robust to illumination variation. A HOG feature is calculated by projecting the image gradient in several orientations, and then performing statistical analysis about the image gradient in the pre-decided orientations. An image is divided into several blocks, and every block is then divided into several smaller regions, called cells. HOG features are calculated in these blocks and cells. And an image is depicted by these HOG features. See it in figure 2.

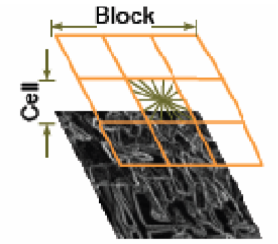
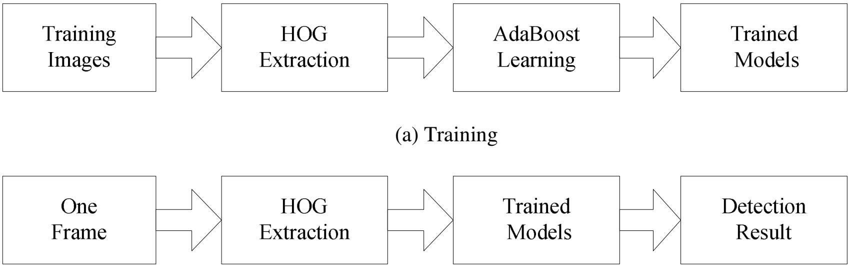


Fig. 2. Blocks and Cells

In the training stage, HOG features are extracted from the training images set and serve as the input of cascade structure based AdaBoost trainer to get an open-hand detecting model. HOG features have advantage over Haar features, since Haar features are sensitive to illumination variation.



(a) Training

(b) Detecting

Fig. 3. Trainning and detecting process

The shape information of an opening hand is relatively unique in the scene. We calculate the HOG features of a new observed image to detect the open hand at different scales and locations. When the hand is detected, we update the hand color model which will be used in hand tracking. To make the algorithm work well, the user should keep his palm opened vertically and statically before getting the detection result.

3 Hand Tracking

In the tracking stage, we use a contour based hand tracker combing two techniques, condensation and partitioned sampling. The tracker is initiated after the hand is detected. And the skin model is calculated at this time.

The hand contour is represented with B-Splines, as shown in Fig. 4. One 14 dimensions state vector is used to describe the dynamics of the hand contour:

$$\chi = (x, y, \alpha, \lambda, \theta_L, l_L, \theta_R, l_R, \theta_M, l_M, \theta_I, l_I, \theta_{Th1}, \theta_{Th2})$$

The sub-vector (x, y, α, λ) is a non-linear representation of a Euclidean similarity transform applied to the whole hand contour template, and (θ_L, l_L) represents the non-rigid movement of the little finger, θ_L means the little finger's angle with respect to the palm, and l_L means the little finger's length relative to its original length in the hand template. (θ_R, l_R), (θ_M, l_M), and (θ_I, l_I), have the same explanation as the sub-vector (θ_L, l_L), but for different fingers. θ_{Th1} represents the angle of the first segment of the thumb with respect to the palm, and the last part θ_{Th2} represents the angle of the second segment of the thumb with respect to the first segment of the thumb.

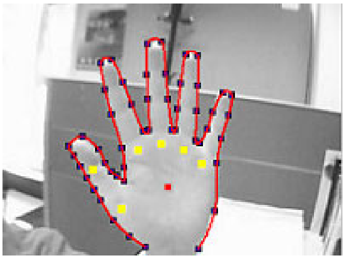


Fig. 4. Hand contour

We use a second-order auto-regressive processes to predict the motion of the hand contour:

$$x_t = A_1 x_{t-1} + A_2 x_{t-2} + B w_t. \quad (1)$$

where A_1, A_2 are fixed matrices representing the deterministic components of the dynamics, B is another fixed matrix representing the stochastic component of the dynamics, and w_t is a vector of independent random normal $N(0,1)$ variants.

In prediction, lots of candidate contours will be produced, we chose the one which can match the image feature best. Usually, to make the condensation filter achieve satisfactorily, more dimensions of the state space are required which increases the computation. To make the algorithm efficient, we use the technique called partitioned sampling which divides the hand contour tracking into two steps: the rigid movement of the whole hand(represented by (x, y, α, λ)), and the non-rigid movement of each finger(represented by angle and length of each finger). We predict the rigid motion first, and then predict the movement of the fingers, which reduces the amount of candidate contours greatly; therefore the efficiency of the tracker is improved.

4 Hand Recognition

In this system, we defined 7 types of hand gesture: left to right, right to left, up, down, question mark, horizontal circles and vertical circles. These hand gestures are recognized using the technique based on hidden Markov models. The hand moving trajectory is gained in the tracking stage. And in the recognition stage, directional features are extracted and used to compute the probability of each gesture type with the trained HMMs.

We trained a HMM for each type of hand gesture. According to the previous experimental results, we choose orientation feature as the main feature in our system. A gesture trajectory is spatio-temporal pattern which consists of centroid point (x_t,y_t). Fig.5 shows some examples of the trajectories. The orientation of each point is determined by the ray from itself and its following point, and is assigned as an integer value (1 to 18) according to the direction of the ray. Those orientations make a vector which is used as the input of the HMMs.

There are three main problems for HMM: Evaluation, Decoding and Training. These three problems can be basically solved with using Forward algorithm, Viterbi algorithm and Baum-Welch algorithm, respectively. The gesture models are trained using BW re-estimation algorithm and the numbers of states are set depending on the complexity of the gesture shape. We collected more than 300 trajectory samples of each isolated gesture for training and about 100 trajectory samples of each isolated gesture for testing.

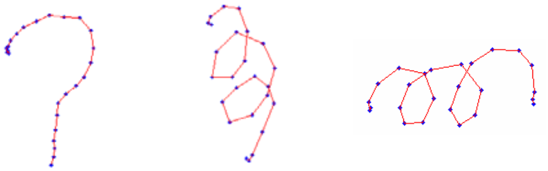


Fig. 5. Examples of hand trajectories

However these trained HMMs cannot effectively model undefined gesture patterns. Therefore, we use another stochastic model called threshold model [16], which uses the trained HMMs as the input. Threshold model well describes the undefined gestures. When the input trajectory fits more threshold model than the trained HMMs, we treat it as an undefined gesture.

We choose left-right banded model (Fig. 6) as the HMM topology, because the left-right banded model is good for modeling order-constrained time-series whose properties sequentially change over time. Since the model has no backward path, the state index either increases or stays the same as time increases.

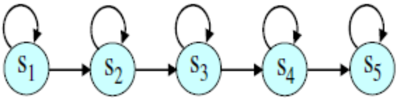


Fig. 6. Left-Right banded model

After finishing the training process by computing the HMM parameters for each type of gesture, then a given gesture is recognized corresponding to the maximal likelihood of seven HMM models by using Viterbi algorithm.

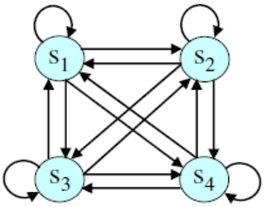


Fig. 7. Ergodic topology

Although the HMM recognizer chooses a model with the best likelihood, we cannot guarantee that the pattern is really similar to the reference gesture unless the likelihood is high enough. Therefore we produced a new threshold model that yields the likelihood value to be used as a threshold. The threshold model is a weak model for all trained gestures in the sense that its likelihood is smaller than that of the dedicated gesture model for a given gesture, and is constructed by collecting the states of all gesture models in the system using an ergodic topology shown in Fig. 7. A gesture is then recognized only if the likelihood of the best gesture model is higher than that of the threshold model; otherwise it is recognized as non-gesture type.

5 Experiment Results

As a real time hand gesture recognition system, users do some predefined hand gestures in front of a normal webcam , then this system can recognize what hand gesture the user has done, and then to drive some other events. The hand detection, hand tracking and hand recognition are all in real time, so this system can fulfill the requirement of many applications of hand gesture recognition.

Fig.8 shows some experiment results of hand detection in the cluttered-background video source. In the detection result, (a) indicates that an open stright hand can be

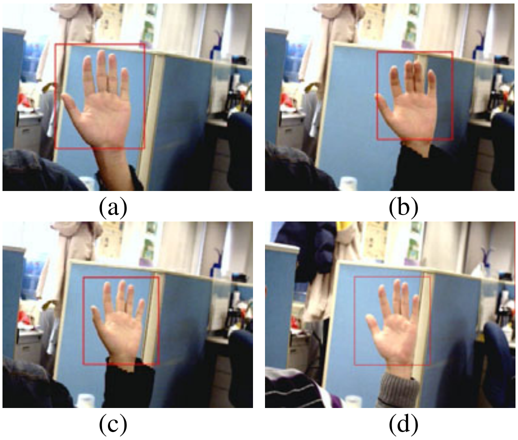
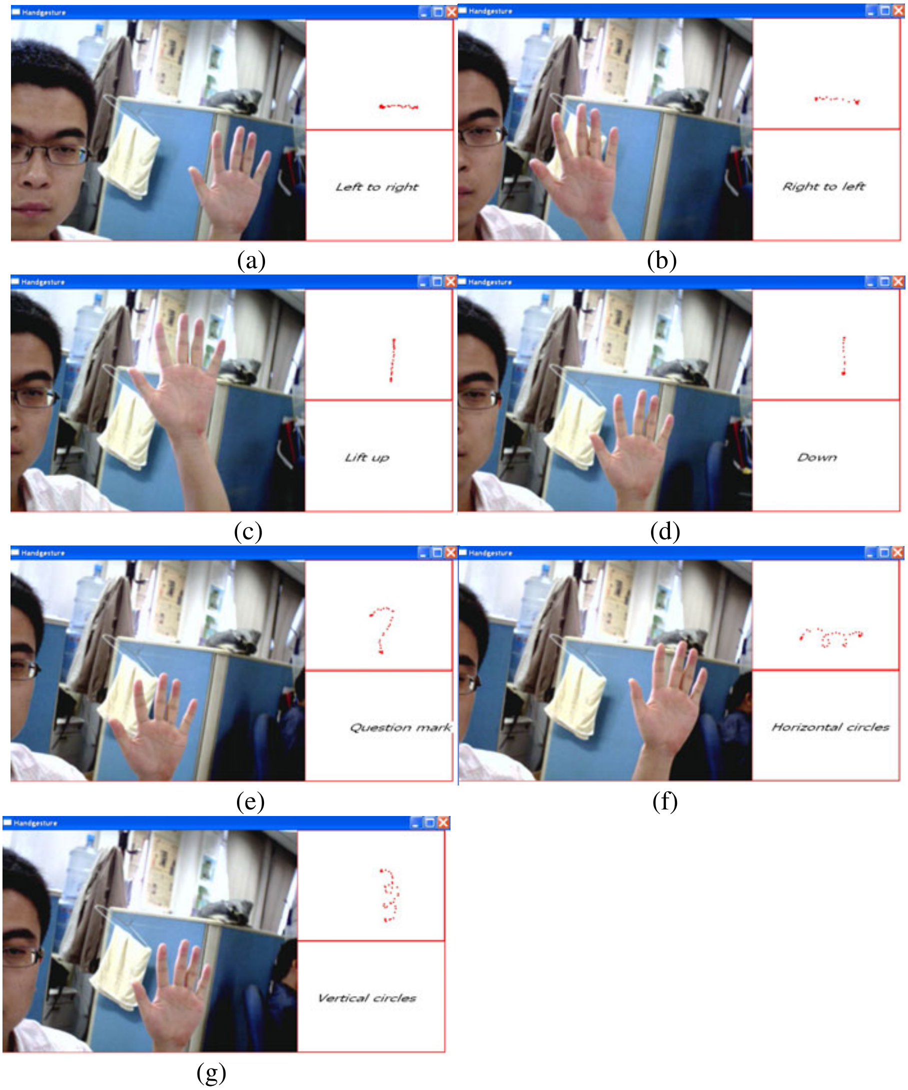


Fig. 8. Hand detection results



(a) (b) (c) (d) (e) (f) (g)

Fig. 9. Hand recognition results

detected, while when the hand is not parallel with the camera, the detector can also locate it, showed in (b)~(d).

Fig.9 shows some results of hand gestures recognition. In (a)~(g) there are the hand gesture trajectories and recogniton results. Here are some statistical data of the recognition of hand trajectory. Type 1 to 7 represents left to right, right to left, up, question mark, down, horizontal circles and vertical circles respectively. And some new gestures can easily be added into the system merely by training several new HMM models. In addition, all these hand gestures can be defined into some manipulate order to the computer or other devices.

Table 1. Recognition rate in the test set

Type	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Rate	98.1%	95.8%	96.0%	98.9%	89%	82%	96%

We collected data from 15 persons to perform training and testing, 300 samples for one gesture, 200 for training and 100 for testing. Among the 15 persons, there are males and females of different ages in order to eliminate the specificity of users. From the tests of the system, in order to improve the reliability of the detection of the hand, only when a hand is detected consecutively for 4 frames can a hand be thought of detected successfully. In the tracking section, a lost-tracking checking process is performed so as to eliminate the wrong decisions in the recognition process. From the statistics data of the test experiments we can see that the overall recognition rate can reach 93.69%. It is a high recognition rate so that the system can be used in practice.

Generally, the tracking of hand can run at 25 frames per second, and the recognition process needs much less time, so this system can run in real time on ordinary PCs. The program is developed and tested on a computer with the Celeron(R) CPU 2.66GHz and 1 GB of RAM, which is a low level configuration in the present time. If the computer is faster, the performance may be better.

6 Conclusion

In this paper, a real time hand gesture detection, tracking and recognition system is proposed. In this system, we use HOG (Histogram of Oriented Gradient) features and AdaBoost algorithm to perform hand detection, which is robust under various illuminations. The condensation and partitioned sampling method is used to implement tracking the contour of the hand. And in the hand tracker, B-Spline is utilized to describe the hand contour, which has many advantages over other curves. The hidden Markov model method is used to recognize the defined several gestures. HMMs are easy to realize yet ready to yield good performance. In addition, it is convenient to include other new hand gestures. From the results of the test experiments, we can see that this system can recognize hand gestures at a high rate. There are some limitations under only one normal web camera to solve computer vision problems; however, if the recognition rate can reach more than 90%, it can be used in practice.

The contribution of this paper lies in that we proposed a framework of hand gestures recognition including hand detection, tracking and recognition under normal camera. Potentially, the system can be used to control the mouse, such as controlling the PPT. As it is very easy to use, the system can also be used in computer games, as well as other virtual reality and augmented reality fields.

Acknowledgement

The authers acknowledge the support by China Hi-Tech Research and Development Programe (2007AA01Z341) and Haidian Park Culture Creativity Industrial Fund.

References

1. Jones, M.J., Rehg, J.M.: Statistical Color Models with Application to Skin Detection. Int. Journal of Computer Vision 46(1), 81–96 (2002)
2. Cootes, T.F., Taylor, C.J.: Active Shape Models: Smart Snakes. In: Proceedings of the British Machine Vision Conference, pp. 9–18. Springer, Heidelberg (1992)
3. Jones, M., Viola, P.: Fast Multi-view Face Detection. Technical Report TR2003-96, MERL (2003)
4. Kolsch, M., Turk, M.: Robust Hand Detection. In: Proc. IEEE Intl. Conference on Automatic Face and Geature Recognition (2004)
5. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, Boston (1990)
6. Liu, N., Lovell, B., Kootsookos, P.: Evaluation of hmm training algorithms for letter hand gesture recognition. In: IEEE International Symposium on Signal Processing and Information Technology (2003)
7. Welch, G., Bishop, G.: An introduction to the Kalmal Filter. In: SIGGRAPH 2001, Course 8 (2001)
8. Isard, M., Blake, A.: CONDENSATION-Conditional density propagation for visual tracking. International Journal of Computer Vision 29(1), 5–28 (1998)
9. Shan, C., Wei, Y., Tan, T., Ojardias, F.: Real Time Hand Tracking by Combining Particle Filtering and Mean Shift. In: Automatic Face and Gesture Recognition, pp. 669–674 (2004)
10. Baltzakis, H., Argyros, A., Lourakis, M., Trahanias, P.: Tracking of Human Hands and Faces through Probabilistic Fusion of Multiple Visual Cues. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) ICVS 2008. LNCS, vol. 5008, pp. 33–42. Springer, Heidelberg (2008)
11. Yuan, M., Farbiz, F., Manders, C.M., Tang, K.Y.: Robust hand tracking using a simple color classification technique. In: VRCAI 2008, Singapore, December 8-9 (2008)
12. Corradini, A.: Dynamic Time Warping for Off-Line Recognition of a Small Gesture Vocabulary. In: Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems RATFG-RTS 2001, p. 82 (2001)
13. Keskin, C., Akarun, L.: Sign tracking and recognition system using input-output HMMs. Pattern Recognition Letters 30(12), 1086–1095 (2009)
14. Huang, D.-Y., Hu, W.-C., Chang, S.-H.: Vision-Based Hand Gesture Recognition Using PCA+Gabor Filters and SVM. In: 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 1–4 (2009)
15. Dalai, N., Triggs, B., Rhone-Alps, I., Montbonnot, F.: Histograms of Oriented Gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, vol. 1 (2005)
16. Lee, H., Kim, J.: An HMM-based threshold model approach for gesture recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(10), 961–973 (1999)

A Virtual Robot Arm Model with Force Feedback and Contact States Identification

Chengjun Chen and Niu Li

School of Mechanical Engineering, Qingdao Technological University, China
ccmilitary@yahoo.com, Liniu8412@163.com

Abstract. This paper modeled a force feedback-based virtual robot arm which can identify contact states between virtual prototypes. In order to enable the virtual robot arm to be driven by haptic device and haptically interact with virtual prototypes in virtual environment, a workspace mapping method based on robot kinematics analysis is proposed. A contact states identification method based on boundary and topology model is presented. In this method the potential contact elements (such as vertex, edge and surface) are obtained by comparing the connectivity degree of the convex, edge and surface of overlaid virtual models. According to the priority of different contact states, different potential contact states are set using potential contact elements of both parts, once the distance and direction between contact elements meet those of some contact state, the contact state between two parts is identified. With the presented virtual robot arm model, operators can drive the virtual robot arm interactively and feel the contact states between virtual prototypes using feedback force.

Keywords: robot arm model, force feedback, virtual teaching, contact states identification.

1 Introduction

With the developments of virtual reality technology, virtual teaching robot is widely studied and has been playing an important role in task planning and programming of articulated robot arm [1]. The key issue to realize virtual teaching robot lies at the human-machine interface, which should provide users with a natural interface to control the virtual robot arm. In this paper, the PHANToM® desktop haptic device is employed to drive an articulated robot arm and display the interaction force of endpoint of the virtual robot arm.

In many virtual reality simulations contact states between the virtual prototypes are needed to be identified. Zhang [2] used Gilbert polyhedron distance calculation method for identifying contact state of two polyhedron models. Takamatsu etc. [3] used rotation theory to express contact state. Fei [4] presented an error-based contact state prediction method， which was used in virtual assembly operation. This paper presented a contact state identification method based on boundary model and polyhedron models with topological information. In this method, the contact elements (such as vertex, edge and surface) of virtual prototypes are figured out by comparing connectivity degree of

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 359–369, 2010.

vertex, edge and surface. Then the contact states are determined by testing different potential contact states.

The PHANToM haptic device is an impedance display device with six-DOF (Degrees of Freedom) input and three-DOF output. Most PHANToM-based haptic applications interact with virtual prototypes in a virtual environment based on a proxy point approach [5-8]. A proxy point is the proxy of the endpoint of the physical PHANToM in the virtual environment. By operating the PHANToM desktop device, the users can change the position of the proxy point to simulate manual operations, such as to touch, pick and move virtual prototypes in the virtual environment. However, this proxy point-based interaction method is not suitable for interactive robot arm operations simulation because the workspace of the virtual robot arm cannot be ensured to coincide with that of the PHANToM at all times.

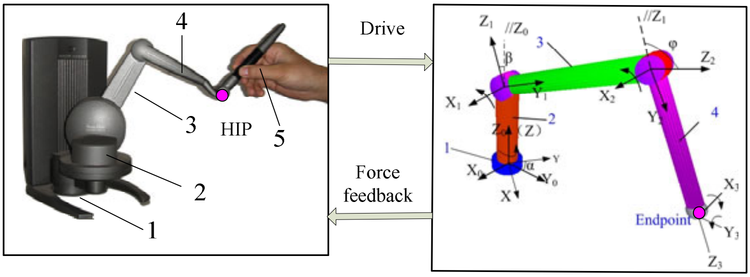
Based on the structural similarity between the PHANToM haptic device and a six-DOF articulated robot arm, this paper designed a six-DOF virtual robot arm that can be driven by the PHANToM device. In order to enable the virtual robot arm to haptically interact with virtual prototypes in the virtual environment, a workspace mapping method based on robot kinematics analysis is proposed. This paper also presented a contact identification method based on boundary model and polyhedron models with topological information. In this method, the contact elements (such as vertex, edge and surface) of virtual prototypes are figured out by comparing connectivity degree of vertex, edge and surface. Then the contact states are determined by numerical calculation.

The reminder of this paper is organized as followings. The section 2 proposes a haptic-based virtual robot arm model. Section 3 presents the contact states identification method based on boundary model and polyhedron models with topological information. Section 4 gives a simple example to verify the presented method. The last section gives a conclusion of this paper.

2 Modeling of a Haptic-Based Virtual Robot Arm

As shown in Figure 1, both the six-DOF robot arm and the PHANToM desktop device have a six-DOF articulated structure. Hence, theoretically, a virtual robot arm model that can simulate the operations of a six-DOF articulated robot arm with different arm lengths and joint mobility can be modeled. This virtual robot arm model is driven by the PHANToM device according to the joint mapping principle, and the virtual force generated on the virtual robot model can be output on the PHANToM device. In practice, the key to modeling the haptic-based virtual robot arm is workspace mapping between the real PHANToM device and the virtual robot arm. The objective of workspace mapping is to map the motion of the PHANToM device to that of the virtual robot arm and the force generated on the virtual robot arm to that of the PHANToM. The main principle and goal of workspace mapping is to align the virtual force with the motion of the virtual robot arm.

In this research, a workspace mapping method between the PHANToM device and the virtual robot arm based on robot kinematics analysis is proposed. This workspace mapping method includes two procedures. The first procedure “motion mapping”,



(a) PHANToM Desktop

(b) Model of 6 DOF virtual robot arm

Fig. 1. Structural similarity between the PHANToM device and six-DOF robot arm

translates the motion of the PHANToM to that of the virtual robot arm; hence the users can operate the PHANToM to drive and move the virtual robot arm in the virtual environment. The second procedure, "force mapping", translates the virtual force generated on the virtual robot arm into that of the PHANToM coordinate system, to be output from the PHANToM device. Through these two mapping procedures, the alignment between the motion and the virtual force of the virtual robot arm can be realized. In other words, the virtual force will drive the endpoint of the virtual robot arm to move, or have the tendency to move along the direction the virtual force.

2.1 Motion Mapping

To ensure that the workspace of the PHANToM coincides with that of the virtual robot arm at all time, motion mapping from the PHANToM to the virtual robot arm should satisfy two basic requirements. Firstly, when driven by the PHANToM, the endpoint of the virtual robot arm can be placed at any point within its workspace, which is termed the reachability of the virtual robot arm. Secondly, any point within the workspace of the PHANToM can only be mapped to one point within the workspace of the virtual robot arm, which is the property of uniqueness. Based on these two requirements, a joint-based motion mapping method is presented. In this method, the current joint angles of the virtual robot arm are obtained by scaling those of the PHANToM according to the specifications of both the PHANToM and the virtual robot arm, and the endpoint position of the PHANToM is estimated using robot forward kinematics.

The specifications of the PHANToM, namely, the joint mobility and the arm length of the PHANToM are constant and known. In this research, the joint mobility of the PHANToM is denoted as (W_0, U_0, V_0), and the arm length of the PHANToM is denoted as (l_0, m_0, n_0). The joint mobility and the arm length of the virtual robot are represented as (W, U, V) and (l, m, n) respectively. If the coordinate systems of the PHANToM and the virtual robot coincide with each other, the current joint angles of the virtual robot arm can be estimated in real-time according to the current joint angles of the PHANToM. In Equation (1), α_0, β_0 and φ_0, which can be accessed using the OpenHaptics API, are the current joint angles of the PHANToM device, and the parameters α, β and φ are the current estimated joint angles of the virtual robot arm.

$$a = \frac{-W}{W_0} a_0, \beta = \frac{U}{U_0} \beta_0, \varphi = -\pi / 2 + \frac{V}{V_0} (\varphi_0 - \beta_0) \tag{1}$$

When the arm lengths and the current joint angles of the virtual robot arm have been obtained, the 3D graph of the virtual robot arm can be drawn in the graphics thread, and the endpoint position P (x,y,z) of the virtual robot arm can be computed according to forward kinematics, which is shown in Equation (2). In the haptic rendering thread, the endpoint of the virtual robot arm model is regarded as the haptic proxy point, which is used to interact with the virtual prototypes in the virtual environment, e.g., to touch, pick and move the virtual prototypes.

$$\left.\begin{array}{l} n\cos\alpha\cos(\beta+\varphi)+m\cos\alpha\cos\beta = x \\ n\sin\alpha\cos(\beta+\varphi)+m\sin\alpha\cos\beta = y \\ n\sin(\beta+\varphi)+m\sin\beta+l = z \end{array}\right\} \tag{2}$$

2.2 Force Mapping

Motion mapping enables a user to control the movement of the virtual robot arm through operating the PHANToM. Hence, various kinds of robot arm interactions, such as touching, picking and moving the virtual prototypes, can be simulated using the endpoint of the virtual robot arm as the haptic proxy. If the virtual robot arm can be endowed with force feedback, the immersiveness of the robot arm interactions can be enhanced. However, there are differences between the joint mobility and the arm length of the PHANToM and the virtual robot arm. Thus, if the virtual forces are exerted directly on the PHANToM device, the haptic interaction will become unstable due to the non-alignment between the motions and the virtual forces at the endpoint of the virtual robot arm. For example, when the endpoint of the virtual robot arm touches a virtual prototype, a virtual force is generated. After this virtual force has been mapped to the PHANToM coordinates, this force will be exerted on the PHANToM device to be displayed to the users. The proper force mapping process in this case should generate an impendence force to prevent the endpoint from touching the surface of the virtual prototype. Hence, a force mapping method based on kinematics analysis is proposed in this research. This method can provide the virtual robot arms with correct force feedback.

As shown in Figure 2, when a virtual force F1 is produced at the endpoint of the virtual robot arm during a virtual interaction, the force mapping method is conducted in the following steps:

(1) Access the current position P_1 of the endpoint of the PHANToM using OpenHaptics API. P_1 is represented in the PHANToM coordinate system.
(2) Within the workspace of the virtual robot arm, select a point Q_2 on the line which direction is the same as that of the virtual force F_1, and passes through the endpoint Q_1 of the virtual robot arm. To maintain the accuracy of feedback force, Q_2 should be sufficiently close to $Q_{1.}$
(3) Suppose the endpoint of the virtual robot arm is placed at point Q_2, use backward kinematics to estimate the current joint angles (α, β, φ) of the virtual robot arm. Next, use Equation (1) to calculate the corresponding joint angles (α_0, β_0 and φ_0) of the PHANToM.

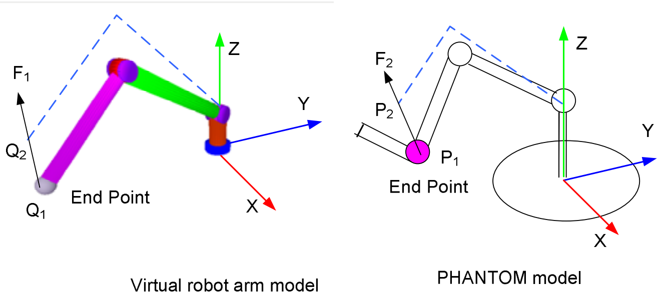
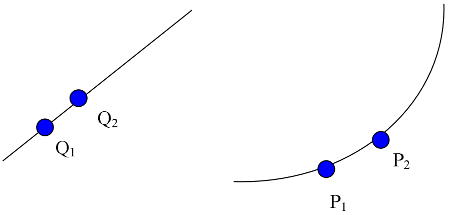


Fig. 2. Force mapping

(4) Use forward kinematics shown in Equation (2) to compute the position P_2 of the endpoint of the PHANToM, where the joint angles of PHANToM are α_0, β_0 and φ_0 respectively.

(5) Use the force F_2 as the feedback force that is about to be exerted on the PHANToM device. The direction of F_2 is from P_1 to P_2, and the magnitude is equal to that of F_1.

It can be seen that above-mentioned method is an approximate method due to the difference in scale of the joint mobility and the arm length. For example, in Figure 3, if the endpoint of the virtual robot arm moves along a line joining Q_1 and Q_2, from Q_1 and Q_2, according to Equation (2), the path of the endpoint of the PHANToM may not be a line, but a curve. However, the line from P_1 to P_2 can also reflect the normal of the curve at point P_1 since the selected point Q_2 is sufficiently close to the point Q_1 and the force mapping process is conducted in a haptic rendering rate, which is 1 *KHz*. Hence, the estimated feedback force is a good estimate of the real value of the actual feedback force.



(a) Virtual robot arm endpoint path (b) PHANToM endpoint path

Fig. 3. Endpoint path of virtual robot arm and PHANToM

3 Contact States Identification

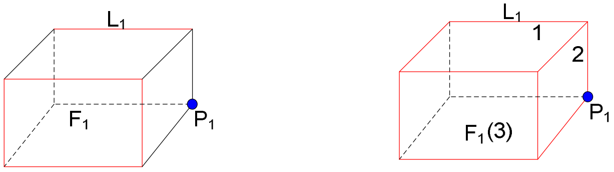
3.1 Related Definitions

Before presenting the presented method, some related definitions needed to be defined and explained.

Definition 1. Connectivity degree (CD): The connectivity degree of vertex, edge, and surface is defined as the number of adjacent surface of respective element in boundary model of part. As shown in Fig. 4 (a), the connectivity degree of vertex P1, edge L1 and surface F1 in cube is 3, 2 and 4 respectively. For virtual prototypes with closed boundary, the connectivity degree of each edge is a fixed value 2, but connectivity degree of both vertex and surface should be calculated according to the boundary of part.

Definition 2. Partial connectivity degree (PCD): The partial connectivity degree of vertex, edge and surface is defined as the number of adjacent surface of respective element in partial boundary model of part. As shown in Fig. 4(b), suppose the partial boundary model of part only includes surface 1, 2 and 3, the partial connectivity degree of P1, L1 and F1 is 1, 1 and 3 respectively. Comparing to CD, the PCD of each edge is not a fixed value. One important thing needed to be pointed out is that the adjacent surfaces of surface F1 include F1 itself if F1 is included into the partial boundary model of part.

Definition 3. Sum of partial connectivity degree (SPCD) of edge: sum of partial connectivity degree of edge is the summary adjacent surface of both edge and vertexes of the edge. As shown in Fig. 4(b), suppose the partial boundary model of part only includes surface 1, 2 and 3, the sum of partial connectivity degree of edge L1 is 2.



(a) (b)

Fig. 4. Related definitions

3.2 Contact States Identification

The contact constraints of virtual prototypes take place among some basic elements such as vertex, edge and surface of virtual prototypes. As shown in Fig. 5, according to difference of contact elements, the contact states can be divided into 5 types, namely point-edge, point-surface, edge-edge, edge-surface and surface-surface contact. The contact constraints can be classified into different priority levels based on the containing relation. The surface-surface contact has the highest priority level, because it includes all other contact constraints. Other contact constraint from highest priority to lowest priority is edge-surface, edge-edge, vertex-surface and vertex-edge.

In virtual environment, contact will takes place between two virtual prototypes when triangle models intersect into or collide with each other. In this paper, the virtual prototypes are represented using both boundary model and polyhedron model with topological information. As shown in Fig. 6, the boundary model includes part level, surface level, edge level and vertex level from top to bottom. Once the boundary model is built, the connectivity degrees of all vertexes, edges and surfaces can be calculated

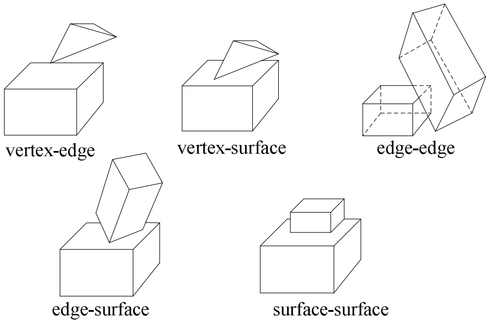


Fig. 5. Type of contact constraints

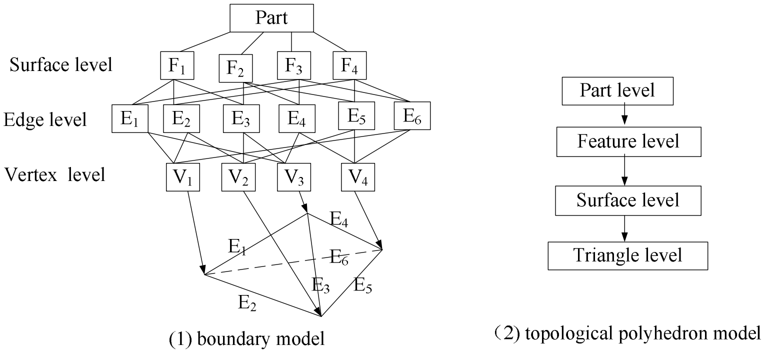


Fig. 6. Model representation of part

and will no longer change during virtual interaction. Using oriented bounding box (OBB) collision detection algorithm [9], the triangle pairs overlaid with each other can be detected. By looking through the topological polyhedron model of part, all surfaces containing triangles overlaid with others can be calculated. These surfaces are named as intersection-surfaces. With the unique coding number of intersection-surface, corresponding surface in boundary model can be fined.

Assume that two virtual parts overlaid with , and the intersection-surface of the first part are denoted as {F_{11}, F_{12} , F_{13}, ……F_{1i}}; the intersection-surfaces of the second part are denoted as {F_{21}, F_{22} , F_{23}, ……F_{2k} }, where i and k represent the number intersection-surface in part 1 and part 2. With the following algorithm, the potential contact elements in two virtual parts can be estimated.

Step 1: Calculating the PCD of the first part. By removing all surfaces except for intersection-surfaces in the boundary model of the first part, a partial boundary model only including intersection-surfaces can be set up. Using aforementioned definition 2 and definition 3, the PCD of the first part and the SPCD of all edges can be calculated.

Step 2: Calculating the PCD of the second part. By removing all surfaces except for intersection-surfaces in the boundary model of the second part, a partial boundary model only including intersection-surfaces can be set up. Using aforementioned definition 2 and definition 3, the PCD of the second part and the SPCD of all edges can be calculated.

Step 3: Estimating potential contact elements of the first part. By looking through all vertexes, edges and surface of the partial boundary model generated at the first step, the potential contact elements of the first part can be determined as follows. If the number of intersection-surfaces is equal to 1, the potential contact element is the intersection-surface. If that is equal to 2, the potential contact element is the common edge of the two intersection-surfaces. If the number of intersection-surfaces is more than two, the following criterions should be taken to estimate the potential elements.

Criterion 1: For each vertex, if the PCD (is calculated using the partial boundary model only including intersection-surfaces) of a vertex is equal to CD of the vertex, which is also equal to the number of intersection-surfaces, then the potential contact element is the vertex.

Criterion 2: For each edge, if the PCD (is also calculated using the partial boundary model only including intersection-surfaces) of edge is equal to 2, the SPCD of the edge is equal to the number of intersection-surfaces, then the potential contact element is the boundary.

Criterion 3: For each surface, if the PCD of the surface is equal to the number of intersection-surfaces, and moreover, the surface itself is an intersection-surface or all its adjacent surfaces are intersection-surface. Then the surface is a potential contact element.

Step 4: Using step 3 to calculate the contact elements of the second part.

Using aforementioned four steps, the potential contact elements of both parts are estimated, and then the contact constraint can be determined. When two parts contact with each other, the contact elements should satisfy some direction and distance constraints. For example, in surface-surface contact, the normal directions of two surfaces are opposite to each other, and the distance between two surfaces is zero. In edge-edge contact, the distance between two edges is zero, and two endpoints of the common perpendicular should lie on the edges. In this paper, different contact constraints are classified into different priority level. The surface-surface contact has the highest priority level, followed by edge-surface, edge-edge, vertex-surface and vertex-edge. Different contact states are set using potential contact elements of both parts by priority from high to low, once the distance and direction between contact elements meet those of some contact state; the contact state between two parts is identified.

4 Experiments

Experiments have been carried out to test the effects of the virtual robot arm model and contact states identification method presented in this paper. Since the force feedback operation can be regarded as a close-loop control system in which the operator is included, the feedback force stability of the human-machine interaction is one of the major evaluating indicators for the workspace mapping method. This paper uses the feedback force stability of a constrained motion to evaluate the proposed workspace mapping method. A good workspace mapping method can constrain the proxy (or the endpoint of the virtual robot arm) to move along the constraint path stably, and the differential coefficient of the feedback force can reflect the constraint states and the stability of the human-machine interaction. So the accuracy and stability of the proposed workspace mapping method are analyzed using the differential coefficient of the feedback force for different types of constraint operations.

The first experiment is a proxy-point constraint. As shown in Fig.7, the endpoint of the virtual robot arm is moved to its destination point by the user with an attractive force from the endpoint to the destination point. The angle β and the distance d are recorded, and the plot is shown in Fig. 8 (a). Similar experiments have also been conducted manually using the workspace mapping method provided by the OpenHaptic toolkit, and the plot is shown in Fig. 8 (b). By comparing the two plots, it can be seen that the convergence of the proposed method have no significant difference from that of the method provided by the OpenHaptic toolkit.

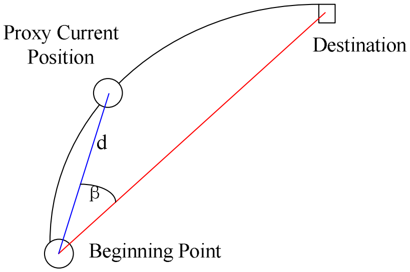


Fig. 7. Experiment of the proxy-point constraint

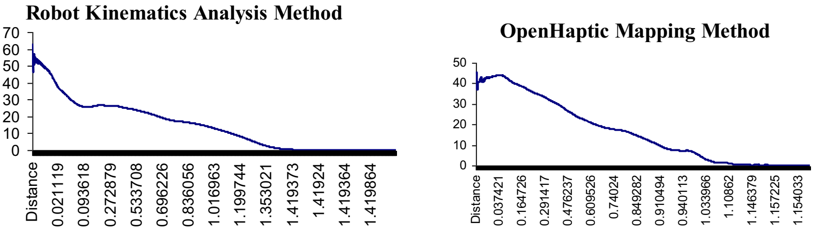


Fig. 8. Plots of (a) the kinematics analysis method, (b) the OpenHaptics method

The second experiment is also a proxy-line constraint. In this experiment, the user leads the endpoint of the robot arm along a path consisting of two lines joining at a point. If the motion of the endpoint is off the path, an attractive force will be generated to force the endpoint to move to the line. In this experiment, the feedback forces and their differential coefficients are recorded. The experiment plots are shown in Fig.9 (a). The differential coefficient of the feedback force plot demonstrates that the operation is stable, and has little vibration. The peak points occur at the joining point of the two lines where the feedback force changes its direction.

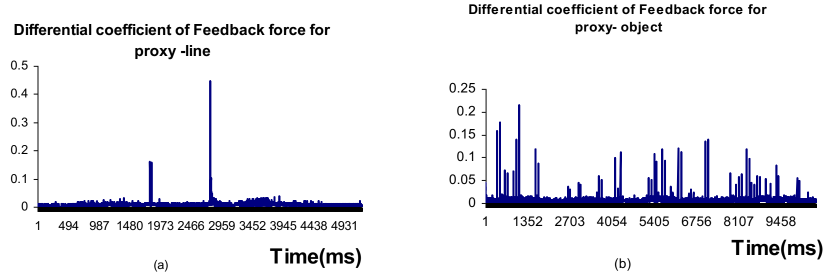


Fig. 9. Experimental plots of proxy-line and proxy-object constraint

The third experiment is a proxy-object constraint. In this experiment, the user operates the PHANToM to touch the surface of a cylinder using the endpoint of the virtual robot arm. When the endpoint penetrates the cylinder, a repulsive force will be generated. The feedback force and its differential coefficient are recorded, and the corresponding plots are shown in Fig. 9(b).

From the experiments conducted, it can be seen that the proposed workspace mapping method based on robot kinematics analysis is effective. Although it is an approximate method, its accuracy and stability are still suitable for virtual robot arm interaction. The alignment between the motion and virtual force of the virtual robot arm can be realized.

To verify the presented contact states identification method, an example is designed in this paper. As shown in Fig. 10, the virtual robot driven by a force-feedback device PHANToM desktop is operating a cube to contact with another cube. The contact state, which is used to calculate feedback force, is dynamically identified using the presented method in this paper.

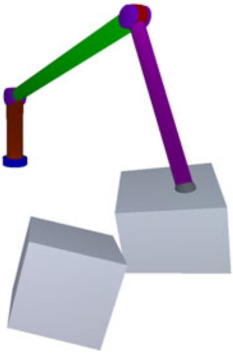


Fig. 10. Contact states identification

5 Conclusions and Future Works

Based on the structural similarity between the PHANToM haptic device and the six-DOF articulated robot arm, a six-DOF virtual robot arm driven by the PHANToM joint is modeled. In order to enable the virtual robot arm to haptically interact with virtual prototypes in a virtual environment, a workspace mapping method based on robot kinematics analysis is proposed. This paper also presented a contact state identification method based on connectivity degree analysis. Compared with some existing methods, the presented method can estimated the potential contact elements of two virtual prototypes by comparing connectivity degree, and then the potential contact state is tested by priority from high to low according to direction and distance constraints of all contact states. The proposed haptic-based virtual robot arm provides a new human-computer interaction method for virtual assembly systems. Potential use of the presented virtual robot arm includes robot path planning and assembly simulation using robot.

Further works will be focus on designing a more complicated example and testing the computational efficiency the contact states identification method presented in this paper. Method for transforming CAD model into model required by the presented should also be explored.

References

1. Harada, K., Kajita, S.: Real time planning of humanoid robot's gait for force controlled manipulation. IEEE2ASME Transactions on Mechatronics 12(1), 53–62 (2007)
2. Zhang, L.X., Xiao, J.: Derivation of Contact States from Geometric Models of Objects. In: IEEE International Symposium on Assembly and Task Planning, vol. 8, pp. 375–380 (1995)
3. Takamatsu, J., Kimura, H., Ikeuchi, K.: Classifying contact states for recognizing human assembly tasks. In: International Conference on Multisensor Fusion and Integration for Intelligent Systems, Taipei, China, pp. 177–182. IEEE Press, Los Alamitos (1999)
4. Fei, Y.Q., Zhao, X.F.: The Prediction Analysis of Assembly Contact States with Errors. Journal of Shanghai Jiaotong University 39(6), 869–872 (2005)
5. Coutee, A.S.: Virtual Assembly and Disassembly Analysis: An Exploration into Virtual Object Interactions and Haptic Feedback, PHD dissertation, Georgia Institute of Technology (2004)
6. Brad, M.H., Judy, M.V.: Desktop haptic virtual assembly using physically based modeling. Virtual Reality 11(4), 207–215 (2007)
7. Iglesias, R., Prada, E., Uribe, A., Garcia-Alonso, A., Casado, S., Gutierrez, T.: Assembly Simulation on Collaborative Haptic Virtual Environments. In: 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp. 241–247 (2007)
8. Van Strijp, C.J., Langen, H.H., Onosato, M.: The application of a haptic interface on microassembly. In: 14th Symposium on Haptics Interfaces for Virtual Environment and Teleoperator Systems, pp. 289–293 (2006)
9. Gottschalk, S., Lin, M.C., Manocha, D.: OBBTree: A Hierarchical structure for rapid interference detection. Computer Graphics 30(8), 171–180 (1996)

Improving Reading Comprehension Using Knowledge Model

Yue Chen

School of Computer, Beijing Institute of Technology,
Beijing, China
chenyue@bit.edu.cn

Abstract. This paper describes the work on reading comprehension system, which accepts arbitrary articles as input and then generates answers according to the questions about the article. A new method to implement reading comprehension system is proposed in this paper. There are three steps in this system. First, the article will be parsed on the paragraph, sentence and phrase level. Second, the information is extracted from all sentences, and then appended to the knowledge model. Finally, the questions are answered by using knowledge model. With the experimental corpus the accuracy rate of knowledge matching is 62.5%, and accuracy rate of question answer is 64.8% with the system knowledge model.

Keywords: Natural Language Processing; reading comprehension; knowledge model; knowledge representation.

1 Introduction

Reading comprehension system is used to understand an article like human. A simple knowledge representation method is proposed where knowledge is stored in knowledge model. Reading comprehension system is designed to test the ability for information extraction from an article. Every article is followed by some questions. Different from the question answering (QA) system, reading comprehension system is consist of article parsing, knowledge building, and answer generation, while QA is consist of question analyses, information retrieval and answer extraction. The study of knowledge model is useful for the development of Natural Language Processing (NLP), and structured information extraction is needed in many areas of NLP.

In QA system, questions can be formalized to query that related to the knowledge structure using the technology of knowledge model [1]. Machine translation (MT) system will benefit from knowledge model because it can generate more easily translated sentences for the input articles [2]. Information Extraction (IE) mainly using template method, while the knowledge model maintains the knowledge node which includes the characteristics of knowledge. Knowledge model can improve the accuracy and robustness of IE system [3]. In information Retrieval (IR), the document stored in database can be changed into knowledge structure with knowledge model technology so that improve the performance as well as precision of search result [4].

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 370–379, 2010.

In Summarization, the center of an article can be extract effectively by understanding the knowledge of article with knowledge model. It can extract a summary from the chapter, paragraph, or sentence level for different needs, and rely on natural language generation to get a more accurate digest [5]. In paraphrasing, since that the process of paraphrasing can be considered as translation between the same languages, the part of natural language generation in knowledge node will improve the accuracy of paraphrasing through making the sentences and articles into knowledge representation [6].

2 Related Work

The research of knowledge model technology originated in the seventies of last century, the introduction of this technology is aimed at trying to make a computer answer a number of questions according to an article. These questions designed manually, and there are clear answers to some questions in the original text, while some questions need knowledge or reasoning in order to get the results from the context. Hirschman [7] studies on reading comprehension system using technology of natural language understanding. MITRE Corporation builds the Remedia corpus for reading comprehension system evaluation. Remedia contains 115 English articles, and divided into different grades according to the degree of difficulty. Each article contain 20 sentences and five questions (the type are who, where, when, what, why) on average, also defines the HumSent accuracy which means the proportion of the number of right answer for the questions. In Remedia corpus named entities, trunk of sentences and pronouns information have been tagged. Sentence is the unit in the corpus with a unique number, and each question has been marked the correct answer sentence.

Ellen Riloff and Michael Thelen [8] develop a rule-based reading comprehension system, and designed a large number of rules of thumb to determine the similarity between a candidate sentence and question. Hwee Tou Ng [9] use features to train a classification model with C4.5 learning algorithm. The feature includes “whether the sentence is the title”, “whether the sentence contains the names of persons, organization name, place name, date, time”, and “the number of words in sentence that matching with the question”.

Dagan and Glickman start the research from the relationship between the content of articles and semantic [10]. Textual entailment is defined as a binary relation between a natural language text T and a hypothesis H. If H can be reason out from T, then T implies H. Recognizing Textual Entailment (RTE) begins in 2005 and has held five sessions so far (RTE-1 ▯ RTE-5). The main method is to calculate the similarity between T and H, such as vocabulary similarity, syntactic similarity and so on [11].

Researchers in Chinese Academy of Sciences develop a QA system on people relationship [12]. Through logical reasoning mechanism, the system output description about people relationship using the knowledge from articles.

Knowledge model techniques is discussed in this paper, approaches on building knowledge structure mainly include knowledge-matching, knowledge markup, semantic reasoning of vocabulary. This article will introduce the main process of reading comprehension system first, and then describe the experiment with results analysis, and finally is the conclusion.

3 Reading Comprehension System

3.1 Knowledge Process

Knowledge model is used to describe the content of article in reading comprehension system. So it is impartment to parse the sentence correctly to build the knowledge model. The data structure of knowledge node is consists of knowledge feature, tag set and knowledge description. Knowledge structure is composed of a collection of knowledge nodes.

Theorem: Suppose F is a set of knowledge features, M is the set of knowledge tags, the knowledge node Ks is a triple Ks=<f, m, G> where:

f is a subset of knowledge features,

m is a subset of knowledge tags,

G is the description of the knowledge node.

Knowledge model is designed to describe information that extract from the article, unlike other system which store data as a record in database, knowledge model likes the brain of human being. The knowledge model is designed using data structure of direct multi-graph. The knowledge model is shown in Fig.1.

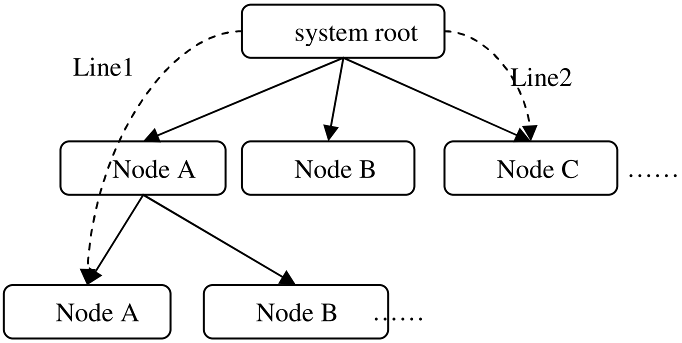


Fig. 1. Knowledge model

As shown in Fig.1. The root node of the graph is the system itself, contains some properties which are used to describe the knowledge model. The child nodes of root are the knowledge nodes, also there are some properties contained in the child nodes. Every child node stores information about other nodes. For example, the line "Line1" which is connecting "system root → Node A → Node A" in the graph means that what is the view for the root that Node A think about itself. And the line "Line2" which is connecting "system root → Node C" means that what is the view of Node C for root.

The system can only understand the sentences that match to knowledge node in knowledge structure, so the knowledge node need to include a description of knowledge feature. Knowledge feature can be characterized as many forms, the particle size of feature can be a single character, or a word. Take knowledge of "COLOR" for example. "COLOR" refers to the property of color of object; the corresponding knowledge features include “color”, “red”, “black”, “blue” and so on. A sentence that response to the age property clearly will be included a few elements of the feature set of “COLOR”. For example, “The color of the flower is red.” contains the feature of “color”, “red”, while the phrase “What is the color of the flower?” contains the characteristics of “color”.

In reading comprehension system we use knowledge node to indicate the properties of object, the relationship between objects, and event. The collection of tags in the knowledge nodes is used to describe objects, attributes, relationships and other information. There are nearly two hundred tags in our system. Take knowledge “HEIGHT” as example. Features in “HEIGHT” is consisting of “<NAME>“, “<HEIGHT>“. If a sentence that describes the height information of a person, the knowledge tag of “HEIGHT” will be used to mark the sentence. For example “小王的身高是180” will be marked as “HEIGHT: <name> 小王 </name> <height> 20 </height>”.

When users need the description information or the answers of questions about knowledge, the system will reply these queries with some sentences. It is necessary to include the information of natural language generation. The information of NLG in knowledge node use manual rules and fixed templates, coupled with knowledge tag, to generate the answer as answer. For example, knowledge description of “HEIGHT” is “<NAME>[<TIME>]身高是<NUMBER>”. Assuming that user query the question of “小王多高?”, the output is “小王身高是180.”

The knowledge structure of “AGE” is shown in Fig. 2.

KNOWLEDGE NODE

Name:HEIGHT

Feature: 身高 NUMBER TIME

Tag: <name></name> <height></height>

Description: <name>[<time>]身高是<num>

Fig. 2. Style of knowledge node

3.2 Knowledge Node

The reading comprehension system analyzes input sentence and generates the set of knowledge nodes. The sentence will be marked by semantic label at first, and then add knowledge that extract from sentence to system model. The key issues of the process as follows:

Decide which semantic label will add to the sentence. After the process of word segmentation the sentence can be considered as a linear sequence or words. Some of the words are the features due to their ability to distinguish the semantic information in the sentence, while there are still some words affect the system uptime because they have no semantic meaning.

More than one result are returned when calculate the knowledge nodes that correspond to the sentence. The sentence commonly corresponds to a knowledge node when dealing with fewer areas; while the sentence corresponds to some knowledge nodes when increase the areas. It is unclearly which is the right knowledge node unless make further judgments.

Parse sentences with knowledge tags. In the process of sentence analysis, the input sentence should be marked according to the knowledge tags in knowledge node if the sentence is related with certain, or uncertain knowledge node.

Therefore, in reading comprehension system it is necessary to take use of features of sentence while calculating the similarity between sentence and knowledge, the methods are as follows.

Extract the keywords in sentence and find out the words that exist in F, the collection of knowledge feature, and remove the words which are not feature as stop words.

Compare the similarity with a threshold, if similarity is greater than threshold, then add the knowledge node into a temporary collection. The sentence will be marked according to the set of knowledge tag in every node in the temporary collection. It will be the right knowledge node that matching the sentence if marked the sentence successfully. If all the knowledge nodes in temporary collection failed to mark on the sentence, that is, the sentence cannot be identified in this system.

In tagging process, different tags are used in different process. The process of information extractation is the same with named entity recognition, such as time, place, institutions, names and other information.

The similarity algorithm is shown as follows.

```
{Input: sentence
Output: Knowledge representation of sentence}
program Knowledge_representation;
  var   sentence: string;
         I, Length : Integer;
         similarity, threshold: float;
         tag : Boolean;
  begin
    {get sentence from input}
    sentence := getInputSentence();
    {get the length of knowledge}
    Length := getKnowledgeLength();
    for I := 1 to Length do
    begin
     {get the similarity between  sentence and
     knowledge node I}
     similarity := getSimilarity(sentence, I);
     if( similarity > threshold ) then
     begin
       tag := tagSentence( S );
       if( tag = TRUE ) then
       begin
         kname := getKnowledgeName( I );
        WriteLn("Sentence belongs to knowledge :",kname);

       end;
     end;
   end.
```

3.3 System Knowledge Model

In reading comprehension system the input texts usually have serval paragraphs. An articles can be decomposed into four levels, including articles, paragraphs, sentences, and phrases. The system starts from the phrase level and generates semantic knowledge of phrase, and then generates the knowledge of the sentence, paragraph, and article using recursive method. When we get one useful data, add it to the system knowledge model using graph structure. Firstly, there is only one graph node, store the information that the system itself, which can be considered as a central node. New knowledge will refresh the graph and the structure of graph makes it easier to process objects with multiple relationships. The algorithm of adding knowledge to system model is as follows.

```
{Input: set of knowledge representation
Output: System knowledge model}
program Knowledge_model;
  var   object, type: string;
         I, Size : Integer;
  begin
    {get the size of the set of knowledge}
    Length := getKnowledgeSize();
    for I := 1 to Size do
    begin
     {get the object from knowledge representation I}
     object := getObject( I );
     {whether the object in system model}
     if( ObjectInModel( object ) = FALSE ) then
     begin
       AddObjectToModel( object );
     end;
     type := getKnowledgeType( I );
     if( type = "ATTRIBUTE" ) then
     begin
       AddAttributeToObject( object );
     end;
     if( type = "RELATIONSHIP" ) then
     begin
       AddRelationshipToObject( object );
     end
     WriteLn("Add object :", object, "to system
model");
   end;
 end.
```

4 Experiment

4.1 Experiment Setting

In reading comprehension system experiment is used to test the ability of understanding an input sentence. We use the PFR corpus for our experiment, the corpus is made of People's daily with segmentation and part of speech (POS) tagging in the first half of 1998. Every word in the corpus has a POS tag. Besides the 26 basic parts of speech marks, the corpus has increased some proper nouns (names, places, organizations, and other proper nouns) for the perspective of applications. Now the number of mark in this corps is more than 40 with linguistic marks. The experiment corpus consists of 400 sentences which are extracted manually from the PFR corpus, while the process of anaphora resolution for pronouns has been achieved taking advantage of corpus tags. Every sentence has been given its knowledge representation. For each sentence, we designed a question that used to ask the relevant knowledge of the contents. The type of knowledge includes object attributes, object relations and description of events. The distribution of knowledge is shown in the table 1.

Table 1. Distribution of knowledge

Type of knowledge	Amount	Percentage
Attribute	240	60%
Relationships	80	20%
Events	80	20%

4.2 Experimental Results and Analysis

In the process of reading comprehension, the first step is knowledge identification, which means that matching the input sentences with knowledge nodes. We take separate experiment using two kinds of feature representation of single character and word. The recognition precision rate indicates the proportion of sentences which can be identified clearly in the system in all input sentences.

$$\text{precision} = \frac{\text{the number of identified sentences}}{\text{th}e\text{ total number of sentences}}\,. \tag{1}$$

The recognition precision of two kinds of knowledge representation in respective application is shown in Table 2 and Fig.3.

Table 2. Precision of sentence recognition

Knowledge	Single character		Word	
	Sentence	percentage	Sentence	Percentage
Attribute	128	53.3%	161	67.1%
Relationships	40	50.0%	42	52.5%
Events	39	48.8%	47	58.6%
Totals	207	51.8%	250	62.5%

Experimental results show that the precision using words as particle size of characteristics is better than using single character, especially in dealing with the sentences belongs to presentation of object attributes. From the kind of knowledge type, the sentences of object attribute have highest precision, while the recognition precision of the sentences regarding object relationships is low.

Some characters may appear in a number of features of knowledge nodes when using single character as the particle size. For example, character "好" appears in knowledge "AGE", "HEIGHT", "SOUND", "COLOR" and other knowledge, which makes the sentence related to many knowledge nodes, so the system need more works to deal with the knowledge tags, and may lead to a recognition failure when the tag does not match with the meaning of the sentence. Table 3 shows the relationship between the characteristics of feature in knowledge node and the number of knowledge nodes which are relevant to the sentence.

Table 3. Sentences and knowledge nodes correlation table

Nodes count	Number of sentence	
	Single character	word
0	0	0
1	31	195
2	33	88
3	80	61
4	78	39
>4	168	17

As shown in Table 3, every input sentence has at least one knowledge node that associated with it. And the sentence can be quickly positioned to the relevant knowledge node when using word as the feature.

After the analysis of sentence, system will add the representation of knowledge to the system knowledge model. We use a number of questions to test the performance of reading comprehension system. Different from most of reading comprehension system, the output of our system is precise answer for each question, while other systems return the original sentence or a sentence number from the article. Because of the sentences that without correct identification, we only use the sentences that identified correct in the system for the experiment. Here precision means the proportion of questions which get the right answer.

Table 4 shows that the precision for input questions in reading comprehension system.

As shown in Table 4, the precision is 64.8% when using words as the particle size of feature, which is higher than the method that using single character. Because of the analysis error of the sentence, system does not reply the correct answer.

Table 4. Precision of question answering

Knowledge	Single character		Word	
	Sentence	percentage	Sentence	Percentage
Attribute	76	59.4%	112	69.6%
Relationships	19	47.5%	22	52.4%
Events	21	53.8%	28	59.6%
Totals	116	56.0%	162	64.8%

The method of using words as the particle size of feature get higher precision than using single character in knowledge matching and question answering in reading comprehension system. The method with characteristic of words can be quickly positioned to the knowledge node. Segmentation error is the main reason which affects the precision. And the step of segmentation requires large-scale lexicon and the corresponding program support to deal with the process which needs segmentation, such as the collection of knowledge features, article understanding and question analysis. Therefore reading comprehension system that using words as feature is typically used for practical applications which require of high precision, and not take the processing speed into account. The reason that impacts the precision of method using single character as the feature is a number of knowledge nodes involved in the calculation. Because the method does not have the process of segmentation, it can be used for applications that require rapid development and high response speed.

5 Conclusions and Future Work

This paper studies the reading comprehension system based on knowledge model. A new method to representation knowledge is proposed in this paper. And the system knowledge model uses a graph structure to store the knowledge that extracted from sentence. PFR corpus is used in out experiment, from which we extracted 400 sentences to form the text corpus. The type of knowledge includes object attributes, relationships, and descriptions of events. With the experimental corpus the precision of knowledge matching is 62.5%, and accuracy rate of question answering is 64.8% with the system knowledge model. Experimental results show that the knowledge model can represent the article in semantic level, and the knowledge structure has strong scalability.

The future research will mainly in the following directions: We will focus on using rational data structures to describe the characteristics of knowledge, not just using the collection of keywords to indicate the knowledge feature. And we will deal with sentence which contains more than one of knowledge, and the relationship between knowledge.

References

1. McKeown, K.R.: Paraphrasing using given and new information in a question-answer system. In: Proc. of the ACL, pp. 67–72. Association for Computational Linguistics, Morristown (1979)
2. Mitamura, T., Nyberg, E.: Automatic rewriting for controlled language translation. In: Proc. of the NLPRS, pp. 1–12 (2001)
3. Shinyama, Y., Sekine, S., Sudo, K.: Automatic paraphrase acquisition from news articles. In: Proc. of the HLT, pp. 40–46. Morgan Kaufmann Publishers Inc., San Francisco (2002)
4. Zukerman, I., Raskutti, B.: Lexical query paraphrasing for document retrieval. In: Proc. of the COLING, pp. 1–7. Association for Computational Linguistics, Morristown (2002)
5. McKeown, K.R., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J.L., Nenkova, A., Sable, C., Schiffman, B., Sigelman, S.: Tracking and summarizing news on a daily basis with Columbia's newsblaster. In: Proc. of the HLT, pp. 280–285. Morgan Kaufmann Publishers Inc., San Francisco (2002)

6. Iordanskaja, L., Kittredge, R., Polguère, A.: Lexical selection and paraphrase in a meaning-text generation model. In: Paris, C.L., Swartout, W.R., Mann, W.C. (eds.) Natural Language Generation in Artificial Intelligence and Computational Linguistics, pp. 293–312 (1991)
7. Lynette, H., Marc, L., Eric, B., John, D.: Burger. Deep Read: A Reading Comprehension System. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pp. 325–332 (1999)
8. Ellen, R., Michael, T.: A Rule-based Question Answering System for Reading Comprehension Tests. In: Proceedings of ANL P/NAACL 2000 Workshop on Reading Comprehension Tests as Evaluation for computer-Based Language Understanding Systems, pp. 13–19 (2000)
9. Hwee, T.N., Leong, H.T., Jennifer, L.P.K.: A Machine Learning Approach to Answering Questions for Reading Comprehension Tests. In: Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 124–132 (2000)
10. Dagan, I., Glickman, U.: Probabilistic textual entailment: Generic applied modeling of language variability. In: Proc. of the PASCAL (2004)
11. Ferrandez, O., Micol, D., Munoz, R., Palomar, M.: A perspective-based approach for solving textual entailment recognition. In: Proc. of the Workshop on Textual Entailment and Paraphrasing, pp. 66–71. Association for Computational Linguistics, Morristown (2007)
12. Wang, S.X., Liu, Q., Bai, S.: An expert system about human relationship question answering. Journal of Guangxi Normal University (Natural Science) 21(1), 31–36 (2003)

Modular Robot Path Planning Using Genetic Algorithm Based on Gene Pool

Huaming Zhong[1], Zhenhua Li[1,2], Hao Zhang[1], Chao Yu[1], and Ni Li[1]

[1] School of Computer Science, China University of Geosciences, Wuhan 430074, China
[2] State Key Laboratory of Geological Processes and Mineral Resources,
China University of Geosciences, Wuhan 430074, China
zhli@cug.edu.cn

Abstract. As a new generation of robotics, a modular robot is flexible enough to achieve self-replication by attaching a new modular, or perform self-assembly by transferring into different shapes. However, the path planning for modular robots, the fundamental function is seldom studied until now. In this paper, we improve the path schedule method of Molecubes, by designing a gene pool, to speed the convergence and avoid the uncertain of the original genetic algorithm (GA). Experiments show that the gene-pool based GA outperforms the old one in both success rate and speed in planning the long path.

Keywords: Modular robots, genetic algorithm, gene pool, path planning.

1 Introduction

The birth of the self-replication robots [3] in Cornell University in 2005 marks modular robot had entered a new field. Following that, the United Kingdom, Japan, Korea and other countries also raised the research results of self-replication robots or self-assembly robots. European scientists started a symbiotic robot project in 2008, focusing on capacity development of the evolution of symbiotic organism robots. It was to design and develop the ultra-large-scale popularization of the robot system and a high degree of autonomy to enable them to adapt to the dynamic environment.

The development of science and technology provides for the robot with hardware and software, and the evolutionary algorithm for the intelligent and automatics. As it is known, while solving complicated problem, the evolution algorithm needs very long time; what's more, the module robot's state of motion is indefinite, if we used the evolutionary algorithms to plan the path for the modular robot, we have to wait a long time to get the result, even that as the quantity of the module increased and the structure of the robot was once complicated, the modular robots control would become very complicated, so we have to look for a good method to solve this problem.

Molecubes [1][13]is built on AGEIA PhysX and OGRE, AGEIA PhysX was selected as the physical simulator, and OGRE was employed as the graphics rendering engine. Both libraries are free for non-commercial use. But the Molecubes is not all suitable for this experiment; so we have to revise the platform to make it suitable for the research.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 380–389, 2010.

In this article, in order to solve these problems, we will use grid to program the sport orbit of the robot at first, then use evolution algorithm to train the gene of the direction which the robot would come , and build the gene pool, finally I will use the gene pool to control the movement of the modular robot. At the end of this article, we will revise the molecubes platform then do the simulation and compare the results, according to the path planning with the gene pool, and the path planning without it; the gene pool is competitive one.

2 Modular Robot Motion Control and Path Planning

2.1 Gene Design

The movement of a single cube is very simple as rotating clockwise and counter-clockwise and halt, so I assigned 4 commands for a single Molecube for a single time interval [5]:

0: remain the current states

1: halt

2: rotate clockwise

3: rotate counter clockwise

When arrange these four commands in a sequence in order, the movement of a single cube can be descript as following:

321010320

For a sequence of commands which describes the movement of a single cube, I call it GENE.

By observing walking of human, I realized that it is actually repeating some kind of simple cycle movement. However, before people start walking, some special movement is necessary like standing up. For this reason, I added several digits of setup commands before the cycle commands.

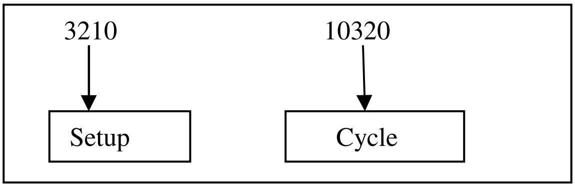


Fig. 1. Components of a Gene

After the gene has been designed, as program running, the codes in the gene are executed from the left to the right. Once all codes has been run and left some execution time, the program will execute the cycle commends again. For example, in a runtime of 20 second with 1 second time intervals. For that cube assigned this gene to, it will behave as 3210 10320 10320 10320 1, as show on table 1.

In the module system because each gene only actuates a module, therefore, if in the multi-module system, needs to design many genes, those genes form a genome. The movement of the multi-module system and the single module system are the same in the similitude.

Table 1. Execution of a single gene

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C	3	2	1	0	1	0	3	2	0	1	0	3	2	0	1	0
MS	cc l	cl	H				cc l	cl		H		cc l	cl		H	

T is time,C is command,MS is move state,ccl is counter clockwise cl is clockwise, H is halt.

2.2 Fitness Function and Fitness Value Design

In this experiment, we want the robot to reach the designated place. So at the beginning of the algorithm, choose the direction in which the robot would come. In order to enable the robot move toward this direction, we choose one point in this direction as P, the coordinate of P is set as (x_0, z_0) , when the experiment begins ,the robot gets the present coordinate of the modular robot p_i (x_i, z_i) and calculate the distance between P and p_i .

$$d(x,z) = \sqrt{(x_i - x_0)^2 + (z_i - z_0)^2} \tag{1}$$

In order to get minimum displacement as the fitness value, we define the fitness function as:

$$fit(f) = \max - d; \max = 20; \tag{2}$$

For each genome, the evolution program will run it run it and get the displacement of the module from the point I choose at the beginning and the point at the end of experiment time, if the displacement is minimum under some genome's actuation, and the fitness is maximal, then this genome is the superior solution and has the very big opportunity to inherit to the offspring generation.

2.3 Cross and Mutation

In this experiment, the genetic factors we used come with the platform, after the program executes all genomes in the parent generation and gets the fitness for them. Then, genomes will be passed down with four kinds of randomly happened mutation, including big-cross, small-cross, single digit and length mutation. To guarantee the genome with the best fitness can be passed down, the program copies the best genome to the first two individuals in the offspring generation. For each of rest individuals, the program selects a parent individual by fitness proportional selection. If mutation randomly not happens, the genome of this selected parent simply copies its genome to an offspring individual. If mutation chosen to happen, a second parent is selected for the use of mutation. The following the flow chart as show in fig.2 clearly shows the genomes is passed down from parent generation to offspring generation.

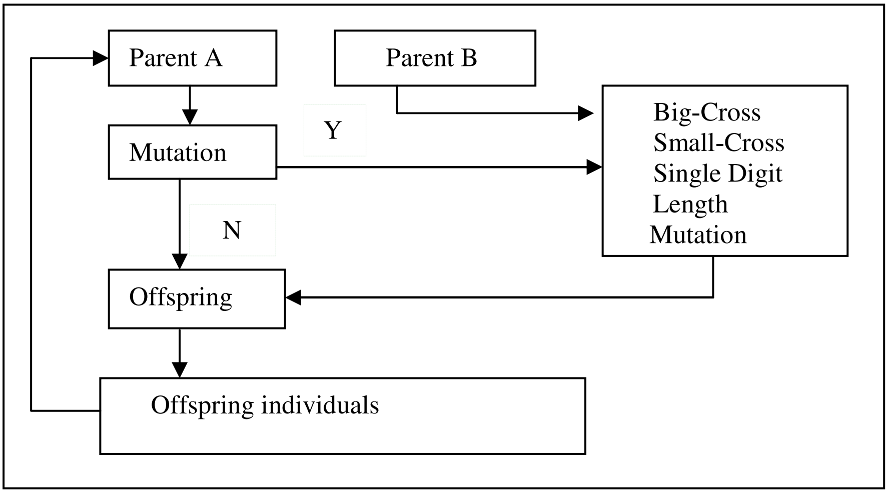


Fig. 2. Inherits Flow

I have mentioned four kinds of randomly happened mutation above, which were described in article Movement Evolution of Molecubes under Simulation, Computational Synthesis Laboratory, Hang Li [5].

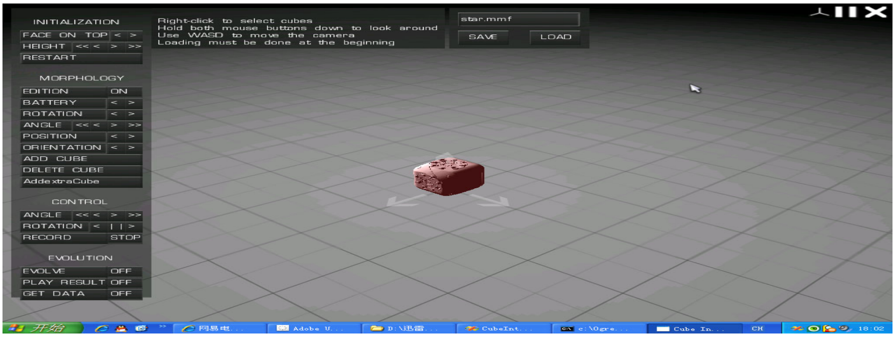
2.4 Parameter Setting

As mentioned above, we have to declare five parameters for the genetic factors in this experiment. pr as possibility to regenerate a new genome, and its value is 0.1; Pc as Small-Cross possibility, and its value is 0.7; Pbc as Big-Cross possibility, and its value is 0.7; Pm as single digit mutation possibility, and its value is 0.05; Pi as length mutation possibility, and its value is 0.05.

3 The Gene Pool

After the gene was designed, we have to build the gene pool which we use to control the robots. Firstly, we want to introduce the molecubes platform, as you see in fig.3, Molecubes are modular devices that can be attached one to another to form a larger robot. A Molecube in its initial state is simply a cube, but each half of the Molecube is capable of rotating along the diagonal axis (from a corner to the opposite corner). Thus, by rotating 120 degrees, the faces of the Molecubes change relative positions, while the Molecube as a whole retains its cube shape. In this way, a system of Molecubes is capable of actuated movement, giving the overall system the ability to perform certain tasks.

As can be seen from Fig.4, suppose the robot locates in the coordinate (0,0) , has 8 directions of movement, encoded as integral values in{1,2,3,4,5,6,7,8}. But in order to simplify control, we choose only four directions as 2, 4, 6, 8; modules are able to



Right-click to select cubes
Hold both mouse buttons down to look around
Use WASD to move the camera
Loading must be done at the beginning

star.mmf

EDITION ON
RECORD STOP
EVOLVE OFF
PLAY RESULT OFF
GET DATA OFF

Fig. 3. Molecubes Interface

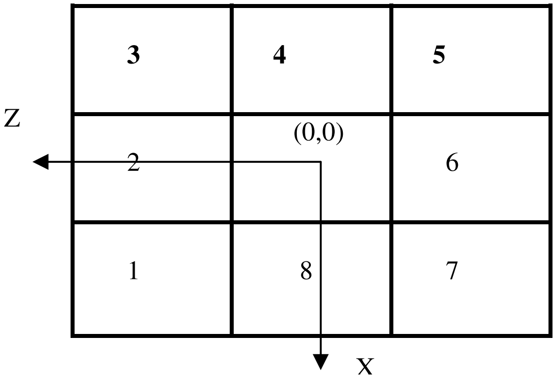


Fig. 4. The Control Grid

move only in the four directions: east, north, west, and south, and the robot may arrive at grid 2, 4, 6, 8 in the first step. If the robot needs to move to grid 1, the robot firstly moves in west direction, arriving at grid 2, then in the south direction, arriving at grid 1; or the first step walks to the south, arrives at grid 8, then the second step walks to the west, arrives at grid 1.

In this experiment, suppose the module in the middle of the modular robot is the first one, stored in halfCubes as halfCubes[0].

3.1 Building

For each direction we run the gene algorithm to get the gene string. The evolution program will run it and get the displacement of the Molecubes from the point I choose at the beginning and the current point. If the displacement is minimum under some genome's actuation, to the system, this genome is the superior solution, and has the very big opportunity to inherit to the offspring generation. In this process, we store the module number and commands. Consider the modular robot as a point and the gravity of the robot is halfCubes[0] locate at the point(0,0), suppose the robot has to move from (0, 0) to (0, 1), in that case, only need to let halfCubes[0] move from (0,0) to (0,1).

After run the gene algorithm, we store the gene string like that:

3
2
0011022031220021020001022001220011020001220021120001020001023011120001321021220001020011023011120001321021220001020011023011120001321021…

Where 3 in the first line is the size of the robot, 2 in the second line is the direction. The first 0 in the third line is the command, The second one is the subscript of the module stored in the halfCubes, the robot will run like that halfCubes[0] driven by command 0, and halfCubes[1] driven by command 1, and halfCubes[0] driven by command 2 and so on.

As described above, the modular robot is considered as a point; however, the robot has its own length. If the module in the front of the robot has to move to p(x, z), only need to let halfCubes[0] move to (x-d, z) or (x, z-d),where d is the distance between the module and halfCubes[0], and then rotating around halfCubes[0], until the module in the front arrive at p.

According to the description above, we need to create 5 gene strings for each size of the modular robot, the four directions: east, north, west, south and the rotating gene strings. For each direction the gene string is not unique, there may be many gene strings, we select the fast one.

3.2 Use the Gene Pool

After the gene pool has built, we use the gene pool to control the movement of the robot. Suppose the robot has to move from p (x_0 , z_0) to p_1 (x_1 , z_1), the flow like that:

```
 if ( $x_0 < x_1$ ) the robot uses the direction 8 gene, it
  will move in the north direction.

else if ( $x_0 > x_1$ ) the robot would pick up the direction 4
gene, it will move in the south direction.

 else if ( $x_0 == x_1$ && $z_0$ ! = $z_1$ )

{

 if ( $z_0 < z_1$ ) the robot uses the direction 2 gene, it
 will move in the west direction.

else if ( $z_0 > z_1$ ) the robot would pick up the direction
6 gene, it will move in the east direction.

}

else

{

for (int i=0; i<halfCubes.size();i++)

halfCubes[i].halt();

}
```

4 Simulation

As a result of the uncertainty gene algorithms and the flexibility modular robot, we have to do something else to control the movement of the robot. In order to simplify the experiment, after the robot run a gene in the gene pool, we reload the modular robot at the current point, to reset the initial state of the robot.

During the simulation, we design two groups of experiments, the robot is required to move to a given point (10,-10) from the initial point (0, 0), one uses the gene pool, and the other one just uses the general genetic algorithm. Each experiment we built three-module chain robot (the modular robot is shown in fig.5) and run ten times to get the success rate.

When uses the gene pool, the robot can fast reach the given point, it costs 237s, the result is shown in table 2, and the path planning is shown in fig.6; but the other one without the gene pool, the robot move slowly, it would cost a long time to get to the given point, the best time is shown on table 3. (This experiment runs on windows XP sp3, Pentium(R) CPU 2.93GHZ, 768M memory.)

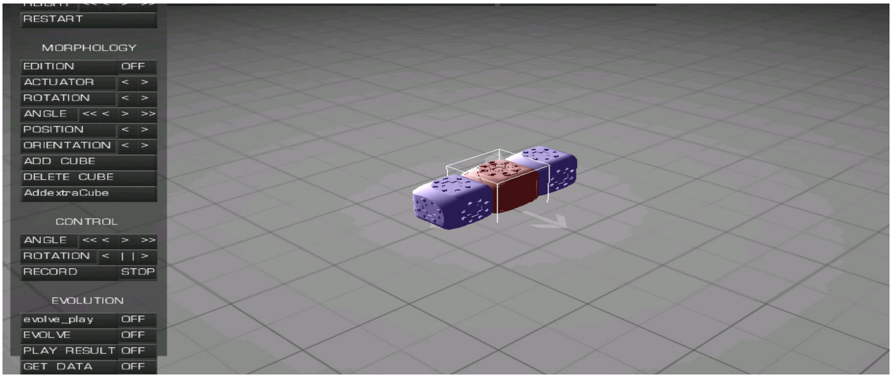


Fig. 5. The Modular Robot

Table 2. The Path Planning with Gene Pool

N	Time (s)	X	Z
1	20	1.84864	-0.253437
2	40	4.42272	0.633729
3	60	6.81199	1.50699
4	80	9.41975	2.2928
5	85	10.0005	2.44869
6	115	9.70217	-1.7377
7	145	10.3311	-3.55071
8	175	9.9396	-5.06618
9	205	10.1708	-7.51781
10	237	9.9735	-10.0003

Table 3. The Path Planning without Gene Pool

N	Time (s)	X	Z
1	1000	0.0779874	-0.342252
2	2000	-0.179848	-1.75601
3	3000	1.33099	0.200457
4	4000	0.473854	-0.148396
5	5000	2.35368	-2.03296
6	6000	2.61556	-3.18925
7	7000	2.66748	-0.612637
8	8000	3.12336	-1.32183
9	9000	2.8942	-2.10905
10	10000	3.41054	-3.5706
11	10500	4.93688	-3.44163
12	11000	7.04466	-5.84199
13	15000	10.9128	-9.00608
14	15193	10.0403	-9.95004

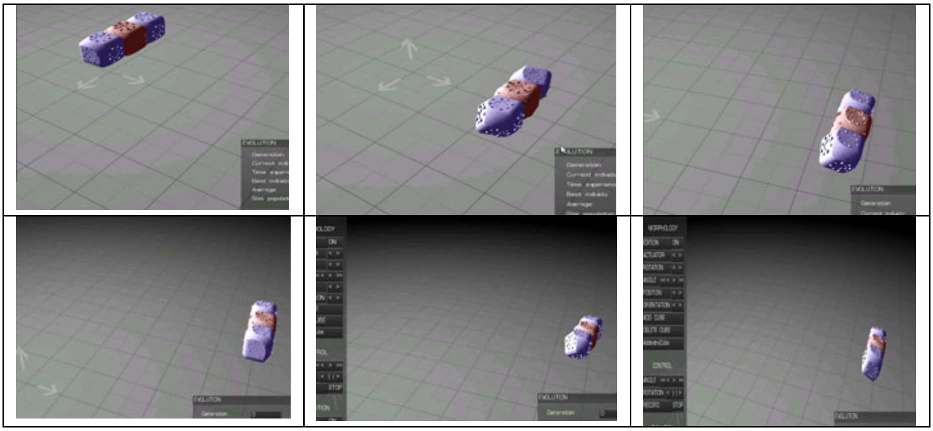


Fig. 6. Path Planning for Modular Robot Use Gene Pool

Table 4. The Success Rate

Points	Success Rate (%)	
	Gene pool based GA	Simple GA
(0,1)	100	100
(1,0)	100	100
(-1,0)	100	100
(0,-1)	100	100
(5,-5)	100	60
(10,-10)	100	20

From table 4 we know that, when the point is near the origin point, both of two get the same success rate, they both perform well. However, when the point is far from the origin point, the gene pool get higher success rate.

According to the results, within the specified time, the robot uses the gene pool can get better performance. But the path planning used simple gene algorithm is flexible, it can accomplish more.

5 Conclusion

In experiments, when the robot stays near the origin point, both the gene pool GA and the simple GA perform same well; even that, since the simple GA is flexible, it would runs better than the former, but when the robot stays far from the origin point the gene pool gets better results. The reason is that there may be some deviations during the movement, when the robot near the given point, the robot need to call the gene strings of different directions many times, it would cost a long time, at that moment the simply GA is better .

However, what we do now is three modular robots, as the number of module increasing, the robot control becomes more complex, and the robot's structures are too complicated to get the gene string. In the future, still, we will study the quick path planning of more modular robotics and adaptive morphology.

Acknowledgement

We wish to acknowledge thee support of the Fundamental Research Funds for the Central Universities, China University of Geosciences(Wuhan)(No. CUG090109) and State Key Laboratory of Geological Processes and Mineral Resource (No. GPMR200715).

References

1. Molecubes_Software, http://128.253.249.235/cubes/index.php?title=Molecubes_Software
2. Evolutionary_robotics, http://en.wikipedia.org/wiki/Evolutionary_robotics
3. Zykov, V., Mytilinaios, E., Adams, B., Lipson, H.: Self-reproducing machines. Nature 435(7038), 163–164 (2005)
4. Zykov, V., Mytilinaios, E., Desnoyer, M., Lipson, H.: Evolved and Designed Self-Reproducing Modular Robotics. IEEE Transactions on Robotics 23(2), 308–319 (2007)
5. Li, H.: Movement Evolution of Molecubes under Simulation. Computational Synthesis Laboratory, Cornell University
6. Lipson, H.: Evolutionary Robotics and Open-Ended Design Automation, pp. 12–20. Cornell University, Ithica
7. Yim, M., Roufas, K., Duff, D., Zhang, Y., Eldershaw, C., Homans, S.B.: Modular reconfigurable robots in space applications. Auton. Robots 14(2-3), 225–237 (2003)
8. Yim, M., Zhang, Y., Duff, D.: Modular robots. IEEE Spectrum 39(2), 30–34 (2002)
9. Whitesides, G.M., Grzybowski, B.: Self-assembly at all scales. Science 295(5564), 2418–2421 (2002)

10. Nielsen, J.: User Configurable Modular Robotics Design and Use. The Maersk Mc-Kinney Moller Institute University of Southern Denmark, pp. 128–155 (2008)
11. Groß, R., Bonani, M., Mondada, F., Dorigo, M.: Autonomous Self-Assembly in Swarm-Bots. IEEE Transactions on Robotics 22(6), 1115–1128 (2006)
12. Griffith, S., Goldwater, D., Jacobson, J.M.: Self-replication from random parts. Nature 437(7059), 636 (2005)
13. Chen, A.: Modeling Molecubes with AGEIA. PhysX (2007)

Pathfinder Based on Simulated Annealing for Solving Placement and Routing Problem

Zhangyi Yu[1], Sanyou Zeng[1], Yan Guo[1], Nannan Hu[1], and Liguo Song[2]

[1] School of Computer Science, China University of Geosciences, China
430074 Wuhan, China
[2] The Microelectronics Technology Institute of Beijing, China
100076 Beijing, China
zhangyi_yu2010@163.com, sanyou-zeng@263.net,
guoyanwuhan@yahoo.com.cn,
nannan_Hu@163.com, LI-guo_song@163.com

Abstract. The placement and routing is a hot topic in Evolvable Hardware, the work of the placement and routing in this paper is as follows: (1) Combining with FPGA's and the VPR's "placement and routing" two-stage optimization model, the designed "random placement and optimal routing" model meets the tasks of placement and routing in PEA that is N*N array of PE. (2)The designed pathfinder based on simulated annealing is a solution for the "placement and optimal routing" cycle model, it uses obstacle avoidance to solve the placement and routing problem. (3)By numerical test experiments, we verify that the success rate of pathfinder based on simulated annealing is higher than the commonly used "depth-first search" under the PEA framework.

Keywords: Evolvable hardware; Obstacle avoidance; simulated annealing; Pathfinder; Placement and routing.

1 Introduction

In the circuit design process, we often relate to how the resource block allocate (i.e. placement) and the routing resource allocate (i.e. wiring), it makes for that a class of large-scale circuits can achieve efficient use of routing resources rate. Generated in the placement of the case, if the wiring resources of chip in certain areas are too few, many of the wiring circuit may fail. However, if the wiring resources are too rich then it may result in waste of resources in routing.

Conditions in the search for a priori knowledge that is not dependent and manual intervention, Hardware evolution [1] achieves the adaptive function with expected circuit and system architecture through the simulated evolution. [2]

In order to achieve coarse-grained circuit design to solve the needs of project development, we designed the PEA array structure [3], which is composed of the N * N PE, each PE in the array is a single chip, it can run a program or a subtasks, the segment of program is reconfigurable and exclusive, each PE can only assign a program or sub-task.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 390–401, 2010.

PE uses the NN connection (i.e. "grid" type connections, each PE connects to it's around PE) to achieve high-speed transfer of data between PE. The connection resources are reconfigurable and are exclusive between the PE; it has a bus connection of interconnection in two different directions, that each bus can only be used once. For example, when N = 4, we give the 4 * 4 PEA array model is as follows:

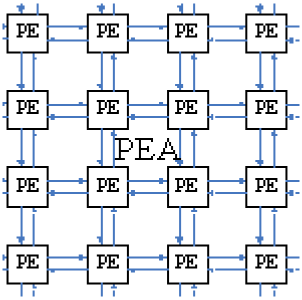


Fig. 1. The 4 * 4 structure of PEA

In the PEA, the connection resources between PE have 48 routing resources. PE is a series of internal configurable directives and functions which consists of operators, in accordance with the command of configurable instructions; it selects the appropriate functional operator.

Combination of these computing unit functions, they have two input / output ports. Although you can set up more, but two is more reasonable. The external part of PE is achieved data path routing through the MUX.

Because of this structure relatived to the FPGA, the resources are fewer. In the placement and routing process, it is more prone to fail. So in this case, we propose pathfinder based on simulated annealing algorithm to solve the placement and routing problem.

2 Placement and Routing Model

2.1 "Placement + Optimal Routing" Cycle Model

In the FPGA, due to relatively available resources are numerous, so it mainly uses the "optimal placement and optimal routing," two-stage optimization model, the process is as follows:

This placement and route model includes the two stages of optimal placement and optimal routing. Firstly it cycles to produce the placement plan and evaluates the placement until it meets the termination conditions of placement, so it gets the best placement program. Then according to the optimal placement program, it gets the corresponding collection of lines network, and then it cycles to generate routing options and assess routing options, the process continues until it meets the termination conditions and it exits routing, so the optimal placement of the program gets the optimal routing scheme. Through two parallel loop structures, it generates optimal plan.

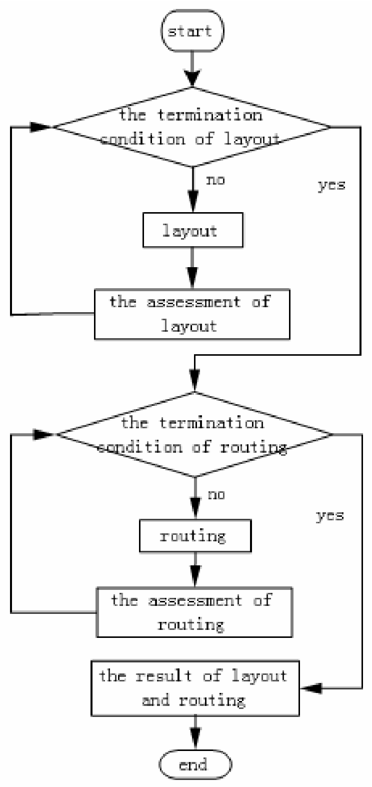

start
the termination condition of layout
no
yes
layout
the assessment of layout
the termination condition of routing
no
yes
routing
the assessment of routing
the result of layout and routing
end

Fig. 2. "Optimal placement and optimal routing" two-stage optimization model

This method is characterized by estimating in advance the performance of the placement. As the actual value of placement can only be determined after the routing in the model and the two parts are separated, therefore, it often need the performances of pre-placement estimation to get the optimal placement based on the objective function. Usually simulated annealing algorithm or genetic algorithm is used to find the optimal placement and then it uses pathfinder algorithm to find the optimal routing.

However, the success rate on the placement and route, that is, the quality also depends on the order of line wiring net, if the order of distribution line network of placement is reasonable and the use of resources is less, then the chances of success of the placement and routing is large, otherwise, the remaining route of lines network will affect the overall, the successful probability will be small.

In this article, because of the needs of coarse-grained circuit design, the PEA framework we designed has the following limitations:

1. The designed PEA array is small, relatively few resources are available.
2. PE resources and routing resources are exclusive (if run out of time no longer used)

3. Placement and routing of real-time requirements (i.e. within the given time to satisfy the condition of the placement and routing programs)

Therefore, the optimal routing of objective function based on estimation may not succeed. Therefore, based on the "optimal placement and optimal route," the basis of two-stage evaluation model, we propose a placement and routing program, which satisfies the framework for the circuit with our resources. It is “placement and optimal routing " cycle model, in which we mainly focus on the success rate of placement and routing and real-time requirements, the flow chart is as follows:

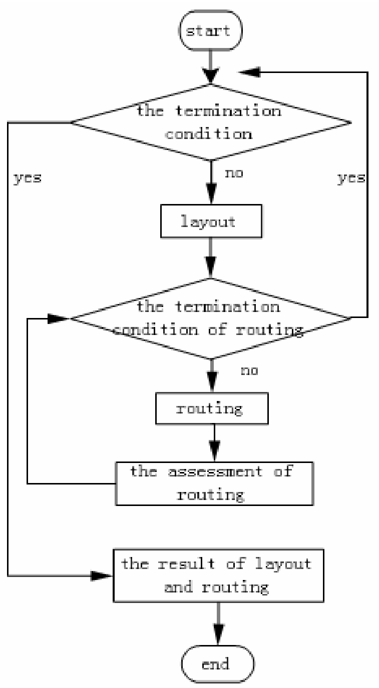

start
the termination condition
yes
no
yes
layout
the termination condition of routing
no
routing
the assessment of routing
the result of layout and routing
end

Fig. 3. "Placement and optimal routing" cycle model

This model is also a cycle route process. First, it uses simulated annealing to create a placement program, then it gets a set of corresponding line network under the placement and recycling uses the pathfinder algorithm to produce routing plan, make routing evaluation, until it meets the termination conditions, then it exits. So it gets the optimal corresponding routing program of the placement. This process is ongoing, when the out termination condition is met; it gets the optimal placement and routing program. This model adopts a double-nested loop method, shown in Figure 3.

The characters of this program is as follows: first of all ,it meets the requirements of real time (i.e., gives time of the first successful route),then for further optimization on the basis, it makes the final placement and routing programs to get to use the least resources.

3 Placement and Routing Program

3.1 The Traditional Route Programs (That Is, Depth-First Search)

In the N*N array of PEA given by us, the method commonly used for placement and routing program is "depth-first search-and-route" ,that is "to see step by step". According to the task requirements of the route network given, it starts to search the available PE resources and routing resources around the node, select a PE resource in all the searched resources, and then put it in the queue. And then to center of the second PE resource, it once again looks for the available PE resources and routes resources around it. If it could not find any related resources, then go back to the previous node, and then find the available resources (except the node) around the previous node. In the ongoing back and into the queue, if it finds the number of the PE block to meet the number of task, it will directly use the final PE and the output of PE to find a path with the Dijkstra algorithm, then the algorithm ends; if it does not find the number of the PE block to meet the number of task, then it continues to go back, when back to the first element in queue, it can not find the available resources, that route fails. Corresponding pseudo code is as follows:

```
FOR i=1 TO the number of nets &&flag=0
{
   int start=the input of nets[i]  ;
   int end=the output of nets[i]  ;
   empty the queue of points ;
   Put the start into the queue;
   int point_count=1; int mark=0;
   While (true)
   {
    Search the available PE and routing sources
    around the points [mark];
     if (find the available sources)
    {
      Randomly select one of the available resources;
      mark=points_count;
      points_count=points_count+1;
       if(the numbers of available sources==the task-
      Num of nets[i])
       Break;
   }
else
{
  if (not go back to the head of queue, that is mark=0)
  {
   points_count=points_count-1;
   mark=points_count-1;
  }
  else
  {
     flag=1; Break;
  }
}
}//for (while)
if (flag==1)
{
   Break;
}
else
{
 Find the way between be the end of queue elements of
 points and the end with Dijkstra.
}
}//for(i)
```

This method is simple and easy to understand, it is generally used method, because of pseudo-code, it randomly select the available resources around each node at each time, so it is easy to form a loop, and because all resources are exclusive, making a lot of resources that have been occupied and wasted and making the success rate of placement and routing is relatively low. It will have great impact on the remaining lines of route network, therefore the success rate of placement and routing of multi-mission line network is very low, and many cases will not find the route to meet the requirements of the program.

3.2 Pathfinder Based on Simulated Annealing Algorithm

Based on the N * N of the PE array structure consisting of PEA that we propose, and because of the route in terms of resources and many restrictions, so we study in this paper mainly for "placement and optimal routing” cycle model in the "Placement and routing model”. Based on this model, we propose pathfinder based on simulated annealing algorithm to solve the placement and routing problem of this structure. First, it creates a placement using simulated annealing program, and then it gets a set of corresponding line network under the placement. It recycling uses pathfinder to produce routing plans and make routing evaluation until it meets the termination conditions then it exits. It gets the optimal corresponding routing program of the placement program. This process is ongoing, when the out termination condition is met; it gets the optimal placement and routing program. This model adopts a double-nested loop method.

3.2.1 Simulated Annealing Placement Algorithm

The placement algorithm is to determine the position of each logical unit block that required achieving circuit functions in the PEA. Its optimal goal is to connect the logical unit block near, to extend to reduce the required routing resources earnestly.

The placement mainly has three devices, they are the minimum cut placement device (based on the classification) [4], placement analysis device (usually also need local iteration) [5] and the placement of devices based on simulated annealing. In the above three devices in the placement, the placement of devices based on simulated annealing has added new goals or constraints, it is easier than the minimum cut placement device and the analytical device. The major study of the placement algorithm based on Simulated Annealing following.

Schematic diagram of the algorithm shown in Figure 4:

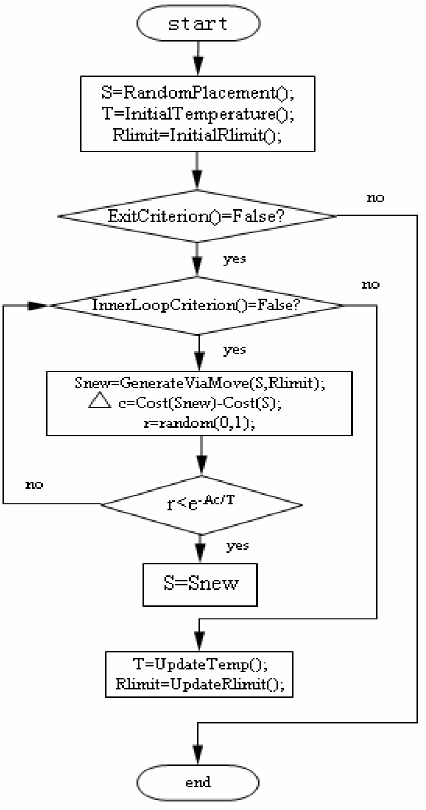


Fig. 4. Devices of placement based on simulated annealing typical

Be noted that in this paper, in use of simulated annealing process, the algorithm's parameters are difficult to control; the main issues are as follows:

(1) Initial parameter (T) and terminal parameter (Tmin) setting, which in fact can be concluded in one point about initial parameter setting. If the initial parameter is high, and times of circulating outsides is high accordingly, as a result, the probability to find the best solution is larger. However, it is at the expense of large computing time.

(2) The speed of annealing setting, which means setting with m. This is the point about when to stop circulating insides, which is necessary to sufficient annealing at some T. The idea is similar to Gibbs Algorithm which makes Markov reach stability. In practical problems, it is determined through repeated simulation.

(3) Management of T, the extent of reduction every time. The conventional methods are as follows: Offset down, Proportional decrease, Descent method based on distance parameters and so on. According to the characters of PEA Array and success rate and the real-time of completing route, this paper proposes two function.

Real-time cooling function:

$$Tnew = Told * e^{\frac{-\lambda * Told}{\sigma}} \tag{1}$$

Increase the success rate and the cooling circuit complexity of optimization functions:

$$Tnew = [1 - \frac{\alpha - 0.44}{40}] * Told \tag{2}$$

Where λ usually adapts to 0.7, and σ means standard deviation to accepted move below T_{old} temperature. Meanwhile, set initial T with 20σ .

The annealing of Lam and Dlosme adapt to feedback controlling. It monitors the standard deviation of cost function, cost function, the average values and proportion α accepted of τ times in the past. Usually τ =100; in the pseudo-code of simulated annealing. $R_{\lim it}$ is large, and allow exchange two modules over a long distance in chips. In the annealing process, try to adjust $R_{\lim it}$, and make the accepted moves approach 0.44 at any temperature. If α , accepted move, is less than 0.44, Lower $R_{\lim it}$. If α is more than 0.44, increase $R_{\lim it}$.

In the equation (4), 0.44 is expected accepting ratio, 40 is the inhibitory factor in order to avoid intense vibration at some temperature.

By controlling these parameters and functions, we finally want to get a better placement program through repeated simulated annealing process, which in the placement of circuit complexity is relatively low; the each PE function blocks' place is relatively close in the PEA thus saving the cost of routing.

3.2.2 Pathfinder Routing Algorithm

In the FPGA routing, people often use a directed graph describing FPGA's routing architecture, namely, routing resource graph. Wiring one line is equivalent to finding a path for connecting logical unit block pin in the routing resources. In order to avoid excessive use of FPGA limited routing resources, this route is as short as possible. In addition, wiring one line doesn't occupy the routing resources of other line to the greatest extent, so there is a set of strategies to avoid crowding for solving the problems of routing resource contention in most of the FPGA's routing device .Every core in the routing devices adopts to the derived method of the maze router to connect the endpoint of every line, essentially Dijkstra algorithm, which finds the shortest path for network source and drain side in the routing resource diagram.

There occurs congestion in the routing process, therefore, Pathfinder is needed for repeated dismantling and re-distribution of every line, at last, all of the congestion issues are resolved. Each process of dismantling and re-distribution one line in the circuit is called as a routing iteration, and the circuit of re-distribution resource shall be Invalid. Each routing iteration is at the expense of increasing routing resource cost to enhance the probability of solving congestion problems. In this paper, we propose negotiation congestion routing algorithm to solve routing problems in circuit module architecture.

```
Algorithm Pathfinder
{
    WHILE (OverusedResourcesExist == TRUE)
    {
       FOR i=1 TO NumberOfNets
       {
            Rip Up (RoutingTree[i]);
           RoutingTree[i] = SourceNode (i);
           FOR j=1 TO NumberSinks (i);
           {
              PriorityQueue=RoutingTree[i]  the  node  of
              cost is zero.
              WHILE (Not Reached (Sink (i, j)) == TRUE)
              {
                 M=FindLowestCostNode (PriorityQueue);
                 Delete Node (PriorityQueue, M);
                 FOR k=1 TO Fanout (M)
                 {
                    N = FanoutNode (M, k);
                    PathCost [N] =Cost[N] + PathCost[M];
                    Add Node  (PriorityQueue,  N)  at  cost
                    PathCost [N];
                 }
              }
            FOR k=1 TO PathNodes (Source (i), Sink (i, j))
              {
                 N =GetPathNode (Source(i),Sink(i,j), k );
                 Cost [N] = UpdateCost (N);
                 Add Node (RoutingTree[i], N);
              }
           }
       }
    }
  }
```

Fig. 5. Pathfinder algorithm flow chart

Mathematically, the costs associated with each node in structural connection diagram are given as:

$$Cost_n = (b_n + h_n) \times p_n \tag{3}$$

h_n Reflects the congestion history of the node n. Every time, the node n is used as h_n increases, and routing device will keep the congestion record. p_n is the current congestion cost of the node n; if using the node doesn't contribute to reusing, p_n is equivalent to 1. In Figure 5, if a node is added to the priority queue, the cost of the path is equivalent to the sum of this node and nodes on the path.

According to the setting of the Circuit architecture on this paper, when a line is dismantled and re-distributed, the values of p_n and h_n is determined by the equation to update the current penalty factor of congestion degree. Equations are given as:

$$p_n = p_n + 1\ ;\quad h_n = h_n + 1; \tag{4}$$

4 Experiment and Analysis

The experiments was performed using three mission line （0->4, 8->12, 3->7） to test associated performance. In the following diagram, IO 1.1and IO 1.2 represents the input and output of the first mission line respectively; IO 2.1and IO 2.2 represents the input and output of the second mission line respectively; IO 3.1 and IO 3.2 represents the input and output of the third mission line respectively.

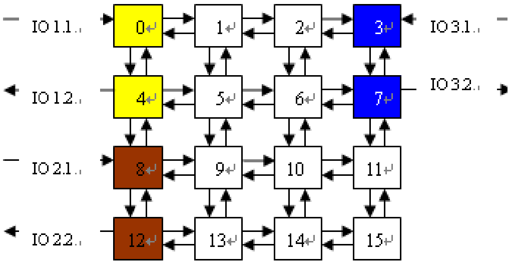


Fig. 6. The three mission lines

Based on the above three mission lines, the two algorithms used in the comparison are the general placement and routing program and negotiation congestion routing algorithm based on simulated annealing .Average success time and the rate of success in the placement and routing are compared. Three mission lines are available to test in the required PE length of 3, 4, 5 under three conditions, and in each case, we set up a certain number of PE damage successively, so the two algorithms are performed on the basis of fault avoidance.

Three mission lines （0->4, 8->12, 3->7， PE are 4 for every line validly, it taks1000 times for each PE damage case.） The result of the two algorithms as table 1:

Table 1. The result of the two algorithms

Number of damaged PE	The traditional route programs		Pathfinder based on simulated annealing algorithm	
	Average time (ms)	Success rate	Average time (ms)	Success rate
0	0.1672	5.6%	0.2259	18.7855%
1	0.1111	3.1%	0.2948	15.6546%
2	0.156	1.6%	0.3771	15.6564%
3	0.276	0.5%	0.3582	10.2467%
4	No	0.0%	0.2925	8.7286%

Three mission lines （0->4, 8->12, 3->7, PE is 5 for every line validly, it taks1000 times for each PE damage case.） The result of the two algorithms as table 2:

Table 2. The result of the two algorithms

Number of damaged PE	The traditional Route programs		Pathfinder based on simulated annealing algorithm	
	Average time （ms）	Success rate	Average time （ms）	Success rate
0	0.1553	0.5%	0.4958	1.4385%
1	No	0.0%	0.4823	1.0180%

Three mission lines （0->4, 8->12, 3->7, PE is 3 for every line validly, it taks1000 times for each PE damage case.） The result of the two algorithms as table 3:

Table 3. The result of the two algorithms

Number of damaged PE	The traditional route programs		Pathfinder based on simulated annealing algorithm	
	Average time （ms）	Success rate	Average time （ms）	Success rate
0	0.1037	19%	0.1662	80.93%
1	0.1032	15.9%	0.1949	75.05%
2	0.1381	10%	0.0976	70.778%
3	0.1070	9.5%	0.0981	66.328%
4	0.1053	6.1%	0.1068	55.3131%
5	0.1111	3.9%	0.2702	45.1613%
6	0.1232	2.2%	0.2686	38.7096%
7	0.1332	0.7%	0.2845	30.5863%

From the above test results, the success rate of pathfinder based on simulated annealing algorithm is higher than the traditional placement and routing algorithm (namely depth-first-search placement and routing), meanwhile, there is little difference between two algorithms in time consumption.

From the table, we find that the traditional placement and routing algorithm (namely depth-first-search placement and routing) can't find a matching route program in some cases, because as for the traditional placement and routing algorithm, when performing the search of available PE resources and routing resources around the PE node, it will randomly choose one resource from the searched result and save it into the path. Then keep searching around the node which just joint in the path. Therefore , it easily formed a loop in our N * N design of the PEA architecture, meanwhile variety resources are exclusive ,which contributes to a lot of resources being occupied .In addition ,the resources in the PEA framework are relatively less , which offers little room to success for the latter mission line in the placement and routing.

In our pathfinder based on simulated annealing algorithm, it doesn't form a loop during the searching path process .Negotiate and penalize to every node, which make each mission line can realize routing and placement successfully at last.

5 Conclusion

In this paper, on the basic of PEA that is N*N array of PE and on the basic of FPGA's and the VPR's placement and routing model with more abundant traditional resources, we propose a placement and routing model under the PEA architecture, which is called as "placement and optimal routing" cycle model, it is in order to meet optimization and the real-time requirements of circuit design.

So far, there are fewer relevant experts and scholars who study the placement and routing algorithm about this structure of PEA, it is owing to that this structure is relatively new .Therefore, we compare the traditional routing program (depth-first search) with the pathfinder based on simulated annealing.

Meanwhile we take the example of PEA array with N = 4 as a test .By comparison, we can draw a conclusion that the pathfinder based on simulated annealing is indeed better than the traditional placement and routing program. Furthermore, the algorithms make it simpler for circuit design. When time is enough and on the premise of meeting the real-time requirement, it can take advantage of rest time to optimize circuit and reduce the use of resource in the routing and placement. Therefore, the algorithm can reduce power consumption and running time and improve efficiency at last.

The future work for us mainly includes how to improve success rate and lower average time consuming under the model of this PEA array.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (No's: 60871021, 60473037).

References

1. Cohen, H.F., Andreou, A.G.: Analog CMOS integration and experimentation with auto adaptive independent component analyzer. IEEE Transactions on Circuits and Systems I Analog and Digital Signal Processing 42(2), 65–77 (1995)
2. Stoica, A., Zebulum, R.: Reconfiguration of Analog Electronics for Extreme Environments: Problem or Solution. In: International Conference on Intelligent Computing and Technology (2005)
3. Ferguson, M.I., Stoica, A., Keymeulen, D., Zebulum, R.S., Duong, V.: An evolvable hardware platform based on DSP and FPTA. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), pp. 145–152. AAAI Press, Menlo Park (2002)
4. Liguo, S.: Coarse-grained Reconfigurable Computing Architecture and its application. Beijing University of Aeronautics and Astronautics (2007)
5. Garis, H.D.: Evolvable Hardware: Genetic Programming of a Darwin Machine. In: Albrecht, R.F., Reeves, C.R., Steels, N.C. (eds.) Artificial Neural Nets and Genetic Algorithms, pp. 441–449. Springer, Heidelberg (1993)

Application of Data Mining in Multi-Geological-Factor Analysis

Jing Chen[1,2], Zhenhua Li[2], and Bian Bian[3]

[1] Faculty of Resource, China University of Geosciences, China
[2] School of Computer, China University of Geosciences, China
[3] Lanning and Designing Institute, East China Branch, SINOPEC, China
chenjing@cug.edu.cn

Abstract. Oil well productivity classification and abundance prediction are important for estimating economic benefit of a well. However, it is difficult to predict because well logs are complex and the amount of data collected today has far exceeded our ability to refine and analyze without the use of automated analysis techniques. In response to the problem above mentioned, data mining technology in recent years has shown the ability for discovering information and effectively extracts information from massive observational data sets that can be used to decisions. Especially, classification and prediction methods, are receiving increasing attention from researchers and practitioners in the domain of petroleum exploration and production (E&P) in China. Therefore, data mining is regarded as one of the ten key techniques for challenging problem of oil exploration and development. In this paper, four distinct kinds of classification and prediction methods in data mining, including decision tree (DT), artificial neural network (ANN), support vector machine (SVM) and Bayesian network are used to two real-world case studies. One is hydrocarbon reservoir productivity classification with 21 samples from 16 wells logging data in Karamay Oilfield 8th district reservoir. The results show that SVM and Bayesian are superior in the classification accuracy (95.2%) to DT, ANN and SVM, and can be considered as a prominent classification model. Another is reservoir abundance prediction with 17 mature accumulation systems samples in JiYang depression basin. The results show that SVM is superior in the prediction accuracy (91.92%) to DT, ANN and Bayesian, and can be taken as an excellent prediction model.

Keywords: data mining, productivity classification, abundance prediction, DT, ANN, SVM, Bayesian.

1 Introduction

Well logging and seismic exploration play important roles in petroleum industry. However, analyzing data from well logs and seismic is often a complex and laborious process because a physical relationship can't be established to show how the data are correlated. At present, especially most mature fields have reached the middle or later stage of exploration and production in China, and the main efforts have been focusing on exploring unconventional reservoirs in the future. Nevertheless, several practical

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 402–411, 2010.

problems such as low resistivity, low porosity and permeability, water-flooding, etc. may be faced in evaluating these complex formations during this period. It is widely admitted that there are a lot of limitations of conventional data analysis ways in oil and gas industry. Traditional methods in petroleum engineering are knowledge-driven and often neglect some underlying factors. On the contrary, many new techniques such as artificial intelligence, soft computing, and data mining, etc. are to deal with mass of data and never overlook any important phenomena. Data mining can be use for uncertainty analysis, risk assessment, data analysis and interpretation, and knowledge discovery from diverse data such as 3D seismic, geological data, well logs, and production data and is regarded as one of the ten key techniques for challenging problem of oil exploration and development.

There have been several studies examining the performance of data mining techniques on petroleum industry problems. Focus of most of the studies has been on classification problems. Several data mining techniques have been applied to solve classification problems, including statistical techniques such as Bayesian, discriminant analysis, and logistic regression, and machine learning techniques such as decision tree or rule induction, support vector machine and neural network. Data mining classifiers have been developed for a variety of applications such as identification of sedimentary facies, or fluid type, lithology discrimination, reservoirs evaluation and oil well production prediction, etc. These techniques automatically induce predication models, called classifiers, based on historical data about previously solve problem cases consist of various types of well log properties including conventional logging measurements, such as Gamma ray, density, neutron, resistivity, etc., and geological properties, such as porosity, permeability, hydrocarbon saturation, etc.[1,2]

In order to extensively investigate the effects of various data mining algorithms on the hydrocarbon reservoir classification and prediction performance, this paper adopts four distinct algorithms such as decision tree (DT), artificial neural network (ANN), support vector machine (SVM) and Bayesian network on two real-world examples including hydrocarbon reservoir productivity classification in Karamay Oilfield 8th district and mature accumulation systems abundance prediction in JiYang depression basin. This work may be beneficial to researchers and practitioners attempting to choose appropriate data mining method for hydrocarbon reservoir evaluation and prediction and can serve a little for future researches.

This paper is organized as follows. Section 1 introduces a brief discussion of related work in the domain of petroleum exploration and production (E&P) in China, and puts forward the new technique data mining. Section 2 describes four data mining methods used in this study. Section 3 presents two case studies and section 4 provides explanation of the results and discussion. Section 5 summarizes the works of this study and section 6 outlines tasks for future research.

2 Data Mining Classification and Prediction Methods

There are a variety of classification and prediction methods developed in the fields of statistical pattern recognition, machine learning and artificial neural network. According to their characteristics and application in petroleum industry, in this paper, we have examined and described the following widely used classification and prediction methods available.

2.1 Decision Tree (DT)

Decision tree is a predictive model that generally refers to a mapping of observations about an item to conclusions about the item's target value, each of which usually involves a single attribute x_i. Different decision tree learning algorithms inducer mainly differ in the goodness measure used to select the splitting attribute at each intermediate tree node, such as the maximum information gain in ID3, gain ratio in C4.5, Gini index in CART, chi-square test in CHAID, etc.[7]

2.2 Artificial Neural Networks (ANN)

Artificial neural networks widely using for classification are highly interconnected networks, which learn by adjusting the weights of the connections between nodes on different layers. A artificial neural network has an input layer corresponding to the attribute vector X, an output layer corresponding to the class y_in_j, and one or more hidden layers possibly. The train of a neural network is often significant; it takes experience and experimenting to adjust the parameters, such as the number of nodes on a hidden layer and the learning rate.

Assume we have n input units, $X_1, X_2, \cdots, X_n$ with input signals $x_1, x_2, \cdots, x_n$. When the network receive the signals (x_i) from input units (X_i), the net input to output (y_in_j) is calculated by summing the weighted input signals as follows:

$$\sum_{i=1}^{n} x_i w_{ij} \tag{1}$$

The matrix multiplication method for calculating the net input is shown in the equation below:

$$y\_in_j = \sum_{i=1}^{n} x_i w_{ij} \tag{2}$$

where w_{ij} is the connection weights of input unit x_i and output unit y_j.

The network output (y_j) is calculated using the activation function $f(x)$. In which, $y_j = f(x)$, where x is y_in_j. The computed weight from the training is stores and will become the information for the future application.

2.3 Support Vector Machine (SVM)

The support vector machine (SVM) is a machine learning technique adapted to separate two classes by a function, which is induced from available examples. This was proposed by Vapnik and others in the 1990s' and is based on statistical theory.

Finding an optimal separating hyperplane that maximizes the separation margin is the main goal of this classifier. [8]

Given a set of training data:

$$X_{train} = \{(x_i, y_i) | i = 1,2,\cdots,total\}, \quad x_i \in R^d, y \in \{+1,-1\}$$

A hyperplane as shown in Eq. (3) can be found to separate these two classes:

$$w \cdot x + b = 0 \tag{3}$$

Above set of vectors is said to be optimally separated by the hyperplane if it separated without error and the distance between the closest vectors to the hyperplane is maximal.

Vapnik introduced a canonical hyperplane, where the parameters $\mathrm{w, b}$ are constrained by Eq. (4):

$$\min_{\mathrm{i}} \left| \mathrm{w} \cdot \mathrm{x}^{\mathrm{i}} + \mathrm{b} \right| = 1 \tag{4}$$

From above set of equations, given the optimal separating hyperplane

$$\mathrm{w}^* = \sum_{\mathrm{i}=1}^{\mathrm{l}} \propto_{\mathrm{i}} \mathrm{y}_{\mathrm{i}} \mathrm{x}_{\mathrm{i}} \tag{5}$$

$$\mathrm{b}^* = -0.5\langle \mathrm{w}^*, \mathrm{x}_{\mathrm{r}} + \mathrm{x}_{\mathrm{s}} \rangle \tag{6}$$

Where $\propto_{\mathrm{i}}$ is the Lagrange multiplier. $\mathrm{x_r}$ and $\mathrm{x_s}$ are any support vectors from each class satisfying $\propto_{\mathrm{i}} > 0, \mathrm{y_r} = -1; \propto_{\mathrm{s}} > 0, \mathrm{y_s} = 1$.

2.4 Bayesian Network

Bayesian describes how the conditional probability of a set of possible causes for a given observed event can be computed from knowledge of the probability of each cause and conditional probability of the outcome of each cause. It relates the conditional and marginal probabilities of stochastic events A and B Bayesian is

$$\mathrm{P(A|B)} = \frac{\mathrm{P(B|A)P(A)}}{\mathrm{P(B)}} \tag{7}$$

In its most general form, Bayesian states that

$$\mathrm{P(A_i|B)} = \frac{\mathrm{P(B|A_i)P(A_i)}}{\mathrm{P(B|A_1)P(A_1) + P(B|A_2)P(A_2) + \cdots + P(B|A_n)P(A_n)}} \tag{8}$$

where i is any number between 1 and n. [9]

3 Experiments

3.1 Case Study 1: Productivity Classification

Select 21 samples from 16 wells logging data in Karamay Oilfield 8th district reservoir, of which 8 samples are learning samples which well numbers are 150th, 151th, 152th, the rests 13 samples are taken as the prediction. There are 6 inputs variables such as depth (m), porosity (%), permeability (10^{-3}um^2), resistivity (Ω•m), daily output (t•d^{-1}), productivity index and 1 output which is variable productivity class (Table1).

Table 1. Karamay Oilfield 8th district reservoir Logging and Tested parameters[6]

Well	Section h/m	Depth	Logging parameter			Tested productivity		
number		d/m	φ/%	k/(10^3μm^2)	R_t/(Ω·m)	Q/(t·d)	Index	Class
150	1794.0~1810.0	12.0	15.7	32.0	20.0	9.500	0.126	3
150	1832.0~1839.0	7.0	16.2	40.1	25.5	8.550	0.224	3
150	1878.5~1880.5	2.0	10.3	3.4	21.5	0.480	0.048	4
150	1926.0~1934.0	6.0	17.5	67.7	26.0	25.270	1.121	2
151	2073.0~2075.5	2.5	19.2	143.1	65.0	12.090	1.469	1
151	2101.0~2110.6	5.5	17.0	54.1	28.0	13.280	0.892	2
152	2111.0~2112.5	1.5	15.7	32.0	23.0	3.225	0.716	2
152	2161.0~2165.0	4.0	11.2	4.9	13.0	0.100	0.012	4
8239	2055.0~2073.0	8.0	16.1	37.2	9.0	12.600	0.504	3
8217	2070.5~2079.0	5.0	17.2	58.3	43.0	14.00	0.926	2
8252	2016.5~2022.5	4.0	15.7	32.0	5.0	7.900	0.553	3
8259	2132.5~2138.5	6.5	15.7	32.0	26.0	5.500	0.351	3
8211	2043.4~2075.5	16.0	17.2	58.0	36.0	9.700	0.191	3
8227	2099.0~2104.0	5.0	15.7	32.0	17.00	13.300	0.971	2
8271	1926.0~1930.0	4.0	14.8	22.0	25.0	5.600	0.312	3
8220	2116.0~2118.0	2.0	18.1	84.7	40.0	9.700	1.870	1
8235	2070.2~2074.8	4.6	16.6	46.6	24.0	18.100	1.295	2
8228	2104.2~2107.2	3.0	10.1	3.2	28.0	0.100	0.012	4
8247	2059.6~2062.8	3.2	8.5	1.6	20.0	0.100	0.010	4
8241	1992.0~1994.2	2.2	18.4	98.4	40.0	12.000	1.423	1
8201	2133.0~2134.8	1.8	17.2	58.3	30.0	5.400	1.236	2

Note: '1' means high productivity, '2' means middle productivity, '3' means low productivity, '4' means dry.

3.2 Case Study 2: Abundance Prediction

Select 17 mature accumulation systems in JiYang depression basin, of which 10 samples (1th~10th) are learning samples, the rests are used to predict. There are 6 input variables such as distance to source rock (km), depth(m), TOC(%), sandstone(%), porosity(%), permeability(%) and 1 output variable which is reservoir abundance (Table 2).

Table 2. JiYang depression basin accumulation systems geological parameters[5]

NO.	Distance /km	Depth /m	TOC /%	Sandstone /%	Porosity /%	Permeability /%	Abundance
1	5.0	1200	2.652	37.2	24.7	1132	101.4
2	11.3	646	2.448	39.0	26.3	998	38.1
3	15.0	367	1.412	44.2	21.45	1153	53.5
4	10.0	463	1.555	51.0	26.5	782	7.3
5	4.0	51	0.730	43.0	24.9	403	5.2
6	8.8	525	1.744	40.7	22.0	505	139.8
7	7.5	615	2.221	42.7	23.3	540	151.2
8	6.0	714	2.388	42.3	26.5	2049	90.2
9	5.8	152	0.822	71.0	21.8	210	5.0
10	7.5	526	1.730	54.1	22.5	244	45.9
16	10.0	428	2.217	53.5	18.2	100	4.7
18	9.0	350	2.469	53.4	19.0	189	56.0
19	8.0	343	1.255	36.1	20.8	1202	58.4
21	10.0	448	2.379	46.6	24.9	535	43.8
22	11.0	801	3.021	32.9	22.9	403	140.1
23	9.0	189	1.338	40.9	30.6	1538	77.4
26	3.0	236	2.093	50.5	23.9	274	59.9

3.3 Preprocessing

Values of variables are firstly normalized to [0, 1]:

$$x = \frac{x - \min(x_i)}{\max(x_i - \min(x_i)}, i = 1 \cdots n \tag{9}$$

These values are further normalized to have zero mean and unit standard deviation by lines transformation:

$$\overline{x_i} = \frac{1}{n}\sum_{j=1}^{n} x_i^j$$

$$\sigma_i^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_i^j - \overline{x_i})^2 \tag{10}$$

$$x_i^{*j} = \frac{x_i^j - \overline{x_i}}{\sigma_i}$$

where x_i^* are normalized variables and j denotes the dimension.

The radial basic function is employed as kernels:

$$K(x, x_i) = \exp(-\frac{\|x - x_i\|^2}{2\sigma^2}) \tag{11}$$

where $\|x - x_i\|^2$ is calculated by formula $\|x - x_i\|^2 = \sum_{j=1}^{n}(x^j - x_i^j)^2$ and σ is kernels width.

4 Results and Discussion

Four distinct kinds of classification methods, including decision tree (DT), artificial neural network (ANN), support vector machines (SVM) and Bayesian network are used to classify and predict.

4.1 Classification Results

Study on 21 samples from 16 wells logging data in Karamay Oilfield 8th district hydrocarbon reservoir. The following results show that there are 5 mismatches (150th, 152th, 8271th, 8220th, 8241th) with DT, 3 mismatches (8271th, 8220th, 8241th) with ANN, 1 mismatch (8227th) with SVM and 1 mismatch (8220th) with Bayesian in total 21 samples (Table 3). It shows that SVM and Bayesian are superior in the classification accuracy (95.2%) to DT (76.2%), ANN (85.7%) and can be considered as the prominent classification model (Table 5).

Table 3. Comparison of the tested and predicted from four distinct classification methods

Well	Tested	DT		ANN		SVM		Bayesian	
Num	Class	Class	Match	Class	Match	Class	Match	Class	Match
150	3	3	√	3	√	3	√	3	√
150	3	3	√	3	√	3	√	3	√
150	4	2	×	4	√	4	√	4	√
150	2	2	√	2	√	2	√	2	√
151	1	1	√	1	√	1	√	1	√
151	2	2	√	2	√	2	√	2	√
152	2	2	√	2	√	2	√	2	√
152	4	2	×	4	√	4	√	4	√
8239	3	3	√	3	√	3	√	3	√
8217	2	2	√	2	√	2	√	2	√
8252	3	3	√	3	√	3	√	3	√
8259	3	3	√	3	√	3	√	3	√
8211	3	3	√	3	√	3	√	3	√
8227	2	2	√	2	√	3	×	2	√
8271	3	2	×	2	×	3	√	3	√
8220	1	2	×	2	×	1	√	2	×
8235	2	2	√	2	√	2	√	2	√
8228	4	4	√	4	√	4	√	4	√
8247	4	4	√	4	√	4	√	4	√
8241	1	2	×	2	×	1	√	1	√
8201	2	2	√	2	√	2	√	2	√

4.2 Prediction Results

Study on 17 mature accumulation systems in JiYang depression basin. Table 4 shows the comparison of the tested and predicted from these four distinct prediction methods. The results show that SVM is superior in the prediction accuracy (91.92%) to DT (80.84%), ANN (87.06%) and Bayesian (90.55%) and can be considered as the prominent prediction model (Table 5).

Table 4. Comparison of the tested and predicted from four distinct prediction methods

NO.	Tested	Predicted abundance							
		DT	Error /%	ANN	Error /%	SVM	Error /%	Bayesian	Error /%
1	101.4	95.05	6.26	100.79	0.60	102.06	0.66	103.14	1.71
2	38.1	32.86	13.74	34.27	10.04	35.41	7.06	43.10	13.13
3	53.5	53.50	0	52.99	0.95	53.18	0.60	55.39	3.53
4	7.3	8.03	10.00	6.58	9.84	6.70	8.17	7.61	4.26
5	5.2	6.13	17.85	5.99	15.19	5.77	10.91	6.73	29.39
6	139.8	123.61	11.58	134.67	3.67	135.47	3.10	128.19	8.31
7	151.2	136.08	10.00	147.85	2.21	148.29	1.93	159.27	5.33
8	90.2	66.42	26.36	69.66	22.78	83.61	7.30	83.04	7.94
9	5.0	6.21	24.17	5.27	5.44	6.23	24.54	6.15	23.00
10	45.9	37.48	18.34	36.57	20.34	50.12	9.20	40.52	11.71
16	4.7	8.49	80.71	8.61	83.16	6.52	38.77	3.61	23.14
18	56.0	46.99	16.08	46.25	17.40	53.80	3.92	64.51	15.20
19	58.4	44.11	24.46	58.02	0.65	60.14	2.98	58.75	0.61
21	43.8	35.53	18.88	34.47	21.29	50.56	15.43	38.34	12.47
22	140.1	123.56	11.81	140.49	0.28	140.12	0.05	140.03	0.05
23	77.4	67.72	12.51	76.91	0.64	77.76	0.47	77.38	0.02
26	59.9	73.61	22.89	56.58	5.55	58.50	2.34	103.14	0.84

4.3 Discussion

Experimental results and theoretical analysis show that SVM and Bayesian models have the following advantages.

SVM and Bayesian are powerful for the problem of classifying and predicting with small samples in our research. The accuracy of them in prediction both are more than 90%, even more than 95% in classification (Table 5).

They are convenient for the domain of petroleum exploration and production (E&P) in which parameters are complex to refine and analyze, and would never overlook any underlying factors consisting of various types of well log properties including conventional logging measurements, such as Gamma ray, density, neutron, resistivity, etc., and geological properties, such as porosity, permeability, hydrocarbon saturation, etc. in some especial phenomena, such as such as low resistivity, low porosity and permeability, water-flooding, etc. So analyze and evaluate hydrocarbon reservoir with multi-geological-factor using data mining methods is viable and may achieve excellent performance.

Table 5. Comparison with DT, ANN, SVM, Bayesian

Method	Classification Accuracy/%	Prediction Accuracy/%
DT	76.2%	80.84%
ANN	85.7%	87.06%
SVM	95.2%	91.92%
Bayesian	95.2%	90.55%

This work may be beneficial to researchers and practitioners attempting to choose appropriate data mining methods for hydrocarbon reservoir evaluation and prediction and can serve a little for future researches.

5 Conclusions

Data mining techniques are obtaining increasing attention from researchers and practitioners in the domain of petroleum exploration and production (E&P) in China, especially classification and prediction methods. This paper researches on application of data mining in multi-geological-factor analysis using well logs.

(1) Study the characteristics of four distinct kinds of method, including decision tree (DT), support vector machine (SVM), artificial neural network (ANN) and Bayesian network. To contrast the application results of these four distinct methods, they were applied to 2 case studies.
(2) Experiment on hydrocarbon reservoir productivity classification with DT, SVM, ANN and Bayesian network selecting 21 samples from 16 wells logging data in Karamay Oilfield 8th district reservoir. Inputs are depth, porosity, permeability resistivity, daily output and productivity index, output is productivity class. The results show that SVM and Bayesian are superior in the classification accuracy (95.2%) to DT (76.2%), ANN (85.7%).
(3) Experiment on mature accumulation systems abundance prediction with DT, SVM, ANN and Bayesian network selecting 17 mature accumulation systems in JiYang depression basin. Inputs are the distance to the source rock, depth, TOC, sandstone percent, porosity and permeability, output is abundance. The results show that SVM is superior in the prediction accuracy (91.92%) to DT (80.84%), ANN (97.06%) and Bayesian (90.55%).

6 Future

Our future researches are listed as following:

(1) Instead of focusing on more distinct data mining methods in hydrocarbon reservoir evaluation and prediction at this moment, our future important study is that the incorporation of domain knowledge such as the substantial elements of input well logging or seismic parameters affects different classifiers to different degrees and prediction accuracy, something that has been empirically chosen in recent research.
(2) More studies are needed in different domains to understand what other types of knowledge could influence performance. For example, future study could explore the effects of knowledge feature construction including well logging and seismic data, and subsequent refinement of these mass data through data mining.

References

1. Hongqi, L., Haifeng, G., Haimin, G., Zhaoxu, M.: Data mining techniques for complex formation evaluation in Petroleum Exploration and Production: A Comparison of Feature Selection and Classification Methods. In: Pacific-Asia-Workshop on Computational Intelligence and Industrial Application, vol. 1, pp. 37–43 (2008)
2. Li, X., Chan, C.W.: Application of an enhanced decision tree learning approach for prediction of petroleum production. Engineering Applications of Artificial Intelligence 23, 102–109 (2010)
3. Guangren, S.: Application of Support Vector Machine to multi-geological-factor analysis. Acta Petrolei Sinica 29(2), 195–198 (2008)
4. Guangren, S., Xingxi, Z., Guangya, Z., Xiaofeng, S., Honghui, L.: The use of artificial neural network analysis and multiple regression for trap quality evaluation: A case study of the Northern Kuqa Depression of Tarim Basin in western China. Marine and Petroleum Geology 21(3), 411–420 (2004)
5. Lina, Y., Guojun, X.: Prediction of abundance of hydrocarbon resources with artificial neural network method-—Taking JiYang depression for example. Journal of Oil and Gas Technology 29(1) (2007)
6. Chengqian, T., Narui, M., Chao, S.: Model and method for oil and gas productivity prediction of reservoir. Journal of Earth Sciences and Environment 26(2) (2004)
7. Sinha, A.P., Zhao, H.: Incorporation domain knowledge into data mining classifiers: An application in indirect lending. Decision support System 46, 287–299 (2008)
8. Mitra, V., Wang, C.-j., Banerjee, S.: Text classification: A least square support vector machine approach. Appiled Soft Computing 7, 908–914 (2007)
9. Appavu, S., Balamurugan, A., Rajaram, R.: Effective and efficient feature selection for large-scale data using Bayes' theorem. International Journal of Automation and Computing 6, 62–71 (2009)
10. Elkamel, A.: An artificial neural network for predicting and optimizing immiscible flood performance in hetogeneous reservoirs. Computer Chem. Engng. 22, 1699–1709 (1998)
11. Yan, W., Shao, H., Wang, X.: Soft sending modeling based on support vector machine and Bayesian model selection. Computer and Chemical Engineering 28, 1489–1498 (2004)

Applying an Artificial Neural Network to Building Reconstruction

Dong-Min Woo, Dong-Chul Park, and Hai-Nguyen Ho

Dept. of Electronics Engineering, Myongji University, Korea
dmwoo@mju.ac.kr, parkd@mju.ac.kr, tasabk@gmail.com

Abstract. This paper highlights the uses of Centroid Neural Network for detecting and reconstructing the 3D rooftop model from aerial image data. High overlapping aerial images are used as an input to the method. The Digital Elevation Map (DEM) data and 2D lines are generated and then combined to form 3D lines. The core of the technique is a clustering process using Centroid Neural Network algorithm to classify these 3D lines into groups of lines that belong to the corresponding building areas. This work differs from the previous researches, as it affiliates 3D lines and corners - obtained by applying the Harris corner detector - to automatically extract accurate and reliable 3D rooftop information. The proposed approach is tested with the synthetic images generated from the Avenches dataset of the Ascona aerial images and gives an average error of 0.38m in comparison with the ground truth data. The experiment result proves the applicability and efficiency of the method in dealing with building reconstruction in complicated scenes.

1 Introduction

Extracting rooftop structure of buildings from aerial images is a difficult task due to the complexity and arbitrary nature of terrain. For example, the considered objects could be partly occluded by surroundings such as vegetation, shadows, or roads. Also, line segments happen to be fragmented in making initial edge detection a deficient source of information. Moreover, the shapes of building roofs are very diverse in reality. Usually buildings have flat or gable roofs in industrial areas, but in cases of residential areas the roof shapes are more complicated. Many of them are even neither flat nor composed of simple rectangles. Recent research has shown some achievements though still remained some insufficiency [1][2][3].

Many approaches have been proposed in the literature to deal with this problem from the primary method based on a single image only [2][4] to the advanced methods taking full advantages of the development of the image processing techniques [1][3][5]. A typical specimen of primary method is the work of Lin and Nevatia [4] for detecting buildings from a monocular image. At first, linear lines are detected from the image. Next parallelogram hypotheses are selected based on wall and shadow evidence before employing perceptual grouping to generate 2D roof.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 412–423, 2010.

However, shadow, detected by darker appearance, is not always available or reliable information.

Nevatia et al. [6] made an improvement on the primary method by using multiple images, but did not employ stereo image analysis. They detected rectilinear buildings and constructed their 3D shapes from each single image. Some structures may be more reliably detected in some views depending upon conditions, such as viewing direction, illumination direction, and building orientation. This approach integrated the results, taking their own advantages to achieve a more accurate model. If a building is correctly detected in one view, supporting evidence for it should be found in the other views. On the other hand, if an incorrect hypothesis has been made, it should be unlikely to find much supporting evidence from the other views. Based on this observation, a better decision can be made by integrating all evidence from all available views. Nevertheless, this solution is rather difficult and most of the researchers recently focus on multiple stereo images [1][3], which is available with only small extra cost.

Cord et. al. suggested an approach for modeling urban buildings [7]. By use of stereo images a digital elevation model (DEM) is created. On the foundation of DEM, the system set a threshold to discriminate the ground region (road, bush) and above-ground region (building, high vegetation). Then buildings are selected from the above-ground region based on the density, accuracy and reliability of their DEM. They computed the histogram on the regions. For a vegetation region the histogram is scattered; otherwise, there are some privileged directions in the histogram of building region. This method, however, depends too much on altitude information for building detection.

Fischer et al. [8] suggested a model-based approach. This method includes three main tasks. The first step matches hypothesized 2D corners of different images leading to 3D corners. The 3D corners are taken to find building parts prior to projecting back into different images as for verification. This approach has advantage over other methods using single images. The final result, however, critically depends on the matching feature step between 2D images and the selection of the correct building template in verification process.

In this paper, the proposed approach can be presented in three steps, as shown in Fig.1. We initially extract 3D lines from aerial images by associating the DEM data and the extracted 2D line information. The 3D lines are grouped to reduce fragmentation and duplication during extraction step. Simultaneously, 3D corner points are detected using Harris corner detector [9]. Secondly, the Centroid Neural Network (CNN) [10] algorithm is employed to group above lines and corners into groups corresponding to each building. Finally, each 3D rooftop shape is reconstructed based on information of lines and corners in each group via rooftop modeling process. Many researches have studied on grouping 2D line segments [11], [12]. One of the most popular approaches is to solve perceptual grouping problem. In this paper, we apply grouping method to 3D corners affiliating with 3D lines by using the CNN algorithm. The advantages of using the CNN algorithm, which was verified as an excellent unsupervised learning algorithm, are requirements of neither a predetermined schedule

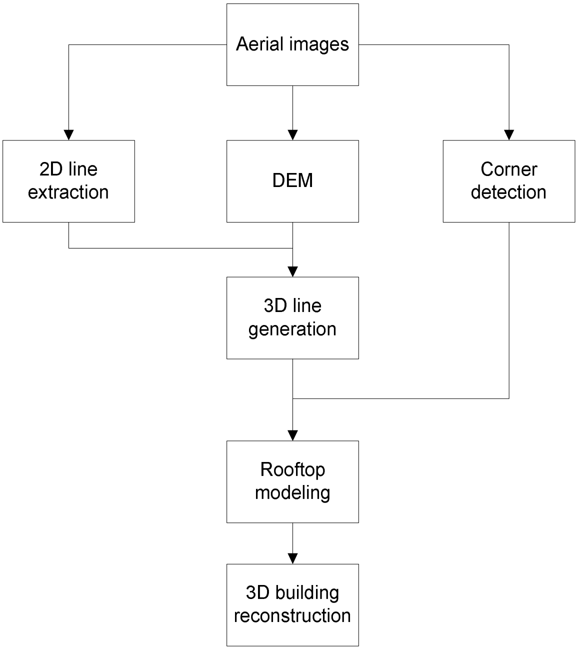


Fig. 1. An overview of the building reconstruction approach from aerial images

for leaning coefficients nor a total number of iterations for clustering while it converges much faster than conventional algorithms with compatible outcome.

The remainder of this paper is organized as follows: Section 2 presents a summary (background theory) of CNN algorithm, while section 3 a 3D line grouping algorithm using the CNN. The modeling process from 3D lines and corner points into rooftop model is stated in section 4. Experimental results and conclusion are presented in section 5 and 6 respectively.

2 Centroid Neural Network

Centroid Neural Network is an unsupervised competitive learning algorithm based on the classical k-means clustering algorithm. CNN introduces the definitions of "winner neuron" and "loser neuron" that are clusters in the clustering process. A neuron is a winner when it wins the data in the current presentation but not in the previous presentation. On the contrary, a loser neuron wins the data in the previous presentation but loses in the current presentation. Only the centroids of winner and loser clusters are changed, so we just need to recalculate the weights of them at each epoch, instead of all clustered data. An "epoch" is a presentation of whole data vectors in the data set to the network.

Generally, the weights of winner and loser neurons are updated as follows, assumed that a data vector x is presented to the network at time n:

$$w_i(n+1)=\frac{1}{N_i+1}[N_i w_i(n)+x]=w_i(n)+\frac{1}{N_i+1}[x-w_i(n)] \tag{1}$$

$$w_j(n+1)=\frac{1}{N_j-1}[N_j w_j(n)-x]=w_j(n)-\frac{1}{N_j-1}[x-w_j(n)] \tag{2}$$

where $w_i(n)$, $w_j(n)$ are the weights of winner neuron i and loser neuron j at time n; N_i, and N_j are the numbers of data vectors in cluster i and cluster j respectively.

3 Grouping of 3D Lines

3.1 3D Line Extraction

The 3D line extraction process is carried out to generate 3D lines, which will be classified into groups of parallel 3D lines in the 3D line grouping module. As for 3D line extraction, there exist two main branches of research: area-based method and feature-based method. The area-based method can achieve a precise terrain model. This method, however, is not quite successful in 3D line extraction for the building reconstruction, since it generates smoothed elevation around the edges of buildings. The feature-based method based on 2D line segment, corner point, edge, etc. seems effective. Nevertheless, when dealing with complex images, the number of possible 2D line matching increases cause the high possibility of false line matching, as well as significant computational cost.

In this research, we construct 3D lines by using line fitting of elevation data on 3D line extracted from ortho-image [13]. The accuracy of this method was evaluated almost 10 times compare to raw elevations obtained by area-based method. This method requires 2D line and DEM information as a prerequisite. The 2D lines are detected from input images using Canny edge detector [14] and Boldt algorithm [15]. For the DEM, firstly, the epipolar images are generated from the aerial images by epipolar re-sampling process. The disparity map between epipolar pair is obtained by stereo matching prior to generate DEM.

3.2 3D Line Clustering

Because of the error in extracting 3D lines, a true line may be constructed more than one time, and create many corresponding 3D line segments lying close and parallel together. As a result, this phenomenon causes the ambiguity. Also, some line segments can be parts of the same linear structure in the image, but fragmented during extraction process. Examples of the fragmented line and the ambiguity are shown in Fig.2a and Fig.2b respectively.

This grouping step employs the CNN algorithm to regroup those segments and replace them by the corrected lines. The possibility of regrouping two segments depends on their collinearity, nearness, and their own length. The collinearity can be measured as the difference in direction of two line segments, whereas the nearness is calculated in terms of the distance between their end points. All these factors are

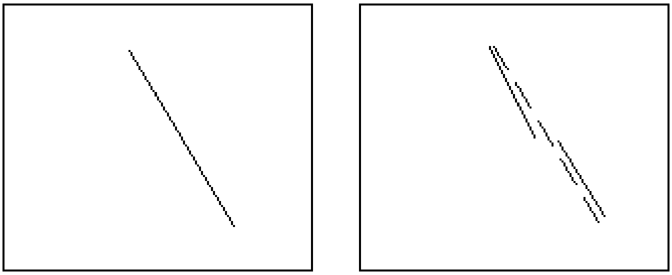


Fig. 2a. Example of fragmented line

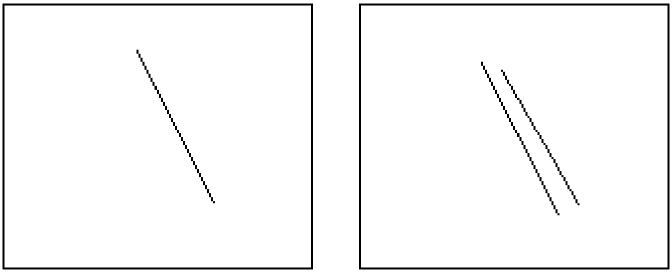


Fig. 2b. Example of ambiguity

formulated into a metric for line segments [16], which is adopted as the distance measure function for CNN.

The distance function between a line segment u and a neuron w - is also a line segment - is defined as:

$$d(u,w) = \frac{1}{f(u,w)} \tag{3}$$

Where: $f(u,w) = G_{\sigma_{angle}}(\theta_w - \theta_u) * G_{\sigma_{length}\sigma_{width}}(R_{\alpha_u}^{-1}(x_w - x_u, y_w - y_u))$ (4)

With:

$\theta_w - \theta_u$: the difference in direction

$x_w - x_u, y_w - y_u$: the distance between the midpoints of two segments

$G_\sigma(x)$: Gaussian function for the orientation component

$G_{\sigma_x \sigma_y}(x, y)$: Gaussian function for the displacement component

The angle and the length of a neuron are updated similarly to equations Eq.1 and Eq.2. A neuron wins a line segment if it has the smallest distance to that line segment among all the line segments. Since equation Eq.3 only applies to 2D line, when using for 3D line it requires another condition that the height values of two considering line segments must not differ so much from each other.

4 Rooftop Modeling Process

4.1 Detecting Corner Points

The rooftop modeling process requires corner points, as well as 3D lines. In this paper, corner points are detected by employing Harris corner detector [9]. This corner detector is robust against noise, and very efficient due to its strong invariance to rotation, scale and illumination variation [17]. The Harris corner detector is based on the local auto-correlation function of a signal which measures the local changes of the signal with patches shifted by a small amount in different directions.

Given a shift $(\Delta x,\ \Delta y)$ and a point (x, y), the auto-correlation functions is defined as

$$c(x,y) = [\Delta x \quad \Delta y] C(x,y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{4}$$

where matrix $C(x,y)$ captures the intensity of the local neighborhood. Let λ_1, λ_2 be the eigenvalues of matrix $C(x,y)$. There are three cases to be considered:

(1) If both λ_1 and λ_2 are small, the local auto-correlation function is flat. Consequently, the windowed image region has approximately a constant intensity

(2) If one eigenvalue is high and the other low, the local auto-correlation function is ridge shaped. Consequently, we have only local shift in one direction, which indicates an edge.

(3) If both eigenvalues are high, the local auto-correlation function is sharply peaked. Consequently, a shift in any direction will result in a significant increase, which indicates a corner.

Application of corner detector to complicated images causes a large number of false corners. For example, vegetation, trees and roads generates false corners, as well as corners of buildings. In this context, the number of error corners is reduced by selecting only ones that lie closely to the extracted 3D lines. About 90 percent of false corners can be eliminated until this step.

4.2 Clustering 3D Corner Points and 3D Lines

The CNN algorithm is used to cluster 3D corner points and 3D lines into the group corresponding to each building. Corner points are clustered by using Euclidean metric as measuring weight. The winner cluster is the one that has neuron coordinate closest to the coordinate of considering point. Weights of the winner neuron *i* and loser neuron *j* on dimension *x* are updated as follows:

$$x\_center_i(n+1) = x\_center_i(n) + \frac{1}{N_i + 1}[x\_point - x\_center_i(n)] \tag{5}$$

$$x\_center_j(n+1) = x\_center_j(n) - \frac{1}{N_i - 1}[x\_point - x\_center_j(n)] \qquad \text{(Eq.6)}$$

where *x_center* indicates value on dimension *x* of the neuron.

Similar equations are applied for dimension *y* and *z*. After finishing this clustering process, the group of corner points with their neuron centers is defined. Each 3D line is distributed to a group that its center is closest to center point of the line. As far as this, each group is a collection of corner points and lines, which corresponds to each building.

5 Modeling 3D Rooftop

There have been some researches on modeling rooftop [3][5] using "hypotheses and verification" paradigm based on perceptual grouping. Generally, these methods try to find out parallel pairs of lines, and afterwards search for another one upright to them so that they can form a rectangle – a usual shape of rooftop. This approach thus faces difficulties in detecting rooftops that are not exactly rectangular. In this paper, a new approach is introduced to solve that problem.

Lines in each clustered group are filtered as following conditions:

(1) Define a sub-group as the lines, which are long enough and closely parallel together, and compute their average angle.

(2) Define another sub-group as the lines, which are perpendicular to above angle with the small amount of error.

(3) For the others that do not belong to the previous two sub-groups, ignore lines shorter than a specific threshold.

The angle error in condition 2 and threshold in condition 3 are set as ±10 degrees and 20 pixels respectively.

The selected lines are extended until cutting together creating lines' intersections. If distance from a lines' intersection to any corner points in group is larger than a threshold, it is disregarded; otherwise, it is accredited as a correct intersected point in reality. An example of lines' intersections is showed in Fig.3. A circle in the figure describes a corner point and an area around it with a radius equal to the threshold. Only lines' intersection within the circle is validated.

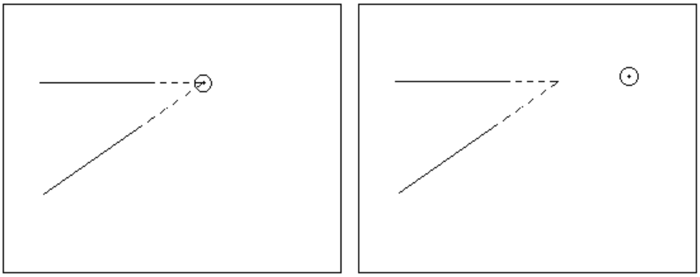
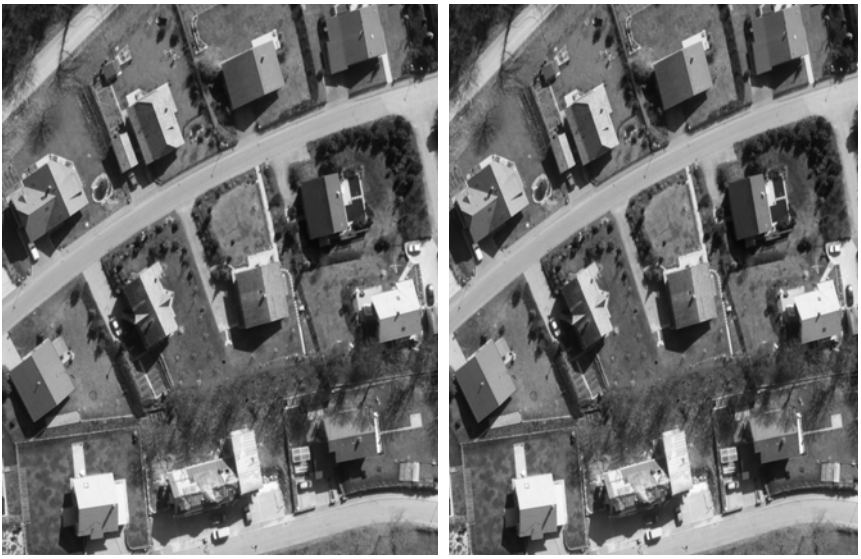


Fig. 3. Validated and disregarded cases of lines' intersections

6 Experimental Results

To evaluate the proposed method, we set up an experiment based on Ascona aerial images of the Avenches area. The sizes of these images are 1024 x 1024. They contain several buildings with different shapes and heights amid many other objects such as trees, roads and shadows which partly occlude the buildings, as shown in Fig.4.

In order to extract the 3D lines, we firstly detect 2D lines (Fig.5) and generate DEM information (Fig.6). The number of 2D lines detected is too large, about 3800 segments, due to the complexity caused by interfering objects. Not only edges of the buildings but also boundaries of the roads and small structures are also extracted. Based on elevation data, we can filter out road boundaries. Also, small line segment can be removed according to its length. The number of lines is decreased significantly - approximately 710 remaining lines - bring about the simplification and reduction in computational cost of grouping process.



(a) (b)

Fig. 4. Two Ascona aerial images

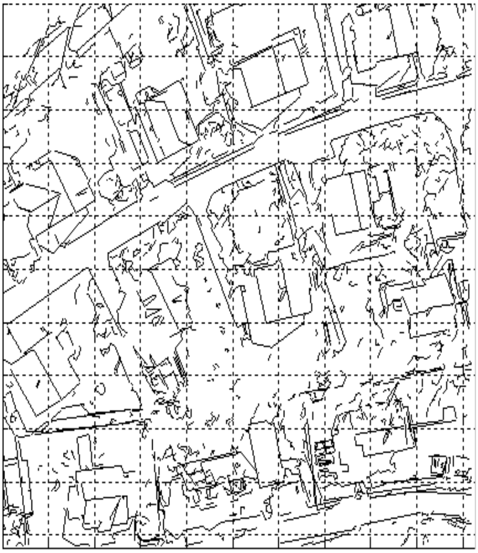


Fig. 5. 2D line detection result

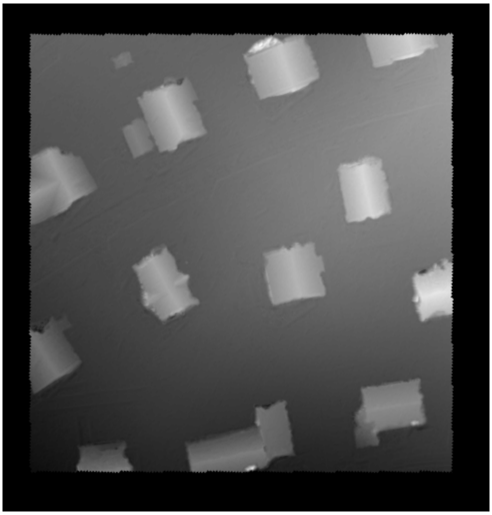


Fig. 6. Digital Evaluation Map

Operating 3D line fitting method as mentioned above, associating with the grouping process using CNN, the final result of 3D lines has been generated as shown in Fig.7.

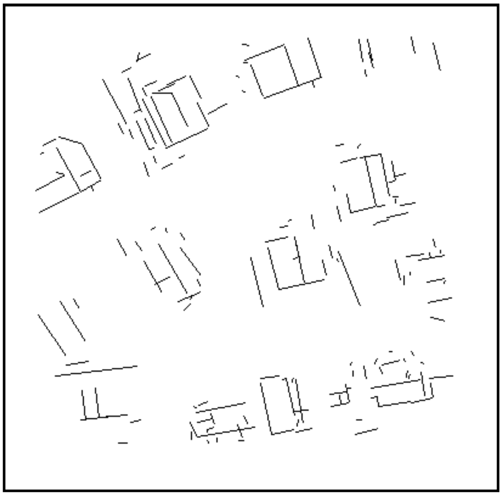


Fig. 7. Extracted 3D lines

CNN is employed again to cluster corner points into the group corresponding to any building. The stability and efficiency of CNN algorithm are represented in table 1 via a comparison with SOM and FCM algorithms. Each algorithm has been executed for three times with the same number of epoch is 43. In that experimental condition, CNN has been proved to be the fastest in three algorithms with the average error values are smallest and same for all times. The errors of clustering process are computed as Eq. 7.

$$err = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N_i} dist(e_{ij},e_{ic})}{N} \qquad \text{(Eq. 7)}$$

where M: number of clusters

N_i: number of corner points in cluster i

e_{ij}: coordinate of corner points j of cluster i

e_{ic}: coordinate of neuron center of cluster i

N: *total number of corner points*

The result of clustering process of corner points and 3D lines are showed in Fig.8 and Fig.9 respectively.

Table 1. Compare CNN with SOM and FCM

	CNN			SOM			FCM		
Time (CPU clock)	46	78	47	125	219	109	11468	9032	10797
Error (pixel)	78.19	78.19	78.19	81.00	71.79	106.52	167.98	168.253	167.71

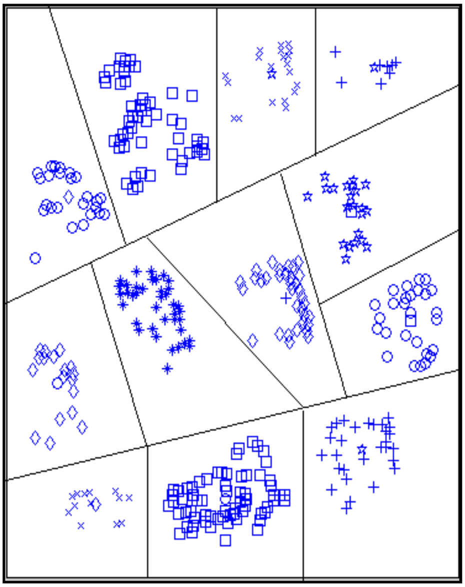


Fig. 8. Result of clustered corner points

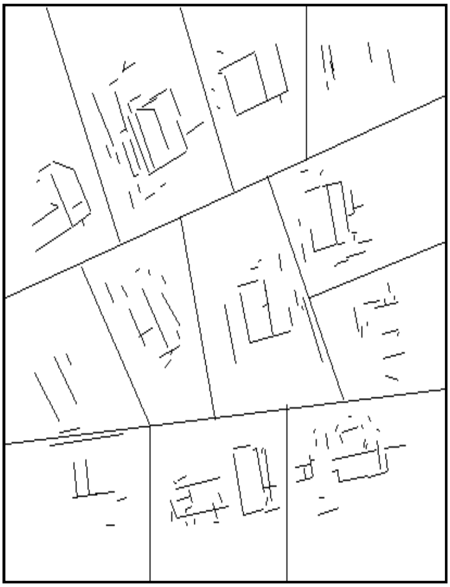


Fig. 9. Result of clustered 3D lines

Rooftop modeling is performed for each group by extending the 3D lines to cut together and validating the distance from intersection to any corner points within the threshold of 15 pixels. Fig.9 provides the rooftop detection result of the entire area.

From the detected rooftop and the known geometric parameters of image acquisition, we reconstruct 3D buildings, as shown in Fig.10.

The experimental result shows that all buildings, which are detected completely in DEM generation phase, are clustered perfectly into rooftop model. The quantitative accuracy of 3D building reconstructed is only 0.38 meter different from the ground truth data obtained by calculating the average distance between the extracted 3D line and the ground truth line as in Eq.8.

$$\frac{\sum \frac{e_{1i}+e_{2i}}{2} * d_i}{\sum d_i} \qquad (8)$$

where, e_{1i} and e_{2i} are the distances from the starting point and end point of line segment i to the ground true 3D line respectively, and d_i is the length of line segment i.

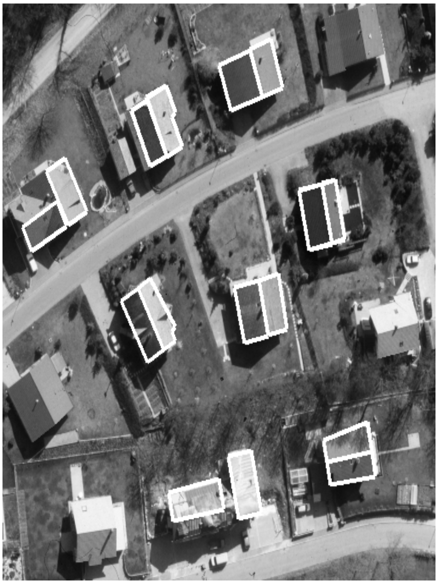


Fig. 10. Example of rooftop

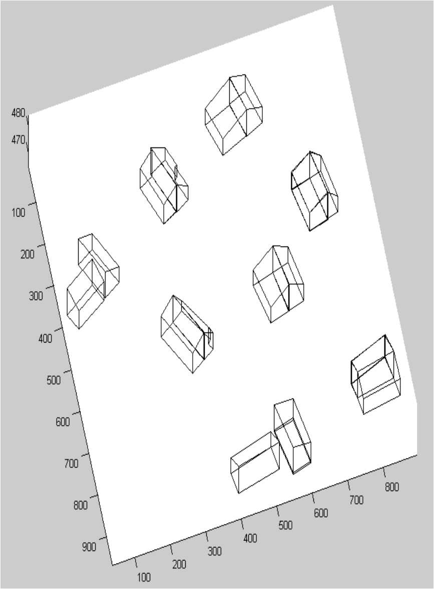


Fig. 11. A 3D view of reconstructed buildings

7 Conclusion

A new method to detect rooftop from two aerial images using the Centroid Neural Network is proposed. In this approach, the DEM data is used to remove the unnecessary lines and corner points prior to executing clustering process. The CNN algorithm is applied to classify corner points and lines into groups corresponding to each building. Rooftops are successfully obtained by connecting these lines and verifying with corner points. The experimental result proved that the proposed method can be utilized efficiently in rooftop detection task and building reconstruction.

References

[1] Collins, R.T., Jaynes, C.O., Cheng, Y.Q., Wang, X., Stolle, F., Riseman, E.M., Hanson, A.R.: The Ascender system: automated site modeling from multiple aerial images. Computer Vision and Image Understanding (72), 143–162 (1998)

[2] Huertas, A., Nevatia, R.: Detecting buildings in aerial images. Computer Vision, Graphics, and Image Processing 41, 131–152 (1988)

[3] Noronha, S., Nevatia, R.: Detection and modeling of buildings from multiple aerial images. IEEE Trans. Pattern Analysis and Machine Intelligence 23, 501–518 (2001)

[4] Lin, C., Nevatia, R.: Building detection from a monocular image. In: Firschein, O., Strat, T.M. (eds.) RADIUS: Image Understanding for Imagery Intelligence, pp. 153–170. Morgan Kaufmann Publishers, San Francisco (1997)

[5] Woo, D.M., Nguyen, Q.D., Park, D.C.: 3D rooftop extracting using perceptual organization based on fast graph search. In: Proceedings of the International Conference on Signal Processing, pp. 1317–1320 (2008)

[6] Nevatia, R., Lin, C., Huertas, A.: A system for building detection from aerial images. In: Automatic Extraction of Man-Made Objects from Aerial and Space Images (1997)

[7] Cord, M., Jordan, M., Cocquerez, J.P., Paparoditis, N.: Automatic extraction and modeling of urban buildings from high resolution aerial images. In: IAPRS 1999 (September 1999)

[8] Fischer, A., Kolbe, T.H., Lang, F.: Integration of 2D and 3D Reasoning for Building Reconstruction Using a Generic Hierarchical Model. In: Forstner, W. (ed.) Proc. Workshop Semantic Modeling for the Acquisition of Topographic Information, Bonn, Germany (1997)

[9] Harris, C., Stephens, M.J.: A combined corner and edge detector. In: Alvey Vision Conference, pp. 147–152 (1988)

[10] Park, D.C.: Centroid neural network for unsupervised competitive learning. IEEE Trans. on Neural Network 11, 520–528 (2000)

[11] Perona, P., Freeman, W.: A factorization approach to grouping. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 655–670. Springer, Heidelberg (1998)

[12] Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)

[13] Woo, D.M., Han, S.S., Park, D.C., Nguyen, Q.D.: Extraction of 3D line segment using digital elevation data. In: Proceedings of the 2008 Congress on Image and Signal Processing, vol. 02, pp. 734–738 (2008)

[14] Canny, J.: A computational approach to edge detection. IEEE Trans. on Pattern Analysis and Machine Intelligence 8, 679–698 (1986)

[15] Boldt, M., Weiss, R., Riseman, E.M.: Token-based extraction of straight lines. IEEE Trans. Systems, Man and Cybernectics 19, 1581–1594 (1989)

[16] Nacken, P.F.M.: A metric for line segments. IEEE Trans. on pattern analysis and machine intelligence 15, 1312–1318 (1993)

[17] Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. International Journal of Computer Vision 37(2), 151–172 (2000)

Fast Dimension Reduction Based on NMF

Pavel Krömer, Jan Platoš, and Václav Snášel

Department of Computer Science
Faculty of Electrical Engineering and Computer Science
VŠB – Technical University of Ostrava
17. listopadu 15, 708 33 Ostrava – Poruba, Czech Republic
{pavel.kromer,jan.platos,vaclav.snasel}@vsb.cz

Abstract. Non-negative matrix factorization is an important method in the analysis of high dimensional datasets. It has a number of applications including pattern recognition, data clustering, information retrieval or computer security. One of its significant drawback lies in its computational complexity. In this paper, we discuss a novel method to allow fast approximate transformation from input space to feature space defined by non-negative matrix factorization.

1 Introduction

With the growing dimension of the data in a dataset, its efficient analysis becomes more and more complicated. Increasing dimension leads to more sparse data. With very sparse data, the points located in different dimensions can be seen as equally distanced and any distance based analysis becomes meaningless [1].

Data processing methods such as clustering algorithms, are mostly designed for low dimensional samples. In sparse high dimensional datasets, only a few dimensions are usually relevant to certain task and other dimensions contain mostly noise [1].

To deal with high dimensional datasets, various methods of feature selection or feature transformation are deployed to reduce the dimension of the data. Feature selection methods aim to eliminate irrelevant dimensions while feature transformation methods project data onto smaller space so, that the relative distance between data points remains the same. Data transformation techniques modify the attributes (e.g. by linear combination) and aim to discover hidden structure of the datasets. Moreover, the data projected to lower dimensional feature spaces can be processed by traditional data mining techniques more efficiently [1]. Feature transformation techniques include matrix factorization methods such as singular value decomposition (SVD) and non-negative matrix decomposition (NMF).

2 Non-negative Matrix Factorization

In general, matrix factorization (or matrix decomposition) is an important task in data analysis and processing. A matrix factorization is the right side matrix product in

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 424–433, 2010.

$$A \approx F_1 \cdot F_2 \cdot \ldots \cdot F_k \tag{1}$$

for the matrix A. The number of factor matrices depends usually on the requirements of given application area. Very often, $k = 2$ or $k = 3$. There are several matrix decomposition methods reducing data dimensions and simultaneously revealing structures hidden in the data. Such methods include SVD and NMF.

Non-negative matrix factorization [2,3] is popular unsupervised learning algorithm for efficient factorization of real matrices satisfying the non-negativity constraint. NMF approximates real $m\times\ n$ matrix A as a product of two non-negative matrices W and H of the dimensions $m \times r$ and $r \times n$ respectively. Moreover, it applies that $r << m$ and $r << n$.

$$A \approx W \cdot H \tag{2}$$

There are several algorithms for NMF computation based on iterative minimization of certain cost function [3]. The original NMF algorithm involved minimization of the Frobenius norm [4] defined by formula (3).

$$\|A - WH\|_F^2 = \sum_{ij} |A_{ij} - (WH)_{ij}|^2 \tag{3}$$

Other investigated cost measures include square of the Euclidean distance between V and its approximation (4) or Kullback-Leibler divergence D (5). For every cost function, there are update rules (multiplicative or additive) applied iteratively in order to reduce the distance between original matrix V and its model [3,4].

$$\|A - WH\|^2 = \sum_{ij} (A_{ij} - (WH)_{ij}) \tag{4}$$

$$D(A \parallel WH) = \sum_{ij} (A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij}) \tag{5}$$

Other interesting NMF algorithms are based on gradient descent methods (GDM) or, extending the GDM, on alternating least square computation [2,4].

2.1 NMF Applications

NMF can be in general used in problems that involve feature extraction, dimension reduction and object (pattern) recognition [5,6]. NMF has been deployed in several problem domains including signal processing, data clustering, spectroscopy and microarray data analysis [7].

NMF has been used for the separation of pitched musical instruments and drums from polyphonic music[5]. An extended nonnegative tensor decomposition was considered for musical sound source separation [8]. Constrained NMF was used as a tool for source spectra separation from magnetic resonance chemical shift imaging of human brain and multichannel EEG signals were analyzed by

NMF. Non-negative matrix factorization has been also studied as a tool for face and facial expression recognition.[5].

NMF has been used several times in text analysis. It was considered e.g. for topic detection in an email corpus [2] and gene tree labeling in biomedical literature [9]. The analysis of computer network traffic data by NMF has been shown as an efficient tool for intrusion detection [10].

2.2 NMF Drawbacks

NMF suffers from a number of drawbacks. NMF algorithms approach the factorization task as an optimization problem and there is no algorithm that would lead to a globally optimal solution of the factorization. Moreover, the factors W and H are usually initialized randomly, which leads to different factors of the same data matrix every time NMF is computed. Several studies have suggested different approaches for better factor initialization. Spherical $k-$means clustering, SVD-based initialization and initialization based on information from the original dataset were considered [5].

The selection of optimal r for a NMF application is unknown. The dimension of feature space r should correspond to the inner dimension of the input data, but it is usually unknown and hard to estimate.

Computing NMF for large data is a time consuming procedure. In most applications, new computation of the factors W and H or at least its parts is necessary whenever new data is processed. Because it typically takes from tens to thousands of iterations for common gradient descent NMF algorithms to converge to acceptable solution, it prevents NMF significantly from the use in applications sensitive to response time, such as online applications. In this paper, we present a novel method for transformation from high dimensional input data space to low dimensional feature space defined by a NMF factorization of the training dataset.

3 Fast Transformation to Feature Space

The key concept of the algorithm for fast transformation to feature space is generalized inverse. First, we define generalized inverse and summarize how to compute generalized inverse of a data matrix H.

3.1 Generalized Inverse

A generalized inverse of a matrix H is a matrix $H^\dagger$ associated in some way with H that exists for a class of matrices larger than the class of nonsingular matrices. It has some of the properties of the usual inverse and reduces to the usual inverse when H is nonsingular [7].

The generalized inverse can be also seen as a general way to find the solution to a system of linear equations

$$b = Hy \quad b \in \mathrm{R}^m, y \in \mathrm{R}^n, H \in \mathrm{R}^{m \times n} \tag{6}$$

The generalized inverse was first studied as general reciprocal by E. H. Moore in 1910s and rediscovered by R. Penrose in 1955. The generalized inverse is also known as Moore-Penrose pseudoinverse or simply pseudoinverse [7].

Definition 1. *Penrose equations*
For every finite matrix A (square or rectangular) of real or complex elements, there is a unique matrix $A^\dagger$ satisfying:

$$AA^\dagger A = A \qquad \text{(MP01)} \tag{7}$$

$$A^\dagger AA^\dagger = A^\dagger \qquad \text{(MP02)} \tag{8}$$

$$(AA^\dagger)^* = AA^\dagger \qquad \text{(MP03)} \tag{9}$$

$$(A^\dagger A)^* = A^\dagger A \qquad \text{(MP04)} \tag{10}$$

where A^* denotes the conjugate transpose of A [7].

The properties of generalized inverse include[7]:

- If $m = n$ and A is nonsingular, then $A^\dagger = A^{-1}$.
- If $m > n$ (there are more equations than variables in the system) then the pseudoinverse gives the solution y such that Ay is closest (in a least-squared sense) to the desired solution vector b. It minimizes

$$||b - Ay||$$

  and so provides approximate solution to an inconsistent linear system.
- If $m < n$, there is generally an infinite number of solutions, and the Moore-Penrose pseudoinverse is the solution whose vector $2-$norm is minimal.

3.2 Construction of Generalised Inverse

A generalised inverse can be constructed with the help of singular value decomposition. Some more restricted solutions also exist [7].

Definition 2. *Singular value decomposition*
Let $A \in \mathrm{R}^{m\times n}$. Then there are orthogonal matrices $U \in \mathrm{R}^{m\times m}$ and $V \in \mathrm{R}^{n\times n}$ such that

$$A = U\Sigma V^T \tag{11}$$

where $\Sigma \in \mathrm{R}^{m\times n}$ is a diagonal matrix

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & \sigma_p & 0 \end{bmatrix}$$

and

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0, \quad p = min\{m, n\}$$

are singular values of A.

Generalized inverse can be computed by a number of techniques, including methods based on matrix decompositon. The method based on SVD is numerically stable general algorithm to find pseudoinverse of any matrix A.

Definition 3. *Computation of generalized inverse*
Let $A = U\Sigma V^T$. Then

$$A^\dagger = V\Sigma^\dagger U^T \tag{12}$$

and $\Sigma \in \mathrm{R}^{n\times m}$

$$\Sigma^\dagger = \begin{bmatrix} \frac{1}{\sigma_1} & 0 \ldots 0 \\ 0 & \frac{1}{\sigma_2} \ldots 0 \\ \vdots & \vdots \ddots \vdots \\ 0 & \ldots\ldots \frac{1}{\sigma_p} \\ 0 & 0 \ldots 0 \end{bmatrix}$$

if any of σ_i is equal to 0, 0 *value is placed to the $i-th$ position on main diagonal of Σ.*

3.3 Transformation to Feature Space

Consider a $m \times n$ data matrix V containing high dimensional data records in every *row*. The factors $W^{m\times r}$ and $H^{r\times n}$ can be interpreted as a weighting matrix and a base matrix respectively. In another words, each row of H represents a base vector of the $r-$dimensional feature space and each row of W represents weights (coordinates) of corresponding point v in the $r-$dimensional feature space.

Then, we can derive the formula for computation of feature space weights:

$$V^{m\times n} \approx W^{m\times r} H^{r\times n} \tag{13}$$

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} H \tag{14}$$

$$v_i^T = w_i^T H \tag{15}$$

$$v_i^T = w_i^T H H^\dagger H \qquad \text{by (MP01)} \tag{16}$$

$$v_i^T H^\dagger = w_i^T H H^\dagger H H^\dagger \tag{17}$$

$$v_i^T H^\dagger = w_i^T H H^\dagger \qquad \text{by (MP02)} \tag{18}$$

where $H^\dagger$ is generalized inverse of H.

The coordinates w_l of every new data point v_l in the feature space defined by matrix H can be computed using

$$w_l^T H H^\dagger = v_l^T H^\dagger \tag{19}$$

Moreover, if the rows of H are linearly independent, $HH^\dagger \approx I$ and so (19) transforms to

$$w_l^T I = v_l^T H^\dagger \tag{20}$$

$$w_l^T \approx v_l^T H^\dagger \tag{21}$$

The formula (21) defines *pinvNMF*, an algorithm for fast transformation from input data space to feature space based on NMF and generalized inverse. The transformation consists of a multiplication of incoming data vector by Moore-Penrose pseudoinverse of the matrix H found by the factorization of training data. Needless to say, such vector matrix multiplication is significantly faster than new NMF computation for each new incoming data point.

4 Computational Experiments

We have implemented pinvNMF and performed computational experiments on a real world data collection to perform an initial verification of the ideas behinf pinvNMF and to investigate whether it can produce meaningfull feature vectors. Fot the experiments was used a subset of the MNIST handwritten digits collection [11]. It contains 70000 images of handwritten digits collected from 250 writers divided to training and testing dataset. The images are resized to 28×28 pixels, antialliased and centered on black background. As a result of antialliasing, the images contain black, white and grayscale pixels[11].

4.1 Comparing NMF and pinvNMF Feature Vectors

We took first 3000 images from the MNIST training dataset and filtered out zeros and ones. In the resulting training set were 624 images consisting of 339 ones and 285 zeros. Therefore, we obtained training data matrix V consisting of 624 rows and 784 columns. Each row in V contained one image from the training dataset. First 25 training images are shown in Figure 1.

The matrix V was subject to NMF for $r = 2$ to obtain factors W and H. According to our interpretation of W and H, the rows of H are base vectors of

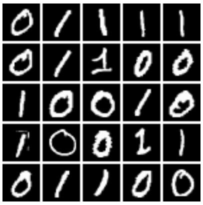


Fig. 1. First 25 0s and 1s in the training collection

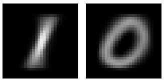


Fig. 2. Base images found by NMF

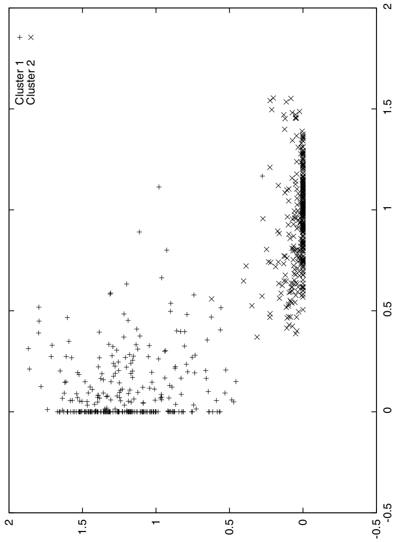


Fig. 3. The training data transformed to feature space

the feature space. The visualisation of rows of H shown in Figure 2 illustrates that NMF has found correct features in the training data, corresponding to two classes in the input data.

Because the feature space is 2−dimensional, we can easily visualise also the matrix W that contains the feature vectors (coordinates in feature space) of training data. Figure 3 shows that the points in feature space form two clusters, one for feature vectors with first coordinate greater than the second and one for feature vectors with second coordinate greater than the first. It is also apparent that some some training images are not clearly recognised as part of one or other cluster. Such vaguely classified data are represented in feature space by points that do not have dominant coordinate (their distance from both clusters is approximatelly the same). An example of such images is shown in Figure 4. Obviously, they are not typical representants of their respective classes and they can be considered as outliers in the training dataset.

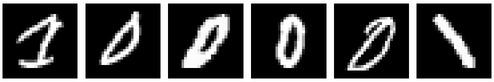


Fig. 4. Most outlying images in the training data set

We have computed generalized inverse of the matrix H. To observe the difference between features obtained by NMF and pinvNMF, we have computed the feature vectors (weights) w_i for all rows of the training matrix V. The feature space obtained by pinvNMF is shown in Figure 5. In contrast to the original feature vectors, the feature vectors created by pinvNMF contain also negative values. Nevertheless, the feature vetors also in this case form two clear clusters similar to the original clusters from Figure 3.

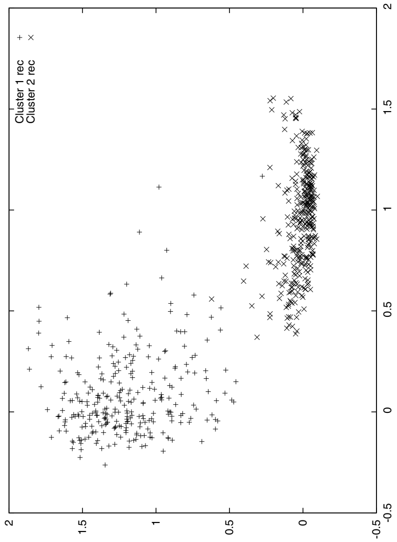


Fig. 5. Tranining data transformed to feature space by pinvNMF

This results suggest that pinvNMF can provide meaningfull feature vectors that are good approximation of the feature vectors obtained by the traditional NMF.

4.2 Classification of 0s and 1s from the MNIST Collection by pinvNMF

We have used pinvNMF for online classification of a subset of images (zeros and ones) from the MNIST collection. Similarly as in the previous experiment, first 624 zeros and ones were taken as training data and NMF was performed to compute matrix H. Then, next 5760 images in the dataset consisting of 2676 zeros and 3084 ones were classified using pinvNMF.

The clustering was simple because we have reduced the data to 2 dimensions. A feature vector was assigned to a cluster corresponding to its dominant coordinate when the absolute difference between its coordinates was larger than a threshold t. Otherwise, the image was labeled as an outlier and left unclassified.

The results of classification experiments for different values of threshold t are shown in Table 1. For lowest value of threshold t, the total classification error (percent of misclasified and unclassified images) was 1.23. The error was growing with larger threshold t due to increasing number of unclassified images.

Table 1. The results of pinvNMF classification

Threshold	Classification	Outlier	Total error
t	error	count	[percent]
0.1	26	45	1.233
0.2	18	95	1.962
0.3	9	170	3.108
0.4	7	297	5.278
0.5	5	544	9.531
0.6	2	853	14.84
0.7	2	1224	21.28

5 Conclusions

This paper introduces the pinvNMF, a fast dimension reduction method based on NMF and generalized matrix inverse. The notion of pinvNMF allows approximate transformation from data space to feature space by multiplication of data vector by generalized inverse $H^{\dagger}$of the matrix H obtained by NMF of training data set. The rows of the matrix H define base vectors of the feature space and the feature vectors represent images of points from high dimensional data space in the feature space.

In contrast to previous approaches that required new computation of the iterative gradient descent algorithm for new input data, pinvNMF allows fast tranformantion to feature space once the matrix H has been learned by classic nonnegative factorization of training data and $H^{\dagger}$ has been computed. The transformation consists of simple vector matrix multiplication only.

A simple real world high dimensional dataset was used to verify the ideas behind pinvNMF. We have computed feature vectors of high dimensional data points by pinvNMF. They contain both, positive and negative values, however they form in the feature space meaningfull clusters similar to those found by classic NMF. Further experiments have shown that the pinvNMF can be used to classify noisy images from two classes with error 1.23.

Numerical experiments have shown that pinvNMF is a promising algorithm for dimension reduction. Was able to find representative feature vectors for a testing collection. This ability together with its simplicity and speed opens for the matrix factorization task new application possibilities in areas such as online high dimensional data clustering, hig dimensional streaming data classification or online intrusion detection.

In our future work, we aim to thoroughly evaluate the pinvNMF algorithm on a classification and outlier detection tasks in real world data collections and its comparison with state of the art algorithms.

Acknowledgement

This work was supported by the Ministry of Industry and Trade of the Czech Republic, under the grant no. FR-TI1/420.

References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)
2. Berry, M.W., Browne, M., Langville, A.N., Pauca, P.V., Plemmons, R.J.: Algorithms and applications for approximate nonnegative matrix factorization
3. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS, pp. 556–562 (2000)
4. Shahnaz, F., Berry, M.W., Pauca, V.P., Plemmons, R.J.: Document clustering using nonnegative matrix factorization. Inf. Process. Manage 42(2), 373–386 (2006)
5. Buciu, I.: Non-negative matrix factorization, a new tool for feature extraction: Theory and applications. In: Proceedings of ICCCC, pp. 67–74 (2008)
6. Cichocki, A., Mørup, M., Smaragdis, P., Wang, W., Zdunek, R.: Editorial: Advances in nonnegative matrix and tensor factorization. Intell. Neuroscience 2008, 1–3 (2008)
7. Ben Israel, A., Greville, T.N.E.: Generalized Inverses. Theory and Applications, 2nd edn. Springer, Heidelberg (2003)
8. FitzGerald, D., Cranitch, M., Coyle, E.: Extended nonnegative tensor factorisation models for musical sound source separation. Intell. Neuroscience 2008, 15 pages, Article ID 872425 (2008)
9. Heinrich, K.E., Berry, M.W., Homayouni, R.: Gene tree labeling using nonnegative matrix factorization on biomedical literature. Intell. Neuroscience 2008, 12 pages, Article ID 276535 (2008)
10. Snášel, V., Platoš, J., Kröomer, P., Abraham, A.: Matrix factorization approach for feature deduction and design of intrusion detection systems. In: IAS 2008, Naples, Italy, pp. 172–179. IEEE Computer Society, Los Alamitos (2008)
11. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)

Frequent Words' Grammar Information in Chinese Chunking

Quan Qi, Li Liu, and Yue Chen

School of Computer Science, Beijing Institute of Technology,
Beijing, China
{qi_quan,greatliuli,chenyue}@bit.edu.cn

Abstract. In Chinese, frequent words, which always contain no significant information for information extraction, play an important role in the grammar structure of sentences. But the grammar information of these words is always ignored in Chinese segmentation. In this paper, for Chinese chunking, we dsesign an experiment to integrate the grammar information of frequent words and investigate the effect of this information on the chunking. We use conditional random fields for chunking, and rewrite the frequent words in the corpus to make them contain sentence structure information. The results show that the grammar information of frequent words, the number of which can be very small, can significantly increase the accuracy of the Chinese chunking.

Keywords: frequent words; grammar information; Chinese chunking; conditional random fields.

1 Introduction

In this paper, we investigate how frequent words and their grammar information affect Chinese segmentation, in particular the Chinese chunking task. Frequent words, which always have no useful information for information extraction, always play an important role in the sentence grammar analysis. These words' grammar information is always the key clue to understand and learn the meaning of sentences. Quantitative methods and corpus-based data collection have been used extensively in the study of language acquisition, language processing, historical linguistics and sociolinguistics, but they have been systematically excluded from the representations and the methods used in the study of formal grammars [1]. The grammar information of frequent words is always ignored in natural language processing (NLP) based on statistical methods. Knowing the effects of this information could help us solve many problems in NLP, such as chunking in informal text, like email and text on forum, which traditional chunking methods cannot gain a satisfying performance. Our experiments will investigate how these frequent words' grammar information improves the performance of statistical methods. Here, we select Chinese chunking as our task, and use conditional random fields (CRF), which are commonly-used for sequence segmentation, to finish this task.

In recent years, Chinese chunking problem, one of Chinese segmentation problems, has received a lot of attention in Chinese natural language processing area.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 434–443, 2010.

Chunking task is identifying chunks which have semantic meanings, as well as their belonging classes. Since Steven P. Abney first introduced the definition of chunks in 1991[2], much work has been done in this field. Many machine learning methods are utilized for chunking, such as Maximum Entropy, hidden Markov model [3] and conditional random fields [4]. Chinese chunking is first introduced by Li et al [5]. After that, a lot of work has been done on this task. Chinese chunking is a hard work, and now only some supervised learning methods can achieve satisfactory performance.

Conditional random fields are the supervised methods which are always applied on Chinese chunking. Comparing with widely-used hidden Markov model, CRF is a powerful sequence labeling model which modeling the conditional probabilities instead of generative probabilities. Now, CRFs are widely used in natural language processing field. CRFs allow us to utilize a large number of observation features which are not required to be irrelevant. Tan et al. [6] use CRFs for Chinese chunking and their experiments show that CRF-based method can get better results than Hidden Markov Model.

In this paper, we use the training and testing data distributed by the CIPS-ParsEval-2009 shared task. Section 2 lays out the basic theory of CRF models in detail. Section3 shows our CRF-based Chinese chunking method, and how to integrate the frequent words' grammar information into the system. Section 4 presents our experiments and explains the experimental results. Finally, Section 5 offers some conclusions regarding the results and some future work plans.

2 Conditional Random Fields

Conditional Random Fields (CRFs) are undirected graphical models used to calculate the conditional probability of values on designated output nodes given values assigned to other designated input nodes [7]. Comparing with other generative models such as hidden Markov Model, a key advantage of CRFs is their great flexibility to include a wide variety of arbitrary, non-independent features of the input (McCallum). Furthermore, it focuses on the modeling of tag sequences given input sequence, but not of the given sequences themselves. Comparing with other discriminative models such as MEMM, CRFs also overcomes the label bias shortcomings.

Formally, in our application, we let $x = (x_1, x_2, ..., x_T)$ denote observed Chinese word sequences which will be marked, and $y = (y_1, y_2, ..., y_T)$ the output of corresponding label sequences which shows the marking results. We use Linear-chain CRFs as our training model for its efficient training and decoding. The conditional probability $p(y \mid x)$ is given by

$$p(y \mid x; \Lambda) = \frac{1}{Z(x;\Lambda)} \exp\{\sum_k \lambda_k \sum_{t=1}^{T+1} f_k(y_{t-1}, y_{t,} x, t)\} . \quad (1)$$

The partition function $Z(x;\Lambda)$, as the normalization factor, is given by

$$Z(x;\Lambda) = \sum_y \exp\{\sum_k \lambda_k \sum_{t=1}^{T+1} f_k(y_{t-1}, y_t, x, t)\} . \quad (2)$$

Here, $f_k(y_{t-1}, y_t, x, t)$ are feature functions, and $\Lambda = \{\lambda_k\}$, as the estimation weights of the feature functions, are the parameters of our training CRF. Two types of features are commonly used in linear-chain CRF: a transition feature which indicates whether a transition between two nodes is performed, and an emission feature which indicates whether the observed feature is occurred when the given state is occurred. Equation (3) gives an example of transition features, and equation (4) gives an emission feature example.

$$f_k(y_{t-1}, y_t, x, t) = \delta(y_{t-1} = i)\delta(y_t = i) \quad . \tag{3}$$

$$f_k(y_{t-1}, y_t, x, t) = \delta(x_t = x)\delta(y_t = i) \quad . \tag{4}$$

In training process, our aim is to assign weights to λ_k, and make the log-likelihood get the maximum given the training corpus. To avoid overfitting, model parameters should be regularized. Equation (5) shows the log-likelihood. The second part of equation (5) is used for smoothing.

$$L_\Lambda = \sum_{i=1}^{m} \log p(y^{(i)} \mid x^{(i)}; \Lambda) - \frac{\|\Lambda\|^2}{2\delta^2} \quad . \tag{5}$$

The objective can be optimized using L-BFGS algorithm, and dynamic programming algorithm is used for decoding.

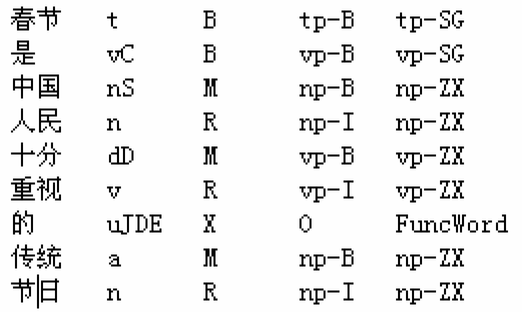
3 Chinese Chunking Using Frequent Words' Grammar Information

Our experiments are composed of four parts. The first part is getting the frequency of each word in the corpus. The second is extracting the grammar information of each word in sentences from the grammar tree corpus. The third is setting the size of the frequent words' set, and rewriting the sentences which contain the words belonging to the frequent words' set in the corpus. The last is training CRFs for Chinese chunking using the different corpus modified according to the size of the frequent word set, and testing their performance.

3.1 Corpus

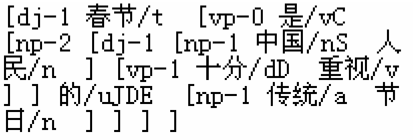
We use the training and testing data distributed by the CIPS-ParsEval-2009 shared task. Chinese Information Processing Society of Parsing evaluation proposes several Chinese parsing tasks. Two tasks' data are related to our experiments. One is Chinese basic chunk analysis. It is composed of 125 articles from China Daily. The other is Chinese grammar tree analysis. Our experiments are designed based on the chunking task. The corpuses from grammar tree tasks are used for extracting the grammar information of words. The two corpuses are all collected from Tshinghua Chinese Treebank (TCT). Data TCT are segmented, POS tagged, chunk annotated, grammar analyzed manually.

The format of the data in chunking corpus is shown in Fig. 1, and the format of the data from grammar tree corpus is shown in Fig. 2. Fig. 3 shows the tree representation of a sentence in the grammar tree corpus. Every sentence in the chunking corpus has its corresponding sentence in the grammar tree corpus; therefore, the grammar structure of sentences in the chunking corpus could be obtained from the grammar tree corpus.



春节	t	B	tp-B	tp-SG
是	vC	B	vp-B	vp-SG
中国	nS	M	np-B	np-ZX
人民	n	R	np-I	np-ZX
十分	dD	M	vp-B	vp-ZX
重视	v	R	vp-I	vp-ZX
的	uJDE	X	O	FuncWord
传统	a	M	np-B	np-ZX
节日	n	R	np-I	np-ZX

Fig. 1. The format of data in the chunking corpus. Column 1 shows the words in the sentence, Column 2 words' POS-tag, and Column 4 the tokens of basic chunks set to the words. There are 15 tokens for Chinese basic chunking [8].



```
[dj-1 春节/t  [vp-0 是/vC
[np-2 [dj-1 [np-1 中国/nS  人
民/n  ] [vp-1 十分/dD  重视/v
] ] 的/uJDE  [np-1 传统/a  节
日/n  ] ] ] ]
```

Fig. 2. The format of data in the grammar tree corpus

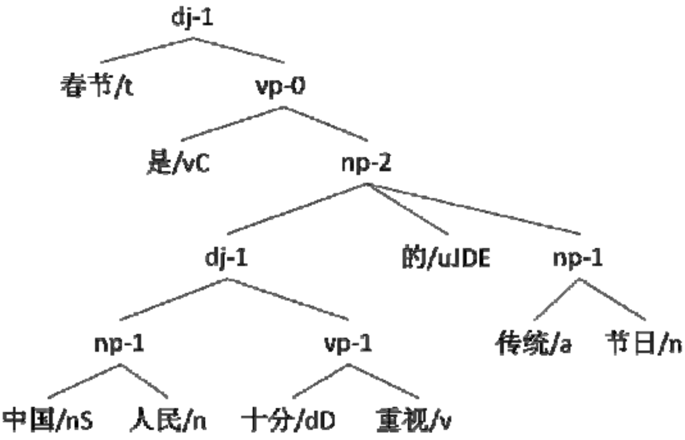


Fig. 3. The tree representation of the sentence in grammar tree corpus. The leaf nodes in the tree describe the words and their POS-tags used in the sentence, and the non-leaf nodes describe the grammar structure of the sentence.

3.2 Getting the Frequency and the Grammar Information of Words in Corpus

For getting the frequent words and their grammar information, we first count the frequency of each word in chunking corpus, and then extract the words' grammar information from the grammar tree corpus. Table 1 shows the top 10 frequently used words and their frequency in grammar tree corpus. In our experiments, punctuation symbols are also regarded as a kind of words for they often have grammar function. According to the analysis of corpus, the frequency of words in grammar tree corpus is roughly approximated with a Zipfian distribution [9], which means most words in corpus occur rarely.

Table 1. 10 most frequently used words in grammar tree corpus

Words	Frequency in corpus
，	6783
的	2888
、	2327
了	1791
一	1594
和	1445
在	1406
是	1307
不	843
个	836

In our experiments, grammar information of words refers to words' position in grammar trees. For each sentence in chunking corpus, its corresponding sentence can be extracted from the grammar tree corpus. Every word in the sentence could get its left sibling node, right sibling node, or both. We use a binary group to represent a word's grammar information in a sentence. One element in the binary group represents the word's left sibling identity and the other represents right sibling identity. If a word doesn't have left sibling or right sibling, the corresponding element value is -1. For example, in Fig.3, "的" has the binary group [dj, np], while "春节"'s binary group is [-1, np]. Table 4 shows "了"'s grammar information in the grammar tree corpus.

There are 15066 words in chunking corpus. Words with high frequency always have some kinds of frequently-occurring grammar information in corpus. As shown in Table 2, "了" with information [word, -1] is a good example for this phenomenon.

Table 2. The grammar information of "了" and corresponding frequency

Grammar information	Frequency in corpus
[word,-1]	1505
[vp,-1]	110
[word, word]	8
[dj,-1]	158
[ap,-1]	10

But frequent words also always have some kinds of grammar information that are rare in corpus. "了" with information [word, word] also gives such an example. This fact gives us a cue that frequent words always have some rarely used usage. Due to the low frequency, words with low frequency also have little grammar information.

3.3 Setting the Size of the Frequent Words' Set and Rewriting the Sentences in Chunking Corpus

For testing the effects of frequent words' grammar information on the performance of Chinese chunking, we integrate different quantity of words' grammar information. First, the words occurring in the corpus are sorted in reverse order according to the frequency. The n words in front of the queue are selected to generate a set of frequent words. We let S_n to denote this set. Then the chunking corpus is modified as followings: for every sentence in the corpus, every word in the sentence will be checked whether it is in S_n. If it can be found in S_n, rewrite this word as the word plus its grammar information in this sentence. How to get the word's grammar information are mentioned in 3.2. For example, if "的" shown in Fig. 1 is in S_n, "的" will be replaced in the sentence by "的[dj, np]" according to the sentence's corresponding grammar tree structure shown in Fig. 2. Thus, a modified chunking corpus which contains frequent words' grammar information is obtained.

3.4 Training CRFs for Chinese Chunking Using the Modified Corpuses

In our experiments, two sets of CRFs for Chinese chunking are trained. In Set 1, only words are utilized as features, and in Set 2, words and corresponding POS-tags are utilized as features. Sentences in chunking corpus are appropriately chunked and token-marked. Words in corpus are also appropriately POS-tagged. In the training process, words and corresponding POS-tags are extracted from sentences and used for calculating feature functions. Tokens for chunking in sentences are used to maximize the log-likelihood. In the decoding process, sentences with correct POS-tags are as the model input. Chunks will be tagged according the features extracted from sentences and the weights of feature functions obtained in training process.

We use CRF++ 0.53 [10] to train CRFs and do decoding. We modify the sample feature template for Chunking in CRF++, and use these templates as our experimental feature templates. Feature templates used for training CRFs in Set 1 are shown in Table 3, and feature templates for Set 2 are shown in Table 4.

Table 3. Feature templates used for training CRFs in Set 1

Word-related feature template	U00:%x[-2,0],U01:%x[-1,0] U02:%x[0,0],U03:%x[1,0] U04:%x[2,0] U05:%x[-1,0]/%x[0,0] U06:%x[0,0]/%x[1,0]
Bigram template	B

Table 4. Feature templates used for training CRFs in Set 2

Word-related feature template	U00:%x[-2,0],U01:%x[-1,0] U02:%x[0,0],U03:%x[1,0] U04:%x[2,0] U05:%x[-1,0]/%x[0,0] U06:%x[0,0]/%x[1,0]
POS-tag -related feature template	U10:%x[-2,1],U11:%x[-1,1] U12:%x[0,1],U13:%x[1,1] U14:%x[2,1], U15:%x[-2,1]/%x[-1,1] U16:%x[-1,1]/%x[0,1] U17:%x[0,1]/%x[1,1] U18:%x[1,1]/%x[2,1] U20:%x[-2,1]/%x[-1,1]/%x[0,1] U21:%x[-1,1]/%x[0,1]/%x[1,1] U22:%x[0,1]/%x[1,1]/%x[2,1]
Bigram template	B

The format of templates in Table 3 and 4 are represented as %x[row,col], in which %x represents the current word, the row represents the relative positions of the current words, and the column represents the absolute position of the column. B means that only combinations of previous output token and current token are used as bigram features [10].

For testing the effects of frequent words' grammar information on the performance of Chinese chunking CRFs, the modified corpuses by the different frequent words set S_n are created. For each corpus, two CRFs are trained. The first CRF belongs to Set 1, which uses words-related unigram feature templates and one bigram feature template (shown in Fig. 1); the second belongs to Set 2, which uses words-related and POS-tag-related unigram feature template. Each CRF in the same CRF set uses the same features but different training corpus. The experiments' results for testing the performance of CRFs will be shown in next section.

4 Experimental Results and Analysis

In our experiments, we train CRFs for Chinese basic chunking. From the above training corpus, we collect 80 files as our training data and 40 files for testing. Our training data contains 7090 sentences and 11296 words, and testing data contains 3470 sentences and 7228 words.

According to experiments' design in Section 3 and the counts of words in our training data, we create 100 modified corpuses. The frequent words sets corresponding to these corpuses are $S_{50 \cdot i}$, where $i = 0,1,2,...,100$, which means that 0, 50, 100... 4950 frequent words in the corpus are used to generate 100 frequent words sets. Correspondingly 100 corpuses are created according to the sets. In these corpuses, the word in Column 1 shown in Fig.1 is rewritten as the word plus its grammar information if it belongs to the frequent words set. For each modified corpus, two CRFs are trained using them. One model uses feature templates in Table 3, and the other one uses templates in Table 4. Thus we get the two set of CRFs as mentioned in Section 3.We use these two sets of CRFs to decode the testing data. For the testing data is from the training corpus of TCT, which contain the correct sequences of chunking tokens. We compare the chunking token sequences obtained from decoding process with these correct token sequences.

Fig. 4 and Table 5 show the change of precision, recall and F-1 values in the first CRF set, Set 1, and Fig. 5 and Table 6 show these values in the second set, Set 2. According to Fig. 4 and Table 5, the F-1 of Chinese chunking CRF, which is trained with no grammar-information-involved corpus, can only gain 0.7088. But when 50 most frequent words' grammar information are integrated into the corpus, the F-1 score for Chunking CRF increases greatly (0.0546). While integrating more words grammar information into the corpus, the performance of CRF increases slightly. Some number increasing even causes the performance's reduction. The reason may be that due to rare used words' low frequency and less grammar information, integrating their information may not change the word sequences greatly in the corpus. CRF trained using slightly changed corpus cannot gain an obvious performance increasing. This experiment gives us an idea that integrating the grammar information of a small number of the most frequent words can gain profitful rewards. CRFs in Set 2 utilize words and POS-tags as features. These features are commonly used for chunking. According to Fig.5 and Table 6, when involving frequent word grammar information into training corpus, the F-1 value for CRFs increase more than 0.02, which is a big increasing for Chinese chunking tasks.

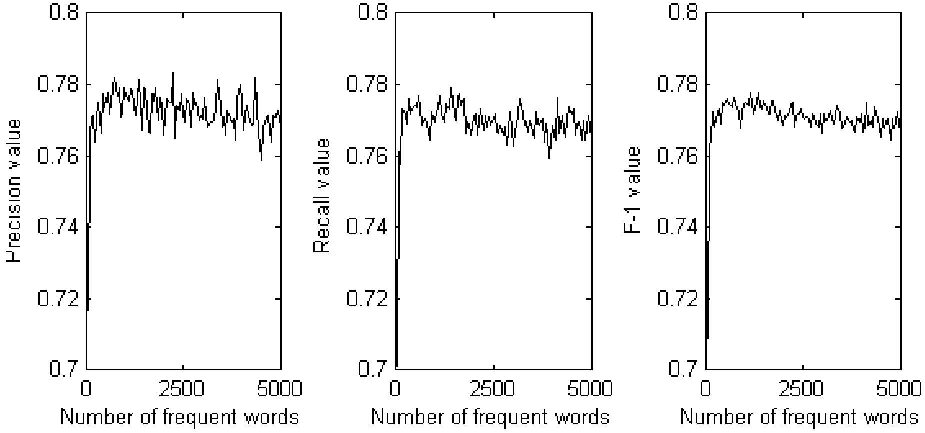


Fig. 4. This shows precision, recall and F-1 of CRFs which utilize words as feature. The number of words represents the how many words are in the frequent words set. The grammar information of words in this set are integrated into the chunking corpus.

Table 5. The Performance of the first 10 CRFs in Fig. 4

Number of frequent Words	Precision value	Recall value	F-1 value
0	0.7166	0.7012	0.7088
50	0.7656	0.7612	0.7634
100	0.7709	0.7577	0.7642
150	0.7715	0.7731	0.7723
200	0.7642	0.7720	0.7681
250	0.7749	0.7692	0.7721
300	0.7703	0.7758	0.7730
350	0.7661	0.7719	0.7690
400	0.7772	0.7732	0.7752
450	0.7738	0.7741	0.7739

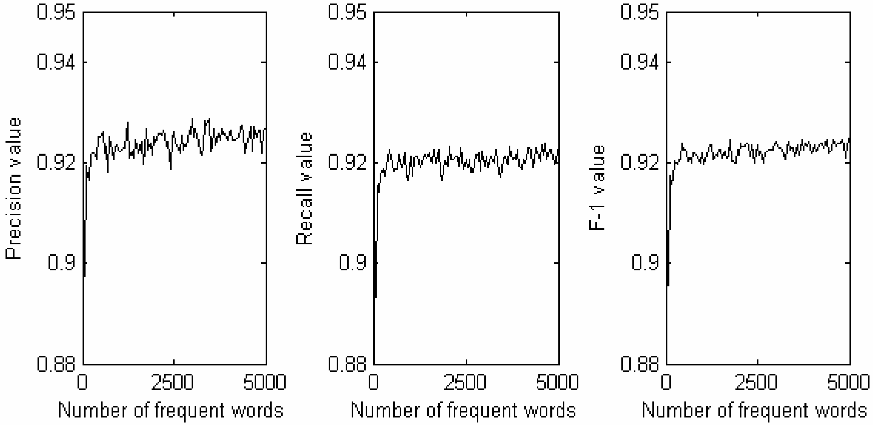


Fig. 5. This shows precision, recall, and F-1 of CRFs which utilize words and corresponding POS-tags as feature. The number of words represents the how many words are in the frequent words set. The grammar information of words in this set are integrated into chunking corpus.

Table 6. The Performance of the first 10 CRFs in Fig. 5

Number of frequent Words	Precision value	Recall value	F-1 value
0	0.8976	0.8934	0.8955
50	0.9194	0.9156	0.9175
100	0.9173	0.9141	0.9157
150	0.9164	0.9174	0.9169
200	0.9217	0.9188	0.9202
250	0.9219	0.9175	0.9197
300	0.9217	0.9192	0.9205
350	0.9205	0.9210	0.9207
400	0.9250	0.9226	0.9238
450	0.9250	0.9198	0.9224

5 Conclusions

In this paper, we devise an experiment which integrates the grammar information of frequent words for Chinese chunking task, one of Chinese segmentation tasks, and investigate how this information affects the results when using CRFs for training and decoding. The experiment results show that integrating only a small number of the most frequent words' grammar information can increase the performance of Chinese chunking greatly, but integrating a large number of the most frequent words' grammar information cannot bring much more performance increase. We believe that this approach holds great promise beyond its already demonstrated success.

In the future, we will find ways to calculate the grammar information of frequent words in sentences instead of extracting it from the grammar tree corpus. For this task will not meet data sparseness problem, obtaining frequent words' grammar information may be not very difficult. We also plan to check how frequent grammar information of frequent words affects the results of chunking on informal Chinese, such as email and text on forum. Informal text processing is a challenging task, for the traditional chunking methods cannot gain a satisfying performance on it.

Acknowledgments. We thank CIPS-ParsEval-2009 for providing the corpus used in this paper.

References

1. Merlo, P., Stevenson, S.: Structure and Frequency in Verb Classification. In: The Thirtieth Incontro di Grammatica Generativa, Venice, Italy, pp. 43–61 (2004)
2. Abney, S.: Parsing by Chunks. Principle-Based Parsing, pp. 257–278. Kluwer Academic Publishers, Dordrecht (1991)
3. Zhou, G.D., Su, J., Tey, T.G.: Hybrid text chunking. In: Proceedings of the CoNLL 2000, pp. 163–165. Association for Computational, Lisbon (2000)
4. Lafferty, McCallum, J., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. ICML 2001, Williamstown, MA, USA, pp. 282–289 (2001)
5. Li, H., Webster, J.J., Kit, C., Yao, T.: Transductive hmm based Chinese text chunking. In: Proceedings of IEEE NLPKE 2003, Beijing, China, pp. 257–262 (2003)
6. Tan, Y., Yao, T., Chen, Q., Zhu, Q.: Chinese chunk identification using svms plus sigmoid. In: Su, K.-Y., Tsujii, J., Lee, J.-H., Kwong, O.Y. (eds.) IJCNLP 2004. LNCS (LNAI), vol. 3248, pp. 527–536. Springer, Heidelberg (2004)
7. McCallum, A.: Efficiently Inducing Features of Conditional Random Fields. In: Proceedings of Conference on Uncertainty in Articifical Intelligence (UAI), Acapulco, Mexico (2003)
8. Chen, W., Zhang, Y., Isahara, Hitoshi.: An empirical study of chinese chunking. In: COLING/ACL 2006 (Poster Sessions), Sydney, Australia (2006)
9. Christopher, D., Manning, Hinrich, S.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
10. CRFsuiteVer.0.53, `http://www.chokkan.org/software/crfsuite/tutorial.html`

Multilabel Classification Using Error Correction Codes

Abbas Z. Kouzani

School of Engineering, Deakin University,
Geelong,Victoria 3217, Australia
kouzani@deakin.edu.au

Abstract. This paper presents a multilabel classification method that employs an error correction code together with a base ensemble learner to deal with multilabel data. It explores two different error correction codes: convolutional code and BCH code. A random forest learner is used as its based learner. The performance of the proposed method is evaluated experimentally. The popular multilabel yeast dataset is used for benchmarking. The results are compared against those of several exiting approaches. The proposed method performs well against its counterparts.

Keywords: Multilabel data, classification, error correction codes, ensemble learners, random forests.

1 Introduction

In some classification problems, the input patterns can belong to multiple classes simultaneously. Singlelabel classification is defined as learning from a collection of instances that each is related to only one label *l* from a set of class labels *L*. Multilabel classification is defined as learning from a set of instances that each is associated with a set of labels $Y \subseteq L$ [1]. A sample multilabel data is shown is Table 1. It consistsof-five instances. Each instance contains six features. There are five classes *L = {C1, C2, C3, C4, C5}*. Each instance belongs to one class or multiple classes. For example, Instance 3 belongs only to Class 4, whilst Instance 1 belongs to both Classes 1 and 3.

Conventional classification algorithms cannot directly deal with multilabel data. The reason is that those algorithms can only predict a single class attribute. Multilabel classification has received some attentions in the past several years. Many methods have been developed for classification of multilabel data. Tsoumakas et al. have published several excellent review articles on multilabel classification and ranking [1-2]. Several existing multilabel classification methods are described in the following.

Brinker et al. [3] state that Binary Relevance (BR) considers the prediction of each label as an independent binary classification task. It trains a separate binary relevance model for each possible label using all examples related to the label as positive examples and all other examples as negative examples. For classifying a new instance, all binary predictions are obtained and then the set of labels corresponding to positive relevance classification is associated with the instance.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 444–454, 2010.

Table 1. Example of multilabel data

Instances	Features						Classes				
							C1	C2	C3	C4	C5
1	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	1	0	1	0	0
2	F_{21}	F_{22}	F_{23}	F_{24}	F_{25}	F_{26}	1	1	0	0	1
3	F_{31}	F_{32}	F_{33}	F_{34}	F_{35}	F_{36}	0	0	0	1	0
4	F_{41}	F_{42}	F_{43}	F_{44}	F_{45}	F_{46}	1	1	1	0	0
5	F_{51}	F_{52}	F_{53}	F_{54}	F_{55}	F_{56}	0	0	0	0	1

Zhang and Zhou [4] report a multi-label lazy learning approach which is derived from the k-Nearest Neighbour (kNN) and named ML-kNN. For each unseen instance, its *k* nearest neighbors in the training set are identified. Then, based on statistical information gained from the label sets of these neighboring instances, maximum a posteriori principle is used to determine the label set for the unseen instance.

Zhang and Zhou [5] present a neural network-based algorithm that is Backpropagation for Multi-Label Learning named BP-MLL. It is based on the backpropagation algorithm but uses a specific error function that captures the characteristics of multi-label learning. The labels belonging to an instance are ranked higher than those not belonging to that instance.

Tsoumakas and Vlahavas [6] propose RAndom K-labELsets (RAKEL) which is an ensemble method for multilabel classification based on random projections of the label space. An ensemble of Label Powerset (LP) classifiers is trained on smaller size of label subset randomly selected from the training data. RAKEL takes into account label correlations by using single-label classifiers that are applied on subtasks with manageable number of labels and adequate number of examples per label.

Nasierding et al. [7] present a multi-label classification framework for problems with large numbers of labels. It comprises an initial clustering phase that breaks the original training set into several disjoint clusters of data. It then trains a multilabel classifier from the data of each cluster. Given a new test instance, the framework first finds the nearest cluster and then applies the corresponding model.

Tsoumakas et al. [8] present Hierarchy Of Multilabel classifiERs (HOMER) that include a number of classifiers each one dealing with a much smaller set of labels and a more balanced example distribution. HOMER first organizes labels into a tree-shaped hierarchy by recursively partitioning the set of labels into a number of nodes using a clustering algorithm. It next builds a multilabel classifier at each node.

Several other important works can be also found in the literature [9-14]. The multi-label classification algorithms are categorised into two groups [2]: problem transformation and algorithm adaptation. The methods belonging to the first group are independent of the classification algorithm. They transform the learning task into one or more single-label classification tasks. Thus, the exiting single-label classification algorithms can be employed to tackle the task. On the other hand, the methods belonging to the second group extend specific single-label classification algorithms so that they can directly handle multilabel data. Majority of the multilabel classification work reported in the literature belong to the problem transformation group.

The motivation of the work reported in this paper is the need to improve the performance of the multilabel classification methods. The paper explores the use of error correction techniques for multilabel classification. Error correction techniques can correct certain transmission errors. The encoder sends out the information symbols appended with additional redundant symbols. The decoder utilises those extra redundant symbols to correct the transmission errors if any.

This paper presents a problem transformation method that employs an error correction technique together with a base ensemble learner to form a method that can deal with multilabel classification problems improving the performance of several exiting methods. It explores two different error correction techniques: a convolutional coding technique and a block coding technique. The description of the theoretical framework as well as the proposed method is given in the following sections.

2 Error Correction Codes

Error correction codes are categorised into a number of groups including convolutional codes and block codes. The convolutional codes depend on the current set of input symbols and some of the past input symbols. Thus, they include memory devices in their architecture. On the other hand, the block codes are memoryless systems whose computations rely only on the current information symbols. In this paper, a convolutional code and a block code are used to form a problem transformation method for multilabel classification.

2.1 Convolutional Code

Convolutional codes map message to code bits sequentially by convolving a sequence of message bits with generator sequences. A binary convolutional code [15] is represented by a three-tuple (n, k, m), where k denotes the number of bit per message symbol, n represents the number of bits per code symbol, and m denotes the number of previous k-bit message blocks that are stored within memory devices. The n bits of the code are linear combinations of the k bits of the message as well as the previous $m \times k$ bits of the message. m is referred to as the memory order of the code. The encoders of convolutional codes can be represented as follows:

$$\boldsymbol{v}_j = \sum_{i=1}^{k} \boldsymbol{u}_i * \boldsymbol{g}_j^{(i)} \tag{1}$$

where * denotes the convolution operation, $\boldsymbol{u}$ is the message sequence, $\boldsymbol{v}$ is the codewod, and $\boldsymbol{g}_j^{(i)}$ denotes the impulse response of the i-th message sequence with the response of the j-th output. The impulse responses are called the generator sequences of the encoder. The message and code sequences can be arranged as follows:

$$\begin{aligned} \boldsymbol{u} &= (u_{1,0}, u_{2,0}, \cdots, u_{k,0}, u_{1,1}, u_{2,1}, \cdots, u_{k,1}, \cdots, u_{1,l}, u_{2,l}, \cdots, u_{k,l}, \cdots) \\ \boldsymbol{v} &= (v_{1,0}, v_{2,0}, \cdots, v_{n,0}, v_{1,1}, v_{2,1}, \cdots, v_{n,1}, \cdots, v_{1,l}, v_{2,l}, \cdots, v_{n,l}, \cdots) \end{aligned} \tag{2}$$

The relation between $\boldsymbol{u}$ and $\boldsymbol{v}$ can be expressed as follows:

$$\boldsymbol{v} = \boldsymbol{u}.\boldsymbol{G} \tag{3}$$

where $\boldsymbol{G}$ is referred to as the generator matrix of the code:

$$G = \begin{bmatrix} G_0 & G_1 & G_2 & \cdots & G_m & & \\ & G_0 & G_1 & \cdots & G_{m-1} & G_m & \\ & & G_0 & \cdots & G_{m-2} & G_{m-1} & G_m \\ & & & \ddots & & & \ddots \end{bmatrix} \tag{4}$$

including $k \times n$ sub-matrices

$$G_l = \begin{bmatrix} g_{1,l}^{(1)} & g_{2,l}^{(1)} & \dots & g_{n,l}^{(1)} \\ g_{1,l}^{(2)} & g_{2,l}^{(2)} & \cdots & g_{n,l}^{(2)} \\ \vdots & \vdots & & \vdots \\ g_{1,l}^{(k)} & g_{2,l}^{(k)} & \cdots & g_{n,l}^{(k)} \end{bmatrix} \tag{5}$$

where $g_{j,l}^{(i)}$ is the impulse response of the i-th input in relation to j-th output.

2.2 Bose, Ray-Chaudhuri, Hocquenghem Code

Bose, Ray-Chaudhuri, Hocquenghem (BCH) Code is a multilevel, cyclic, error-correcting, variable-length digital code that can correct errors up to about 25% of the total number of digits [16-17]. The original applications of BCH code were limited to binary codes of length 2^m-1 for some integer m. These were extended later to the non-binary codes with symbols from Galois field GF(q). Galois field is a field with a finite field order (number of elements). The order of a Galois field is always a prime or a power of a prime number. GF(q) is called the prime field of order q where the q elements are 0,1, …, q-1. BCH codes are cyclic codes and can be specified by a generator polynomial. For any integer $m \geq 3$ and $t < 2^{m-1}$, there exists a primitive BCH code with the following parameters:

$$\begin{aligned} & n = 2^m - 1 \\ & n - k \leq mt \\ & d_{min} \geq 2t + 1 \end{aligned} \tag{6}$$

The code can correct t or fewer random errors over a span of 2^m-1 bit positions. The code is called a t-error-correcting BCH code over GF(q) of length n. This code is specified as follows:

1. Specify the smallest m such that GF(q^m) has a primitive nth root of unity β.
2. Select a nonnegative integer b. Frequently, b=1.
3. Form a list of 2t consecutive powers of β: $\beta^b, \beta^{b+1}, \dots, \beta^{b+2t-1}$. Determine the minimal polynomial with respect to GF(q) of each of these powers of β.
4. The generator polynomial $g(x)$ is the least common multiple (LCM) of these minimal polynomials. The code is a $(n, n\text{-} deg(g(x)))$ cyclic code.

Due to the fact that the generator is constructed using minimal polynomials with respect to GF(q), the generator $g(x)$ has coefficients in GF(q), and the code is over GF(q). Two fields are involved in the construction of BCH codes. GF(q) is where the generator polynomial has its coefficients and is the field where the elements of the codewords are. GF(q^m) is the field where the generator polynomial has its roots. For encoding purpose, it is adequate to work only with GF(q). However, decoding requires operations in GF(q^m). For binary BCH codes, let α be a primitive element in GF(2^m). For $1 \leq i \leq t$, let $\Phi_{2i-1}(x)$ be the minimum polynomial of the field element

α^{2i-1}. The degree of $\Phi_{2i-1}(x)$ is m or a factor of m. The generator polynomial g(x) of t-error-correcting BCH codes of length 2^m-1 is given by:

$$g(x) = LCM\{\Phi_1(x), \Phi_3(x), \ldots, \Phi_{2t-1}(x)\} \quad (7)$$

The first explicit decoding algorithm for binary BCH codes was Peterson's algorithm. Berlekamp introduced the first truly efficient decoding algorithm for both binary and nonbinary BCH codes. This was further developed by Massey and is usually called the Berlekamp-Massey decoding algorithm.

Consider a BCH code with $n = 2^{m-1}$ and generator polynomial $g(x)$. Suppose a code polynomial $c(x) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1}$ is transmitted. Let $r(x) = r_0 + r_1 x + \ldots + r_{n-1} x^{n-1}$ be the received polynomial. Then, $r(x) = c(x) + e(x),$ where $e(x)$ is the error polynomial. To check whether $r(x)$ is a code polynomial, $r(\alpha) = r(\alpha^2) = \ldots = r(\alpha^{2t}) = 0$ is tested. If yes, then $r(x)$ is a code polynomial, otherwise $r(x)$ is not a code polynomial and the presence of errors is detected. The decoding procedure includes three steps: syndrome calculation, error pattern specification, and error correction.

3 Random Forest Learner

The random forest [18-19] refers to a collection of tree structures classifiers $\{h(x, \Theta), k = 1\}$ where $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x. For each observation, each individual tree votes for one class and the forests predicts the class that has most votes. The user has to specify the number of randomly selected variables m_{try} to be searched through for the best split at each node. The node is split using the best among a subset of predictors randomly chosen at that node. The largest tree possible is grown and is unpruned. The root node of each tree in the forest contains a bootstrap sample from the original data as the training set. The observations that are not in the training set are referred to as "out-of-bag" observations. The out-of-bag instances for classifier $h_k(x)$ are represented as $O_k(x)$. $Q(x,y_j)$ is voted as the out-of-bag proportion for class y_j based on input x and the approximation of $P(h\ (x) = y_j)$ [20]:

$$Q(x, y_j) = \frac{\sum_{k=1}^{K} I\left(h_k(x) = y_j\,; (x, y) \in O_k\right)}{\sum_{k=1}^{K} I\left(h_k(x)\,; (x, y) \in O_k\right)} \quad (8)$$

where $I(.)$ is the indicator function. The generalisation error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. The margin function is calculated to find the extent which the average vote for the right class y exceeds the average vote for any other class:

$$mr(x, y) = P(h(x) = y) - \max_{\substack{j=1 \\ j \neq y}}^{c} P\left(h(x) = y_j\right) \quad (9)$$

$Q(x,y)$and $Q(x,y_j)$ are used to estimate the function. The expected margin is known as strength. The strength formula is:

$$s = \frac{1}{n} \sum_{i=1}^{n} \left(Q(x_i, y) - \max_{j \neq y} Q_{j=1}^{c}(x_i, y_j) \right) \tag{10}$$

The variance of the margin over the square of the standard deviation of the forests is known as the average correlation as follows:

$$\bar{\rho} = \frac{\mathrm{var}\,(mr)}{sd\,(h(\,))^2} = \frac{\frac{1}{n} \sum_{i=1}^{n} \left(Q(x_i, y) - \max_{j \neq y} Q_{j=1}^{c}(x_i, y_j) \right) - s^2}{\left(\frac{1}{k} \sum_{t=1}^{K} \sqrt{p_k + \hat{p}} + (p_k - \hat{p})^2 \right)^2} \tag{11}$$

where $p_k = \frac{\sum_{(x_i, y) \in O_k} I\,(h_k(x) = y)}{\sum_{(x_i, y) \in O_k} I\,(h_k(x))}$ is the out-of-bag estimate of $P(h_k\ (x) = y)$, $\hat{p}_k = \frac{\sum_{(x_i, y) \in O_k} I\,(h_k(x) = \hat{y}_j)}{\sum_{(x_i, y) \in O_k} I\,(h_k(x))}$ is the out-of-bag estimate of $P(h_k(x) = \hat{y}_j)$, and $\hat{y}_j = \mathrm{argmax}_{j=1, j \neq y}^{c} Q(x, y_j)$ is the estimation for x instance in the training set represented as $Q(x,y_j)$. The training data are run down each tree. If observations i and j both end up in the same terminal node, the similarity between i and j is increased by one. At the end of the forest construction, the similarities are symmetrised and divided by the number of trees. The similarity between an observation and itself is set to one.

4 Proposed Multilabel Classification Method

The proposed multilabel classification method is a problem transformation approach that employs an error correction code together with a base ensemble learner to deal with multilabel data. For the error correction code, two codes are explored: the convolutional code and the BCH Code. For the base ensemble learner, the random forest learner is employed.

4.1 MultiLabel Convolutional Random Forest

The first classifier is called MultiLabel Convolutional Random Forest (ML-CRF). The block diagram description of ML-CRF is shown in Fig. 1(a). The method first transforms the set of labels L using the convolutional encoding algorithm. For a dataset with c classes, each set of labels that is associated with an instance containing c binary values is treated as a message codeword and is transformed into a t-bit binary values where $t > c$. The t-bit binary word is called the encoded message. Then, the multilable classification problem is decomposed into t binary classification problems. Next, t random forests classifiers are developed one for each binary class. After that, the t classification decisions of the t binary classifiers are transformed using the convolutional decoding algorithm and again c binary values are obtained. Therefore, the advantage of the error-correcting properties of the convolutional code is incorporated into the system that helps correct possible misclassification of some individual t binary classifiers. For classification of a new instance, its features are independently presented to the t binary classifiers. Then the t classification decisions of the t binary classifiers are transformed into *c* binary values using the convolutional decoding algorithm. The error-correcting is applied during this transformation that helps correct possible misclassification of some individual *t* binary classifiers. The bits of the *c* resultant binary values that are '1' indicate that the instance belong to the associated class.

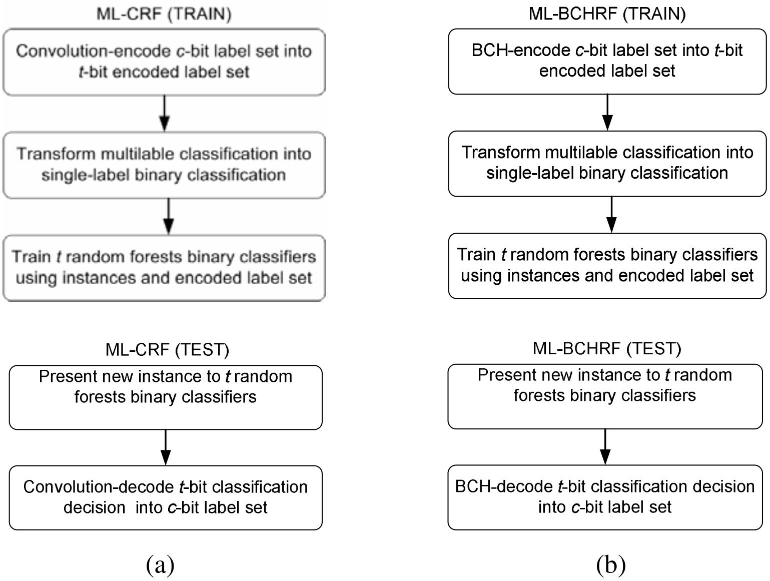

ML-CRF (TRAIN)
Convolution-encode c-bit label set into t-bit encoded label set
Transform multilable classification into single-label binary classification
Train t random forests binary classifiers using instances and encoded label set
ML-BCHRF (TRAIN)
BCH-encode c-bit label set into t-bit encoded label set
Transform multilable classification into single-label binary classification
Train t random forests binary classifiers using instances and encoded label set
ML-CRF (TEST)
Present new instance to t random forests binary classifiers
Convolution-decode t-bit classification decision into c-bit label set
ML-BCHRF (TEST)
Present new instance to t random forests binary classifiers
BCH-decode t-bit classification decision into c-bit label set
(a)
(b)

Fig. 1. Description of the training and the test stages of (a) ML-CRF and (b) ML-BCHRF

4.2 MultiLabel Bose, Ray-Chaudhuri, Hocquenghem Random Forest

The second classifier is called MultiLabel Bose, Ray-Chaudhuri, Hocquenghem Random Forests (ML-BCHRF). The block diagram description of ML-BCHRF is shown in Fig. 1(b). The method first transforms the set of labels *L* using the BCH encoding algorithm. Each set of labels is treated as a message codeword and is transformed into a *t*-bit binary values where $t > c$. Next, *t* random forests classifiers are developed one for each binary class. After that, the *t* classification decisions of the *t* binary classifiers are transformed using the BCH decoding algorithm and again *c* binary values are obtained. The advantage of the error-correcting properties of the BCH code is incorporated into the system that helps correct possible misclassification of some individual *t* binary classifiers. For classification of a new instance, its features are presented to the *t* binary classifiers. Then, the *t* classification decisions are transformed into *c* binary values using the BCH decoding algorithm. The bits of the *c* resultant binary values that are '1' indicate the classes that the instance belongs to.

5 Results

To evaluate ML-CRF and ML-BCHRF, their performances were evaluated against a number of exiting methods on the multilabel Yeast dataset. This dataset is among the

Table 2. Characteristics of the yeast dataset

Name	Features	Classes	Train	Test
Yeast	103	14	1500	917

popular benchmark datasets for multi-label classification. Its characteristics are presented in Table 2.

The evaluation measures for multilabel classification are different from those of singlelabel classification. Hamming-Loss [2] is a popular measure that have been used in most of the multilabel classification works. Smaller values of Hamming-Loss are indicative of better performances of the multilabel classifier. In this work, Hamming-Loss is used for performance comparison of the reported multilabel classification approaches.

We trained and tested both ML-CRF and ML-BCHRF on the Yeast dataset. ML-CRF used the convolutional feedback encoder shown in Fig. 2. On the other hand, ML-BCHRF employed the (63, 16) BCH encoder where two dummy bits were added to the label set making it have 16 binary bits.

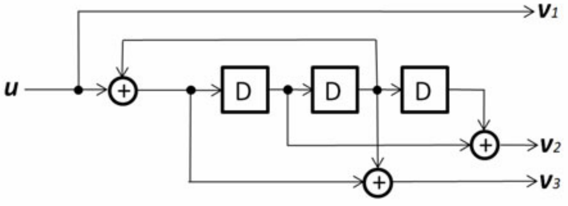


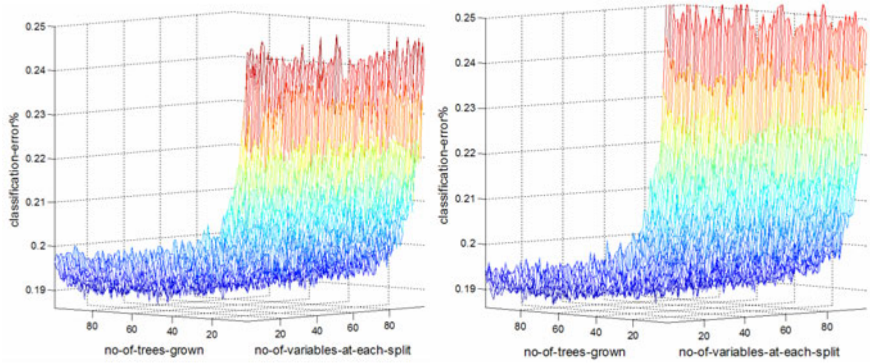
Fig. 2. Block diagram description of the convolutional feedback encoder used in ML-CRF

Both ML-CRF and ML-BCHRF employ the random forest learner as their base classifier. The random forest learner has two important parameters, called number-of-trees-to-grow and number-of-variables-at-each-split, that can be varied to get the best number of tree within the forest for the specific training data. Using the 1500 example in the training set, the two parameters were varied as follows. The first parameter, no-of-trees-grown, was varied from 2 to 99 with an increment of 1. For each tree grown, the second parameter, no-of-variables-at-each-split, was also varied from 2 to 99 with an increment of 1. For each trained classifier that was made of a specific number of trees and variables, the 917 test examples were applied and the classification error was calculated. Fig. 3 shows the achieved classification errors for ML-CRF and ML-BCHRF for the test examples. The best three classification errors for ML-CRF and ML-BCHRF are given in Table 3.

The best results for ML-CRF and ML-BCHRF were compared against those of MMP [9], AdaBoost.HM [10], ADTBoost.HM [11], LP [6], BR [3], RankSVM [12], ML-KNN [4], RAKEL [6], 1vsAll SVM [13], and ML-PC [14] found in the literature. Table 4 shows these results.

The experimental results show that ML-CRF and ML-BCHRF have performed better than their reported counterparts. The best performance has been achieved by ML-BCHRF for no-of-trees-grown = 89 and no-of-variables-at-each-split = 26.

The associated Hamming-Loss is 0.1862. The reason for the demonstrated performance relates to the mixture of: (i) the error correcting capability of the convolutional and BCH codes, and (ii) the g performance of the random forest learner.



(a) (b)

Fig. 3. Classification errors for the 917 test examples: (a) ML-CRF and (b) ML-BCHRF

Table 3. The best three classification errors for ML-CRF and ML-BCHRF

Method	1st	2nd	3rd

ML-CRF	Hamming-Loss=0.1874
	No-of-Trees=86
	No-of-Variables=42
		Hamming-Loss=0.1876
		No-of-Trees=59
		No-of-Variables=15
			Hamming-Loss=0.1876
			No-of-Trees=95
			No-of-Variables=14

ML-BCHRF	Hamming-Loss=0.1862
	No-of-Trees=89
	No-of-Variables=26
		Hamming-Loss=0.1871
		No-of-Trees=89
		No-of-Variables=11
			Hamming-Loss=0.1872
			No-of-Trees=53
			No-of-Variables=43

Table 4. Results

Method	Hamming-Loss
MMP	0.297
ADTBoost.HM	0.215
AdaBoost.HM	0.210
LP	0.202
BR	0.199
RankSVM	0.196
ML-KNN	0.195
RAKEL	0.193
1vsAll SVM	0.191
ML-PC	0.189
ML-CRF	0.187
ML-BCHRF	0.186

5 Conclusions

A method was proposed that encodes labels using an error correction code, and then transforms the problem into binary classification. A random forests classifier is developed for each binary class. The classification decisions are decoded using the convolutional and BCH decoding algorithms and again binary values are obtained. The experimental results show that the proposed method has performed better than its reported counterparts when tested on the Yeast benchmark dataset.

Acknowledgments. The author would like to thank Ms Gulisong Nasierding for various discussions on multilabel classification systems.

References

[1] Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining 3(3), 1–13 (2007)

[2] Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, 2nd edn. (2010)

[3] Brinker, K., Furnkranz, J., Hullermeier, E.: A united model for multilabel classification and ranking. In: Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006), Riva del Garda, Italy, pp. 489–493 (2006)

[4] Zhang, M.-L., Zhou, Z.-H.: A k-nearest neighbour based algorithm for multi-label classification. In: Proceedings of IEEE GrC 2005, Beijing, China, pp. 718–721 (2005)

[5] Zhang, M.-L., Zhou, Z.-H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering 18(10), 1338–1351 (2006)

[6] Tsoumakas, G., Vlahavas, I.: Random k-labelsets: An ensemble method for multilabel classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 406–417. Springer, Heidelberg (2007)

[7] Nasierding, G., Tsoumakas, G., Kouzani, A.Z.: Clustering based multi-label classification for image annotation and retrieval. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, San Antonio, Texas, USA, October 11-14, pp. 4514–4519 (2009)

[8] Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and efficient multilabel classification in domains with large number of labels. In: Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD 2008), Antwerp, Belgium (2008)

[9] Crammer, C., Singer, Y.: A family of additive online algorithms for category ranking. Machine Learning Research 3, 1025–1058 (2003)

[10] Schapire, R.E., Singer, Y.: BoosTexter: A boostingbased system for text categorization. Machine Learning 39, 135–168 (2000)

[11] Comite, F.D., Gilleron, R., Tommasi, M.: Learning multi-label alternating decision tree from texts and data. In: Perner, P., Rosenfeld, A. (eds.) MLDM 2003. LNCS, vol. 2734, pp. 35–49. Springer, Heidelberg (2003)

[12] Elisseeff, A., Weston, J.: A kernel method for multilabelled classification. In: Proceedings of NIPS 2002, Cambridge (2002)

[13] Platt, J.: Probabilistic Outputs for Support Vector Machines and Comparison to Regularized Likelihood Methods. In: Adv. in Large Margin Classifiers, pp. 61–74. MIT Press, Cambridge (1999)

[14] Petrovskiy, M.: Paired comparisons method for solving multi-label learning problem. In: Proceedings of the Sixth International Conference on Hybrid Intelligent Systems (2006)

[15] Han, Y.S.: Introduction to binary convolutional codes. Notes, Graduate Institute of Communication Engineering, National Taipei University

[16] Wei, C.-H.: Channel coding notes, `http://cwww.ee.nctu.edu.tw/course/channel_coding/`

[17] Rudra, A.: BCH codes. Lecture 24, Error Correcting Codes: Combinatorics, Algorithms and Applications (2009)

[18] Boinee, P., De Angelis, A., Foresti, G.L.: Meta random forests. International Journal of Computational Intelligence 2, 138–147 (2006)

[19] Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)

[20] Robnik-Sikonja, M.: Improving random forests. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 359–370. Springer, Heidelberg (2004)

Test-Cost Sensitive Classification Using Greedy Algorithm on Training Data

Chang Wan

School of Information Science and Technology,
Sun Yat-sen University,
Guangzhou Guangdong, China
wanchang@mail2.sysu.edu.cn

Abstract. Much work has been done to deal with the test-cost sensitive learning on data with missing values. There is a confliction of efficiency and accuracy among previous strategies. Sequential test strategies have high accuracy but low efficiency because of their sequential property. Some batch strategies have high efficiency but lead to poor performance since they make all decisions at one time using initial information. In this paper, we propose a new test strategy, GTD algorithm, to address this problem. Our algorithm uses training data to judge the benefits brought by an unknown attribute and chooses the most useful unknown attribute each time until there is no rewarding unknown attributes. It is more reasonable to judge the utility of an unknown attribute from the real performance on training data other than from the estimation. Our strategy is meaningful since it has high efficiency(We only use training data so GTD is not sequential) and lower total costs than the previous strategies at the same time. The experiments also prove that our algorithm significantly outperforms previous algorithms especially when there is a high missing rate and large fluctuations of test costs.

1 Introduction

Much work has been done to minimize the misclassification errors in last decades [1][2][3][4] . However, in real world applications, different types of misclassification errors can cost quite differently. As a result, cost-sensitive learning has been proposed to lower the misclassification costs. Many previous works have focused on minimize the total of misclassification costs. However, misclassification costs are not the only type of costs to consider in practice. When performing classification on a new case, values for some attributes may be missing. In such case, we have to make decisions whether to perform additional tests in order to obtain missing values for these attributes. We could not ignore that additional tests may incur more costs while the additional values from the tests may greatly improve the classification accuracy. So, it is necessary for us to analyze whether a test is worthwhile for our classification. As an example, consider the task of a medical practice that examines patients for a certain illness. In medical diagnosis(see Table 1), medical tests are like attributes in machine learning whose

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 455–464, 2010.

values may be obtained at a certain cost(test costs), and misdiagnoses are like the misclassifications which may also bear a cost(misclassification costs), the doctor is considered as the classifier using the information to make decisions. When there is an incoming patient, the doctor has to diagnose the illness for the patient. Usually, some information of the patient is unknown; for example, the blood tests or the X-ray tests. At this point. the doctor has to make the judgments appropriately: performing these tests will incur extra costs, but some tests may provide useful informative benefits to decrease the classification costs. After all, it is the balancing act of the two types of costs–namely, the classification costs and the test costs– that will determine which tests will be done.

Table 1. An Example of a New Case Containing Missing Values and Their Associated Costs for Getting a Value

Tests:	Blood Test	X-ray Test	CAT Scan	Ultrasound	MRI Scan	Diagnosis
Known:	?	Normal	?	?	Abnormal	To be decided
Costs:	\$500		\$200	\$300		MC + TC

MC is the misclassification cost, TC is the test cost.

2 Related Work

Much work has been done in machine learning on minimizing the classification errors. This is equivalent to assigning the same cost to each type of classification errors (for example, FP and FN), and then minimizing the total misclassification costs. Among a variety of costs two types of costs are very important in cost sensitive learning. In [5], Turney considered the following types of costs in machine learning:

1. Misclassification costs: These are the costs incurred by misclassification errors. Works such as [6][7][8] considered machine learning with nonuniform misclassification costs.
2. Test costs: These are the costs incurred for obtaining attribute values. Some previous work such as [9][10] considered the test costs alone without incorporating misclassification costs. As pointed out in [5], it is obviously an oversight.

In [11], Qiang Yang et al. designed sequential strategy and batch strategy. These strategies are proved to perform better than previous ones. Sequential strategy performs well in improving the accuracy of classification and reducing the total costs. However, it has low efficiency. The batch strategy makes up the above shortcomings of sequential strategy but its performance is usually worse than the sequential strategy because little information can be acquired at first.

In [12], Mumin Cebe et al. proposed a new strategy "Based on Conditioned Loss Functions". It takes into consideration the misclassification costs together with the test costs of attributes utilizing the consistency behavior for the first time. Unfortunately this strategy also chooses the attributes to be tested sequentially. What is more, Mumin Cebe et al. put forward an action“reject” to refuse to classify. In many real world applications, it is not permitted to reject for the simple reason of presenting misclassification.

In this paper, we proposed a strategy to make decisions about the combination of attributes to be tested at the first time based on making full use of training data. The novelty of our work can be seen from several angles, as follows:

1. Most previous strategies which perform well have low efficiency because they are sequential. As our strategy only makes use of training data, it can make decision initially. As a result, our strategy shows obviously high efficiency than most previous work.
2. As our strategy use the results obtained by real tests on training data, it is more possible to outperform previous strategy based on estimating. The experiments results have proved our conjecture as well.

After all, our strategy has been proved to reduce the total costs and improve the efficiency at the same time.

3 Classifier

3.1 The Standard Naïve Bayes Classifier

Naïve Bayes classifier is shown to perform well in practice to minimize classification errors, even in many domains containing clear attribute dependence [7]. The standard Naïve Bayes algorithm computes the posterior probability $P(c_j|x)$ of sample x belonging to class c_j according to the Bayes rule:

$$P(c_j|x) = \frac{P(x|c_j)P(c_j)}{P(x)} \tag{1}$$

where

$$P(x|c_j = \prod_{m=1}^{\|A\|} P(A_m = v_{m,k}|c_j) \tag{2}$$

is the product of $\|A\|$ likelihoods.$\|A\|$ is the number of attributes, A_m stands for attribute m and $v_{m,k}$ is one possible value of A_m. x is predicted to belong to the class c_{j^*}. where $j^* =$ arg $max_j P(c_{j^*}|x)$. When there exists missing values in sample x, the corresponding attributes are simply left out in likelihood computation and the posterior probability is computed only based on the known attributes.

However, classification errors are not the only criteria in evaluating a learned model. In practice, costs involved during classification are even more important in deciding whether the model is effective in making correct decisions. Therefore, the standard Naïve Bayes classifier should be extended to be cost-sensitive.

3.2 Test-Cost Sensitive Naïve Bayes Classifier

In this paper, we consider two types of costs in Naïve Bayes classification: misclassification costs and test costs. Misclassification costs are considered when there are different types of classification errors and the costs they bring are different. The standard Naïve Bayes algorithm (NB) can be extended to take the misclassification costs into account. Suppose that C_{ij} is the cost of predicting a sample of class c_i as belonging to class c_j . The expected misclassification costs of predicting sample x as class c_j is also known as the conditional risk [13], which is defined as:

$$R(c_j|x) = \sum_i C_{ij} * P(c_i|x) \tag{3}$$

$P(c_i|x)$ is the posterior probability given by the standard NB classifier. Sample x is then predicted to belong to class c_{j*} which has the minimal conditional risk:

$$R(c_{j*}|x) = min_j R(c_j|x). \tag{4}$$

4 Test Strategies

In this paper, we put forward a new test strategy called"Greedy on Training Data" (GTD) and apply it on cost sensitive naïve Bayes classifier. When incoming the test data with missing values, we aimed at selecting a subsets of unknown attributes seemed to be most useful. There comes a key problem: how to judge an unknown attribute's utility for classification. In this study, we make judgments by testing attributes' utility based on the training data. When incoming the test data, we suppose that the training data has the same attributes' situation with the test data(if an attribute is unknown in test data, it will also be unknown in training data, otherwise it will be known in training data). We make the classification without testing any unknown attributes at the beginning and get average total costs. Afterwards, for each unknown attribute, we first make it revealed in training data and test the average total costs in training data. Then we compare the result with the original one's(the total costs without the value of this attribute). If the present one's result is smaller than the original one's, it means that this attribute is useful and we can use how much it reduces the average total costs to evaluate its utility.

Before proposing the GTD in detail, we first define some notations used in subsequent analysis.

Definition 1. *We denote $S = \langle T1, T2 \ldots Tn \rangle$ is the subset of training data, n is the number of training data sets. $F_d = \langle f_{d1}, f_{d2} \cdots f_{d\|A\|} \rangle$ denotes the attributes' situation of data D. If f_{di}=1, it means the ith attribute is known, otherwise, it is unknown. $\|A\|$ stands for the number of attributes. $F_{now} = \langle f_1, f_2 \ldots f_{\|A\|} \rangle$ denotes the present attributes' situation.*

Definition 2. *We use D_u to represent the test data, $K = \langle k_1, k_2 \ldots k_{\|A\|} \rangle$ denotes the attributes' situation of D_u. $R = \langle r_1, r_2 \ldots r_{\|A\|} \rangle$ to denotes the subsets of selected unknown attributes to be tested(if r_i=1, it means the ith attribute is selected to be tested, otherwise it is not selected. The default value of r_i is 0.d*

Definition 3. *Util=$\langle u_1, u_2 \ldots u_{\|A\|}\rangle$denotes the utility of the attributes. The default value of u_i is 0.*

First, we let F_{now} equals to K. The cost sensitive naïve Bayes algorithm(CSNB) is applied on S based on F_{now} to get the original average total costs(ATC_o). Then, for each $k_i(1 <= i <= \|A\|)$, if k_i equals to 0, we let f_i=1, and apply CSNB on S based on F_{now} to get a new result(ATC_{new}). We set $u_i = ATC_o - ATC_{new}$. Then the f_i is set to 0 again. In this way, we get the utility of every unknown attribute in the present situation F_{now}. Then, we use the greedy thought to choose the i_*th unknown attributes having the largest utility($u_{i^*} = max_{i^*} u_{i^*}$) since it implies that this attribute is the most useful attribute at present. If the u_{i^*} is not positive, we stop choosing unknown attributes, otherwise the r_{i^*} is set to 1. The f_{i^*} is also set to 1, denoting the i^*th attribute is known in training data. Afterwards, we use the above method iteratively until there is no useful unknown attribute(u_{i^*} is not positive). The total costs of classifying D_u with class label i to class label j can be got as follows:

$$C_{sum}(i) = C_{ij} + \sum_{m} C_{test}(m); r_m = 1; \tag{5}$$

$C_{sum}(i)$ is the total costs of D_u. $C_{test}(m)$ stands for the test costs of Am.

In order to make full use of S, we run a ten-fold cross validation on it. From the above, it is obvious that we use the greedy thought to choose the most"useful" unknown attribute which leads to the lowest average total costs on S iteratively. If not have one, we stop testing and use the initial known attributes and the attributes decided to be tested to classify data. Another thought is that trying every possible subsets of unknown attributes and choose the subset which leads to minimum average total costs on training data. But, it is not feasible in practice because it is a NP Complete problem. It is easy to prove that such a combinatorial optimization problem can not be solved in polynomial time. As a result, our GTD is more appropriately in reality. GTD is given in Algorithm 1 as follows.

5 Experiments

In order to evaluate the performance of GTD algorithm, several experiments were carried out on data sets from the UCL ML repository [14]. The eight data sets used are listed in Table 2. These data sets were chosen because the number of their attributes, instances and class varies in a wide range. Hence, we can test whether our algorithms can be used in many kinds of situations. Discretization were applied on the numerical attributes in data sets to change it to 5 intervals with equal length. For convenience, we try to use the data sets without missing values.

For algorithm comparison, we use other four methods besides GTD algorithm, namely lazy naïve Bayes(LNB), extracting naïve Bayes(ENB), cost-sensitive naïve Bayes with sequential strategy(NBS) and cost-sensitive naïve Bayes with batch strategy(NBB). LNB is termed Lazy naïve Bayes, as given in [13], since

Algorithm 1. GTD algorithm(S,CSNB,D_i,K_i,$\|A\|$)

Input:

S– Training data sets without missing values
CSNB– a cost sensitive naïve Bayes classifier
D_i– the data to be tested
K_i– an array to denote the known attributes of D_i ; $k_{ij} = 1$ means the jth attribute of D_i is known, otherwise the value of it is missed
$\|A\|$– the number of attributes of D_i

Output:

KM– KM=$\langle km_{i1}, km_{i2}, \ldots km_{i\|A\|}\rangle$ to denote which attributes to be tested;

Steps:

```
use CSNB to classifier $S$ based on $K_i$
  $las \leftarrow$ calculate the average total costs;
$bes \leftarrow las$
$Klas \leftarrow K_i$
$Kbes \leftarrow K_i$
while true
  for $j \leftarrow 1$ to $\|A\|$
    if $klas_{ij} = 0$
      then $Ktmp \leftarrow Klas$
        $ktmp_{ij} = 1$
        use CSNB to classifier $S$ based on $Ktmp$
        $tmp \leftarrow$ calculate the average total costs;
        if $tmp < bes$
          then $bes \leftarrow tmp$
            $Kbes \leftarrow Ktmp$
  if $Kbes = Klas$
    then break;
    else $las \leftarrow bes$
      $Klas \leftarrow Kbes$
for $j \leftarrow 1$ to $\|A\|$
  if $k_{ij} = 0$
    then if $klas = 1$
          then $km_{ij} \leftarrow 1$
          else $km_{ij} \leftarrow 0$
    else $km_{ij} \leftarrow 0$
return $KM$
```

Table 2. Data Sets Used in the Experiments

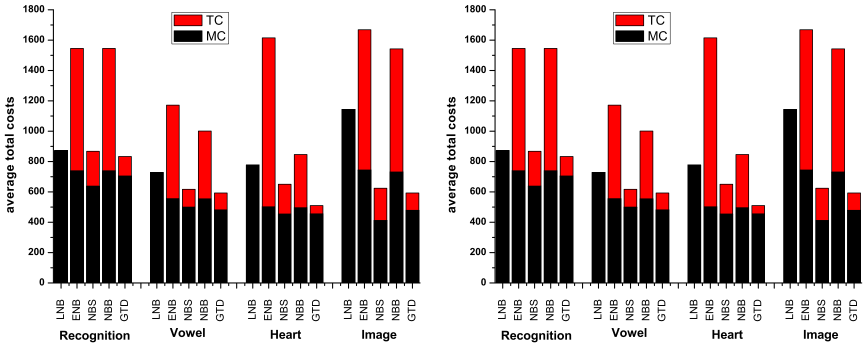
Name of data sets	No. of instances	No. of attributes	No. of class labels
Ecoli	336	7	8
Heart	267	22	2
Parkinsons	197	22	2
Image	2310	20	7
Contraceptive	1473	9	3
Monk	432	6	2
Recognition	10992	16	10
Vowel	990	13	11

it simply predicts class labels based on the known attributes and requires no further tests to be done on unknown ones. The second one is the naïve Bayes classifier extended further from LNB. It requires all the missing values to be made up before prediction. Since this classifier allows no missing values, it is therefore termed Exacting naïve Bayes. NBS and NBB are introduced in[11].

We ran a three-fold cross validation on the data sets. For the testing samples, a certain percentage (missing rate) of attributes were randomly selected and marked as unknown. If we decided to test an unknown attribute, then the real value of it was to be revealed. The performance of the algorithms was measured by the total costs, the sum of test costs and misclassification costs.

5.1 Comparison of Performance among Five Methods

We first compare these five methods with a given missing rate, misclassification costs and test costs. In these experiments, the percentage of unknown attributes (missing rate) was 97 percent which means that we hardly have any information about the case we have to predict. We choose this missing rate in order to make a clear comparison of these five methods since more missing values let test strategies play more important roles in classification. The misclassification costs are all set to $70n_a$ where n_a is the number of attributes. The test costs of every attribute is randomly set between 40-60.



MC is the misclassification costs, TC is the test costs.

Fig. 1. Experiments on 8 data sets

The results are shown in Fig.1. Each group of 5 bars represents the runs of 5 algorithms on one particular data set. The height of a bar represents the average total costs and, therefore, the lower the better. Each bar consists of two parts: The lower dark portion stands for the misclassification costs while the upper red portion stands for the test costs. We can see the large different of average total costs from the figures.

There are some very interesting observations from these experiments.

1. Sometimes even we known all the values of missing attributes, we can not get the minimum misclassification costs. It makes sense from the feature selection prospective as some features are useless. Knowing all the attributes may have negative effects on classification accuracy.
2. Despite its lazy nature, the LNB method performs surprisingly well, sometimes very closely to NBS and GTD.
3. From the results, we can see that our algorithm GTD performs best among the 5 algorithms. It is not unexpected for the following reasons. GTD selects attributes to be tested based on the classification results on training data. It uses the fact to decide whether an unknown attributes is worth to be tested. It is more reasonable and appropriate than making decisions based on estimation used by previous test strategies. In addition, every time it chooses the most useful unknown attribute based on its utility on reducing the average total costs of training data. This greedy method makes it possible for GTD to choose the most valuable unknown attributes.

5.2 Comparison of Performance with Changing Missing Rates

To investigate the impact of the percentage of unknown attributes on the total costs, experiments were carried out on average total costs with the increasing percentage of unknown attributes. In order to obtain the effects of missing rates changes clearly we set the range of it from 0 to 97 percent(it is impossible to apply classification on data without any attribute values). From the experiments above, we can find that the comparable better test strategies are LNB, NBS and GTD. As a result, the following comparison only focus on these three strategies while ignore the other two strategies. Fig. 3 shows the results on the 8 data sets. As we can see, when the missing rate is low, the performance of these 3 strategies does not have clearly difference. However, when the missing rate increases to a point, the average total costs of LNB increases significantly and we can see the obviously difference on these 3 strategies. Again, GTD is better than the other two over the whole range.

5.3 Comparison of Performance with Changing Test Costs Range

Another set of experiments was conducted to study the impact of test costs range change on strategy performance. The comparison was only between NBS and GTD since LNB has no relationship with test costs. In these experiments, We make the range of test costs from 40-60 to 0-100. We try to make the range larger while the average test costs is not changed. In this case, we can make a clear understanding of how “sensitive” the strategies are to the test costs fluctuations. There are two observations worth mention:

1. There are not obvious disciplines about the average total costs of these two strategies with the range of test costs becoming larger. It makes sense because the average of test costs is not changed while the range becomes larger. We can imply that the range change of test costs does not directly affect the total costs.

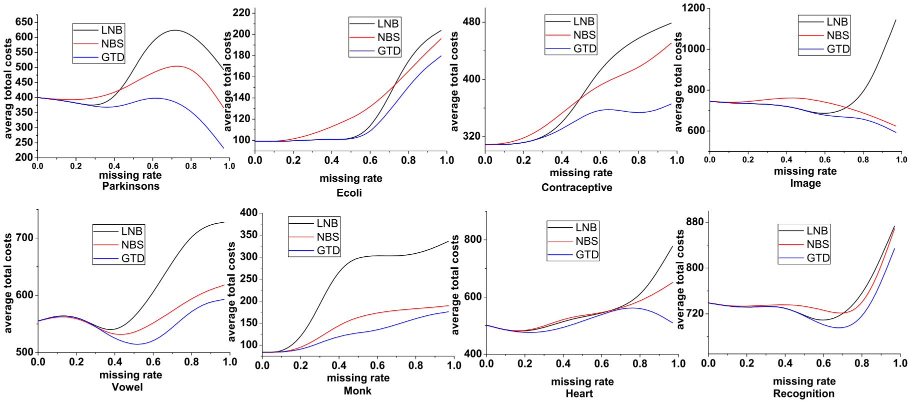


Fig. 2. Comparison of Performance with Changing Missing Rates

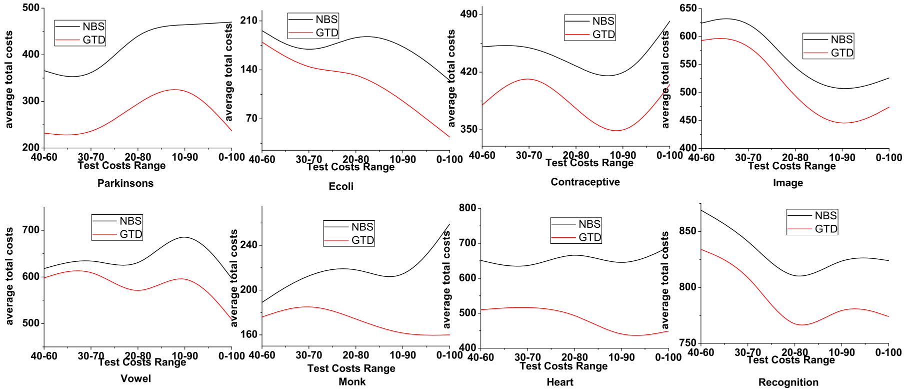


Fig. 3. Comparison of Performance with Changing Test Costs Range

2. With the range of test costs becoming larger, the difference of performance between these 2 strategies becomes more obviously. In other words, the average total costs of GTD is much lower than that of NBS with larger range of test costs. As a result, we can imply than our strategy GTD is more sensitive to the test costs fluctuations than NBS.

6 Conclusion and Future Work

In this paper, we proposed a new test strategy GTD to deal with test-cost sensitive classification on data with missing values. Our strategy is very valuable because the strategy has high efficiency and leads to lower total costs than previous strategies at the same time. This makes our strategy is more appropriate in the real world applications. Experiments show that our strategy outperform other competing algorithms. In addition, the experiments imply that when the missing rate is high our algorithm outperforms other algorithms more significantly. The experiments also tells that our algorithm is more sensitive to the fluctuations of test costs than other compared algorithms. In the future, more studies can be made on how to select the subsets of unknown attributes to be revealed to get the average total costs on training data in order to get higher quality subsets of unknown attributes.

Acknowledgement

I am really grateful to Mr. Ce Guo, an undergraduate student from department of Computer Science at Sun Yat-sen University, for giving me so many recommendations and advices about this study.

References

1. Mitchell, T.M.: Machine Learning. McGraw Hill, New York (1997)
2. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco (1993)
3. Juang, B.-H., Katagiri, S.: Discriminative learning for minimum error classification. IEEE Transactions on Signal Processing 40(12) (1992)
4. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3) (1995)
5. Turney, P.D.: Types of cost in inductive concept learning. In: Workshop Cost-Sensitive Learning at the 17th Int'l. Conf. Machine Learning (2000)
6. Elkan, C.: The foundations of cost-sensitive learning. In: 7th Int'l. Joint Conf. Artificial Intelligence, pp. 973–978 (2001)
7. Domingos, P., Pazzani, M.: On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning, 103–130 (1997)
8. Kai, M.T.: Inducing cost-sensitive trees via instance weighting. In: Żytkow, J.M. (ed.) PKDD 1998. LNCS, vol. 1510, pp. 139–147. Springer, Heidelberg (1998)
9. Nunez, M.: The use of background knowledge in decision tree induction. Machine Learning, 231–250 (1991)
10. Tan, M.: Cost-sensitive learning of classification knowledge and its applications in robotics. Machine Learning J., 7–33 (1993)
11. Yang, Q., Ling, C., Chai, X., Pan, R.: Test-cost sensitive classification on data with missing values. IEEE Transactions on Knowledge and Data Engineering (5) (2006)
12. Cebe, M., Gunduz-Demir, C.: Test-cost sensitive classification based on conditioned loss functions. Machine Learning (2007)
13. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley and Sons, Inc., Chichester (2001)
14. Blake, C.L., Merz, C.J.: Uci repository of machine learning databases (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html

3-D Magnetotelluric Adaptive Finite-Element Modeling

Changsheng Liu[1,2], Yan Yu[3], Zhengyong Ren[1,4], and Qi Wu[1]

[1] School of Info-physics and Geometrics Engineering, Central South University, Changsha, China, 410083
[2] Changsha Aeronautical Vocational and Technical College, Changsha, China, 410014
[3] School of Civil Engineering and Construction, Wuhan University of Technology, Wuhan, China, 430070
[4] Institute of Geophysics, ETH-Swiss Federal Institute of Technology, Zurich, Switzerland
lcs888_2002@163.com, yy@msdi.cn, renzhengyong@gmail.com, wt83111@163.com

Abstract. Aiming at three principal problems existing in traditional node-based finite-element method when modeling MT problems, this paper proposed an adaptive vector-based finite-element method with unstructured mesh. On the basis of 3-D finite-element formulae, it introduces a residual-based posteriori error estimator. Then, it presents the adaptive refinement strategy and computation flow. The test of a homogenous earth model and two anomalies with high resistivity model show that the numerical solutions enable the convergence to the exact results with a maximal average error less than a percentage value of 1%. Thus, it can be concluded that the h-adaptive vector finite-element method can guarantee the accuracy and efficiency of complicated models, through which its great potential is displayed.

Keywords: Magnetotelluric three-dimensional modeling; unstructured mesh ; vector finite-element method; residual-based error estimate; h-adaptive finite-element method.

1 Introduction

Since Coggon[1] first reported the finite-element method of electric and electromagnetic filed, it has showed its great potential in geo-electromagnetic forward modeling. Yan[2] studied the forward modeling problem of controlled source audio frequency magnetotelluric (CSAMT) filed by using vector finite-element method(VFEM). Shi et al.[3] presented its application to 3-D magnetotelluric modeling through considering the divergence-free condition. Then, Myung and Hee et al.[4] utilized vector finite-element method to implement 3-D magnetotelluric forward modeling. Mao and Wang[5] addressed its application to borehole ground electric field. Sun et al.[6] simulated the electromagnetic response of inclined inhomogeneous layered earth. Wang [7] accomplished 3-D magnetotelluric forward modeling in high frequencies. Liu [8] presented some numerical results in 3-D magnetotelluric filed.

Currently, there exit three principal problems in geo-electromagnetic modeling with finite-element method needed to be addressed: (1) the spurious solutions can not

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 465–473, 2010.

be avoided; (2) the error of discretization caused by model subdivision should be particularly considered especially when terrain topography is simulated; (3) The accuracy of standard finite-element method is determined by one time model subdivision. The edge-based vector finite-element method is a useful tool to solve first mentioned problem [7, 8]. Moreover, the unstructured mesh can fit any complicated models to greatly reduce the error of discretizaiton [4,8,9,10,11]. In addition, the adaptive finite-element method is much superior to avoid the third mentioned problem by implementing the adaptive refinement. Since 1995, Demkowicz and Rachowicz have done a systematical research in the adaptive hp finite-element method of direct current (DC) and geo-electromagnetic field. The application to the hole-hole model in petroleum prospecting has given the high accurate apparent resistivity results. Key [12] (2006) presented an adaptive finite-element algorithm used in 2-D magnetotelluric modeling. Utilizing the adaptive finite-element method, Li [13] reported some numerical results of 2-D controlled source audio frequency magnetotelluric filed. However, the 3-D adaptive finite-element geo-electromagnetic modeling has not been discussed yet.

In this paper, we give the adaptive vector finite-element algorithm and the corresponding implementation using unstructured mesh. It is a worthy discussion for the application of adaptive finite-element method in geo-electromagnetic modeling.

2 3D MT Vector Finite-Element Model

3-D MT model is depicted in Fig.1, where, V_a denotes the air region (electric conductivity σ_0=0), V_e means the earth region (electric conductivity $\sigma_1 \neq 0$). V_s is the interface of air and earth, **n** is the outward unit normal vector of V_s and outside boundary of the model, the electric conductivity of the anomaly beneath the earth $\sigma_2 \neq 0$, S_t denotes the upper boundary of the air region, S_b means the lower artificial boundary of the earth region, S_s is the four surrounding boundaries of the whole region.

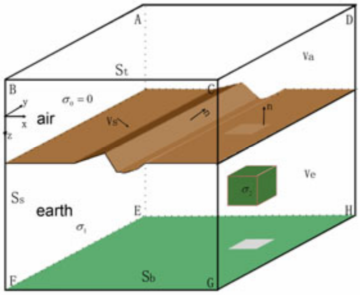


Fig. 1. MT model of three-dimensional structure of general earth electrical

Thus, in region Va and Ve, the electromagnetic filed satisfy the following equation[14]:

$$\nabla\times\nabla\times\mathbf{E} - i\omega\mu\sigma\mathbf{E} = \mathbf{0} \tag{1}$$

The Direlet boundary condition [4](Nam et al., 2007) is adopted in this paper. Moreover, the electromagnetic values on the boundaries of the horizontally layered earth are enforced on the outside boundary.

$$\mathbf{E}\times\mathbf{n} = \mathbf{E}_0\times\mathbf{n} \tag{2}$$

Equation (1) and Equation (2) are called the magnetotelluric boundary value problem. By terms of the vector identity formula $\mathbf{B}\cdot(\nabla\times\mathbf{A}) = \mathbf{A}\cdot(\nabla\times\mathbf{B}) + \nabla\cdot(\mathbf{A}\times\mathbf{B})$ and Green's integral formula, its variation expression can be easily deduced as:

$$b(\mathbf{E},\mathbf{V}) = f(\mathbf{V}), \quad \mathbf{V}\in H(curl) \tag{3}$$

To solve the electric filed of equation (3), using the vector finite-element method, we divide the ertire computing domain into a set of irregular tetrahedron elements. The difference between VFEM and the standard finite-element method is the tangential component of the electric filed is defined on the tangential direction of each edge. Thus, the electric filed within arbitrary tetrahedron **e** can be written into:

$$\mathbf{E}^e = \sum_{i=0}^{5} E_i^e \mathbf{\Phi}_i^e \tag{4}$$

Where E_i^e and $\mathbf{\Phi}_i^e$ denote the tangential electric field and the vector shape function on the ith th edge of the ith element, respectively.

As the divergence of $\mathbf{\Phi}_i^e$ is zero, the requirements for free divergence on the current density are easily satisfied and since $\mathbf{\Phi}_i^e$ is only continuous along the tangential component of edges, the continuity of the tangential component of the electric field normal discontinuity remains. Additionally, based on the vector-shape function, the tetrahedral element can also efficiently suppress the spurious physical solutions. In contrast, an extra penalty term must be added to the node-based finite-element method to avoid the spurious solutions which results in increased computation complexity.

The derived system of equations for MT vector finite-element modeling is expressed as following

$$\mathbf{AU = B} \tag{5}$$

Where **U** is the unknown vector at the tetrahedron element edge. Matrixes **A** and **B** can are

$$\begin{aligned} \mathbf{A}^e_{ij} &= \int_{\Omega^e} \nabla\times\mathbf{\Phi}_i\cdot\nabla\times\mathbf{\Phi}_j - i\omega\mu\sigma\mathbf{\Phi}_i\cdot\mathbf{\Phi}_j d\Omega \\ \mathbf{B}^e_i &= \oint_{\partial L^e} \mathbf{\Phi}_i\cdot\mathbf{E}_0 d\Gamma \end{aligned} \tag{6}$$

where $\mathbf{B}_i$ is nonzero just at the edges of the outer Dirichlet boundary.

3 A Posteriori Error Estimates Based on RESIDUAL

We utilize the Nédélec tetrahedral linear element. The solution process can be expressed as: define the T_k as a series of regional units tetrahedral mesh generation on the calculation of regional Ω , the F_k as a series of triangular surface grid. On the finite element space T_{k}, define U_k as[15] （Chen Zhiming et al.）:

$$U_{\mathrm{k}} = \{\mathbf{u} \in \boldsymbol{H}(\nabla\times,\Omega) : \mathbf{u}\times n = n\times\mathbf{u}_0,\ \mathbf{u}\,|_T = a_T + b_T \times x \\ \text{with } a_T, b_T \in R^3,\ \forall T \in T_{\mathrm{k}}\} \tag{7}$$

Where on the T_{k}, finite-element method calculationcan be expressed as : solution $\boldsymbol{E}_k \in \boldsymbol{U}_k$, to meet:

$$a(\boldsymbol{E}_k, \mathbf{v}) = \int_\Omega \mathbf{f}\cdot\mathbf{v} + \int_{\partial\Omega} \mathbf{g}\cdot\mathbf{v}_t \quad \forall\mathbf{v}\in\boldsymbol{U}_k \tag{8}$$

Definition of error of numerical solutions:

$$\mathbf{e}_k = \boldsymbol{E} - \boldsymbol{E}_k \tag{9}$$

Can be derived for the MT model of a posteriori error estimates:

$$\|\mathbf{e}_k\|_{\mathbf{H}(\nabla\times,\Omega)} \le C(\sum_{T\in T_k}\eta_T^2 + \sum_{F\in F_k}\eta_F^2)^{1/2} \tag{10}$$

4 Adaptive Finite-Element Strategy

We utilize the h-type finite-element method to implement 3-D MT numerical modeling. The constrained Delaunay tetrahedron is adopted for model subdivision and the corresponding mesh refinement. For arbitrary complicated model, once the initial model and the global terminal condition are given, the algorithm will automatically execute the h-type adaptive refinement without any artificial interface.

On the mesh T_k , $\mathbf{E}_k$ can be easily obtained by the above motioned algorithm. By using the error convergence theory of references [15], the error estimate for the current mesh can be derived, and the element error indicator can be written into:

$$\eta^2{}_T = \sum_{T\in T_k}\left\{h_T\,\|k^2\mathbf{E}_k - \nabla\times\nabla\times\mathbf{E}_k\|^2{}_{0,T} + \|\nabla\cdot(k^2\mathbf{E}_k)\|^2{}_{0,T}\right\} \\ + \sum_{F\in F_k}\left\{h_F^{1/2}\,\|[\mathbf{v}\times\nabla\times\mathbf{E}_k]\|^2{}_{0,T} + \|[k^2\mathbf{E}_k\cdot n]\|^2{}_{0,T}\right\} \tag{11}$$

Correspondingly, the global error indicator and the maximal error indicator are denoted by

$$\eta_h = (\sum_{T\in T_k}\eta_T^2)^{1/2} \tag{12}$$

$$\eta_{\max} = \max_{T\in T_k}\eta_T \tag{13}$$

If the computed element error exceeds the specified error, the corresponding refinement is needed according to the following steps.

Step 1: give a small positive value $\varepsilon > 0$, assume the initial mesh is T_0 and $T_h = T_0$. Solve equation (8) on T_0.

Step 2: compute the local error estimate η_T, global error indicator η_h and the maximal error estimate $\eta_{\max}$ for each element $T \in T_0$.

Step 3: if $\eta_h \le \varepsilon$, go to Step 4; otherwise：

if $\eta_T > \beta\eta_{\max}, \beta \in [0.5, 1.0)$, execute the refinement for all the elements satisfying this condition；

Then, solve the formula (8) on the refined T_h；

Compute the element error estimate η_T for each element $T \in T_0$,the global error indicator η_h and the maximal error estimate $\eta_{\max}$；return to Step 3.

Step 4: post processing

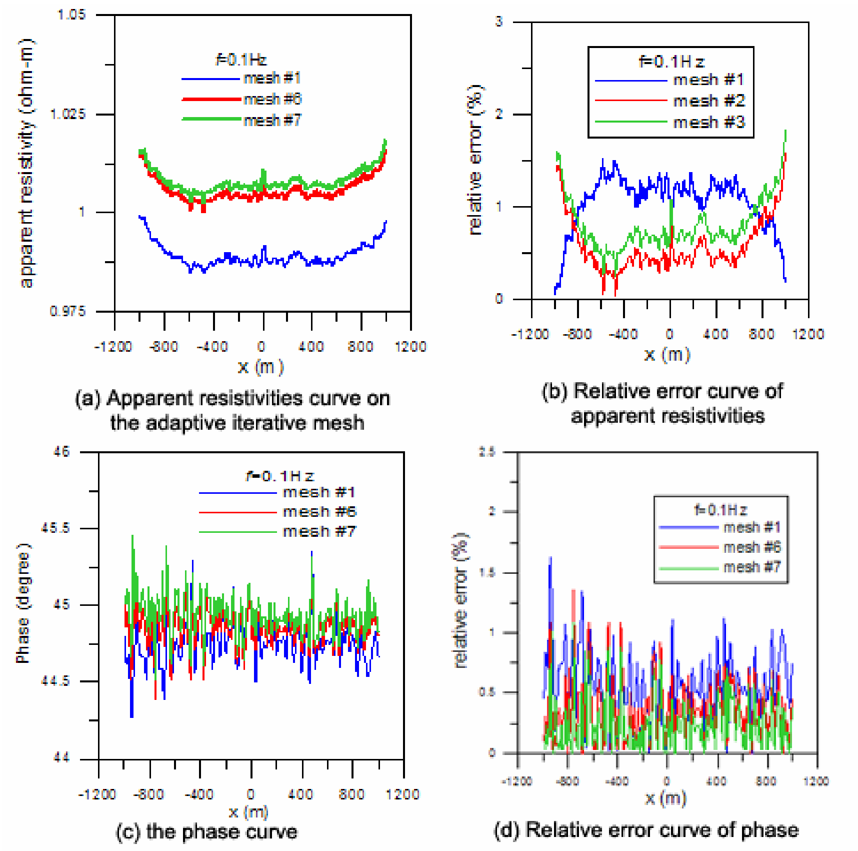
In the algorithm steps, we can find that the small value ε and refinement factor β both have a great effect on the refinement. Generally, ε is less than 1, and β is proper in the range of between 0.5 and 0.7. For different problems, the difference of ε is large, $\varepsilon \in [1e-10, 0.1]$. Moreover, in practical application, the maximal iteration time Its and the maximal consumed memory are often used as terminal condition, generally $Its = 10$, $M = 500MB$.

5 Numerical Results

5.1 Homogenous Earth Model

Firstly, a homogenous earth model with 1ohm-m resistibility is considered and that of air layer is 10^{10}ohm-m. Our MT measuring stations are located in range of -1000m to 1000m along with x-axis with 10m intervals. The measuring lines are located on the x axis and y axis, respectively, the range is between 0 km and 3 km. 21 measuring points are observed, the working frequency is f=0.1HZ. The terminal conditions are $\varepsilon = 0.01$, $\beta = 0.5$, $Its = 10$, $M = 500MB$. The algorithm stops after 7 iterations; the maximal consumed memory is 40.5MB. The number of the nodes on the initial mesh is 9407 with 58741 elements and 136476 edges. The number of the nodes on the six time refined mesh is 11148 with 67755 elements and 158450 edges. On the final mesh, the number of the nodes is 13333, the number of the element is 79475 and the number of the element is 186648.

The element error on the initial mesh is large, the average relative error is 56.10%, and the maximal error is 149%. Thus, the large error distribution makes the adaptive refinement executed on the elements with large errors. This can be easily seen from the six-time refined mesh and seven-time refined mesh. In the six-time refined mesh, the average element error is reduced to 2.94%, the maximal error becomes 8.98%; in the seven-time refined mesh, the average element error is greatly decreased to 0.932%, and the maximal error becomes 4.86%, the average error is less than 1% of a given, so the refined process to terminate.



(a) Apparent resistivities curve on the adaptive iterative mesh

(b) Relative error curve of apparent resistivities

(c) the phase curve

(d) Relative error curve of phase

Fig. 2. The apparent resistivities curves, phase curves and there relative error curve of uniform measurement points on the surface with frequency of 0.1Hz MT model

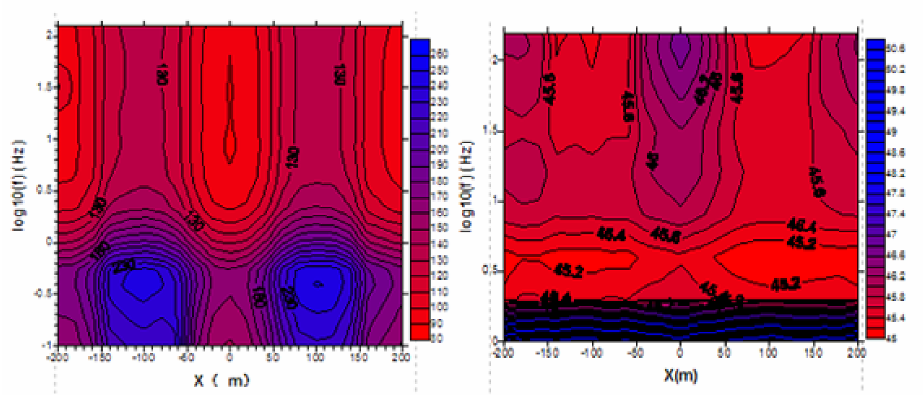
Moreover, we can find that the elements nearly the surveying line are not greatly refined along with the adaptive process. This results in a small accuracy improvement near the surveying line (see in Fig.2): the average error of the apparent resistivities is 1.062% while it is 0.578% in the six-time refined mesh. However, as the refinement is performed, the average relative error of the apparent resistivities keeps a stable value in the range between 0.55% and 0.80%. The average error of the phases is 0.567%, while it is 0.359% in the twenty-time refined mesh. Similarly, as the refinement is executed, the error doses not change much and keep a value between 0.25% and 0.35%. These results show that the commonly used adaptive strategy can give an improved accuracy for the initial mesh. However, there still exists some deficiency for our geo-electromagnetic modeling which requires more accuracy.

5.2 Two Anomalies with High Resistivities

In the second model, two anomalies both have a size of 100m×100m×100m. The distance in the x-direction is 100m, the buried depth is 50m and the distance between arbitrary axis plane and the outside boundary is 10km. The surveying line is located on the x-axis, from -200m to 200m and the spacing is 20m. The resistivities of the two anomalies are both 10000 ohm-m and the background earth has a resistivity of 100 ohm-m. The working frequency is 8Hz. On the initial mesh, the maximal error for the total measuring line is 199.15%, while on the final refined mesh it becomes 0.403%; the consumed time is 4549.527s, the average cost time for all iterations is 454.9527s.

On initial mesh, 1940 nodes are generated with 11154 elements and 26314 edges. The average error is 80.23%. On the eight-time refined mesh, 14434 nodes are generated with 78300 elements and 188030 freedoms. The average error is 0.61% and maximal error is 1.26%. On the final refined mesh, 54810 nodes are generated with 302317 elements and 721306 freedoms. The average error becomes 0.18% and the maximal error is 0.41%.

The numerical results from the final refined mesh are shown in Fig.3. From this figure it is seen that the spurious cross section of phases can give a clear description of the locations of the two anomalies.



(a) Spurious cross section of the apparent resistivities

(b) Spurious cross section of the phases

Fig. 3. The numerical results from the final adaptively refined mesh

6 Discussions and Summaries

In this paper, we first present the 3-D MT adaptive vector finite-element method with unstructured mesh. Based on this, we derive the residual based posteriori error estimator for 3-D MT case. Then the adaptive refinement strategy is given and the adaptive process is depicted. To validate our algorithm, two models are simulated: homogenous earth model and two anomalies buried in the homogenous earth. The numerical results show that the algorithm can achieve the expected accuracy. However, as the refinement is executed, the improved of the accuracy is very small. This may be due to the selection of the global refinement operator. If the more efficient progress is expected for the application of the adaptive finite-element method in geo- electromagnetic filed, the local refinement operator should be considered.

Acknowledgements

This work is financially supported by the National High Technology Research and Development Program of China (No.2006AA06Z105, No .2007AA06Z134) , Science Technology Program of Hunan province, China(No.2008FJ4181) and the Science Research Project of Education Department of Hunan province, China (No.08D008). Moreover, the great thanks also go to Dr. Hang Si and Researcher Zhang Ling Bao for the distinctive mesh package, and the team providing the mesh visualization package Paraview.

References

[1] Coggon, J.H.: Electromagnetic and electrical modeling by the finite element method. Geophysics 36(1), 132–155 (1971)

[2] Shu, Y.: Studies on the Electrical and Electromagnetic Prospecting Based on Three-dimensional Finite Element Numerical Modeling. Xi'an Jiaotong Unviersity, Xi'an (2003) (in Chinese)

[3] Shi, X., Utada, H., Wang, J., Siripunvaraporn, W., Wu, W.: Three dimensional magnetotelluric forward modeling using vector finite element method combined with divergence corrections (VFE++). In: 17th IAGA WG 1.2 Workshop on Electromagnetic Induction in the Earth Hyderabad, India, October 18-23 (2004)

[4] Nam, M.J., Kim, H.J., Song, Y., Lee, T.J., Son, J.-S., Suh, J.H.: 3D magnetotelluric modelling including surface topography. Geophysical Prospecting 55, 277–287 (2007)

[5] Li-Feng, M., Xu-Ben, W., Zhan-Xiang, H.: Application of an edge-based finite-element method to model 3D AC borehole-to-surface forward problem. Chinese Geophysical, 651–652 (2006) (in chinese)

[6] Xiang-Yang, S., Nie, Z.-P., Zhao, Y.-W., Li, A.-Y., Xi, L.: The electromagnetic modeling of logging-while-drilling tool in tilted anisotropic formations using vector finite method. Chinese Journal of Geophysics 51(5), 1600–1607 (2008)

[7] Ye, W.: A study of 3D high frequency magnetotellurics modeling by edge-based finite element method. Central South University, Changsha (2008) (in chinese)

[8] Changsheng, L., Zhengyong, R., Jingtian, T., Yan, Y.: Three-dimansional magnetotellurics modeling using edge-based finite element unstructured meshes. Applied Geophysics 5(3), 170–180 (2008)

[9] Nelson, E.M.: Advances in 3D Electromagnetic finite element modeling. In: Proceedings of the IEEE 1997 Particle Accelerator Conference, Vancouver, Canada, pp. 1837–1840 (1998)

[10] Sugeng, F.: Modeling the 3D TDEM response using the 3D full-domain finite-element method based on the hexahedral edge-element technique. Exploration Geophysics 29(4), 615–619 (1998)

[11] Yoshimura, R., Oshiman, N.: Edge-based finite element approach to the simulation of geoelectromagnetic induction in a 3-D sphere. Geophysical Research Letters 29(3), 1039–1045 (2002)

[12] Key, K., Weiss, C.: Adaptive finite-element modeling using unstructured grids: The 2D magnetotelluric example. Geophysics 71(6), 291–299 (2006)

[13] Li, Y., Key, K., Constable, S.: An adaptive finite element modeling of 2-D marine controlled-source electromagnetic fields. In: 18th IAGAWG 1.2Workshop on Electromagnetic Induction in the Earth, El Vendrell, Spain, September 17-23, pp. S3–E14 (2006)

[14] Harrington, R.F.: Time Harmonic Electromagnetic Fields. McGraw-Hill Book Co., New York (1961)

[15] Chen, Z., Wang, L., Zheng, W.: An adaptive multilevel method for time-harmonic waxwell equations with singualrities. SIAM J. Sci. Comput. 29(1), 118–138 (2007)

A Study on Modeling Evolutionary Antenna Based on ST-5 Antenna and NEC2

Yuanyuan Fan[1,2], Qingzhong Liang[2], and Sanyou Zeng[2]

[1] School of Computer Science, Wuhan University, China
[2] School of Computer Science, China University of Geosciences, China
yyfan@cug.edu.cn

Abstract. Plentiful applications of evolutionary algorithm (EA) to antenna design have formed a new hot research topic - evolutionary antenna. A novel antenna has been designed through EA in NASA Ames Research Center, and has been successfully applied in ST-5 mission [1]. It has represented the first evolved hardware in space, and the first deployed evolved antenna [1, 2]. We present a mathematical model and an algorithmic model for the evolutionary antenna, according to the requirements for the ST-5 antenna and the characteristics of NEC2. And also we show the experiment results and make a conclusion.

Keywords: evolutionary antenna; modeling; NEC2; ST-5.

1 Introduction

Since F.Braun designed the first wire antenna in 1898, a variety of wire antennas have been designed, such as dipole, monopole over a ground plane, rhombic, beverage, Yagi, log periodic, loop, helix and spiral antennas[3], which have vastly applied in communications, radar, remote sensing systems and many other fields.

Conventional methods of antenna designs by simulation or analysis are time- and labor-intensive, limit complexity, increase the cost and time expended, and require that antenna engineers have significant knowledge of the universe of antenna designs. Local optimization methods are not much better, since an initial guess that is close to the final design must be provided [4].

Evolutionary algorithm has already been shown to be capable of searching the vast, unknown solution space so that it has revolutionized many fields of design [3]. Evolutionary algorithm allows the computer to find the satisfied designs only with the information of the desired performance. And it does not require an initial guess, and even the amount of the knowledge can be very minimal.

Researchers have been investigating the application of evolutionary algorithm to electromagnetic design and optimization since the early 1990s [5, 6]. While with the improvement of the computers' capability and the simulators' precision, the research in evolutionary antenna is flourishing recently. Especially many types of wire antenna have been studied, such as Yagi, crooked-wire genetic antenna [3, 4], antenna array [7, 8], quadrifilar helical antennas [9] and so on.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 474–486, 2010.

NEC (Numerical Electromagnetics Code) is a software that can simulate the electromagnetic response of antennas and metal structures [10]. An input file containing the information of an antenna's configuration is red by NEC, and then an output file will be generated using the Method of Moments that contains the antenna's radiation pattern, polarization and other properties. During an optimization, EA generates an input file for each design in population and call NEC to generate an output file accordingly. Then EA analyzes the simulation results in output file, and scores the design's performance through the cost function. Our experiments are carried out with NEC2, which is released to the public and now is available on most computing platforms [10].

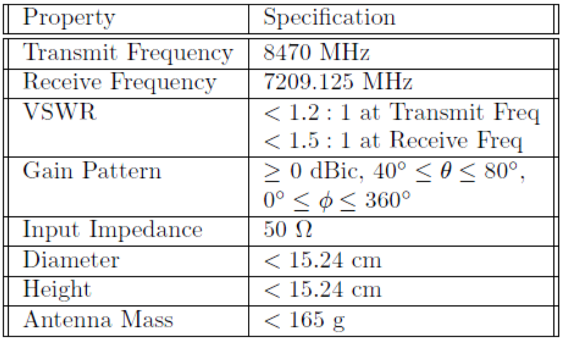
To achieve the specified electromagnetic properties, we must scheme out antenna's structure and size. Since the mapping between antenna's configuration and electromagnetic properties satisfies Maxwell's equations, the problem of antenna design is eventually a kind of function optimization. For the ST-5 antenna, we take into account many factors such as antenna's structure and size, pattern, VSWR (voltage standing wave ratio), etc., and then model the ST-5 antenna optimization as a constrained multi-objective optimization problem. Based on this mathematical model, we devise the chromosome representation, the inequality constraints and the cost function, so that a model of evolutionary algorithm is presented. And then some satisfied designs have achieved using a self-adaptive DE algorithm. At last, we show the results of the experiments and make a conclusion.

2 Model of Evolutionary Antenna

2.1 ST-5 Mission Antenna

NASA's Space Technology 5 mission (ST-5) is one of NASA's New Millennium Program missions to measure the effect of solar activity on the Earth's magnetosphere in the harsh environment of space. There are three satellites in the mission with two antennas on each of them which are used to communicate with a 34 meter ground-based dish antenna. The key ST-5 mission antenna requirements are summarized in Table 1.

Table 1. Key ST-5 Antenna Requirements [2]



Property	Specification
Transmit Frequency	8470 MHz
Receive Frequency	7209.125 MHz
VSWR	$< 1.2 : 1$ at Transmit Freq $< 1.5 : 1$ at Receive Freq
Gain Pattern	≥ 0 dBic, $40^\circ \leq \theta \leq 80^\circ$, $0^\circ \leq \phi \leq 360^\circ$
Input Impedance	50 Ω
Diameter	< 15.24 cm
Height	< 15.24 cm
Antenna Mass	< 165 g

According to these requirements and by reference to the DS Linden's "crooked-wire genetic antenna" [11], the experts in NASA Ames Research Center have evolved a novel antenna which is radically different from any typical antenna. And the result shows that these antennas meet the ST-5 antenna requirements very well.

There are three kinds of chromosome representations for this type of antenna. Binary chromosome [12] and real chromosome [2, 13] have been used in the non-branching situation, which encode each component of the (x, y, z) coordinate for the terminal points of wire into some bits or a real number. While a LOGO-like antenna constructing programming language was devised for the branching situation [2, 13].

Here the performance of an antenna design is decided by three factors commonly, which are VSWR, gain and outlier. The goal of evolution is to minimize the cost function F, which describe antenna performance. The expressions of the three factors is shown in formula (1) ~ (3), and the cost function F in formula (4). The gain component of the fitness function uses the gain (in dBic) in 5° increments about the angles of interest: from $40° \leq \theta \leq 90°$and $0° \leq \varphi \leq 360°$, where θ is the elevation and φ is azimuth. [2, 13]

$$\begin{aligned}
v_r &= \text{VSWR at receive frequency} \\
v_r' &= \begin{cases} v_r + 2.0(v_r - 1.25) & \text{if } v_r > 1.25 \\ v_r & \text{if } 1.25 > v_r > 1.1 \\ 1.1 & \text{if } v_r < 1.1 \end{cases} \\
v_t &= \text{VSWR at transmit frequency} \\
v_t' &= \begin{cases} v_t + 2.0(v_t - 1.15) & \text{if } v_t > 1.15 \\ v_t & \text{if } 1.15 > v_t > 1.1 \\ 1.1 & \text{if } v_t < 1.1 \end{cases} \\
vswr &= v_r' v_t'
\end{aligned} \tag{1}$$

$$\begin{aligned}
gain_{ij} &= \text{gain at } \theta = 5^\circ i,\ \phi = 5^\circ j \\
gain(i,j) &= \begin{cases} 0 & \text{if } gain_{ij} > 0.5 \\ 0.5 - gain_{ij} & \text{if } gain_{ij} < 0.5 \end{cases} \\
gain &= 1 + 0.1 \sum_{i=8}^{i<19} \sum_{j=0}^{j=72} gain(i,j)
\end{aligned} \tag{2}$$

$$\begin{aligned}
outlier(i,j) &= \begin{cases} 0.1 & \text{if } gain_{ij} < 0.01 \\ 0 & \text{otherwise} \end{cases} \\
outlier &= 1 + \sum_{i=8}^{i<19} \sum_{j=0}^{j=72} outlier(i,j)
\end{aligned} \tag{3}$$

$$F = vswr \times gain \times outlier \tag{4}$$

The EA was halted after 100 generations had been completed, the EA's best score was stagnant for 40 generations, or EA's average score was stagnant for 10 generations. [2, 13]

Since a design is evaluated by a single score and the halt condition is not directly related to whether a satisfied design has found, it cannot guarantee that the satisfied solution must be found in each of the implementation of EA. In addition, the value of every sample point within the specified space would be processed, which will affect the efficiency of the algorithm.

For the two shortcomings, we have proposed an improved mathematical and algorithmic model for ST-5 antenna.

2.2 Mathematical Model for Evolutionary Antenna

The design of antenna is eventually a kind of constrained multi-objective optimization problem, in which many requirements should be taken into account such as structure and size of antenna, pattern, VSWR (voltage standing wave ratio) and so on.

According to the ST-5 antenna requirements and some provisions of NEC2, the main constraints of the antenna optimization include antenna size, antenna mass, VSWR of transmit and receive frequency, gain both in transmit and receive frequency, wire length, angle between adjacent wire. The goal of optimization is to minimize the VSWR while maximize the gain.

Since the gain and VSWR of an antenna can be regarded as a function of the antenna structure, the mathematical model of ST-5 antenna has been abstracted as follows.

The objective functions consist of four functions to be minimized. The minimization of the VSWR of transmit and receive frequency is described as formula (5) and (6). The minimization of the negative gain in transmit frequency and receive frequency is described as formula (7) and (8). The sample space of gain is the same as NASA's.

$$\min(\mathrm{T\_VSWR}(\mathrm{antenna\_structure})) \tag{5}$$

$$\min(\mathrm{R\_VSWR}(\mathrm{antenna\_structure})) \tag{6}$$

$$\min_{\theta,\varphi}(-T\_Gain(antenna\_structure)) \tag{7}$$

$$\min_{\theta,\varphi}(-R\_Gain(antenna\_structure)) \tag{8}$$

The constraints aim at VSWR, gain, and antenna's configurations. The constraints on the VSWR of transmit and receive frequency is as formula (9) and (10). The constraints on the negative gain in transmit and receive frequency is shown in formula

(11) and (12). While the constraints on antenna's configuration is presented in formula (13)~(17), which separately describe the specification on diameter and height of antenna volume, antenna's mass, length of every wire, and angle between the adjacent wires.

$$\text{T\_VSWR(antenna\_structure)} < 1.2 \tag{9}$$

$$\text{R\_VSWR(antenna\_structure)} < 1.5 \tag{10}$$

$$-\underset{\theta,\varphi}{T\_Gain}(antenna\_structure) \le 0 \tag{11}$$

$$-\underset{\theta,\varphi}{R\_Gain}(antenna\_structure) \le 0 \tag{12}$$

$$Diameter(antenna\_structure) < 15.24cm \tag{13}$$

$$Height(antenna\_structure) < 15.24cm \tag{14}$$

$$Mass(antenna\_structure) < 165g \tag{15}$$

$$\frac{1}{10}\lambda \le Wire\_Length(antenna\_structure) \le \frac{1}{2}\lambda \tag{16}$$

$$Angle(antenna\_structure) \ge 20^{\circ} \tag{17}$$

2.3 Algorithmic Model for Evolutionary Antenna

An EA is composed of several elements: the chromosome representation of a design, the mating and mutation operators, the cost function, selection and replacement schemes, and runtime parameters like population size. Among these elements, chromosome representation and cost function would vary with different problems, while other elements can follow the classical method or empirical values.

Since NEC2 was used to simulate the design, we present a real chromosome with the characteristics of NEC2.

The initial feed wire is a thumbnail lead, starting near by origin with a specified length along the positive Z-axis, and z0 is the feed wire length. The other wires start from end of the feed wire, end to end. Each wire can be described by the 3D coordinates (x, y, z) of its two terminal points. The radius of the wire is represented by r. So the chromosome of an antenna with n wires is (r, z0, x1, y1, z1, x2, y2, z2, x3, y3, z3, …, xn, yn, zn), as shown in Fig. 1.

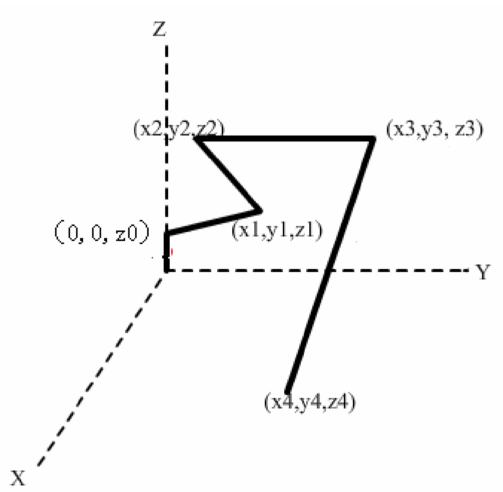


Fig. 1. Chromosome of an antenna with 4 wire

There are three types of constraint handling. The first type is the constraints on the size of the antenna structure, which is handled in the born of a new design. Through specifying the bounds of every gene, it can guarantee that the size of the new-born design must be valid. We also check up the validity of the wire length and the angle as soon as a design is generated before NEC2 is called. If the structure does not meet the constraints, it must be invalid and should be regenerated. This can greatly save computing resources.

The second type is the constraints on antenna structure inherent in NEC2, i.e. the length of every wire must be more than 0.02m, which is represented in formula (18).

$$\text{lengthConstraints[k]} \begin{cases} 0, & \text{if wireLength[k]} \leq 0.02 \\ \text{wireLength[k]-0.02}, & \text{else} \end{cases} \tag{18}$$

The third type is the constraints on electromagnetic properties of antenna. Since all the gains within the specified space must be greater than 0, we can just consider the minimum of the gains. Because as long as the minimum is greater than 0, the gain of every sample point in the space must meet the requirement. The requirement for VSWR and gain both in transmit and receive frequency is represented in formula (19) ~ (22).

$$Vt\ \_constraints \begin{cases} 0, & \text{if } Vt < 1.2 \\ Vt - 1.2, & \text{else} \end{cases} \tag{19}$$

$$Vr\ \_constraints \begin{cases} 0, & \text{if } Vr < 1.5 \\ Vr - 1.5, & \text{else} \end{cases} \tag{20}$$

$$\text{Gt\_constaints} \begin{cases} 0, & \text{if } \min(\text{Gain\_t}) > 0 \\ 0 - \min(\text{Gain\_t}), & \text{else} \end{cases} \tag{21}$$

$$\text{Gr\_constaint} \begin{cases} 0, & \text{if } \min(\text{Gain\_r}) \\ 0 - \min(\text{Gain\_r}), & \text{else} \end{cases} \tag{22}$$

Where Vt and Vr respectively represents the VSWR of transmit and receive frequency, while min(Gain_t) and min(Gain_r) respectively represents the minimum gains within the specified space in transmit and receive frequency.

The electromagnetic properties of an antenna are also regarded as the objectives. The cost function F as in formula (23) is constructed by the weight sum of all the objectives so that the weight can be adjusted to help the optimization of antenna.

$$\begin{aligned} F &= W_{Vt} * Vt\_constriants + W_{Vr} * Vr\_constra\text{int}\,s \\ &+ W_{Gt} * Gt\_constra\text{int}\,s + W_{Gr} * Gr\_constra\text{int}\,s \end{aligned} \tag{23}$$

Antenna optimization is a nonlinear, constrained puzzle with a huge search space and a large number of local optima. Since differential evolution (DE) algorithm is very efficient at solving the global optimal problem [14], and has been employed in many electromagnetic fields [15]. A kind of self-adaptive DE algorithm is employed. The result is satisfied.

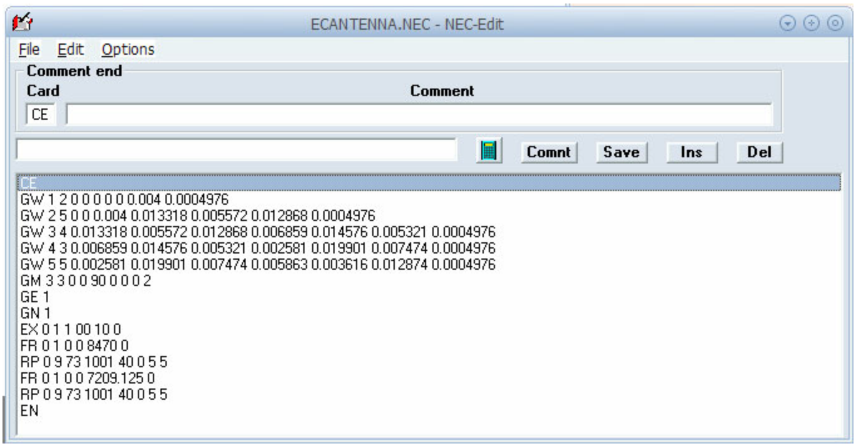
3 Results of Experiment

An infinite ground plane approximation was used in all runs, although in reality antenna's base is always located on a finite ground. That is because the infinite ground setting would provide sufficient accuracy, and the simulation time is still of the order of second. While the simulation time with finite ground would increase several times. Since the simulation time is so vast and the computing is based on single-core, we run the evolution in parallel on a distribution platform to speed up.

A set of experiments were carried out on a PC with four-core and 2GB memory. There were three sub-populations running severally with a population of 50 individuals each. Every 20 generations, the adjacent sub-populations exchanged their current best individual. The iterations in the sub-populations halted after 5000 generations had been completed, or the every value of the constraints of the best individual had attained zero. All the weights of cost function F were set to 1.

We run the evolution 30 times separately based on the algorithmic model. Generally, the best individual would meet all of the constraints after 500~600 generations. Except once the variance of the sub-populations reached zero so that the iteration could not continue, all of the iterations obtained satisfied solution. We have acquired nearly 90 satisfied designs from these experiments, and the results of an evolved antenna as well as the statistic data of all of the results are shown as follows.

The results from NEC2 is presented in figure 2 ~ 4 as follows. The NEC2 input file of the antenna is listed in figure 2, and the 3D structure of the antenna is shown in figure 3.



```
CE
GW 1 2 0 0 0 0 0 0.004 0.0004976
GW 2 5 0 0 0.004 0.013318 0.005572 0.012868 0.0004976
GW 3 4 0.013318 0.005572 0.012868 0.006859 0.014576 0.005321 0.0004976
GW 4 3 0.006859 0.014576 0.005321 0.002581 0.019901 0.007474 0.0004976
GW 5 5 0.002581 0.019901 0.007474 0.005863 0.003616 0.012874 0.0004976
GM 3 3 0 0 90 0 0 0 2
GE 1
GN 1
EX 0 1 1 00 10 0
FR 0 1 0 0 8470 0
RP 0 9 73 1001 40 0 5 5
FR 0 1 0 0 7209.125 0
RP 0 9 73 1001 40 0 5 5
EN
```

Fig. 2. Input file in NEC2

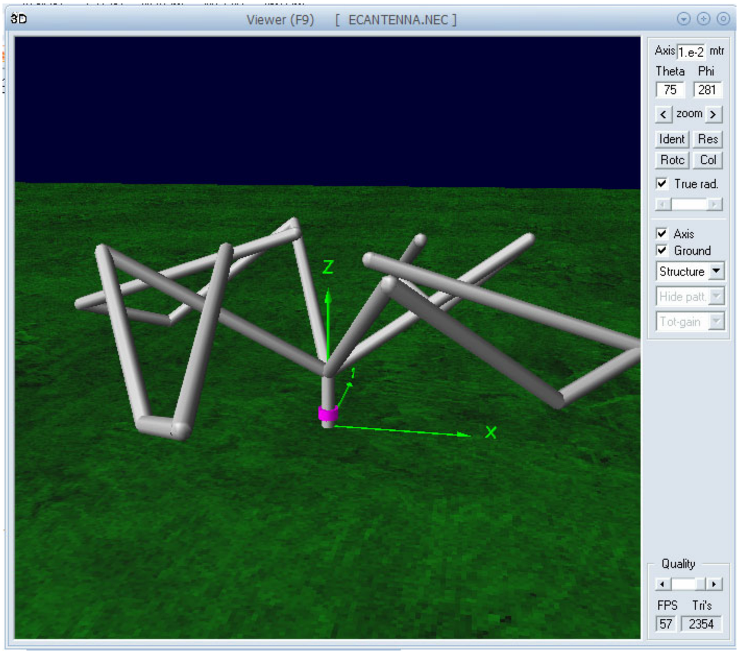
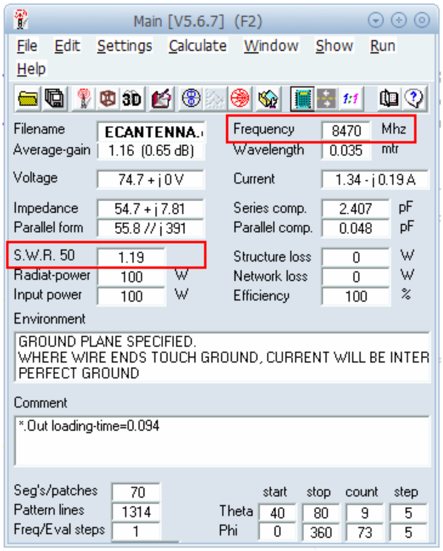


Fig. 3. 3D structure in NEC2

The antenna's VSWR of transmit frequency (8470 MHz) is 1.19, and the VSWR of the receive frequency (7209.125 MHz) is 1.5, whose results from NEC2 are presented in figure 4(a) and figure 4(b).



Filename ECANTENNA.
Average-gain 1.16 (0.65 dB)
Frequency 8470 Mhz
Wavelength 0.035 mtr
Voltage 74.7 + j 0 V
Current 1.34 - j 0.19 A
Impedance 54.7 + j 7.81
Parallel form 55.8 // j 391
Series comp. 2.407 pF
Parallel comp. 0.048 pF
S.W.R. 50 1.19
Radiat-power 100 W
Input power 100 W
Structure loss 0 W
Network loss 0 W
Efficiency 100 %

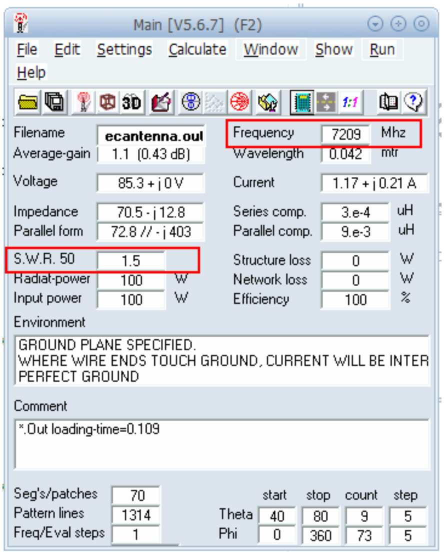
Environment
GROUND PLANE SPECIFIED.
WHERE WIRE ENDS TOUCH GROUND, CURRENT WILL BE INTER
PERFECT GROUND

Comment
*.Out loading-time=0.094

Seg's/patches 70
Pattern lines 1314
Freq/Eval steps 1

	start	stop	count	step
Theta	40	80	9	5
Phi	0	360	73	5

Fig. 4(a). VSWR of transmit frequency



Filename ecantenna.oul
Average-gain 1.1 (0.43 dB)
Frequency 7209 Mhz
Wavelength 0.042 mtr
Voltage 85.3 + j 0 V
Current 1.17 + j 0.21 A
Impedance 70.5 - j 12.8
Parallel form 72.8 // - j 403
Series comp. 3.e-4 uH
Parallel comp. 9.e-3 uH
S.W.R. 50 1.5
Radiat-power 100 W
Input power 100 W
Structure loss 0 W
Network loss 0 W
Efficiency 100 %

Environment
GROUND PLANE SPECIFIED.
WHERE WIRE ENDS TOUCH GROUND, CURRENT WILL BE INTER
PERFECT GROUND

Comment
*.Out loading-time=0.109

Seg's/patches 70
Pattern lines 1314
Freq/Eval steps 1

	start	stop	count	step
Theta	40	80	9	5
Phi	0	360	73	5

Fig. 4(b). VSWR of receive frequency

The antenna's gain in various angles is presented in the curve as figure 5(a) and 5(b), which has shown that all the gains are above zero.

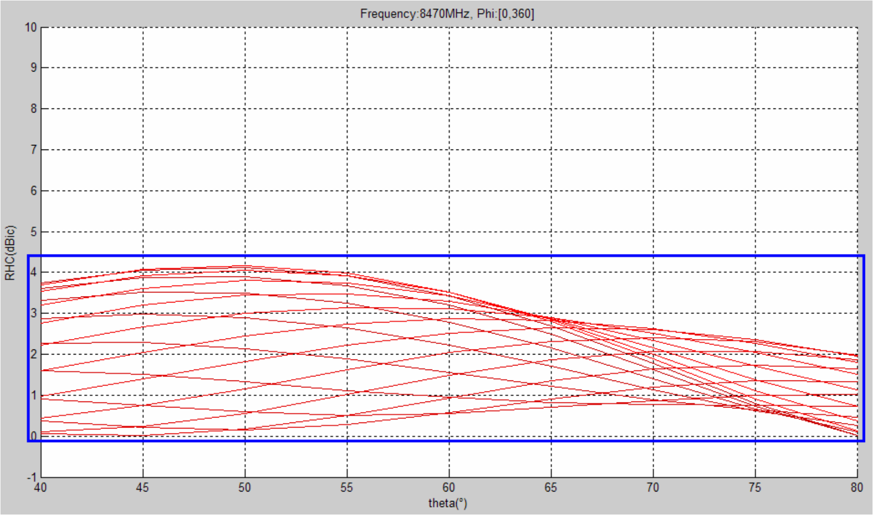


Fig. 5(a). Right-hand circular polarization gain in transmit frequency (8470 MHz)

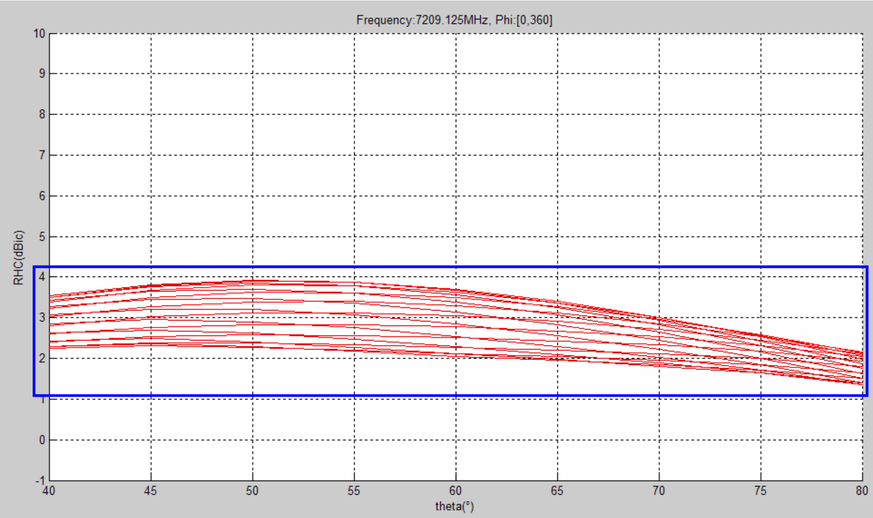


Fig. 5(b). Right-hand circular polarization gain in receive frequency (7209.125 MHz)

The statistic data of the results from the first sub-population in the 30 experiments is listed in table 2 and table 3. The data in table 2 show the speed of the evolution. The lattermost variances give the difference of the individuals in the sub-population, which represent the convergence degree. While the data in table 3 present the electromagnetic properties of the best design in the sub-population. It is noticeable that the sub-populations in No.11 experiment converged at 800 generation round and failed to continue for satisfied solution. and the quality of the evolution.

Table 2. Statistic data of evolution algorithm

No.	iteration number	cost time (minute)	lattermost variance
01	476	123	0.51
02	418	82	1.49
03	351	86	0.375
04	640	192	0.6
05	438	116	0.07
06	344	68	1.62
07	379	105	0.2
08	482	132	0.31
09	795	228	0.14
10	526	162	0.14
11	**720**	**312**	**0**
12	420	102	1.76
13	379	88	0.54
14	520	146	0.15
15	463	117	2
16	185	43	2.33
17	510	126	0.82

Table 2. (*Contiuned*)

18	379	116	0.2
19	832	235	0.35
20	587	167	2.18
21	590	136	1.2
22	231	55	1.9
23	457	107	1.88
24	579	156	0.18
25	456	107	0.17
26	307	71	2.54
27	472	122	0.73
28	461	135	0.27
29	819	241	0.16
30	563	175	0.17

Table 3. Electromagnetic properties of the best design

No.	VSWR (transmit freq)	VSWR (receive freq)	Min-Gain (transmit freq)	Max-Gain (transmit freq)	Min-Gain (receive freq)	Max-Gain (receive freq)
01	1.1928	1.487	0.0209	6.4291	0.0027	5.9303
02	1.1927	1.49735	0.04479	3.116	1.35047	3.3506
03	1.19339	1.49597	0.03969	2.3685	1.39699	3.3592
04	1.1942	1.4958	0.005745	2.844	1.0519	3.0143
05	1.19966	1.49996	0.00943	4.17983	1.01536	3.76219
06	1.19981	1.49965	0.004486	3.12901	1.32617	3.24333
07	1.19912	1.49252	0.044177	2.65602	1.00073	3.069747
08	1.189958	1.49103	0.037184	2.229298	1.2378	3.13355
09	1.199746	1.497889	0.005323	2.60124	1.02926	3.001759
10	1.194839	1.457066	0.0219998	2.362476	1.13946	3.11532
11	**1.383723**	**1.49947**	**0.000612**	**2.46622**	**0.002843**	**4.73149**
12	1.1956	1.495037	0.024746	3.305879	1.45032	3.48033
13	1.1996	1.49489	0.0953	2.2329	1.27269	3.2568
14	1.1877	1.49855	0.009765	2.9246	1.0117	2.97566
15	1.1867	1.49996	0.00636	4.16365	1.07358	3.6953
16	1.1991	1.499107	0.086975	3.32204	1.27064	3.37773
17	1.1950	1.49453	0.07236	2.617398	1.13396	3.10204
18	1.1975	1.49696	0.02393	2.23765	1.19949	3.13623
19	1.1956	1.49724	0.0002831	2.51567	1.05742	3.013237
20	1.1966	1.49141	0.036027	2.420347	1.13492	3.08235
21	1.194985	1.47006	0.02965	6.342362	0.0102	6.0109
22	1.19644	1.49807	0.00972	3.1996	1.4347	3.44607
23	1.1825	1.4999	0.0066	2.1995	1.21277	3.2379
24	1.19678	1.47596	0.05345	2.8855	0.96887	2.9981
25	1.19006	1.49809	3.55161	4.17809	1.0514	3.72584
26	1.17108	1.499077	0.02654	3.23355	1.32603	3.28012
27	1.18922	1.499082	0.007579	2.599009	1.02829	3.137166
28	1.189958	1.49103	0.037184	2.229298	1.2378	3.13355
29	1.191825	1.49784	0.020009	2.53045	1.06639	3.016254
30	1.15362	1.495608	0.003777	2.37381	1.1399	3.09479

4 Conclusion

With the addition of constraints to the mathematical model and the algorithmic model, the requirements for antenna's electromagnetic properties are not only reflected in the cost function, and also in the constraints. As soon as the solution meets all the constraints, it can be guaranteed to meet all the antenna specification. So that as long as the algorithm does not converge, it will continue until the satisfied solution is found. If the goal is not an optimal but a satisfied solution, the evolution will halted automatically as soon as the satisfied design is found with a reasonable halt condition, which can save much computing time. If the halt condition is whether the evolution has converged, the algorithm will continue to search for better designs even the satisfied solution have been achieved.

In addition, since the target of gain involves a variety of sampling points, the method of the comparison between the minimum and the target, instead of the comparison between every sample and the target, has saved computing resource obviously.

To further speed up optimization, we employ a parallel evolution. The sub-population will exchange the evolutionary information every several generations. The evolution has been greatly improved not only in speed, but also in quality.

Regrettably, we can not attain data in a realistic environment without test equipment. But we have confidence on NEC2 whose computational results usually agree well with the actual measurements, since it has been used to simulate many different kinds of antennas for several decades [3]. So we can believe that the mathematical model and the algorithmic model presented here is feasible and effective. We believe the technology of evolutionary antenna will play an increasingly important role in antenna design, especially when producing antennas with somewhat unconventional specifications.

References

1. Hupp, E., Chandler, L.: NASA's Micro-Satellites Complete Technology Validation Mission (2006)
2. Lohn, J.D., et al.: Evolutionary design of an X-band antenna for NASA's space technology 5 mission. In: Proceedings of NASA/DoD Conference on Evolvable Hardware 2003, pp. 155–163 (2003)
3. Linden, D.S., Altshuler, E.E.: Wiring like mother nature [antenna design]. Potentials, IEEE, 9–12 (1999)
4. Linden, D.S., Altshuler, E.E.: Evolving Wire Antennas Using Genetic Algorithms: A Review. In: The First NASA/DOD Workshop on Evolvable Hardware (1999)
5. Michielssen, E., et al.: Design of Lightweight, Broad-band Microwave Absorbers Using Genetic Algorithms. Microwave Theory & Techniques 41, 1024–1031 (1993)
6. Haupt, R.L.: An introduction to genetic algorithms for electromagnetic. In: Antennas and Propagation Magazine, IEEE, pp. 7 –15 (1995)
7. Haupt, R.L.: Genetic algorithm design of antenna arrays. In: Aerospace Applications Conference, IEEE, vol. 1, pp. 103–109 (1996)
8. Globus, A., Linden, D., Lohn, J.: Evolutionary design of a phased array antenna element. In: Antennas and Propagation Society International Symposium 2006, IEEE, pp. 2071–2074 (2006)

9. Lohn, J.D.: Evolutionary optimization of a quadrifilar helical antenna. In: Antennas and Propagation Society International Symposium, IEEE, Los Alamitos (2002)
10. Burke, J., Poggio, A.: NEC (1981)
11. Linden, D.S., Altshuler, E.E.: Automating Wire Antenna Deign using Genetic Algorithms. Microwave Journal 39, 74–86 (1996)
12. Altshuler, E.E., Linden, D.S.: Wire-antenna designs using genetic algorithms. Antennas and Propagation Magazine, IEEE, 33–43 (1997)
13. Lohn, J.D., Hornby, G.S., Linden, D.S.: An Evolved Antenna for Deployment on Nasa's Space Technology 5 Mission. In: Genetic Programming Theory and Practice, vol. II, pp. 301–315. Springer, US (2005)
14. Qing, A.: Fundamentals and Applications in Electrical Engineering, p. 352. Wiley-IEEE Press (2009)
15. Qing, A.: An Introductory Survey on Differential Evolution in Electrical and Electronic Engineering. In: Differential Evolution: Fundamentals and Applications in Electrical Engineering, IEEE, pp. 287–310 (2009)

A Synthesis of Four-Branch Microwave Antenna by Evolution Algorithm and Orthogonal Experiment

Jincui Guo, Jinxin Zou, Yincheng Wang, Xiaojuan Zhao, and Liangjiang Yu

Faculty of Mechanical & Electronic Information, China University of Geosciences,
WuHan, 430074, China
guojc888@vip.sina.com, jinxinzou@163.com

Abstract. In order to meet the design requirements of a wide bandwidth, wide beam four-branch microwave antenna, evolution algorithms and orthogonal experiment are combined for the design simulation of the microwave antenna design. The evolution algorithm is able to search the large space for the optimal solution, while the orthogonal experimental design can search local space further and refine it for even better solution. The former algorithm is executed parallelly, and the NEC2 software based on method of moments is used as numerical calculation method of electromagnetic field. The four-branch antenna which is designed by combining the method of the evolution algorithm with orthogonal experiment in this paper, performs excellent on pattern. This proves that this method can satisfy the design requirements.

Keywords: evolution algorithm; orthogonal design; method of moments; satellite antenna.

1 Introduction

About the optimization problems of microwave antenna synthesis, the possible parameters often have the features of multi-peak, non-linear, shape value change and even discontinuity. Generally, the objective functions of the optimization problems have many different physical meanings and complex relationships, which refer to multi-objective optimization problem (MOP) [1]. The effects of surrounding environment are also needed to be taken into account, which makes the antenna synthesis usually need a large amount of computation operations. Therefore, it is very difficult for antenna designers to use traditional manual antenna design method [2, 3] for antenna synthesis.

Since 1990's, EA (Evolution Algorithm) has been applied to electromagnetic fields to resolve optimization problems [4]. EA only needs an objective function for the optimization problem, and the continuity, derivability of objective function do not need to be considered. On the other hand, EA can also accumulate the knowledge about the search spaces automatically, and thus controls the search process and speeds up the convergence rate. So EA can be applied in solving complex problems [5]. In addition, the populations and individuals of EA are easily to be executed parallelly, which can largely reduce the time for searching the solution of antenna design. Therefore, the application of EA and numerical calculation of electromagnetic field for

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 487–495, 2010.

antenna design can broaden the scope of antenna design and improve the pattern quality, which has become a hot topic in modern antenna research [4].

Orthogonal experiment design [6] is a popular method of researching multivariate or multilevel problems. Only a few representative samples are picked out from comprehensive experiments to carry out the experiment. The selected points are uniform and comparable. Because only selected samples are needed to be experimented, the orthogonal design can greatly reduce the workload and be widely used in many fields. In order to search the local spaces border and refiner in the antenna synthesis, the orthogonal experiment is carried out after the execution of evolutionary algorithm. In this paper, the synthesis method of a four-branch antenna is proposed by combining evolution algorithm and orthogonal experimental design, and the experiment results shows that it can be completed quickly.

2 Antenna Modeling and Evolution Algorithms

The possible solutions to the problems should be modeled out of coding sequences (chromosome) for evolution algorithms. And antenna structure is represented by linear sequences of real-values [7, 8] in this paper.

2.1 Coding of Antenna

The first short section of antenna is a feed wire, with direction of the positive Z axis, originating from the origin. The other four branches start from the end of the feed wire. The whole antenna structure is cylindrical symmetry about the Z axis [3], so only the branch in the first quadrant, where x>0, y>0 and z>0, are needed to be coded. The branch in the first quadrant contains four segments of conductor. The coding sequence is: $(R, F_0, x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4)$, where, R is the radius of wires, F_0 is the length of the feed wire, both are fixed in this evolution. The following parts represent the end coordinates of the branch in the first quadrant, separately.

2.2 The process of Evolution Algorithm

The execution process of evolution algorithm is shown in Figure 1. The select operator keeps the excellent coding results in the next generation [9]. With a larger fitness evaluation f(i), the individuals have a higher probability to appear in the next generation population again. The selection probability of an individual is defined as $P(i)=f(i)/\Sigma f(i)$, where Σf(i) is the sum of fitness values of all the individuals. In order to speed up the convergence rate, the best individual is saved to the next generation. Two individuals are selected from the population by rate $P_{c,}$ and the cross operator exchanges their chromosomes at a random point which attempts to search the whole searching space [9]. The mutation operation is to change the bits of coding string, which is selected from the population with a small mutation rate P_m. In order to avoid premature, adaptive mutation rate is used in this design, which is defined as:

$$P_m=0.03(1.0-H/L)^3+a \tag{1}$$

Where L is the length of chromosome, H is the Hamming distance of its parents individual, the 'a' is a constant and equals to 0.02. Then, the more similar the two individual is (i.e. H is very little), the greater P_m is, which helps to generate more new individuals, and avoids premature.

The EA halt criterion is that 1000 generations had been completed. Different EA parameter setting (i.e., population size, P_c and P_m) will cause different convergence behavior. In order to achieve good results, we set P_c to 45%, and group size to 50 from experience.

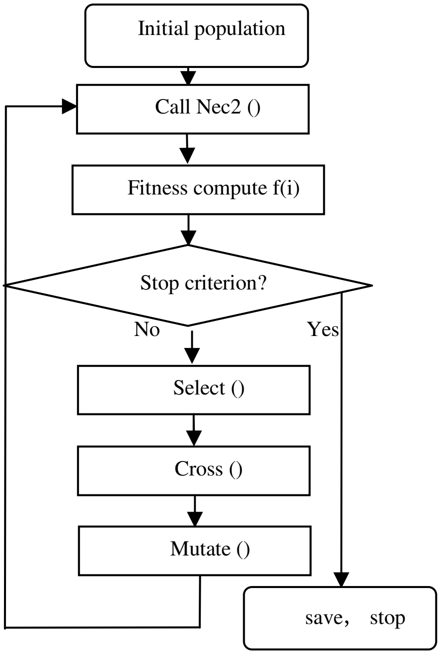


Fig. 1. Chart of evolution algorithms process

2.3 Evaluation of Individual Fitness

The main requirements of the design can be represented as follows: the receive and transmit gain should be larger than 0dBic, where 40°≤θ≤80°, 0°≤φ≤360°; receive VSWR (Voltages Standing Wave Ratio)≤1.2, transmit VSWR≤1.5. The method of exerting pressure on the individual is used instead of fitness in the design. i.e., the worst individual will get the greatest pressure. The relationship between pressure values P_{all} and fitness value f(i) is represent by $f(i)=1/(1+P_{all}(i))$.

The antenna VSWR represents the reflection of input signals. The smaller the VSWR is, the better the matching [10]. Thus when the individual's VSWR is greater than 1.2 (for the receiver is 1.5), strong pressure will be imposed on it, and when the VSWR is less than 1.1 (for the receiver is 1.2), there will be no pressure on it. A little pressure will be imposed on it when VSWR is set between the two values. The total

VSWR pressure equals to the product of receive VSWR pressure and the transmit VSWR pressure. The receipt VSWR pressure values are represented as:

```
If ($V_r$>=1.5) Pr=$V_r$+3.0*($V_r$-1.5);
Else if ($V_r$>=1.2) Pr=$V_r$;
Else $P_r$=1.0;
```

Gain is used to evaluate the radiation intensity of an antenna in a certain direction [3], with the characteristic of the bigger the better (where $40^\circ \leq \theta \leq 80^\circ$). The gain pressure value is the sum of all the field points and the total gain pressure is the product of receiving value and transmitting values. The gain values are stored by G_r[i][j], where i is the increment of θ, j of φ, and the increment is 5°. The receiving gain pressure values are decided by the following formulas:

```
If ($G_r$[i][j]>=0.5) $P_{g\_r}$[i][j]=0;
Else $P_{g\_r}$[i][j]=0.5-$G_r$[i][j];
$P_{gain\_r}$=1+0.1•i•j $P_{g\_r}$[i][j];
```

Since the objectives of the gain is very important, in case that gain is less than 0.2dB, a penalty function is need to make the gain evolution more effective. The receiving penalty function is decided by $P_{enalty}=1+\sum i \sum j\ P_{un}[i][j]$. Finally the total pressure values P_{all} is the product of three parts: gain pressure value, VSWR pressure value and the penalty value. Through the steps above, the multi-objective problem (MOP) is converted to a single-objective problem.

3 Parallel Execution and Orthogonal Experimental Design

3.1 Parallel Execution

Because evolutionary computation is very time-consuming, its execution process is designed parallelly. The parallel computing platform is a Beowulf cluster [4] consisting of 10 PCs connected by Ethernet. A master-slave model [4] is used for task sending. The master node is responsible for task allocation, and the slave nodes are used to complete the assigned task. The MPI (Message Passing Interface, version 1.2.5) [4] is used for message passing, where *mpi_send* and *mpi_recv* is used to send messages or receive messages between the nodes on the MPI communication layer.

3.2 Orthogonal Experimental Design

It takes a very long time to calculate an antenna by Ansoft HFSS [11] software. However, we can get total situation and optimal level combination by analyzing only a few representative samples selected by orthogonal experiment, and thus greatly reduce the workload [6]. The relationship between antenna gain and its structure is complex and unclear. However, when a very small change happens on the antenna structure, its impact of gain will be approximated as a linear and simple. Then the orthogonal design can be used in this case.

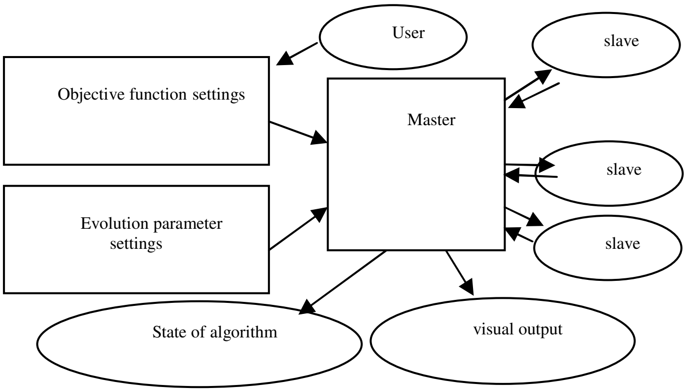


Fig. 2. Master-slave model

Orthogonal table is used to arrange the experiments [1, 6] in order to make sure that each factor combines once with all the other factors in the experiment. Orthogonal table has two characters, depicted as follows:

1) Each level occurs the same times in every column;

2) The permutation of each level is arranged completely and balancedly in every column. Because of the orthogonality of the table, selected samples must be distributed in the overall experiment points evenly. The orthogonal table used in the design has 27 rows, 13colums.

-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1 -1 0 0 0 0 0 0 0 0 0

-1 -1 -1 -1 1 1 1 1 1 1 1 1 1

-1 0 0 0 -1 -1 -1 0 1 0 1 0 1

-1 0 0 0 0 0 0 1 -1 1 -1 1 -1

-1 0 0 0 1 1 1 -1 0 -1 0 -1 0

……

Fig. 3. The first 6 lines of the orthogonal table

The format of evolutionary result is organized as *(F0, x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4)*. Where, '-1' in the orthogonal table means a decrease of Δ (0.3mm) from the evolutionary results, '1' means an increase of Δ (0.3mm) from the evolutionary results, and '0' means keeping the evolutionary results unchanged. The number of the first column shows the length of the feeder of antenna. The following 12 columns represent the 3 coordinate (x, y, z) of the branch in the first quadrant, respectively. After the 27 experiments calculated by Ansoft HFSS, the impact weight of each factor is evaluated. Usually, every factor has a best average score at certain level, and the optimal result can be predicted by combining the best factor.

4 The Design Results and Comparison

The design requirements of the four-branch satellite microwave antenna include:

1) Transmit frequency: 8470MHz; receive frequency: 7209.125MHz; right-hand polarization;

2) VSWR≤1.2 at transmit frequency, VSWR≤1.5 at receive frequency, Input impedance: 50Ω;

3) Gain Pattern >0dBic, with 40°<θ<80°, 0°<φ<360° (including receive and transmit);

4) Geometry size: Diameter<15.24cm, Height<15.24cm, the mass of antenna <165g.

The orthogonal experiment after evolution, which is used for further and refiner search of local spaces, improves the result from EA algorithm. The coordinates of the antenna branch in the first quadrant are:

(0,0,0.004), (0.0017,0.00567,0.019635), (0.014523,0.0017,0.009843), (0.008212,0.009701,0.003267), (0.013275,0.006707,0.020601).

The three-dimensional structure of the four-branch satellite microwave antenna is shown in figure 4. The transmit VSWR of the antenna is 1.19, and the receive VSWR is 1.48(shown in figure 5). Because input impedance is standard 50Ω, the evolved antenna can then be connected to the cable directly without using matching networks.

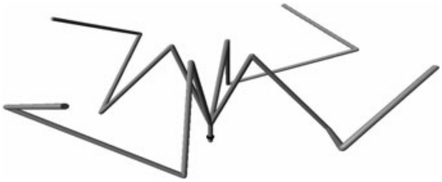


Fig. 4. Three-dimensional figure of the four-branch antenna

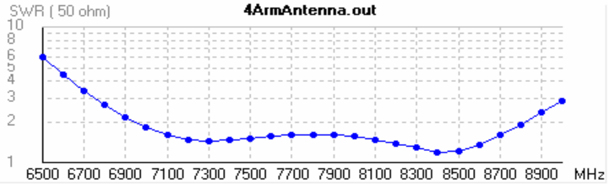


Fig. 5. The relationship between VSWR and frequency

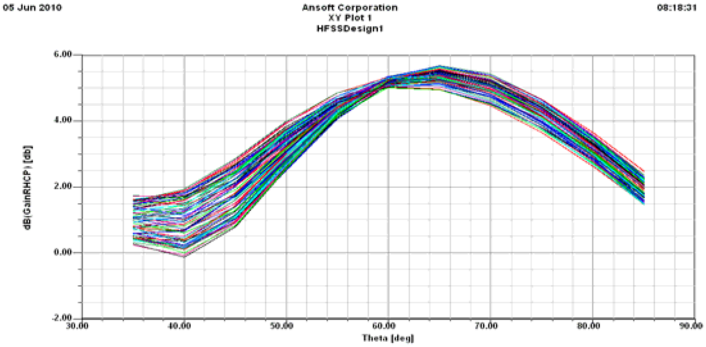


Fig. 6. The right-hand circular polarization gain chart by EA only, F = 8.470GHz(*X-axis isθ, Y-axis is the dB value of right-hand circular polarization gain*)

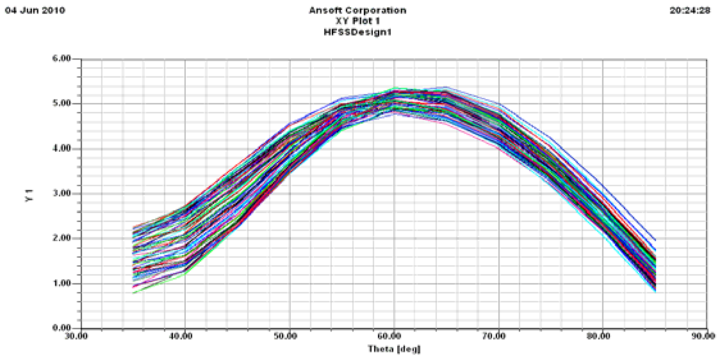


Fig. 7. The right-hand circular polarization gain chart after the improvements of orthogonal design, F = 8.470GHz(*X-axis is θ, Y-axis is the dB value of right-hand circular polarization gain*)

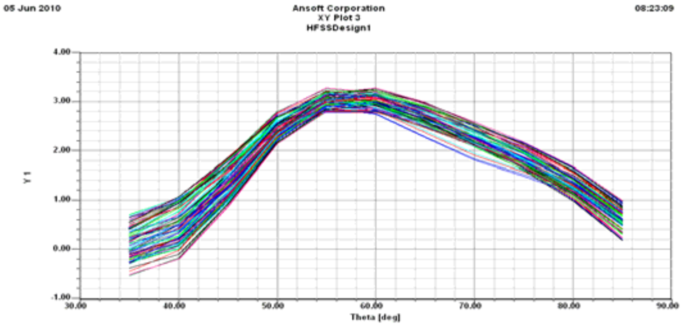


Fig. 8. The right-hand circular polarization gain chart by EA only, F = 7.209125GHz(*X-axis is θ, Y-axis is the dB value of right-hand circular polarization gain*)

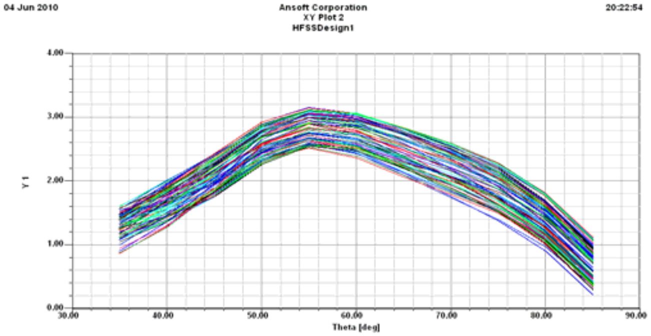


Fig. 9. The right-hand circular polarization gain chart after the improvements of orthogonal design, F = 7.209125GHz(*X-axis isθ, Y-axis is the dB value of right-hand circular polarization gain*)

The simulation results of the antenna gain by using Ansoft HFSS are shown in Figure 6,7,8,9 respectively. We can see that the gains have been largely improved after the orthogonal experimental design. Particularly, the gain improves about 0.6-1.5dB in case of θ=40°. Better results are achieved by the orthogonal experiments with small amount of computation. The four-branch antenna is 100% compliant with the requirements mentioned above, and the gain is as high as 4dBic at some places. Due to its extensive coverage, and its pattern meeting the requirements of Earth-beam matching this wide-beam, the wide-bandwidth microwave antenna can be applied to Satellite communications fields directly.

5 Conclusions

EA is an overall spaces searching algorithm based on probability. Comparing with the traditional manual method, it can largely improve the search range and produce antenna designs which cannot be designed by the traditional method. But the large searching space causes it needing much time to complete the design process. In addition, it has limited capacity for refiner local space search. By contrast, the orthogonal experimental design can search local spaces further and refiner, which is performed after the overall research by EA to reach a better result. The simulation results validate the feasibility of this method for the design of four-branch microwave antenna.

References

1. Zeng, S.Y., Kang, L.S., Ding, L.X.: An Orthogonal Multi-objective Evolutionary Algorithm for Multi-objective Optimization Problems with Constraints. Evolutionary Computation 12(1), 77–98 (2004)
2. Yang, Y., Zeng, S., Long, H., et al.: Automated Wire Antennas Design Using Dynamic Dominance Genetic Algorithm. In: Proceedings of the 2009 NASA/ESA Conference on Adaptive Hardware and Systems, pp. 257–262. IEEE Publisher, Los Alamitos (2009)
3. Cai, Z., Zeng, S., Yang, Y., Kang, L.: Automated Antenna Design Using Normalized Steady State Genetic Algorithm. In: NASA/ESA Conference on Adaptive Hardware and Systems, pp. 125–132 (2008)
4. Chen, X., Ka-ma, H., Zhao, X.: A Study of Applying NEC and Non-Blocking Master-Slave Parallel Genetic Algorithms to Automated Antenna Design. Chinese Journal of Electronics 32(8), 1390–1392 (2004)
5. Zeng, S., Yang, Y., Shi, Y., Yang, X., Xiao, B., Gao, S., Yu, D., Yan, Z.: A micro niche evolutionary algorithm with lower-dimensional-search crossover for optimisation problems with constraints. International Journal of Bio-Inspired Computation 1(3), 177–185 (2009)
6. Gao, S., Zeng, S., Xiao, B., Zhang, L., Shi, Y.: An Orthogonal Multi-objective Evolutionary Algorithm with Lower-dimensional Crossover. In: Proceedings of the 2009 Congress on Evolutionary Computatoin (CEC 2009), Norway, pp. 1959–1964 (2009)
7. Xu, J.-d., Zeng, S.-y., Wang, P., Yan, J.-f.: A Study of Encoding Mode in Automated Four-arm Branching Antenna Design. Microcomputer Information 32(7) (2007)

8. Lohn, J.D., Linden, D.S., Hornby, G.S., et al.: Evolutionary Design of an X-Band Antenna for NASA's Space Technology 5 Mission. In: Proceedings of the 2003 NASA/Dod Conference on Evolvable Hardware, pp. 155–163 (2003)
9. Yu, D., Yan, Z., Zeng, S., Yang, X., Shi, Y., Gao, S., Xiao, B.: A Dynamic Evolutionary Algorithm and Its Application in Automated Antenna Design. In: Proceedings of the 2009 World Summit on Genetic and Evolutionary Computation, Shanghai, China, pp. 929–932 (2009)
10. Hornby, G.S., Globus, A., Linden, D.S., Lohn, J.D.: Automated Antenna Design with Evolutionary Algorithms. In: AIAA Space 2006, San Jose, California, pp. 19–21 (2006)
11. Hou, W.-n., Shao, J.-x.: Application of Ansoft HFSS in teaching experiment of antenna. Digital communication, 87–89 (2009)

Band Structures of Multilayer Films with Randomness

Ping Li[1], Zhuo Li[2], and Yong Liu[3]

[1] School of Optoelectronics
Beijing Institute of Technology, Beijing 100081, China
Henan University of Science and Technology, Luoyang 471003, China
pingli818@163.com
[2] School of Optoelectronics
Beijing Institute of Technology, Beijing 100081, China
[3] School of Computer Science and Engineering
The University of Aizu, Aizu-Wakamatsu, Fukushima 965-8580, Japan

Abstract. Some randomness could likely occur in one way or another in the fabrication of photonic crystals. It is essential to understand how such randomness would affect the performance of the produced photonic crystals. In this paper, the band structures of multilayer films with different levels of randomness were analyzed by plane wave expansion (PWE) with supercell. The results clearly show how the lowest band gap decreased with the increased randomness. Meanwhile, a number of slim band gaps could appear in the perturbed multilayer films.

1 Introduction

In the past two decades, photonic crystals have been attracting a great attention from both research fields and industry due to their super ability of manipulating photons and potential commercial applications, such as photonic crystal fibers possessing enhanced properties over normal optical fibers [1,2]. Photonic crystals were firstly proposed by E. Yablonovitch and S. John in 1987 [3,4], which consist of periodic high and low dielectric constant. The multilayer film is one of the simplest photonic crystals which consists of alternating layers of material with different dielectric constants [2]. Although there are a number of two and three dimensional photonic crystals developed, the applications of higher dimensional photonic crystals have proceeded more slowly, and are still facing great challenges in fabrication of these structures with sufficient precision to preserve their crystal properties. In contrast, one-dimensional photonic crystals are already in widespread applications ranging from low and high reflection coatings on lenses and mirrors to color changing paints and inks.

Photonic crystals affect the propagation of electromagnetic waves in a similar way that a semiconductor crystal affects the motion of electrons through designing allowed and forbidden electric energy bands. Just as semiconductor crystals, photonic crystals also possess photonic band gaps in which light within certain ranges of frequency cannot propagate through. Such photonic band gaps make

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 496–504, 2010.

photonic crystals as ideal optical materials for controlling and manipulating the flow of light. Photonic band gaps are essential to many real-world applications of photonic crystals that depend on the location and width of photonic band gaps. To achieve a certain band structure with a required band gap is often a complex design task because both materials and structure parameters have shown to affect the band gaps, including the lattic geometry, dielectric constants, unit cell size, and filling fraction.

Both analytical solutions and computational photonic methods have been developed for determining band structures, such as transfer-matrix method [5], plane wave expansion (PWE) [6], finite difference time domain (FDTD) [7]. PWE has been used widely for determining the band structure of specific photonic crystal structures. PWE refers to a computational technique in electromagnetics to solve the Maxwell's equations by formulating an eigenvalue problem out of the equation through Fourier transform. PWE is particularly useful in calculating modal solutions of Maxwell's equations over an inhomogeneous or periodic geometry.

This paper applied PWE with supercell in analyzing the band structures of multilayer films with randomness. In the fabrication, it is possible that the produced multilayer film does not have a perfect periodic structure in which the layers with the same dielectric constant might have the different width. Since the values of layer width might appear at random in such perturbed multilayer films, it would be interesting to discover how the band structures could change with increasing randomness. The experiment results suggest that the lowest band gap would narrow down with increasing randomness. Meanwhile, a number of slim band gaps could appear in such nonperiodic crystals.

The rest of this paper is organized as follows. Section 2 describes PWE and the supercell method. Section 3 discusses how different randomness levels would affect the band structures of nonperiodic multilayer films. Finally, Section 4 concludes with some remarks and future research directions.

2 Plane Wave Expansion

Electromagnetic waves in the multiplyer films with a periodic distribution of dielectric constants can be described as the following Maxwell eigenvalue problem,

$$\nabla \times \left(\frac{1}{\varepsilon(r)} \nabla \times H \right) = \frac{\omega^2}{c^2} H \tag{1}$$

where $\varepsilon(r)$ is the position-dependent dielectric function, H is the magnetic field, ω is the frequency, and c is the speed of light in vacuum.

In solving Eq.(1) by PWE for a periodic function $H(r) = H(r+a)$ with period a, H can be represented by an infinite sum of sines and cosines, or in terms of complex exponentials,

$$H(r) = \sum_{G} H(G) e^{i(k+G)\cdot r} \tag{2}$$

where k is the wave vector, and G is the reciprocal lattice vector. Both vectors are in the unit of $2\pi/a$. Similarly, the dielectric function $\varepsilon(r)$ can also be represented in Fourier series,

$$\varepsilon(r) = \sum_{G} \varepsilon(G) e^{iG\cdot r} \tag{3}$$

By taking Eqs.(1), (2), and (3), the following eigen-equation could be obtained

$$\sum_{G'} |k+G||k+G'|\varepsilon(G-G')H(G') = \frac{\omega^2}{c^2}H(G) \tag{4}$$

where

$$\begin{aligned} \varepsilon(G) &= \frac{1}{a}\int_{-\frac{a}{2}}^{\frac{a}{2}} \varepsilon(G) e^{-iGx}\, dx && (5)\\ &= \frac{1}{a}(\varepsilon_1 - \varepsilon_2)\int_{-\frac{b}{2}}^{\frac{b}{2}} e^{-iGx}\, dx + \frac{1}{a}\varepsilon_2 \int_{-\frac{a}{2}}^{\frac{a}{2}} e^{-iGx}\, dx && (6) \end{aligned}$$

ε_1 and ε_2 refer the dielectric constants for the filling layers and background, respectively. $f = b/a$ is called filling rate in the structure.

PWE with single unit cell described in Eqs.(5) and (6) can be extended to a supercell with N unit cells in multilayer films. The basic idea of supercell method is to replace one unit cell with N unit cells. The lattice spacing is now given by Na. For example $N = 5$, a 5 supercell has an overall periodicity. In N supercell for a multilayer film, there is

$$\begin{aligned} \varepsilon(G) &= \frac{1}{Na}\int_{-\frac{Na}{2}}^{\frac{Na}{2}} \varepsilon(G) e^{-iGx}\, dx && (7)\\ &= \frac{1}{Na}\varepsilon_1 \int_{\Omega_1} e^{-iGx}\, dx + \frac{1}{Na}\varepsilon_2 \int_{\Omega_2} e^{-iGx}\, dx && (8) \end{aligned}$$

where Ω_1 is the sum of layers with the dielectric constant ε_1, and Ω_2 is the sum of layers with the dielectric constant ε_2.

Certainly, supercell method is more useful in nonperiodic photonic crystals, such as photonic crystals with defects. In this paper, supercell consists of N unit cells in which each unit has length a. However, in each unit, the length of the two layers is randomly perturbed. It is assumed that this perturbed unit cell would repeat in space so that PWE could be applied.

3 Experimental Studies

Two experiments of PWE with supercell were conducted. The first experiment was to test whether PWE with supercell could give the identical results to those obtained by PWE with single unit cell. The second experiment was to present the band structures of multilayer films with different amount of randomness. 1001 plane waves were used in the two experiments.

3.1 Band Structures of Periodic Multilayer Films

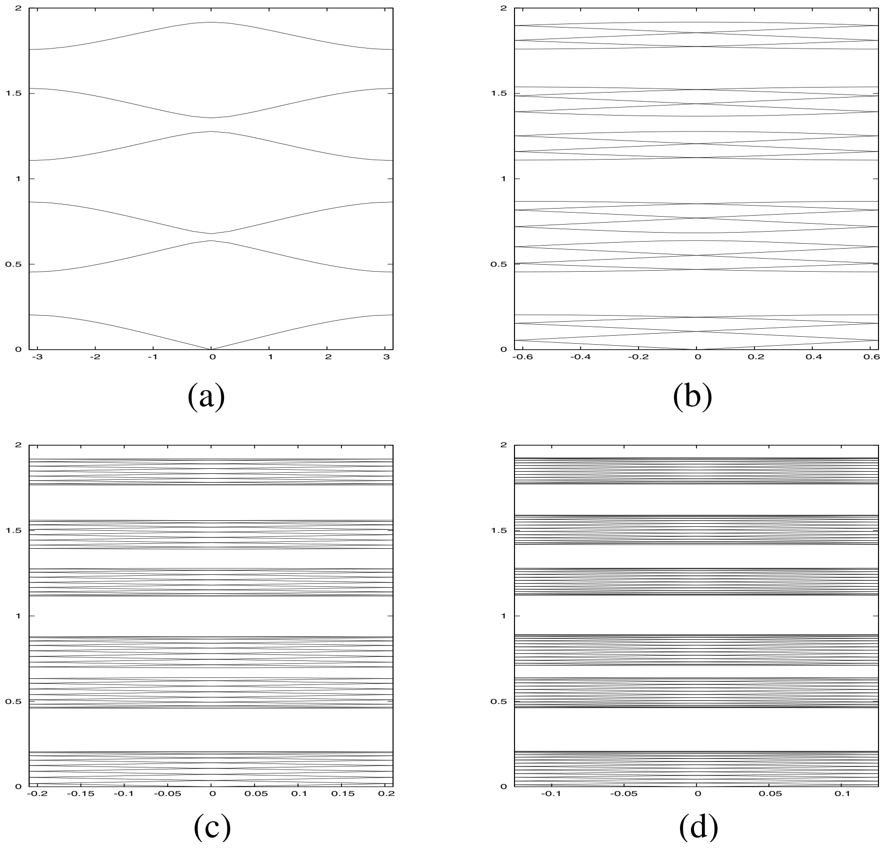
Figs. 1 and 2 describe the band structures of multilayer films obtained by PWE with one unit cell, 5 supercell, 15 supercell, and 25 supercell. Fig. 1 shows the results of the structure with the filling rate 0.2, while Fig. 2 displays the results of the structure with the filling rate 0.5. If the supercell consists of N unit cells, each band will be folded in N bands since the area of the first Brillouin zone is reduced by N. Although the number of supercell affected the macroscopic symmetry of the multiplayer film, the band gaps obtained by the supercell method with different number of unit cells should actually be the same. The very small difference in the band gaps were mearly led by the accuracy of the solutions, while supercell with large number of unit cells should require more number of plane waves.

In comparison between the two structures with the filling rates 0.2 and 0.5 respectively, the lowest band gap in the structure with the filling rate 0.2 is 0.251, that is much wider than 0.106, i.e. the lowest band gap in the structure with the filling rate 0.5. Within a certain range of filling rates, it is also true that the lower filling rate a multilayer film has, the wider the lowest band gap is. For the results of other band gaps, the changes with the filling rates are a little more complicated, but less interested. Therefore, only results of the lowest band gap were discussed in this paper.

3.2 Band Structures of Nonperiodic Multilayer Films

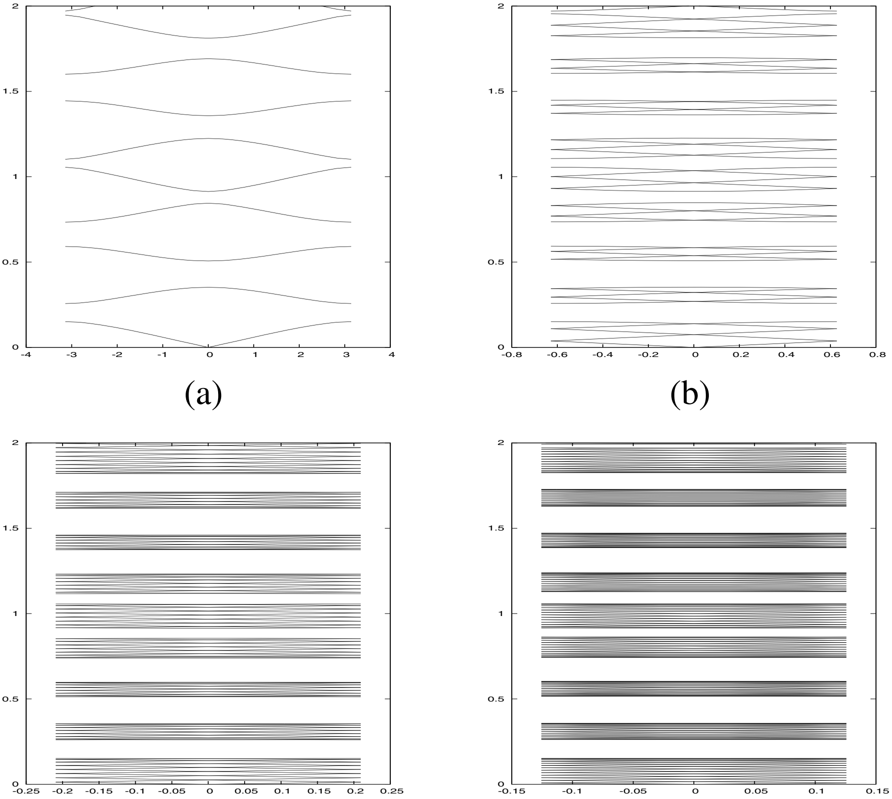
Figs. 3 and 4 describe the band structures of multilayer films with different amount of randomness, i.e. 10%, 25%, 50%, and 75% random changes introduced in the two initial filling rates. p% randomness means that the layer width could be either p% wider or p% narrower. Fig. 3 presents the results of the structure with randomness starting from the initial filling rate 0.2. In comparison, Fig. 4 gives the results of the structure with randomness beginning with the initial filling rate 0.5.

For the structure with the initial filling rate 0.2, the lowest band gap could be reasonably wide even after the level of randomness was increased to 50%. However, when randomness was increased to 75%, the lowest band gap was split into a number of very narrow band gaps. For the structure with the initial filling rate 0.5, the lowest band gap started to split into a number of slim band gaps after randomness reached at 50%. It suggests that the level of randomness could have larger impact on multilayer films with higher filling rate. The amount of changes on the band gap and gap-midgap ratio for the lowest band gap was given in Tables 1 and 2. The second column in the tables represents the results of the periodic structures without randomness.



(a) (b) (c) (d)

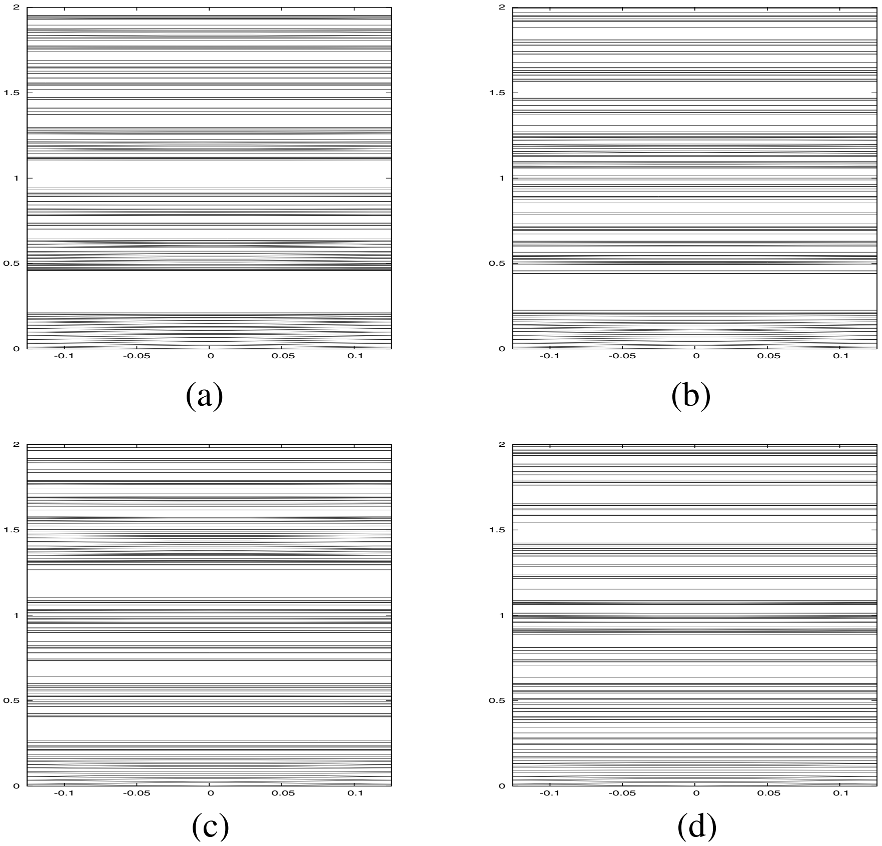
Fig. 1. Band structure with the filling rate 0.2. (a) Results by PWE with one unit cell; (b) Results by PWE with 5 supercell; (c) Results by PWE with 15 supercell; (d) Results by PWE with 25 supercell.



(a) (b)

(c) (d)

Fig. 2. Band structure with the filling rate 0.5. (a) Results by PWE with one unit cell; (b) Results by PWE with 5 supercell; (c) Results by PWE with 15 supercell; (d) Results by PWE with 25 supercell.



(a)

(b)

(c)

(d)

Fig. 3. Changes of band structure with the filling rate 0.2. (a) The layers contains 10% random change; (b) The layers contains 25% random change; (c) The layers contains 50% random change; (d) The layers contains 75% random change.

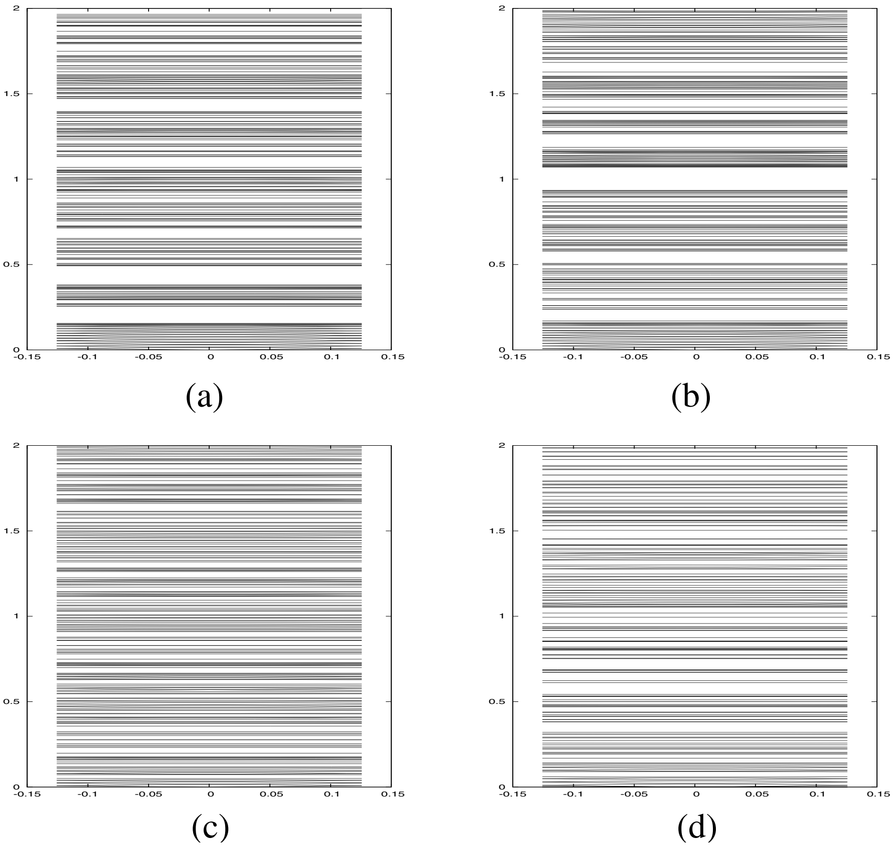


Fig. 4. Changes of band structure with the filling rate 0.5. (a) The layers contains 10% random change; (b) The layers contains 25% random change; (c) The layers contains 50% random change; (d) The layers contains 75% random change.

Table 1. Results of the lowest band gap in the multilayer film with the filling rate 0.2

rate of randomness	0	10%	25%	50%	75%
gap size	0.251	0.248	0.212	0.128	0.051
reduction of gap size	0	2.23%	16.4%	49.4%	79.9%
gap-midgap ratio	0.763	0.742	0.633	0.379	0.169
reduction of gap-midgap ratio	0	1.96%	16.4%	50.0%	77.6%

Table 2. Results of the lowest band gap in the multilayer film with the filling rate 0.5

rate of randomness	0	10%	25%	50%	75%
gap size	0.106	0.099	0.067	0.034	0.024
reduction of gap size	0	6.6%	36.8%	68%	77.4%
gap-midgap ratio	0.52	0.482	0.33	0.158	0.131
reduction of gap-midgap ratio	0	7.3%	36.5%	70.0%	74.8%

4 Conclusions

This paper reported how randomness in the layer width had affected the band structures of the multilayer films. Especially, the relation between the level of randomness and the changes of the lowest band gap were discussed. It was found that the smaller random changes on layers had less impact on the lowest band gap, while larger random changes could produce so enormous impact on the lowest band gap that the lowest band gap could finally be too narrow to be useful.

The comparisons between two multilayer films with different filling rate were also made on the the lowest band gap under the different levels of randomness in the structures. It had been shown that the multilayer films with higher filling rate would be affected greater by randomness. All the results in this paper were produced on the one-dimensional photonic crystals. It would be interesting to extend these results on two or three dimensional photonic crystals with randomness.

References

1. Russell, P.S.J.: Photonic-crystal fibers. Journal of Lightwave Technology 24, 4729–4749 (2006)
2. Joannopoulos, J.D., Johnson, S.G., Winn, J.N., Meade, R.D.: Photonic Crystals: Molding the Flow of Light, 2nd edn. Princeton University Press, Princeton (2008)
3. Yablonovitch, E.: Inhibited spontaneous emission in solid-state physics and electronics. Physical Review Letters 58, 2059–2062 (1987)
4. John, S.: Strong localization of photons in certain disordered dielectric superlattices. Physical Review Letters 58, 2486–2489 (1987)
5. Pendry, J.B., MacKinnon, A.: Calculation of photon dispersion relations. Physical Review Letters 69, 2772–2775 (1992)
6. Ho, K.M., Chart, C.T., Soukoulis, C.M.: Existence of a photonic gap in periodic dielectric structures. Physical Review Letters 65, 3152–3155 (1990)
7. Taflove, A., Hagness, S.C.: Computational Electrodynamics: The Finite-Difference Time-Domain Method, 2nd edn. Artech House Publishers, Boston (2000)

Fast Principal Component Analysis Based on Hardware Architecture of Generalized Hebbian Algorithm

Shiow-Jyu Lin[1,2], Yi-Tsan Hung[1], and Wen-Jyi Hwang[1,⋆]

[1] Department of Computer Science and Information Engineering,
National Taiwan Normal University, Taipei, 117, Taiwan
[2] Department of Electronic Engineering, National Ilan University, I-Lan, 260, Taiwan
sjlin.gm@gmail.com, 697470157@ntnu.edu.tw, whwang@csie.ntnu.edu.tw

Abstract. This paper presents a novel hardware architecture for fast principle component analysis (PCA). The architecture is developed based on generalized Hebbian algorithm (GHA). In the architecture, the updating of different synaptic weight vectors are divided into a number of stages. The results of precedent stages are used for the computation of subsequent stages for expediting training speed and lowering the area cost. The proposed architecture has been embedded in a system-on-programmable-chip (SOPC) platform for physical performance measurement. Experimental results show that the proposed architecture is an effective alternative for fast PCA in attaining both high performance and low computation time.

1 Introduction

Principal component analysis (PCA) [3] finds a linear transformation which reduces m-dimensional feature vectors to l-dimensional feature vectors (where $l < m$) in such a way that the information is maximally preserved in minimum mean squared error sense. This linear transformation is known as PCA transform or Karhunen-Loeve transform (KLT). It can also reconstruct m-dimensional feature vectors from the l-dimensional feature vectors with some finite error known as reconstruction error. The PCA and its variants are mostly used in compression and reconstruction of high dimensional feature vectors. The PCA-based algorithms have been successful in many research areas such as image processing [5].

Although many PCA-based algorithms are effective, the computational complexity of these algorithms are usually very high. This is especially true when the dimension of feature vector is large. Consequently, it may be difficult to adopt the PCA-based algorithms for real-time applications. A number of algorithms [1] have been proposed for accelerating the computational speed of PCA. However, most of these algorithms are implemented by software. Therefore, only moderate acceleration can be achieved. Although hardware implementation of

[⋆] To whom all correspondence should be sent.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 505–515, 2010.

PCA and its variants are possible, large memory consumption and complicated circuit control management are usually required. The resource consumption may become impractically high as the feature vector dimension grows. An alternative for the PCA implementation is based on the PCA neural network, which is also known as the generalized Hebbian algorithm (GHA) [4,2]. Nevertheless, the GHA may converge slowly, and achieving a good accuracy requiring excessive large number of iterations. Long computational time therefore is still required by many GHA-based algorithms.

The objective of this paper is to present a hardware architecture of GHA for fast PCA. Although large amount of arithmetic computations are required for GHA, the proposed architecture is able to achieve fast training with low area cost. The long datapath for the updating of synaptic weight vectors is separated into a number of stages. The results of precedent stages will be used for the computation of subsequent stages for expediting training speed and lowering the area cost. In addition, the direct memory access (DMA) can be employed for the training vector delivery to further reduce the computational time.

To demonstrate the effectiveness of the proposed architecture, a texture classification system on a system-on-programmable-chip (SOPC) platform is constructed. The system consists of the proposed architecture, a softcore NIOS II processor, a DMA controller, and a SDRAM. The proposed architecture is used for finding the PCA transform using the GHA training, where the training vectors are stored in the SDRAM. The DMA controller is used for the DMA delivery of the training vectors. The softcore processor is only used for coordinating the SOPC system. It does not participate the GHA training process. As compared with its software counterpart running on Pentium IV CPU, our system has significantly lower computational time for large training set. All these facts demonstrate the effectiveness of the proposed architecture.

2 Preliminaries

Figure 1 shows the corresponding neural model for the GHA, consisting of m inputs x_1,...,x_m, and l outputs y_1,...,y_l. The output $y_j(n)$ at time n produced in response to the set of inputs $x_1(n)$,...,$x_m(n)$ is given as follows:

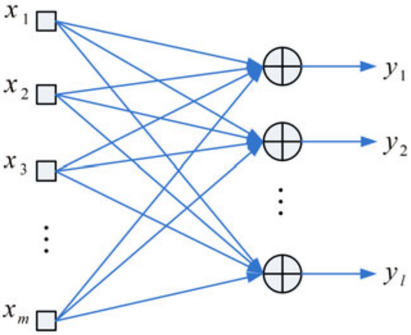


Fig. 1. The neural model for the GHA

$$y_j(n) = \sum_{i=1}^{m} w_{ji}(n)x_i(n). \tag{1}$$

The synaptic weight $w_{ji}(n)$ is adapted in accordance with a generalized form of Hebbian learning, as shown below:

$$w_{ji}(n+1) = w_{ji}(n) + \eta(y_j(n)x_i(n) - y_j(n)\sum_{k=1}^{j} w_{ki}(n)y_k(n)), \tag{2}$$

where η is the learning rate of the network. Let $\mathbf{w}_j(n) = [w_{j1}(n), ..., w_{jm}(n)]^T$ be the j-th synaptic weight vector in the GHA network. It can be shown that $\mathbf{w}_j(n)$ will converge as the eigenvector associated with the j-th principal component λ_j of the inputs, where $\lambda_1 > ... > \lambda_l$. Detailed discussions of GHA can be found in [2].

3 The Proposed GHA Architecture

For sake of brevity, define $\mathbf{x}(n) = [x_1(n), ..., x_m(n)]^T$, and $\mathbf{y}(n) = [y_1(n), ..., y_l(n)]^T$. Figure 2 shows the proposed GHA architecture based on eqs.(1) and (2). As shown in the figure, the GHA architecture can be divided into 3 units: the principal components computation unit, the synaptic weight updating unit, and the memory unit. All the synaptic weights $\mathbf{w}_j(n), j = 1, ..., l$, are stored in the memory unit. The synaptic weights $\mathbf{w}_j(n), j = 1, ..., l$, and the input $\mathbf{x}(n)$ are broadcasted concurrently to the principal components computation unit and synaptic weight updating unit for the computation of $\mathbf{y}(n)$ and $\mathbf{w}_j(n+1), j = 1, ..., l$, respectively. Based on eq.(1), the principal components computation unit is a circuit containing adders and multipliers for computing $\mathbf{y}(n)$, as shown in figure 3. Due to limited bus width for data transfer, the input vector $\mathbf{x}$ may not be transferred to the proposed architecture in single clock cycle. One way to solve the problem is to use the architecture shown in Figure 4, where $\mathbf{x}$ is divided into m/b segments for data

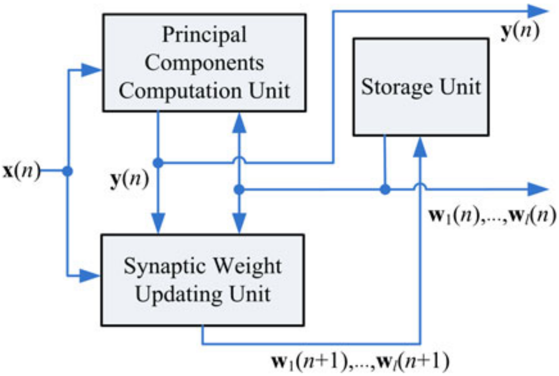

Principal Components Computation Unit
Storage Unit
Synaptic Weight Updating Unit
x(n)
y(n)
y(n)
w1(n),...,wl(n)
w1(n+1),...,wl(n+1)

Fig. 2. The proposed GHA architecture

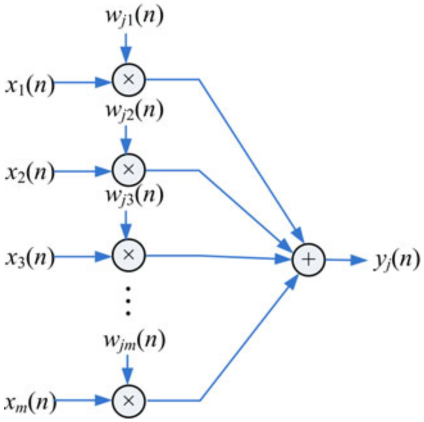


Fig. 3. The architecture of principal components computation unit

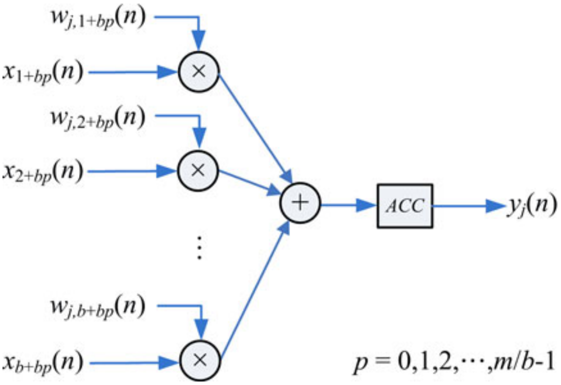


Fig. 4. The architecture of principal components computation unit for limited data bus width

transfer, where each segment contains b elements. We can also observe from the figure that there are only b multipliers in the circuit. This may effectively reduce the area cost when b is small.

The synaptic weight vectors are updated using $\mathbf{x}(n), \mathbf{y}(n)$ and $\mathbf{w}_j(n), j = 1, ..., l$, as inputs. A direct implementation of the synaptic weight updating unit based on eq.(2) is possible. Figure 5 shows the resulting design, which is termed Architecture I in this paper. The Architecture I consists of l modules, where each module j at time n is responsible for computing the synaptic weight vector $\mathbf{w}_j(n+1)$. Each module j in the architecture consists of m sub-modules, as depicted in Figure 6. Each sub-module ji operates in accordance with eq.(2). Figure 7 shows the architecture of each sub-module. The updated synaptic weights are then stored back to the memory unit.

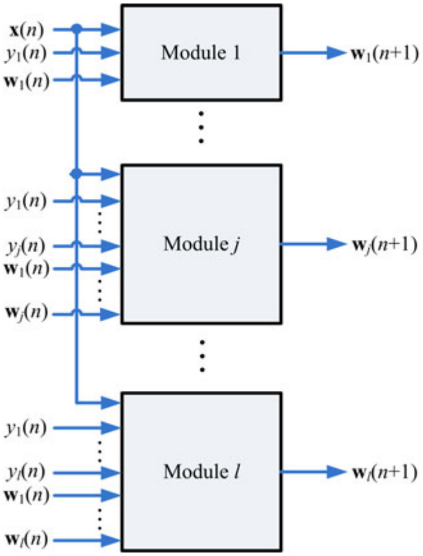


Fig. 5. Architecture I for implementing synaptic weight updating unit based on eq.(2)

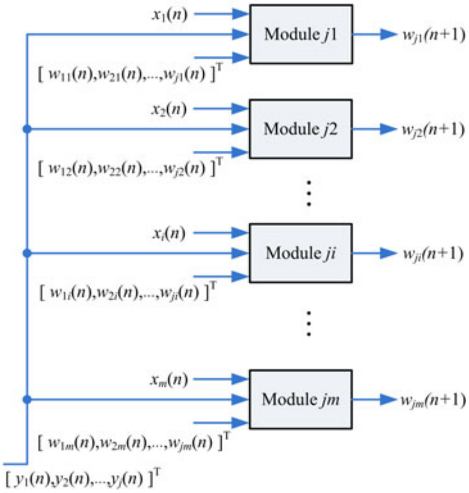

$x_1(n)$
Module $j1$
$w_{j1}(n+1)$
$[\, w_{11}(n), w_{21}(n), \ldots, w_{j1}(n)\,]^T$
$x_2(n)$
Module $j2$
$w_{j2}(n+1)$
$[\, w_{12}(n), w_{22}(n), \ldots, w_{j2}(n)\,]^T$
$x_i(n)$
Module ji
$w_{ji}(n+1)$
$[\, w_{1i}(n), w_{2i}(n), \ldots, w_{ji}(n)\,]^T$
$x_m(n)$
Module jm
$w_{jm}(n+1)$
$[\, w_{1m}(n), w_{2m}(n), \ldots, w_{jm}(n)\,]^T$
$[\, y_1(n), y_2(n), \ldots, y_j(n)\,]^T$

Fig. 6. The architecture of each module j in Architecture I

Although the implementation of Architecture I is simple, the hardware consumption of the architecture is high for large l and/or m. To reduce the area cost, first we observe that eq.(2) can be re-written as

$$w_{ji}(n+1) = w_{ji}(n) + \eta y_j(n)(x_i(n) - \sum_{k=1}^{j} w_{ki}(n) y_k(n)). \qquad (3)$$

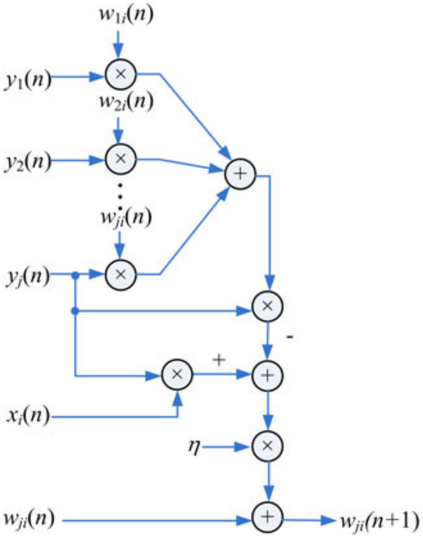


Fig. 7. The architecture of each submodule ji in Architecture I

Define

$$z_{ji}(n) = x_i(n) - \sum_{k=1}^{j} w_{ki}(n)y_k(n), j = 1, ..., l. \tag{4}$$

Substituting eq.(4) into eq.(3), we obtain

$$w_{ji}(n+1) = w_{ji}(n) + \eta y_j(n)z_{ji}(n). \tag{5}$$

Note that the $z_{ji}(n)$ associated with submodule ji can be obtained from $z_{(j-1)i}(n)$ associated with submodule $(j-1)i$ by

$$z_{ji}(n) = z_{(j-1)i}(n) - w_{ji}(n)y_j(n), j = 2, ..., l. \tag{6}$$

When $j = 1$, from eq.(4) it follows that

$$z_{1i} = x_i(n) - w_{1i}(n)y_1(n). \tag{7}$$

By comparing eqs(6)(7), it can be observed that eq.(6) still holds for $j = 1$ by setting

$$z_{0i}(n) = x_i(n). \tag{8}$$

Based on eqs.(5) and (6), each submodule can be significantly simplified. Figure 8 shows the modified architecture of submodule $ji, j = 1, ..., l$, using the computational results $z_{(j-1)i}(n)$ of submodule $(j-1)i$ (with the initial $z_{0i}(n) = x_i(n)$) for reducing the area cost. Be comparing Figures 7 and 8, we see that the number of adders and multipliers are effectively reduced.

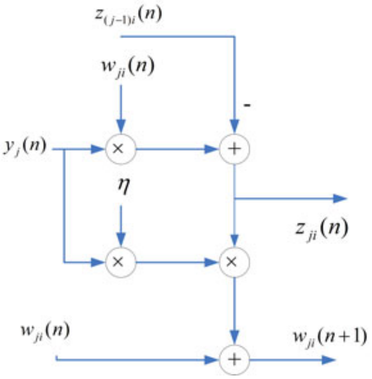


Fig. 8. The architecture of each modified submodule ji

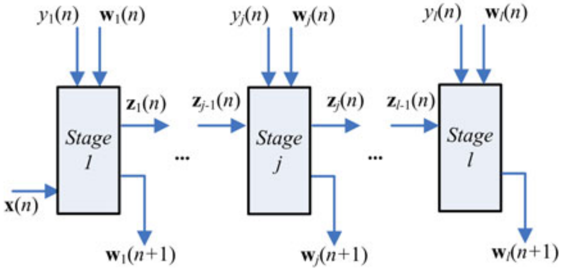


Fig. 9. Architecture II for implementing synaptic weight updating unit

Figure 9 shows the architecture of the new synaptic weight updating unit using the modified submodule depicted in Figure 8. The architecture, termed Architecture II, can be separated into l stages. Each stage j contains the modified submodules $ji, i = 1, ..., m$, shown in Figure 8 for computing $\mathbf{w}_j(n+1)$. Figure 10 depicts the architecture of each stage. In addition to $\mathbf{w}_j(n+1)$, it can be observed from Figure 10 that each stage j produces $\mathbf{z}_j(n) = [z_{j1}(n), ..., z_{jm}(n)]^T$. Consequently, the results of precedent stages will be used for the computation of subsequent stages for expediting training speed and lowering the area cost. Buffers may be added to the output of each stage for lowering the datapath length, thereby enhancing the clock rate.

The proposed architecture is used as a custom user logic in a SOPC system consisting of softcore NIOS CPU [6], DMA controller and SDRAM, as depicted in Figure 11. The set of training vectors is stored in the SDRAM. The training vectors are then delivered to the proposed circuit by the DMA controller. The softcore NIOS CPU runs on a simple software for coordinates different components in the SOPC. The proposed circuit operates as a hardware accelerator for GHA training. The resulting SOPC system is able to perform efficient on-chip training for GHA-based applications.

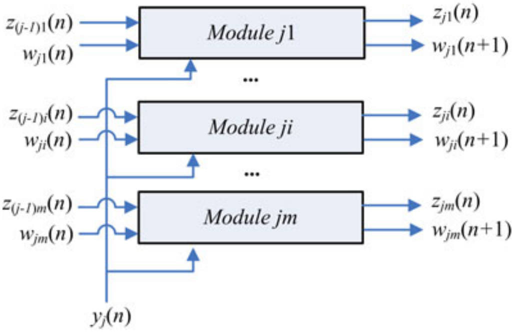


Fig. 10. Architecture of Stage j of Architecture II

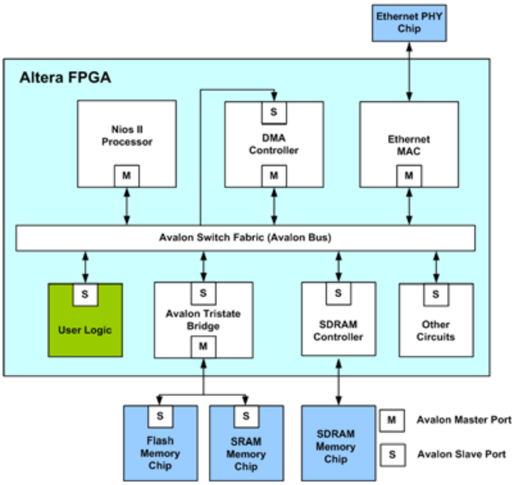


Fig. 11. The SOPC system for implementing GHA

4 Experimental Results

This section presents some experimental results of the proposed architectures. The GHA is used for texture classification in the experiments. Figure 12 shows the 5 textures considered in this paper. The dimension of training vectors is 4×4 (i.e., $m = 16$). There are 32000 training vectors in the training set. Due to limitation of data bus, b elements of each training vector will be conveyed into proposed circuit at a time (i.e., $b = 4$). The target FPGA device for all the experiments in this section is Altera Cyclone III [7].

To demonstrate the effectiveness of the proposed architecture, the principal component based k nearest neighbor (PC-kNN) rule is adopted. Two steps are involved in the PC-kNN rule. In the first step, the PCA is applied to the input vectors to transform the high dimensional data into l principal components. The synaptic weight vectors after the convergence of GHA training are used to form

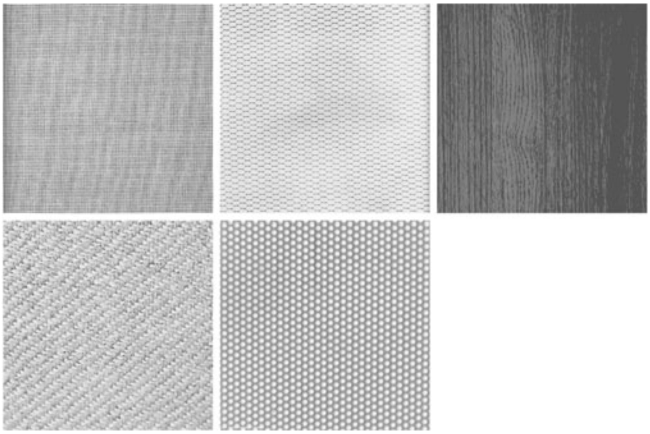


Fig. 12. Textures considered in the experiments

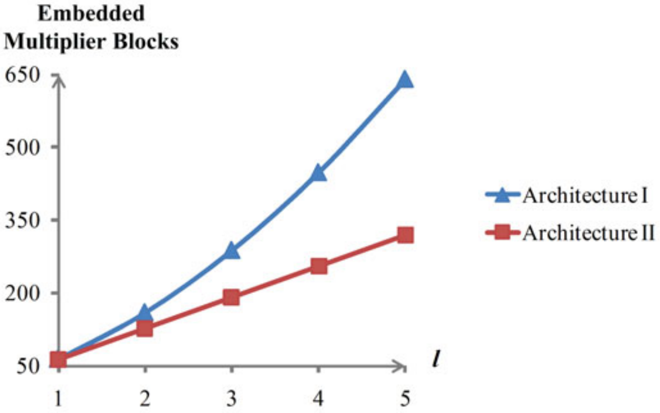


Fig. 13. The consumption of Embedded Multiplier Blocks by Archtectures I and II

the matrix for the linear PCA transform. In the second step, the kNN method is applied to the principal subspace for texture classification. When $l = 4$, the PC-kNN rule is able to achieve 81.7% classification success rate.

Figure 13 compares the area cost of Architecture I and Architecture II for various number of principal components l. In both the architectures, the embedded multiplier blocks are used for realizing the adders and multipliers for updating synaptic weight vectors. Therefore, the area cost considered in Figure 13 is the number of embedded multiplier blocks. It can be observed from Figure 13 that as l becomes large, the embedded multiplier block consumption of Algorithm I is significantly larger than that of Algorithm II. The Architecture II consumes lower hardware resources because the computation at subsequent stages are able to use the results produced by their precedent stages. By contrast, all the

Table 1. Hardware resource consumption of the SOPC system using the proposed GHA architecture as custom user logic

Hardware resource	Consumption
LEs of GHA Circuit	30,898
Embedded Memory Bits of GHA Circuit	0
Embedded Multiplier blocks of GHA Circuit	272
LEs of Entire SOPC System	41,255
Embedded Memory Bits of Entire SOPC System	2,673,968
Embedded Multiplier blocks of Entire SOPC System	276

submodules in Architecture I operate independently, resulting in large hardware overhead for updating synaptic weight vectors.

Table 1 shows the hardware resource consumption of the NIOS-based SOPC system with the proposed GHA architecture embedded as the custom user logic. The number of principal components is $l = 4$. The proposed GHA circuit (containing the principal components computation unit, the synaptic weight updating unit, and the memory unit) consumes 30,898 logic elements (LEs), and 272 embedded multiplier blocks of the FPGA device. The synaptic weight updating unit is implemented by Architecture II. The NIOS softcore CPU [6] itself also consumes hardware resources. The whole SOPC system uses 41,255 LEs. The hardware overhead imposed by the softcore processor is only 10,357 LEs. The operating speed of the system is 85 MHz.

The CPU time of the NIOS-based SOPC system for various numbers of training epoches are shown in Figure 14. For comparison purpose, the CPU time of its software counterpart is also included in the figure. The software training is

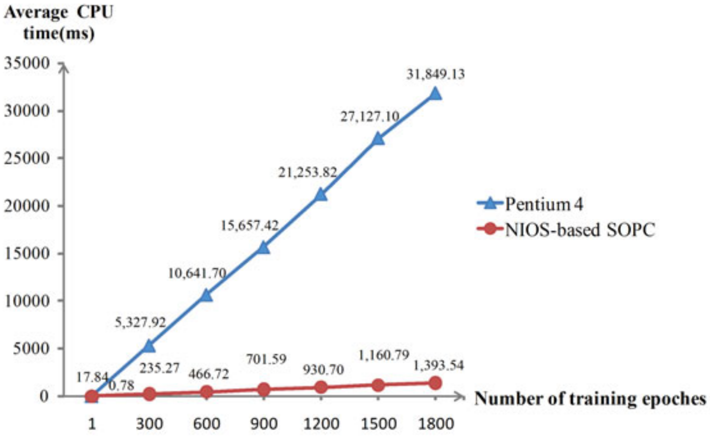


Fig. 14. The CPU time of the proposed hardware architecture and its software counterpart for different number of iterations

based on the general purpose 3.0-GHz Pentium IV CPU. It can be observed from Figure 14 that the SOPC system has significantly lower execution time. In addition, SOPC achieves larger reduction in CPU time as the number of epoches increases. In particular, when the number of training epoches reaches 1800, the CPU time of the proposed SOPC system is 1393.54 ms. By contrast, the CPU time of Pentium IV is 31849.13 ms. The speedup of the proposed architecture over its software counterpart therefore is 22.85. All these facts demonstrate the effectiveness of the proposed architecture.

5 Concluding Remarks

From the experimental results, it can be observed that Architecture II is effective for the hardware implementation of synaptic weight updating unit. It consumes significantly lower area cost as compared with Architecture I. In addition, the CPU time of the SOPC system based on Architecture II is lower than that of the software counterpart running on the general purpose processor. The proposed architecture therefore is an effective alternative for on-chip learning applications requiring both low area cost and high speed computation.

References

1. Gunter, S., Schraudolph, N.N., Vishwanathan, S.V.N.: Fast Iterative Kernel Principal Component Analysis. Journal of Machine Learning Research, 1893–1918 (2007)
2. Haykin, S.: Neural Networks and Learning Machines, 3rd edn. Pearson, London (2009)
3. Jolliffe, I.T.: Principal component Analysis, 2nd edn. Springer, Heidelberg (2002)
4. Karhunen, J., Joutsensalo, J.: Generalization of Principal Component Analysis, Optimization Problems, and Neural Networks. Neural Networks, 549–562 (1995)
5. Kim, K., Franz, M.O., Scholkopf, B.: Iterative kernel principal component analysis for image modeling. IEEE Trans. Pattern Analysis and Machine Intelligence, 1351–1366 (2005)
6. NIOS II Processor Reference Handbook, Altera Corporation 2007 (2007), http://www.altera.com/literature/lit-nio2.jsp
7. Cyclone III Device Handbook, Altera Corporation 2008 (2008), http://www.altera.com/products/devices/cyclone3/cy3-index.jsp

The Data-Based Mathematical Modeling and Parameter Identification in JAK-STAT Signaling Pathway by Using a Hybrid Evolutionary Algorithm

Wei Zhang and Xiufen Zou

School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China
zhang20041252@126.com

Abstract. Based on the published quantitative measurements of EPO receptor and STAT5 phosphorylation in JAK-STAT signaling pathway [1], we build an improved mathematical model and use a hybrid evolutionary algorithm to optimize all rate constants. Numerical results demonstrate that our model is well fit to the measured data and can use to predict the dynamical behavior of the JAK-STAT signaling pathway.

Keywords: Mathematical model; JAK-STAT signaling pathway; Parameter identification; Evolutionary algorithm.

1 Introduction

The Janus kinase/signal transducers and activators of transcription (JAK/STAT) pathway is one of the essential signaling pathways for a wide range of cytokines and growth factors in mammals [2-6], therefore, considerable efforts have been made in the modeling and analysis of the dynamical behavior of the JAK-STAT signaling pathway [7-11]. Because signal transduction through the erythropoietin receptor (EpoR) is important for the proliferation and differentiation of erythroid progenitor cells, several subsequent models and dynamical analysis were reported since a time-delayed mathematical model of the core module of the JAK-STAT signaling pathway based on time-resolved measurements of EPOR and STAT5 phosphorylation was proposed [1,12-16]. In this study, we construct an improved model by adding two reaction processes in nucleus and taking into account the different volume ratio of the nucleus and cytoplasm.

The paper is organized as follows. Section 2 describe the main reaction processes and presents the mathematical model. In Section 3, we propose a hybrid evolutionary algorithm to identify the optimal parameters in the model and make a sensitivity analysis for all rate constants. Finally, some conclusions are addressed in Section 4.

2 The Improved Model of the JAK-STAT Pathway

Here, the published data of Epo-induced JAK-STAT5 pathway is used [1]. When the Hormone (Epo) binding to the EpoR results in activation by phosphorylation of JAK2

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 516–522, 2010.

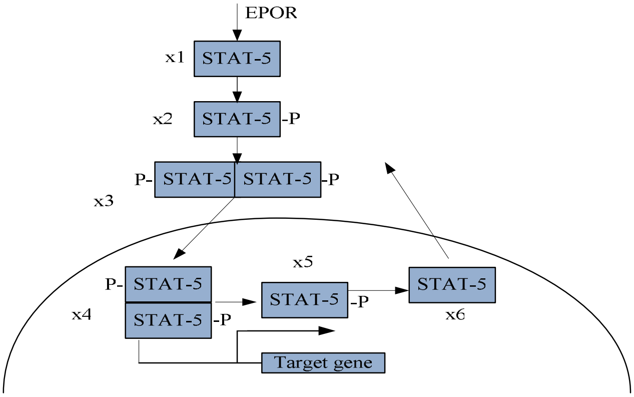


Fig. 1. The EPO-induced JAK-STAT pathway

and subsequently in tyrosine phosphorylation of JAK2 and the EpoR cytoplasmic domain (Fig.1). Phosphotyrosine residues in the EpoR mediate recruitment of monomeric STAT5 (x_1). Upon receptor recruitment, monomeric STAT5 is tyrosine phosphorylated (x_2), dimerizes (x_3), and migrates to the nucleus (x_4), where it binds to the promoter region of the DNA and initiate gene transcription, then the activated STAT5 dimer disintegrate to the monomeric form (x_5), and the activated monomeric STAT5 dephosphorylated into unphosphorylated form (x_6) and exported to the cytoplasm. Because the STAT5 dimer can disintegrate to the monomeric form for its instability in cytoplasm, we add a reverse reaction in the pathway.

Comparing with the time-delayed model presented by Swameye et al.[1], the proposed model is different in the following two aspects.

(1) We include the phosphorylation of monomer STAT-5 and inactive monomer STAT-5 in nuclear instead of the delayed term in literature [1].

(2) We consider the different volume ratio of the nucleus and cytoplasm in the BaF3 cell. A scaling factor kv = V/U (ratio of cytoplasmic and nuclear volumes) has been introduced, which is similar to Lipniacki et al. [1, 17]. The value of kv is approximately set to 2 according to the experiments in literature [18].

The improved mathematical model of the JAK-STAT pathway can be described as follows.

$$\begin{aligned} &dx_1/dt=-k_1x_1EPOR+k_6x_6/kv \\ &dx_2/dt=k_1x_1EPOR-2k_2x_2+2k_{22}x_3 \\ &dx_3/dt= k_2x_2- k_{22}x_3-k_3x_3 \\ &dx_4/dt= kvk_3x_3-k_4x_4 \\ &dx_5/dt= 2k_4x_4-k_5x_5 \\ &dx_6/dt= k_5x_5-k_6x_6 \end{aligned} \tag{1}$$

Where k_1, k_2, k_3, k_4, k_5, k_6, k_{22} are constants with dimension [min-1]. According to [1,14], the initial values of x_2, x_3, x_4, x_5, x_6 are zero, and the initial value of x_1 is a free parameter which is estimated from experiment data. The time course of activation of the EpoR is displayed in Fig.2 with arbitrary unit.

The observed data of experiments in cytoplasm is the amount of tyrosine phosphorylated STAT5 (y_1=s_1(x_2+2x_3)) and the total amount of STAT5 (y_2=s_2(x_1+x_2+2x_3)). As input function that determines the response of the STAT5, EPO induced tyrosine phosphorylation of EpoR measured by y_3 (y_3=s_3EpoR$_A$). The s_1, s_2 and s_3 are the scaling parameters that denote the relative units of protein measured by experiments. These parameters and rate constants are needed to be determined to fit the experimental data.

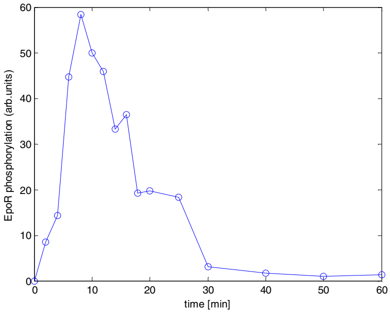


Fig. 2. The measured time series of the amount of EPOR (redrawn from [1])

3 A Hybrid Evolutionary Algorithm for Parameter Estimation

The parameter estimation of the JAK-STAT pathway system can be converted into the minimization of a weighted distance measure J between experimental and predicted values.

$$J(x(t=0), p)=\Sigma^{N}_{i=1}\Sigma^{2}_{j=1}(y_j^{D}(t_i)- y_j^{M}(t_i ;x_1(t=0), p))^{2}/w_{ij.} \tag{2}$$

Where $y_j^D(t_i)$ represents the measured data for component j at time t_i , and $y_j^M(t_i ;x_1(t=0), p)$ is the vector of states that corresponds to the predicted data using the model with starting value $x_1(t=0)$ and the parameter p. w_{ij} corresponds to the different weights taken to normalize the contributions of each term, in this paper we set $w_{ij}=1/\max[y_j^D(t_i)]_{j.}$

We present a hybrid algorithm which combine Multi-parent crossover ([19]) and differential evolution (DE) [20] to solve the optimization problem (2).

```
Procedure Multi-parent-DE algorithm
begin
    parameterize (pop_size,p,cr,M)
    initialize population
    evaluate population
    iteration=0
  while(Iteration<maxIteration)
        sp(t)=select for the parents for recombination
        sp'(t)=recombination(sp(t))
        evaluate (sp'(t)),compare with the best one, if better than the best replace it, else go to   next step
       execute DE operater get the evolved populations P'(t)
       evaluate P'(t) and select the best individual
       iteration++
 end
```

The range of parameters is set to [0.0001, 10]. The population size is set to 50, multi-parent recombination number M=8, parameters in DE algorithm are set to R=0.5 and CR=0.55. The algorithm is run 10 times and the number of the iteration is 400, the optimal value of weighted distance J is 0.0896. The obtained parameters are listed at Table 1 and the convergence is depicted in Fig.3.

Table 1. The obtained parameters by using the hybrid algorithm in model (1)

k_1	k_2	k_3	k_4	k_5	k_6	k_{22}	$x_1(0)$
0.0409	7.1992	0.1729	0.3735	4.6778	0.9417	1.5933	6.2840

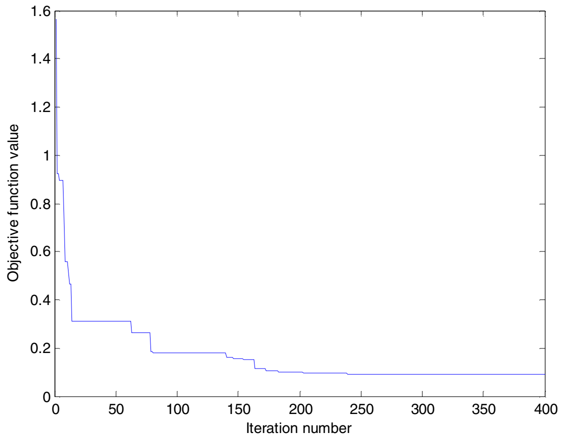


Fig. 3. Convergence curves (objective function verus iteration number)

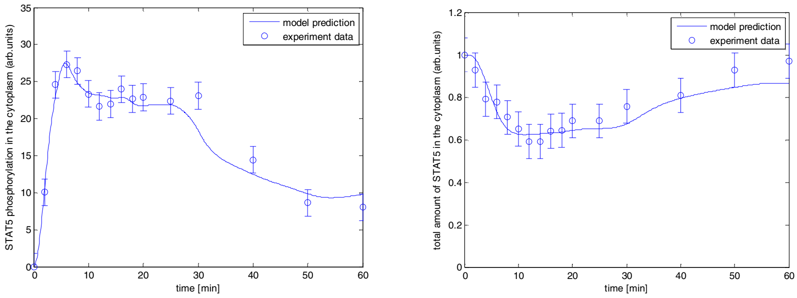


Fig. 4. Comparisons that simulation results with the measured time series of phosphorylated and total STAT5 in cytoplasm (the experimental data is redrawn from [1])

The comparisons that simulation results using our model and the measured time series of phosphorylated and total STAT5 in cytoplasm are presented in Fig.4. The results indicate that our model and algorithm are efficient.

To investigate the effects of parameter changes on the amount of STAT5 populations, we further make the sensitivity analysis of parameters by using an approach proposed by Hasan, K.Khalil et al.[21]. We set the range of the parameter distributions is the random number between [0, 1]. The effects of parameter changes on the amount of unphosphorylated STAT5 in cytoplasm and activated STAT5 in nuclear involved in cycling are showed in Fig. 5. We can see that the perturbations of parameters k1, k3, k4 are most sensitive to the amount of activated STAT5 in nuclear, which are consistent with the results of Swameye et al.[1].

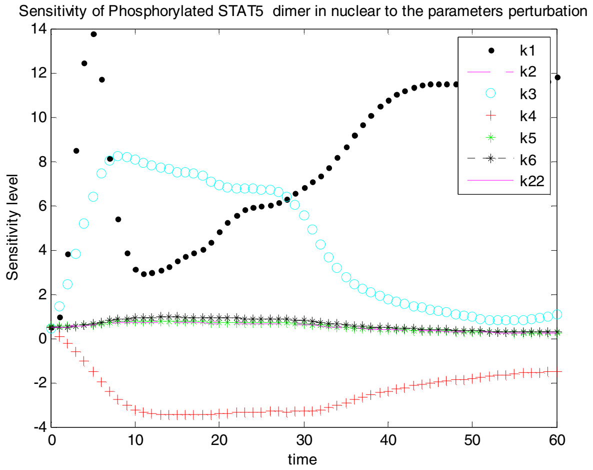


Fig. 5. Sensitivity of the phosphorylated STAT5s in nucleus to the perturbation of parameters

4 Conclusions

Mathematical modeling and parameter estimation are important aspects for analyzing quantitatively dynamical behavior of signaling networks. In this paper, an improved mathematical model is proposed and a hybrid evolutionary algorithm is used to fit the published experimentally measured time series of EpoR and STAT5 phosphorylation in JAK-STAT signaling pathway. Numerical experiments and sensitivity analysis show that our model and obtained rate constants are reasonable. In our future work, we would use the proposed model and approaches to analysis the dynamical behavior of signaling networks and predict the important feature of interactional mechanism between proteins.

Acknowledgments

This work was supported by the new cross-disciplinary project of Wuhan University under Grant 1081001.

References

1. Swameye, I., Muller, T., Timmer, J., Sandra, O., Klingmuller, U.: Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by databased modeling. Proc. Natl. Acad. Sci. U.S.A. 100, 1028–1033 (2003)
2. Kitano, H.: Systems biology: a brief overview. Science 295, 1662–1664 (2002)
3. Kitano, H.: Computational systems biology. Nature 420, 206–210 (2002)
4. Horvath, C.M.: Trends Biochem. Science 25, 496–502 (2000)
5. Darnell Jr., J.: STATs and gene regulation. Science 277, 1630–1635 (1997)
6. Endy, D., Brent, R.: Modeling cellular behavior. Nature 409, 391–395 (2001)
7. Agaisse, H., Perrimon, N.: The roles of JAK/STAT signaling in Drosophila immune response. Immunol. Rev. 198, 72–82 (2004)
8. Zi, Z., Cho, K.H., Sung, M.H., Xia, X., Zheng, J., Sun, Z.: In silico identification of the key components and steps in IFN-gamma induced JAK-STAT signaling pathway. FEBS Lett. 579, 1101–1108 (2005)
9. Haspel, R., Salditt-Georgie, M., Darnell Jr., J.: The rapid inactivation of nuclear tyrosine phosphorylated STAT-1 depends upon a protein tyrosine phosphatase. EMBO J. 156, 6262–6268 (1996)
10. Haspel, R., Darnell Jr., J.E.: A nuclear protein phosphatase is required for the inactivation of STAT-1. Proc. Nat. Acad. Sci. 96, 10188–10193 (1999)
11. Yamada, S., Shiono, S., Joo, A., Yoshimura, A.: Control mechanism of JAK/STAT signal transduction pathway. FEBS Lett. 534, 190–196 (2003)
12. Tao, W.Y., Wen, F., Zhang, H., Liu, G.H.: The signal transduction mediated by erythropoietin and proinflammatory cytokines in the JAK/STAT pathway in the children with cerebral palsy. Brain development 31, 200–207 (2009)
13. Levy, D.E., Darnell Jr., J.E.: Stats: transcriptional control and biological impact. Nat. Rev. Mol. Cell Biol. 3, 651–662 (2002)

14. Timmer, J., Muller, T., Swameye, I., Sandra, O., Klingmuller, U.: Modeling the nonlinear dynamics of cellular signal transduction. Int. J. Bifurcation and Chaos 14, 2069–2079 (2004)
15. Zeng, R., Aoki, Y., Yoshida, M., Arai, K., Watanabe, S.: Stat5b shuttles between cytoplasm and nucleus in cytokine-dependent and independent manner. J. Immunol. 168, 4567–4575 (2002)
16. Lai, S.Y., Childs, E.E., Xi, S., Coppelli, F.M., Gooding, W.E., Wells, A., Ferris, R.L., Grandis, J.R.: Erythropoietin-mediated activation of JAK-STAT signaling contributes to cellular invasion in head and neck squamous cell carcinoma. Oncogene. 24, 4442–4449 (2005)
17. Lipniacki, T., Paszek, P., Brasier, A.R., Luxon, B., Kimmel, M.: Mathematical model of NF–kB regulatory module. J. Theor. Biol. 228, 195–215 (2004)
18. Mayerhofer, M., Gleixner, K.V., Hoelbl, A., Florian, S., Hoermann, G., Aichberger, K.J., Bilban, M., Esterbauer, H., Krauth, M.T., Sperr, W.R., Longley, J.B., Kralovics, R., Moriggl, R., Zappulla, J., Liblau, R.S., Schwarzinger, I., Sexl, V., Sillaber, C., Valent, P.: Unique effects of KIT D816V in BaF3 cells: induction of cluster formation, histamine synthesis, and early mast cell differentiation antigens. Journal of Immunology 180, 5466–5476 (2008)
19. Guo, T., Kang, L.S.: A new evolutionary algorithm for function optimization. Wuhan University Journal of Natural Sciences 4, 409–414 (1999)
20. Storn, R.: A Simple and Efficient Heuristic Strategy for Global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997)
21. Hasan, K.K.: Nonlinear Systems, 3rd edn., pp. 68–71. Pearson Education, Prentice Hall (2002)

The Research on the Intelligent Interpreter for ISO 14649 Programs

Yu Zhang, Yongxian Liu, and Xiaolan Bai

School of Mechanical Engineering & Automation, Northeastern University,
Shenyang Liaoning 110004, China
zy4097534@126.com

Abstract. ISO 14649 is a new data model for CNC (Computerized Numerical Control). It provides rich information for CNC machine tools and tells CNC “what to do”, based on features rather than “how to do” as described by the G&M codes. “Translating” ISO 14649 programs are the first and key step in implementing this new data model. Although some researches about interpretation of the STEP-NC program have been conducted, most of them do not care whether the machining information existing in ISO 14649 programs is adequate for the machining process during the interpretation. Therefore, in this paper, the ISO 14649 interpreter based on the neural network and expert system is proposed for the combination of the expert knowledge. The numeric reasoning capacity of the neural network can be used to analyze the machining information very well. This has been proved effective by the case study.

Keywords: ISO 14649, Expert system, Neural network, STEP-NC.

1 Introduction

Up to now, NC (Numeraical Control) programs driving CNC machines are programmed in ISO 6983, also called the G&M codes. G&M codes are typically generated by CAD/CAM (Computer-Aided Design/Computer-Aided Manufacturing) systems. However, ISO 6983 limits its development because of its own problems [1]. ISO 14649 is a new model of data transfer between CAD/CAM systems and CNC machines. It remedies the shortcomings of ISO 6983 by specifying machining processes rather than machine tool motion, using the object-oriented concept of Workingsteps. Workingsteps correspond to high-level machining features and associated process parameters. CNC is responsible for translating Workingsteps into axis motion and tool operation [2]. Because ISO 14649 programs can’t be used directly by machine tools, the interpretation of ISO 14649 programs is the first and key step. Although many scholars and institutions in some countries [3-10] have done some research about the interpreter, their work has mainly been focusing on extracting the information from the files and didn’t mention the completeness and accuracy of the machining information in ISO 14649 programs. Therefore, this paper aims to apply the neural network and expert system [11-13] in order to acquire the complete, accurate and credible machining information which is extracted from ISO 14649 in ISO 10303-21.

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 523–534, 2010.

The remainder of this paper is described as follows. The framework of the interpreter is proposed in Section 2. Second 3 explains three key technologies. In Section 4, a case study is provided. Conclusions are given at the end.

2 Overview of ISO 14649

At present, STEP-NC is being developed to provide a data model for a new breed of intelligent CNC controllers. There are two different ISO subcommittees working on STEP-NC standard with two different focuses. ISO TC 184/SC1 works on ISO 14649 (also called ARM model), whereas ISO TC 184/SC4 works on STEP AP-238 (also called AIM model). The main difference between these two models is the degree to which they use the STEP representation methods and technical architecture [14].

A part program from ISO 14649 or STEP AP-238 can be described in a Physical File Format, i.e. ISO 10303-21. As shown in Fig. 1 [2] and Fig.2, the first section of the part program in ISO 10303-21 is the header section marked by the keyword "HEADER". In this header section, some general information and comments concerning the part program are given, such as filename, author, date, organization, etc. The second and main section of the program file is the data section marked by the keyword "DATA". This section contains all information about geometry, features and manufacturing tasks. This section also includes a PROJECT entity that serves as an explicit reference for the starting point of the manufacturing tasks.

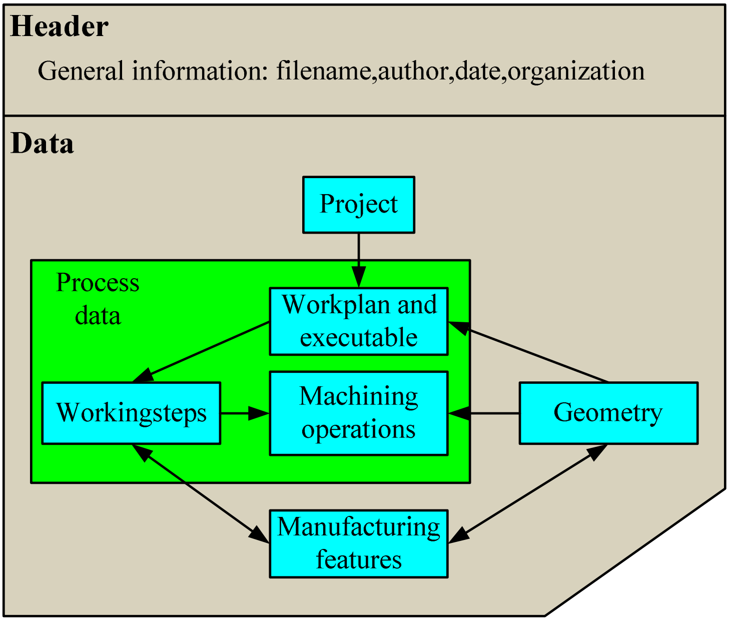


Fig. 1. Data structure of ISO 14649

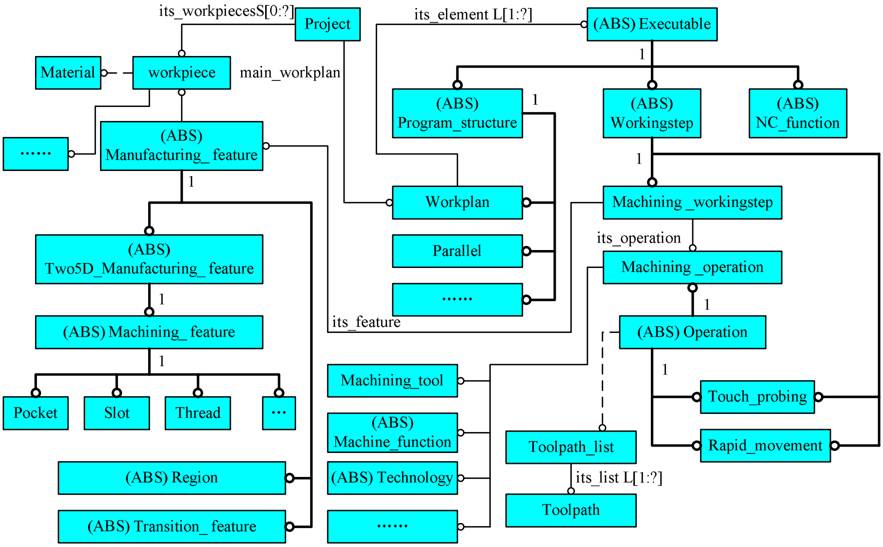


Fig. 2. Simple data model of ISO 14649

3 System Framework

As shown in Fig.3, the construction of this system is based on the rule tree-based expert system and BP (Back Propagation) neural network. The integration of the BP neural network and expert system can not only make the best of expert knowledge but also exert the inference mechanism and self-study capacity of neural network, so that symbol inference and numerical inference are parallelly used to analyze the machining information extracted from ISO 14649 programs. Therefore, the machining data can be optimized.

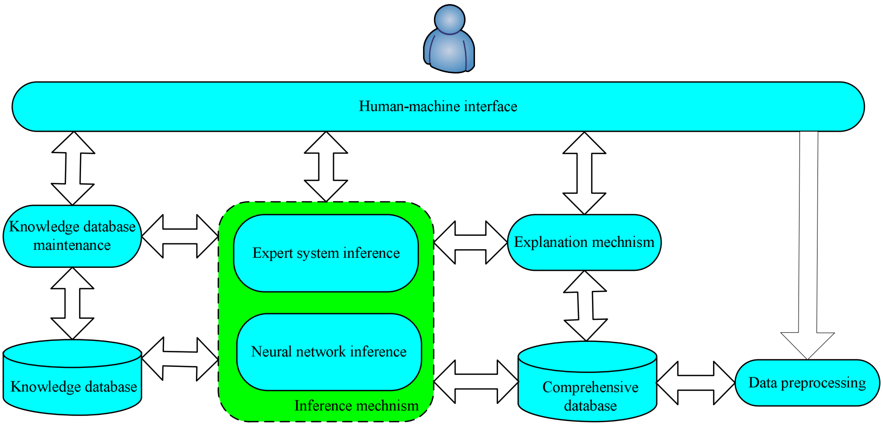


Fig. 3. System framework

3.1 Human-Machine Interface Module and Data Preprocessing Module

The human-machine interface module provides a friendly human-machine interface so that users can conveniently communicate with computers, maintain a knowledge database and see results.

As Fig.4 shows, an ISO 14649 program includes all the machining information for machining the part. As shown in Fig.5, every line in the program consists of row numbers, entity names and attribute lists, and their relationships is logical. Therefore, along with the analysis of accidence and syntax, the data preprocessor module can get the machining information according to logical relationships. And the obtained information is written into a comprehensive database in order to be used later.

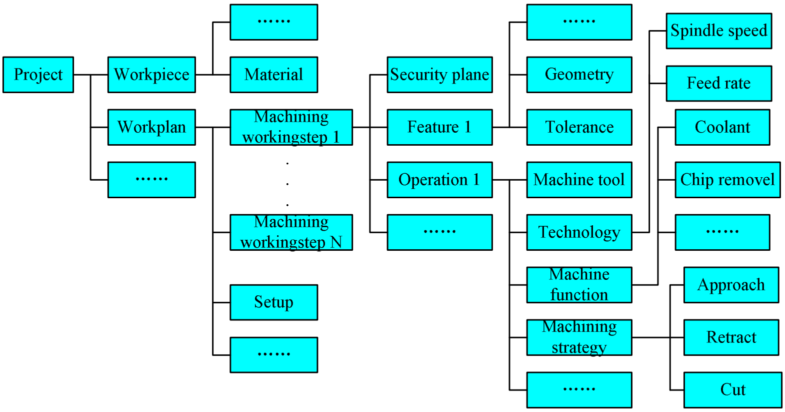


Fig. 4. Information structure in DATA section

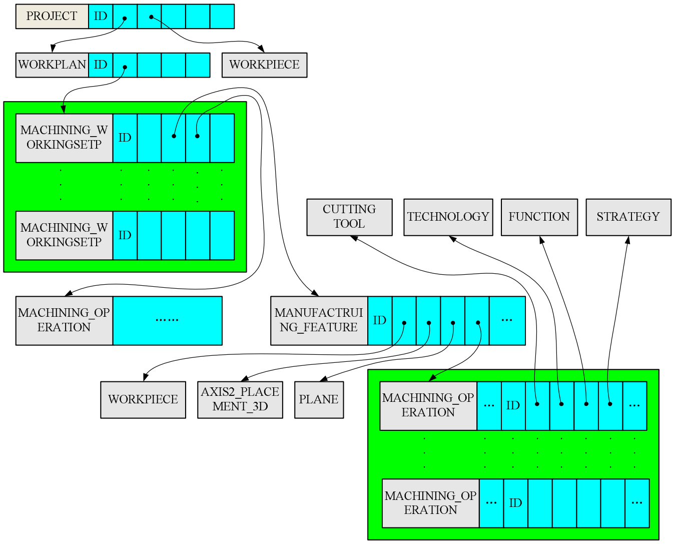


Fig. 5. Logical relationship among entities

3.2 Expert System Inference Module

The expert system is used as one part of the inference mechanism, which can check the data from the data processor. First, based on experts' experience and knowledge, a knowledge database is built in terms of the rule-tree format, i.e. root node stands for the result; leaf nodes stand for initial input and interim nodes stand for analysis data. Then, every rule in the knowledge database is assigned to a number which stands for a condition comprising the reason and results. Therefore, according to the rules in the knowledge database, symbol data in the comprehensive database can be handled by this module and corresponding results also can be given back to the users.

3.3 BP Neural Network Inference Module

The three-layered BP neural network showed in Fig.6 is applied to the interpreter as an effective complement of the expert system inference mechanism. The inference process is outlined as follows. First of all, network structure knowledge and reasoning knowledge as well as weight threshold value are called and initial data is converted and input into the neural network module. Then, after forward calculation is used, the output can be obtained. Finally, through educement rules, the output will be expressed by the logical expression method to be easily understood by the users. Therefore, the inference mechanism of the BP neural network can remedy the shortcoming of expert system and make the analysis of machining data better.

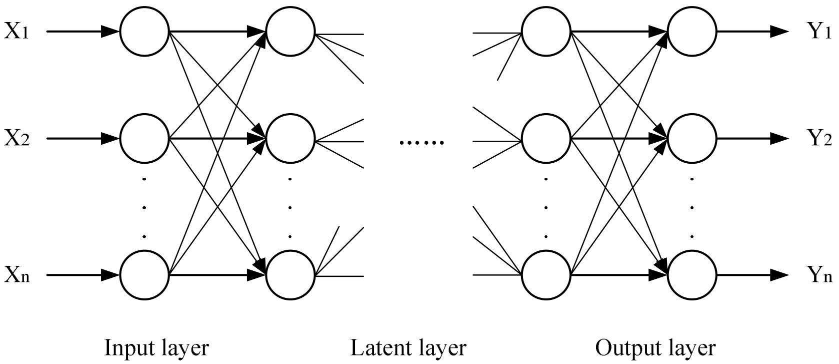


Fig. 6. BP neural network

3.4 Explanation Mechanism Module

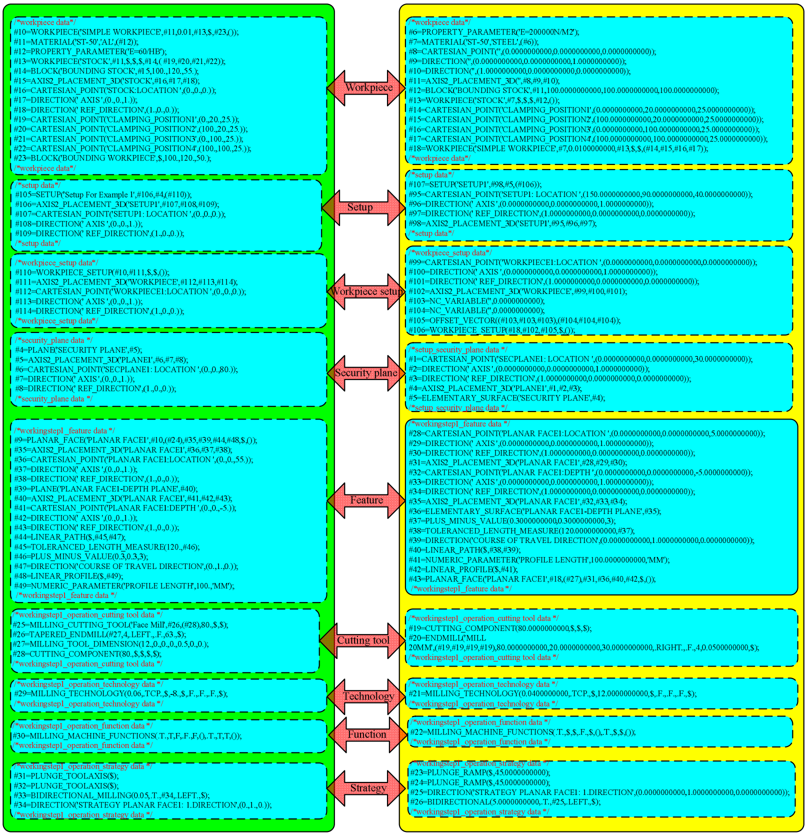
For the expert system, the explanation mechanism can directly get the answers because every rule in the knowledge database is assigned to an exclusive number. For the neural network, there are two methods. One is the case-based explanation method. That is, a case explanation is chosen as a reference for the explanation of the current case and the chosen case should be similar according to certain logic. Then, a fuzzy rule is applied to obtain the explanation. The other one is the network partition technology. That is, to solve bigger problems, the system adopts multiple BP networks and every BP network corresponds to a typical kind of problem. Then, every BP network is regarded as a big rule whose premise corresponds to the input of BP network and whose conclusion corresponds to the output of BP network. Finally, users can know the relevant information. Therefore, the explanation module can answer users' questions and give the results.

3.5 Knowledge Database Maintenance, Knowledge Database and Comprehensive Database

Engineers save experts' experience and knowledge in rule format into a knowledge database. In addition, if the inference meets uncertain factors, engineers will be questioned and the new knowledge will be added to the knowledge database by engineers. Therefore, the knowledge database can be updated and maintained. And the rules of the expert system and the structure of the neural network also are saved into the knowledge database so that they can be used during the inference. The comprehensive database contains original data and middle data to be used later.

4 Case Study

Example 1 in ISO 14649-11 standard [15] is chosen as the research object, whose program is shown in the appendix. Through the analysis of this system, the program is



```
/*workpiece data*/
#10=WORKPIECE('SIMPLE WORKPIECE',#11,0.01,#13,$,#23,());
#11=MATERIAL('ST-50','AL',(#12));
#12=PROPERTY_PARAMETER('E=60/HB');
#13=WORKPIECE('STOCK',#11,$,$,$,#14,( #19,#20,#21,#22));
#14=BLOCK('BOUNDING STOCK',#15,100.,120.,55.);
#15=AXIS2_PLACEMENT_3D('STOCK',#16,#17,#18);
#16=CARTESIAN_POINT('STOCK:LOCATION ',(0.,0.,0.));
#17=DIRECTION(' AXIS ',(0.,0.,1.));
#18=DIRECTION(' REF_DIRECTION',(1.,0.,0.));
#19=CARTESIAN_POINT('CLAMPING_POSITION1',(0.,20.,25.));
#20=CARTESIAN_POINT('CLAMPING_POSITION2',(100.,20.,25.));
#21=CARTESIAN_POINT('CLAMPING_POSITION3',(0.,100.,25.));
#22=CARTESIAN_POINT('CLAMPING_POSITION4',(100.,100.,25.));
#23=BLOCK('BOUNDING WORKPIECE',$,100.,120.,50.);
/*workpiece data*/

/*setup data*/
#105=SETUP('Setup For Example 1',#106,#4,(#110));
#106=AXIS2_PLACEMENT_3D('SETUP1',#107,#108,#109);
#107=CARTESIAN_POINT('SETUP1: LOCATION ',(0.,0.,0.));
#108=DIRECTION(' AXIS ',(0.,0.,1.));
#109=DIRECTION(' REF_DIRECTION',(1.,0.,0.));
/*setup data*/

/*workpiece_setup data*/
#110=WORKPIECE_SETUP(#10,#111,$,$,());
#111=AXIS2_PLACEMENT_3D('WORKPIECE',#112,#113,#114);
#112=CARTESIAN_POINT('WORKPIECE1:LOCATION ',(0.,0.,0.));
#113=DIRECTION(' AXIS ',(0.,0.,1.));
#114=DIRECTION(' REF_DIRECTION',(1.,0.,0.));
/*workpiece_setup data*/

/*security_plane data */
#4=PLANE('SECURITY PLANE',#5);
#5=AXIS2_PLACEMENT_3D('PLANE1',#6,#7,#8);
#6=CARTESIAN_POINT('SECPLANE1: LOCATION ',(0.,0.,80.));
#7=DIRECTION(' AXIS ',(0.,0.,1.));
#8=DIRECTION(' REF_DIRECTION',(1.,0.,0.));
/*security_plane data */

/*workingstep1 _feature data */
#9=PLANAR_FACE('PLANAR FACE1',#10,(#24),#35,#39,#44,#48,$,());
#35=AXIS2_PLACEMENT_3D('PLANAR FACE1',#36,#37,#38);
#36=CARTESIAN_POINT('PLANAR FACE1:LOCATION ',(0.,0.,55.));
#37=DIRECTION(' AXIS ',(0.,0.,1.));
#38=DIRECTION(' REF_DIRECTION',(1.,0.,0.));
#39=PLANE('PLANAR FACE1-DEPTH PLANE',#40);
#40=AXIS2_PLACEMENT_3D('PLANAR FACE1',#41,#42,#43);
#41=CARTESIAN_POINT('PLANAR FACE1:DEPTH ',(0.,0.,-5.));
#42=DIRECTION(' AXIS ',(0.,0.,1.));
#43=DIRECTION(' REF_DIRECTION',(1.,0.,0.));
#44=LINEAR_PATH($,#45,#47);
#45=TOLERANCED_LENGTH_MEASURE(120.,#46);
#46=PLUS_MINUS_VALUE(0.3,0.3,3);
#47=DIRECTION('COURSE OF TRAVEL DIRECTION',(0.,1.,0.));
#48=LINEAR_PROFILE($,#49);
#49=NUMERIC_PARAMETER('PROFILE LENGTH',100.,'MM');
 /*workingstep1_feature data */

/*workingstep1_operation_cutting tool data */
#25=MILLING_CUTTING_TOOL('Face Mill',#26,(#28),80.,$,$);
#26=TAPERED_ENDMILL(#27,4,.LEFT.,.F.,63.,$);
#27=MILLING_TOOL_DIMENSION(12.,0.,0.,0.,0.5,0.,0.);
#28=CUTTING_COMPONENT(80.,$,$,$,$);
/*workingstep1_operation_cutting tool data */

/*workingstep1_operation_technology data */
#29=MILLING_TECHNOLOGY(0.06,.TCP.,$,-8.,$,.F.,.F.,.F.,$);
/*workingstep1_operation_technology data */

/*workingstep1_operation_function data */
#30=MILLING_MACHINE_FUNCTIONS(.T.,T,F,.F.,F,(),.T.,T,T,());
/*workingstep1_operation_function data */

/*workingstep1_operation_strategy data */
#31=PLUNGE_TOOLAXIS($);
#32=PLUNGE_TOOLAXIS($);
#33=BIDIRECTIONAL_MILLING(0.05,.T.,#34,.LEFT.,$);
#34=DIRECTION('STRATEGY PLANAR FACE1: 1.DIRECTION',(0.,1.,0.));
/*workingstep1_operation_strategy data */
```

```
/*workpiece data*/
#6=PROPERTY_PARAMETER('E=200000N/M2');
#7=MATERIAL('ST-50','STEEL',(#6));
#8=CARTESIAN_POINT('',(0.0000000000,0.0000000000,0.0000000000));
#9=DIRECTION('',(0.0000000000,0.0000000000,1.0000000000));
#10=DIRECTION('',(1.0000000000,0.0000000000,0.0000000000));
#11=AXIS2_PLACEMENT_3D('',#8,#9,#10);
#12=BLOCK('BOUNDING STOCK',#11,100.0000000000,100.0000000000,100.0000000000);
#13=WORKPIECE('STOCK',#7,$,$,$,#12,());
#14=CARTESIAN_POINT('CLAMPING_POSITION1',(0.0000000000,20.0000000000,25.0000000000));
#15=CARTESIAN_POINT('CLAMPING_POSITION2',(100.0000000000,20.0000000000,25.0000000000));
#16=CARTESIAN_POINT('CLAMPING_POSITION3',(0.0000000000,100.0000000000,25.0000000000));
#17=CARTESIAN_POINT('CLAMPING_POSITION4',(100.0000000000,100.0000000000,25.0000000000));
#18=WORKPIECE('SIMPLE WORKPIECE',#7,0.0100000000,#13,$,$,(#14,#15,#16,#17));
/*workpiece data*/

/*setup data*/
#107=SETUP('SETUP1',#98,#5,(#106));
#95=CARTESIAN_POINT('SETUP1: LOCATION ',(150.0000000000,90.0000000000,40.0000000000));
#96=DIRECTION(' AXIS ',(0.0000000000,0.0000000000,1.0000000000));
#97=DIRECTION(' REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000));
#98=AXIS2_PLACEMENT_3D('SETUP1',#95,#96,#97);
/*setup data*/

/*workpiece_setup data*/
#99=CARTESIAN_POINT('WORKPIECE1:LOCATION ',(0.0000000000,0.0000000000,0.0000000000));
#100=DIRECTION(' AXIS ',(0.0000000000,0.0000000000,1.0000000000));
#101=DIRECTION(' REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000));
#102=AXIS2_PLACEMENT_3D('WORKPIECE',#99,#100,#101);
#103=NC_VARIABLE('',0.0000000000);
#104=NC_VARIABLE('',0.0000000000);
#105=OFFSET_VECTOR((#103,#103,#103),(#104,#104,#104));
#106=WORKPIECE_SETUP(#18,#102,#105,$,());

/*setup_security_plane data */
#1=CARTESIAN_POINT('SECPLANE1: LOCATION ',(0.0000000000,0.0000000000,30.0000000000));
#2=DIRECTION(' AXIS ',(0.0000000000,0.0000000000,1.0000000000));
#3=DIRECTION(' REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000));
#4=AXIS2_PLACEMENT_3D('PLANE1',#1,#2,#3);
#5=ELEMENTARY_SURFACE('SECURITY PLANE',#4);
/*setup_security_plane data */

/*workingstep1_feature data */
#28=CARTESIAN_POINT('PLANAR FACE1:LOCATION ',(0.0000000000,0.0000000000,5.0000000000));
#29=DIRECTION(' AXIS ',(0.0000000000,0.0000000000,1.0000000000));
#30=DIRECTION(' REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000));
#31=AXIS2_PLACEMENT_3D('PLANAR FACE1',#28,#29,#30);
#32=CARTESIAN_POINT('PLANAR FACE1:DEPTH ',(0.0000000000,0.0000000000,-5.0000000000));
#33=DIRECTION(' AXIS ',(0.0000000000,0.0000000000,1.0000000000));
#34=DIRECTION(' REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000));
#35=AXIS2_PLACEMENT_3D('PLANAR FACE1',#32,#33,#34);
#36=ELEMENTARY_SURFACE('PLANAR FACE1-DEPTH PLANE',#35);
#37=PLUS_MINUS_VALUE(0.3000000000,0.3000000000,3);
#38=TOLERANCED_LENGTH_MEASURE(120.0000000000,#37);
#39=DIRECTION('COURSE OF TRAVEL DIRECTION',(0.0000000000,1.0000000000,0.0000000000));
#40=LINEAR_PATH($,#38,#39);
#41=NUMERIC_PARAMETER('PROFILE LENGTH',100.0000000000,'MM');
#42=LINEAR_PROFILE($,#41);
#43=PLANAR_FACE('PLANAR FACE1',#18,(#27),#31,#36,#40,#42,$,());
 /*workingstep1_feature data */

/*workingstep1_operation_cutting tool data */
#19=CUTTING_COMPONENT(80.0000000000,$,$,$);
#20=ENDMILL('MILL
20MM',(#19,#19,#19,#19),80.0000000000,20.0000000000,30.0000000000,.RIGHT.,.F.,4,0.0500000000,$);
/*workingstep1_operation_cutting tool data */

/*workingstep1_operation_technology data */
#21=MILLING_TECHNOLOGY(0.0400000000,.TCP.,$,12.0000000000,$,.F.,.F.,.F.,$);
/*workingstep1_operation_technology data */

/*workingstep1_operation_function data */
#22=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.F.,$,(),.T.,$,$,());
/*workingstep1_operation_function data */

/*workingstep1_operation_strategy data */
#23=PLUNGE_RAMP($,45.0000000000);
#24=PLUNGE_RAMP($,45.0000000000);
#25=DIRECTION('STRATEGY PLANAR FACE1: 1.DIRECTION',(0.0000000000,1.0000000000,0.0000000000));
#26=BIDIRECTIONAL(5.0000000000,.T.,#25,.LEFT.,$);
/*workingstep1_operation_strategy data */
```

Fig. 7. Comparison between two ISO 14649 programs

optimized. Only the analysis of the first workingstep is shown in Fig.7 because other workingstep's analysis is similar.

As Fig.7 shows, some modification can be found. For workpiece data, the size of the original workpiece bounding geometry is corrected and stock data is added. For setup data, it is involved with four coordinate systems, which are very important to the machining process because they largely impact on the machining time and toolpath. They are the machine tool coordinate system, setup coordinate system, workpiece coordinate system and feature coordinate system, respectively. To get efficient machining, the origin in the setup coordinate system is relocated and the workpiece coordinate system is set the same as the setup coordinate system in this case study. For security plane data, the location of the security plane is changed because of the change of the coordinate system. For feature data, because of the change of the coordinate system and correction of geometric size, corresponding feature data is also modified. For other data, considering the machining condition, workpiece material and cutter material etc., the size of cutting tool, feed-rate and the spindle speed are changed and the direction of spindle in the operation data is corrected.

5 Conclusions

The integration of the expert system with neural network not only fully exerts their advantages, but also makes their disadvantages compensated. Based on the expert system and BP neural network, i.e. the expert knowledge and its symbol inference as well as the numeric inference of the neural network, the interpreter can intelligently estimate whether the machining information obtained from ISO 14649 programs is adequate or not, and can also modify it. Thus, the brainwork intensity for the operators is reduced and the process planning time can be shortened. Furthermore, the machining of parts can be implemented more rapidly and better.

References

1. Xu, X.W., He, Q.: Striving for a total integration of CAD, CAPP, CAM and CNC. Robotics and Computer-Integrated Manufacturing 20(2), 101–109 (2004)
2. International Standards Organization: ISO 14649-1. Industrial automation systems and integration – Physical device control – Data model for computerized numerical controllers – Part 1: Overview and fundamental principles (2003)
3. Suh, S.H., Lee, B.E., Chung, D.H., Cheon, S.U.: Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. Computer-Aided Design 35(12), 1073–1082 (2003)
4. Kramer, T.R., Proctor, F., Xu, X., Michaloski, J.L.: Run-time interpretation of STEP-NC: implementation and performance. Computer Integrated Manufacturing 19(6), 495–507 (2006)
5. Moura, P., Marchetti, V.: Logtalk Processing of STEP Part 21 Files. In: Etalle, S., Truszczyński, M. (eds.) ICLP 2006. LNCS, vol. 4079, pp. 453–454. Springer, Heidelberg (2006)
6. Cang, G.L.: Research on Key Technology of STEP-NC-based NC Mill Machining. PhD Thesis, pp. 71–85 (2006) (in Chinese)

7. Yu, W.J.: Research on Tool Path Planning Approach for Five-Axis Machining Based on STEP-NC PhD Thesis, pp. 79–101 (2007) (in chinese)
8. Chen, X.S.: Study on Key Technologies of Numerical Control Tuning Simulation Based on STEP-NC PhD Thesis, pp. 28–37 (2007) (in chinese)
9. Zhao, F., Xu, X., Xie, S.: STEP-NC enabled on-line inspection in support of closed-loop machining. Robotics and Computer-Integrated Manufacturing 24(2), 208–209 (2008)
10. STEP-NC Machine Information, `http://www.steptools.com/products/stepncmachine/`
11. Giarratano, J., Riley, G.: Expert systems: principles and programming, 4th edn. China Machine Press, Beijing (2006)
12. Hagan, M.T., Demuth, H.B., Beale, M.: Neural Network Design, 1st edn. China Machine Press, Beijing (2002)
13. Ni, N.Z., Cai, Q.S.: Neural network expert system based on knowledge discovery techniques. Journal of Xiamen University (Natural Science) 39(3), 288–292 (2000) (in chinese)
14. Feeney, A.B., Kramer, T., Proctor, F., Hardwick, M., Loffredo, D.: STEP-NC Implementation—ARM or AIM? In: White Paper, ISO T24 STEP-Manufacturing Meeting (2003)
15. International Standards Organization: ISO 14649-11. Industrial automation systems and integration – Physical device control – Data model for computerized numerical controllers – Part 11: Process data for milling, 2004(E)

Appendix

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('EXAMPLE 1'),'1');
FILE_NAME('EXAMPLE1.STP','2004-5-4T15-32-
  8',('AUTHOR'),(''),'WZL ISO10303-PART21 PARSER
PACKAGE','','');
FILE_SCHEMA((('MACHINING_SCHEMA','      MILLING_SCHEMA','
  MILLING_TOOL_SCHEMA'));
ENDSEC;
DATA;
#1=CARTESIAN_POINT('SECPLANE1:                  LOCATION
  ',(0.0000000000,0.0000000000,30.0000000000));
#2=DIRECTION('                                      AXIS
  ',(0.0000000000,0.0000000000,1.0000000000));
#3=DIRECTION('
  REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000)
  );
#4=AXIS2_PLACEMENT_3D('PLANE1',#1,#2,#3);
#5=ELEMENTARY_SURFACE('SECURITY PLANE',#4);
#6=PROPERTY_PARAMETER('E=200000N/M2');
#7=MATERIAL('ST-50','STEEL',(#6));
#8=CARTESIAN_POINT('',(0.0000000000,0.0000000000,0.000000
  0000));
#9=DIRECTION('',(0.0000000000,0.0000000000,1.0000000000));
#10=DIRECTION('',(1.0000000000,0.0000000000,0.0000000000));
#11=AXIS2_PLACEMENT_3D('',#8,#9,#10);
```

```
#12=BLOCK('BOUNDING
  STOCK',#11,100.0000000000,100.0000000000,100.0000000000
  );
#13=WORKPIECE('STOCK',#7,$,$,$,#12,());
#14=CARTESIAN_POINT('CLAMPING_POSITION1',(0.0000000000,20
  .0000000000,25.0000000000));
#15=CARTESIAN_POINT('CLAMPING_POSITION2',(100.0000000000,
  20.0000000000,25.0000000000));
#16=CARTESIAN_POINT('CLAMPING_POSITION3',(0.0000000000,10
  0.0000000000,25.0000000000));
#17=CARTESIAN_POINT('CLAMPING_POSITION4',(100.0000000000,
  100.0000000000,25.0000000000));
#18=WORKPIECE('SIMPLE WORK-
  PIECE',#7,0.0100000000,#13,$,$,(#14,#15,#16,#17));
#19=CUTTING_COMPONENT(80.0000000000,$,$,$);
#20=ENDMILL('MILL
20MM',(#19,#19,#19,#19),80.0000000000,20.0000000000,30.00
  00000000,.RIGHT.,.F.,4,0.0500000000,$)
;
#21=MILLING_TECHNOLOGY(0.0400000000,.TCP.,$,12.0000000000
  ,$,.F.,.F.,.F.,$);
#22=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.F.,$,(),.T.,$,$,()
  );
#23=PLUNGE_RAMP($,45.0000000000);
#24=PLUNGE_RAMP($,45.0000000000);
#25=DIRECTION('STRATEGY PLANAR FACE1:
  1.DIRECTION',(0.0000000000,1.0000000000,0.0000000000));
#26=BIDIRECTIONAL(5.0000000000,.T.,#25,.LEFT.,$);
#27=PLANE_FINISH_MILLING($,$,'FINISH PLANAR
FACE1',10.0000000000,$,#20,#21,#22,5.0000000000,#23,#24,#
  26,2.5000000000,$);
#28=CARTESIAN_POINT('PLANAR FACE1:LOCATION
  ',(0.0000000000,0.0000000000,5.0000000000));
#29=DIRECTION(' AXIS
  ',(0.0000000000,0.0000000000,1.0000000000));
#30=DIRECTION('
  REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000)
  );
#31=AXIS2_PLACEMENT_3D('PLANAR FACE1',#28,#29,#30);
#32=CARTESIAN_POINT('PLANAR FACE1:DEPTH
  ',(0.0000000000,0.0000000000,-5.0000000000));
#33=DIRECTION(' AXIS
  ',(0.0000000000,0.0000000000,1.0000000000));
#34=DIRECTION('
  REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000)
  );
#35=AXIS2_PLACEMENT_3D('PLANAR FACE1',#32,#33,#34);
#36=ELEMENTARY_SURFACE('PLANAR FACE1-DEPTH PLANE',#35);
#37=PLUS_MINUS_VALUE(0.3000000000,0.3000000000,3);
#38=TOLERANCED_LENGTH_MEASURE(120.0000000000,#37);
```

```
#39=DIRECTION('COURSE OF TRAVEL DIRECTION',(0.0000000000,1.0000000000,0.0000000000));
#40=LINEAR_PATH($,#38,#39);
#41=NUMERIC_PARAMETER('PROFILE
  LENGTH',100.0000000000,'MM');
#42=LINEAR_PROFILE($,#41);
#43=PLANAR_FACE('PLANAR
  FACE1',#18,(#27),#31,#36,#40,#42,$,());
#44=MACHINING_WORKINGSTEP('WS FINISH PLANAR FACE1',#5,#43,
  #27,$);
#45=DRILLING_CUTTING_TOOL('DRILL
20MM',(),90.0000000000,20.0000000000,70.0000000000,.RIGHT
  .,.F.,120.0000000000);
#46=MILLING_TECHNOLOGY(0.0300000000,.TCP.,$,16.0000000000
  ,$,.F.,.F.,.F.,$);
#47=DRILLING_TYPE_STRATEGY(75.0000000000,50.0000000000,2.
  0000000000,50.0000000000,75.0000000000
,8.0000000000);
#48=DRILLING($,$,'DRILL
HOLE1',10.0000000000,$,#45,#46,#22,$,30.0000000000,0.0000
  000000,0.0000000000,$,#47);
#49=REAMING_CUTTING_TOOL('REAMER
22MM',(#19,#19,#19,#19),100.0000000000,22.0000000000,60.0
  000000000,.RIGHT.,.F.,$);
#50=MILLING_TECHNOLOGY(0.0300000000,.TCP.,$,18.0000000000
  ,$,.F.,.F.,.F.,$);
#51=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#52=REAMING($,$,'REAM
HOLE1',10.0000000000,$,#49,#50,#22,$,30.0000000000,20.000
  0000000,1.0000000000,$,#51,.T.,5.00000
00000,$);
#53=CARTESIAN_POINT('HOLE1: LOCATION ',(20.0000000000,
  60.0000000000,0.0000000000));
#54=DIRECTION(' AXIS ',(0.0000000000,0.0000000000,
  1.0000000000));
#55=AXIS2_PLACEMENT_3D('HOLE1',#53,#54,$);
#56=CARTESIAN_POINT('HOLE1: DEPTH ',(0.0000000000,
  0.0000000000,-30.0000000000));
#57=DIRECTION(' AXIS ',(0.0000000000,0.0000000000,
  1.0000000000));
#58=DIRECTION('
  REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000)
  );
#59=AXIS2_PLACEMENT_3D('HOLE1',#56,#57,#58);
#60=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND
  HOLE1',#59);
#61=TOLERANCED_LENGTH_MEASURE(22.0000000000,#37);
#62=THROUGH_BOTTOM_CONDITION();
#63=ROUND_HOLE('HOLE1
  D=22MM',#18,(#48,#52),#55,#60,#61,$,#62);
#64=MACHINING_WORKINGSTEP('WS DRILL HOLE1',#5,#63,#48,$);
```

```
#65=MACHINING_WORKINGSTEP('WS REAM HOLE1',#5,#63,#52,$);
#66=MILLING_TECHNOLOGY($,.TCP.,$,20.0000000000,$,.F.,.F.,
  .F.,$);
#67=CONTOUR_BIDIRECTIONAL($,$,$,$,$,$);
#68=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGH
POCKET1',15.0000000000,$,#20,#66,#22,$,$,$,#67,2.50000000
  00,5.0000000000,1.0000000000,0.5000000
000);
#69=MILLING_TECHNOLOGY($,.TCP.,$,20.0000000000,$,.F.,.F.,
  .F.,$);
#70=CONTOUR_PARALLEL(5.0000000000,.T.,.CW.,.CONVENTIONAL.
  );
#71=BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISH
POCKET1',15.0000000000,$,#20,#69,#22,$,$,$,#70,2.00000000
  00,10.0000000000,$,$);
#72=CARTESIAN_POINT('POCKET1:  LOCATION  ',(45.0000000000,
  110.0000000000,0.0000000000));
#73=DIRECTION('                                     AXIS
  ',(0.0000000000,0.0000000000,1.0000000000));
#74=DIRECTION(' REF_DIRECTION',(-1.0000000000,
  0.0000000000,0.0000000000));
#75=AXIS2_PLACEMENT_3D('POCKET1',#72,#73,#74);
#76=CARTESIAN_POINT('POCKET1:    DEPTH    ',(0.0000000000,
  0.0000000000,-30.0000000000));
#77=DIRECTION('     AXIS     ',(0.0000000000,0.0000000000,
  1.0000000000));
#78=DIRECTION('
  REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000)
  );
#79=AXIS2_PLACEMENT_3D('POCKET1',#76,#77,#78);
#80=ELEMENTARY_SURFACE('DEPTH SURFACE FOR POCKET1',#79);
#81=PLANAR_POCKET_BOTTOM_CONDITION();
#82=PLUS_MINUS_VALUE(0.1000000000,0.1000000000,3);
#83=TOLERANCED_LENGTH_MEASURE(1.0000000000,#82);
#84=PLUS_MINUS_VALUE(0.1000000000,0.1000000000,3);
#85=TOLERANCED_LENGTH_MEASURE(10.0000000000,#84);
#86=CARTESIAN_POINT('P1',(0.0000000000,0.0000000000,0.000
  0000000));
#87=CARTESIAN_POINT('P2',(0.0000000000,80.0000000000,0.00
  00000000));
#88=CARTESIAN_POINT('P3',(-
  50.0000000000,80.0000000000,0.0000000000));
#89=CARTESIAN_POINT('P4',(-
  50.0000000000,0.0000000000,0.0000000000));
#90=POLYLINE('CONTOUR OF POCKET1',(#86,#87,#88,#89,#86));
#91=GENERAL_CLOSED_PROFILE($,#90);
#92=CLOSED_POCKET('POCKET1',#18,(#68,#71),#75,#80,(),$,#8
  1,#83,#85,#91);
#93=MACHINING_WORKINGSTEP('WS ROUGH POCKET1',#5,#92,#68,$);
#94=MACHINING_WORKINGSTEP('WS FINISH POCKET1',#5,#92,#71,
  $);
```

```
#95=CARTESIAN_POINT('SETUP1: LOCATION ',(150.0000000000,
  90.0000000000,40.0000000000));
#96=DIRECTION(' AXIS ',(0.0000000000,0.0000000000,
  1.0000000000));
#97=DIRECTION('
  REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000)
  );
#98=AXIS2_PLACEMENT_3D('SETUP1',#95,#96,#97);
#99=CARTESIAN_POINT('WORKPIECE1:LOCATION
  ',(0.0000000000,0.0000000000,0.0000000000));
#100=DIRECTION(' AXIS ',(0.0000000000,0.0000000000,
  1.0000000000));
#101=DIRECTION('
  REF_DIRECTION',(1.0000000000,0.0000000000,0.0000000000)
  );
#102=AXIS2_PLACEMENT_3D('WORKPIECE',#99,#100,#101);
#103=NC_VARIABLE('',0.0000000000);
#104=NC_VARIABLE('',0.0000000000);
#105=OFFSET_VECTOR((#103,#103,#103),(#104,#104,#104));
#106=WORKPIECE_SETUP(#18,#102,#105,$,());
#107=SETUP('SETUP1',#98,#5,(#106));
#108=WORKPLAN('MAIN WORKPLAN',(#44,#64,#65,#93,#94),$,
  #107,$);
#109=PROJECT('EXECUTE EXAMPLE1',#108,(#18),$,$,$);
ENDSEC;
END-ISO-10303-21;
```

Erratum: Feature Selection Using Ant Colony Optimization for Text-Independent Speaker Verification System

Javad Sohafi-Bonab[1] and Mehdi Hosseinzadeh Aghdam[2]

[1] Department of Computer Engineering, Islamic Azad University,
Bonab Branch, Bonab, Iran
[2] Department of Computer Engineering & IT, Payame Noor University,
Bonab, Iran
sohafi.j.b@gmail.com, hosseinzadeh@comp.ui.ac.ir

Z. Cai et al. (Eds.): ISICA 2010, LNCS 6382, pp. 13–24, 2010.

DOI 10.1007/978-3-642-16493-4_54

The paper entitled “Feature Selection Using Ant Colony Optimization for Text-Independent Speaker Verification System’’ by Javad Sohafi-Bonab and Mehdi Hosseinzadeh Aghdam, starting on page 13 of this publication, has been retracted as parts of Sect. 1, Sect. 2 and Sect. 5 were plagiarized from the paper starting on page 421 of LNCS 5099.

The original online version for this chapter can be found at
http://dx.doi.org/10.1007/978-3-642-16493-4_2

Author Index

Aghdam, Mehdi Hosseinzadeh 13, E1

Bai, Xiaolan 523
Bian, Bian 402
Bu, Zhiqiong 338

Cai, Zhihua 86, 112, 312
Chen, Chengjun 359
Chen, Guolong 1
Chen, Jing 402
Chen, Yan 261
Chen, Yue 370, 434
Chen, Yunliang 211

Dai, Guangming 229
Deng, Qiao 64
Deng, Xiaotie 252
Dong, Keren 252

Fang, Yongsheng 181
Fang, Yuankang 329
Fan, Yuanyuan 474
Fey, Dietmar 170

Guan, Jing 112
Guo, Jincui 487
Guo, Lili 240
Guo, Wenzhong 1
Guo, Yan 390

He, Huagang 280
He, Zhouqian 272
Ho, Hai-Nguyen 412
Huang, Jianzhong 211
Huang, Yujuan 289
Huang, Zhiqiu 329
Hu, Nannan 390
Hung, Yi-Tsan 505
Hwang, Wen-Jyi 505

Jiang, Dazhi 25, 76
Jiang, Hua 193
Jiang, Jingjing 261
Jiang, Siwei 312

Kang, Lishan 193
Konstantinidis, Andreas 33
Kouzani, Abbas Z. 444
Krömer, Pavel 424

Lewicki, Arkadiusz 44
Li, Chengjun 134
Li, Huanzhe 123
Li, Jituo 349
Li, Jun 181
Li, Kangshun 25
Li, Ni 380
Li, Niu 359
Li, Ping 496
Li, Qingshan 240
Li, Shen 54
Li, Xia 123
Li, Yuanxiang 54, 304
Li, Zhenhua 380, 402
Li, Zhuo 496
Liang, Qingzhong 474
Limmer, Steffen 170
Lin, Chang 297
Lin, Guangming 200
Lin, Shiow-Jyu 505
Liu, Changsheng 465
Liu, Jian 158
Liu, Kunqi 123
Liu, Li 434
Liu, Songhu 134
Liu, Sundong 200
Liu, Wei 64
Liu, Xiaobo 86
Liu, Yong 496
Liu, Yongxian 523
Luo, Wenjian 148

Ma, Lixiao 123
Mo, Li 229
Monteiro, Glauber Duarte 220
Moreira, Renato Simões 220
Mu, Xiangwei 261

Oliveira, Roberto Célio Limão de 220

Park, Dong-Chul 412
Platoš, Jan 424

Qi, Quan 434
Qian, Ling 280

Rahnamayan, Shahryar 95
Ren, Zhengyong 465

Shi, Lei 349
Shi, Zhangsong 140
Snášel, Václav 424
Soares, Átila Siqueira 220
Sohafi-Bonab, Javad 13, E1
Song, Liguo 390
Sun, Guangfu 134

Tadeusiewicz, Ryszard 44
Tang, Fei 200
Teixeira, Otávio Noura 220

Wan, Chang 455
Wang, Feng 252
Wang, Hui 76, 95, 103
Wang, Huijie 200
Wang, Jing 95, 103
Wang, Lingling 304
Wang, Lu 112
Wang, Yangsheng 349
Wang, Yincheng 487
Woo, Dong-Min 412
Wu, Qi 465
Wu, Zhijian 25, 76, 95, 103

Xiao, Sheng 140
Xie, Changsheng 211
Xing, Changfeng 140
Xu, Xue 64

Yang, Jian 261
Yang, Man 280
Yang, Ming 112
You, Bingyu 1
Yu, Chao 86, 380
Yu, Liangjiang 487
Yu, Yan 465
Yu, Zhangyi 390
Yue, Xuezhi 25
Yun, Haishun 240

Zeng, Sanyou 390, 474
Zhai, Jing 272
Zhang, Chao 329
Zhang, Dongmei 64, 134
Zhang, Guixu 33
Zhang, Haifeng 289
Zhang, Hao 380
Zhang, Min 289
Zhang, Shuqi 193
Zhang, Wei 516
Zhang, Yu 523
Zhao, Chengguang 240
Zhao, Man 272
Zhao, Xiaojuan 487
Zhen, Ziyang 329
Zheng, Bojin 338
Zhong, Huaming 380
Zhou, Aimin 33
Zhou, Qian 148
Zhu, Fei 193
Zhu, Jiankai 229
Zou, Jinxin 487
Zou, Xiufen 516