Ying Tan Yuhui Shi
Carlos A. Coello Coello (Eds.)

LNCS 8795

Advances in Swarm Intelligence

5th International Conference, ICSI 2014
Hefei, China, October 17–20, 2014
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 8795

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

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Alfred Kobsa
University of California, Irvine, CA, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
TU Dortmund University, Germany

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Gerhard Weikum
Max Planck Institute for Informatics, Saarbruecken, Germany

Ying Tan Yuhui Shi
Carlos A. Coello Coello (Eds.)

Advances in Swarm Intelligence

5th International Conference
ICSI 2014, Hefei, China, October 17-20, 2014
Proceedings, Part II

Springer

Volume Editors

Ying Tan
Peking University
Key Laboratory of Machine Perception (MOE)
School of Electronics Engineering and Computer Science
Department of Machine Intelligence
Beijing 100871, China
E-mail: ytan@pku.edu.cn

Yuhui Shi
Xi'an Jiaotong-Liverpool University
Department of Electrical and Electronic Engineering
Suzhou 215123, China
E-mail: yuhui.shi@xjtlu.edu.cn

Carlos A. Coello Coello
CINVESTAV-IPN
Investigador Cinvestav 3F, Depto. de Computación
México, D.F. 07300, Mexico
E-mail: ccoello@cs.cinvestav.mx

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-319-11896-3 e-ISBN 978-3-319-11897-0
DOI 10.1007/978-3-319-11897-0
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014949260

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

© Springer International Publishing Switzerland 2014
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in ist current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This book and its companion volume, LNCS vols. 8794 and 8795, constitute the proceedings of the fifth International Conference on Swarm Intelligence (ICSI 2014) held during October 17–20, 2014, in Hefei, China. ICSI 2014 was the fifth international gathering in the world for researchers working on all aspects of swarm intelligence, following the successful and fruitful Harbin event (ICSI 2013), Shenzhen event (ICSI 2012), Chongqing event (ICSI 2011) and Beijing event (ICSI 2010), which provided a high-level academic forum for the participants to disseminate their new research findings and discuss emerging areas of research. It also created a stimulating environment for the participants to interact and exchange information on future challenges and opportunities in the field of swarm intelligence research.

ICSI 2014 received 198 submissions from about 475 authors in 32 countries and regions (Algeria, Australia, Belgium, Brazil, Chile, China, Czech Republic, Finland, Germany, Hong Kong, India, Iran, Ireland, Italy, Japan, Macao, Malaysia, Mexico, New Zealand, Pakistan, Romania, Russia, Singapore, South Africa, Spain, Sweden, Taiwan, Thailand, Tunisia, Turkey, United Kingdom, United States of America) across six continents (Asia, Europe, North America, South America, Africa, and Oceania). Each submission was reviewed by at least two reviewers, and on average 2.7 reviewers. Based on rigorous reviews by the Program Committee members and reviewers, 105 high-quality papers were selected for publication in this proceedings volume with an acceptance rate of 53.03%. The papers are organized in 18 cohesive sections, 3 special sessions and one competitive session, which cover all major topics of swarm intelligence research and development.

As organizers of ICSI 2014, we would like to express sincere thanks to University of Science and Technology of China, Peking University, and Xi'an Jiaotong-Liverpool University for their sponsorship, as well as to the IEEE Computational Intelligence Society, World Federation on Soft Computing, and International Neural Network Society for their technical co-sponsorship. We appreciate the Natural Science Foundation of China for its financial and logistic support. We would also like to thank the members of the Advisory Committee for their guidance, the members of the International Program Committee and additional reviewers for reviewing the papers, and the members of the Publications Committee for checking the accepted papers in a short period of time. Particularly, we are grateful to Springer for publishing the proceedings in the prestigious series of Lecture Notes in Computer Science. Moreover, we wish to express our heartfelt appreciation to the plenary speakers, session chairs, and student helpers. In addition, there are still many more colleagues, associates, friends, and supporters who helped us in immeasurable ways; we express our sincere gratitude to them all. Last but not the least, we would like to thank all the speakers, authors, and participants for their great contributions that made ICSI 2014 successful and all the hard work worthwhile.

July 2014

Ying Tan
Yuhui Shi
Carlos A. Coello Coello

Organization

General Chairs

Russell C. Eberhart	Indiana University-Purdue University, USA
Ying Tan	Peking University, China

Programme Committee Chairs

Yuhui Shi	Xi'an Jiaotong-Liverpool University, China
Carlos A. Coello Coello	CINVESTAV-IPN, Mexico

Advisory Committee Chairs

Gary G. Yen	Oklahoma State University, USA
Hussein Abbass	University of New South Wales, ADFA, Australia
Xingui He	Peking University, China

Technical Committee Chairs

Xiaodong Li	RMIT University, Australia
Andries Engelbrecht	University of Pretoria, South Africa
Ram Akella	University of California, USA
M. Middendorf	University of Leipzig, Germany
Kalyanmoy Deb	Indian Institute of Technology Kanpur, India
Ke Tang	University of Science and Technology of China, China

Special Sessions Chairs

Shi Cheng	The University of Nottingham, Ningbo, China
Meng-Hiot Lim	Nanyang Technological University, Singapore
Benlian Xu	Changshu Institute of Technology, China

Competition Session Chairs

Jane J. Liang	Zhengzhou University, China
Junzhi Li	Peking University, China

Publications Chairs

Radu-Emil Precup	Politehnica University of Timisoara, Romania
Haibin Duan	Beihang University, China

Publicity Chairs

Yew-Soon Ong	Nanyang Technological University, Singapore
Juan Luis Fernandez Martinez	University of Oviedo, Spain
Hideyuki Takagi	Kyushu University, Japan
Qingfu Zhang	University of Essex, UK
Suicheng Gu	University of Pittsburgh, USA
Fernando Buarque	University of Pernambuco, Brazil
Ju Liu	Shandong University, China

Finance and Registration Chairs

Chao Deng	Peking University, China
Andreas Janecek	University of Vienna, Austria

Local Arrangement Chairs

Wenjian Luo	University of Science and Technology of China, China
Bin Li	University of Science and Technology of China, China

Program Committee

Kouzou Abdellah	University of Djelfa, Algeria
Ramakrishna Akella	University of California at Santa Cruz, USA
Rafael Alcala	University of Granada, Spain
Peter Andras	Newcastle University, UK
Esther Andrés	INTA, USA
Sabri Arik	Istanbul University, Turkey
Helio Barbosa	Laboratório Nacional de Computação Científica, Brazil
Carmelo J.A. Bastos Filho	University of Pernambuco, Brazil
Christian Blum	Technical University of Catalonia, Spain
Salim Bouzerdoum	University of Wollongong, Australia
Xinye Cai	Nanhang University, China
David Camacho	Universidad Autonoma de Madrid, Spain
Bin Cao	Tsinghua University, China
Kit Yan Chan	DEBII, Australia

Mu-Song Chen	Da-Yeh University, Taiwan
Walter Chen	National Taipei University of Technology, China
Shi Cheng	The University of Nottingham Ningbo, China
Leandro Coelho	Pontifícia Universidade Católica do Parana, Brazil
Chenggang Cui	Shanghai Advanced Research Institute, Chinese Academy of Sciences, China
Chaohua Dai	Southwest Jiaotong University, China
Arindam K. Das	University of Washington, USA
Prithviraj Dasgupta	University of Nebraska at Omaha, USA
Kusum Deep	Indian Institute of Technology Roorkee, India
Mingcong Deng	Tokyo University of Agriculture and Technology, Japan
Yongsheng Ding	Donghua University, China
Madalina-M. Drugan	Vrije University, The Netherlands
Mark Embrechts	RPI, USA
Andries Engelbrecht	University of Pretoria, South Africa
Fuhua Fan	Electronic Engineering Institute, China
Zhun Fan	Technical University of Denmark, Denmark
Komla Folly	University of Cape Town, South Africa
Shangce Gao	University of Toyama, Japan
Ying Gao	Guangzhou University, China
Shenshen Gu	Shanghai University, China
Suicheng Gu	University of Pittsburgh, USA
Ping Guo	Beijing Normal University, China
Haibo He	University of Rhode Island, USA
Ran He	National Laboratory of Pattern Recognition, China
Marde Helbig	CSIR: Meraka Institute, South Africa
Mo Hongwei	Harbin Engineering University, China
Jun Hu	Chinese Academy of Sciences, China
Xiaohui Hu	Indiana University Purdue University Indianapolis, USA
Guangbin Huang	Nanyang Technological University, Singapore
Amir Hussain	University of Stirling, UK
Hisao Ishibuchi	Osaka Prefecture University, Japan
Andreas Janecek	University of Vienna, Austra
Changan Jiang	RIKEN-TRI Collaboration Center for Human-Interactive Robot Research, Japan
Mingyan Jiang	Shandong University, China
Liu Jianhua	Fujian University of Technology, China
Colin Johnson	University of Kent, USA
Farrukh Khan	FAST-NUCES Islamabad, Pakistan
Arun Khosla	National Institute of Technology, Jalandhar, India

Franziska Klügl	Örebro University, Sweden
Thanatchai Kulworawanichpong	Suranaree University of Technology, Thailand
Germano Lambert-Torres	Itajuba Federal University, Brazil
Xiujuan Lei	Shaanxi Normal University, China
Bin Li	University of Science and Technology of China, China
Xiaodong Li	RMIT University, Australia
Xuelong Li	Chinese Academy of Sciences, China
Yangmin Li	University of Macau, China
Jane-J. Liang	Zhengzhou University, China
Andrei Lihu	Politehnica University of Timisoara, Romania
Fernando B. De Lima Neto	University of Pernambuco, Brazil
Ju Liu	Shandong University, China
Wenlian Lu	Fudan University, China
Wenjian Luo	University of Science and Technology of China, China
Jinwen Ma	Peking University, China
Chengying Mao	Jiangxi University of Finance and Economics, China
Michalis Mavrovouniotis	De Montfort University, UK
Bernd Meyer	Monash University, Australia
Martin Middendorf	University of Leipzig, Germany
Sanaz Mostaghim	Institute IWS, Germany
Jonathan Mwaura	University of Pretoria, South Africa
Pietro S. Oliveto	University of Sheffield, UK
Feng Pan	Beijing Institute of Technology, China
Bijaya Ketan Panigrahi	IIT Delhi, India
Sergey Polyakovskiy	Ufa State Aviation Technical University, USA
Thomas Potok	ORNL, USA
Radu-Emil Precup	Politehnica University of Timisoara, Romania
Kai Qin	RMIT University, Australia
Quande Qin	Shenzhen University, China
Boyang Qu	Zhongyuan University of Technology, China
Robert Reynolds	Wayne State University, USA
Guangchen Ruan	Indiana University Bloomington, USA
Eugene Santos	Dartmouth College, USA
Gerald Schaefer	Loughborough University, USA
Kevin Seppi	Brigham Young University, USA
Zhongzhi Shi	Institute of Computing Technology, CAS, China
Pramod Kumar Singh	ABV-IIITM Gwalior, India
Ponnuthurai Suganthan	Nanyang Technological University, Singapore
Mohammad Taherdangkoo	Shiraz University, Iran
Hideyuki Takagi	Kyushu University, Japan
Ying Tan	Peking University, China

Ke Tang	University of Science and Technology of China, China
Peter Tino	University of Birmingham, UK
Mario Ventresca	Purdue University, USA
Cong Wang	Northeasten University, China
Guoyin Wang	Chongqing University of Posts and Telecommunications, China
Jiahai Wang	Sun Yat-sen University, China
Jun Wang	Peking University, China
Lei Wang	Tongji University, China
Ling Wang	Tsinghua University, China
Lipo Wang	Nanyang Technological University, Singapore
Qi Wang	Xi'an Institute of Optics and Precision Mechanics of CAS, China
Zhenzhen Wang	Jinling Institute of Technology, China
Man Leung Wong	Lingnan University, China
Shunren Xia	Zhejiang University, China
Bo Xing	University of Johannesburg, South Africa
Ning Xiong	Mälardalen University, Sweden
Benlian Xu	Changsu Institute of Technology, China
Bing Xue	Victoria University of Wellington, New Zealand
Pei Yan	University of Aziz, Japan
Yingjie Yang	De Montfort University, UK
Wei-Chang Yeh	National Tsing Hua University, China
Gary Yen	Oklahoma State University, USA
Peng-Yeng Yin	National Chi Nan University, Taiwan
Ivan Zelinka	FEI VSB-Technical University of Ostrava, Czech Republic
Zhi-Hui Zhan	Sun Yat-sen University, China
Jie Zhang	Newcastle University, UK
Jun Zhang	Waseda University, Japan
Junqi Zhang	Tongji University, China
Lifeng Zhang	Renmin University of China, China
Mengjie Zhang	Victoria University of Wellington, New Zealand
Qieshi Zhang	Waseda University, Japan
Qingfu Zhang	University of Essex, UK
Zhenya Zhang	Anhui University of Architecture, China
Qiangfu Zhao	The University of Aizu, Japan
Wenming Zheng	Southeast University, China
Yujun Zheng	Zhejiang University of Technology, China
Cui Zhihua	Complex System and Computational Intelligence Laboratory, China
Aimin Zhou	East China Normal University, China
Zexuan Zhu	Shenzhen University, China
Xingquan Zuo	Beijing University of Posts and Telecommunications, China

Additional Reviewers

Alves, Felipe
Bi, Shuhui
Chalegre, Marlon
Cheng, Shi
Dong, Xianguang
Gonzalez-Pardo, Antonio
Haifeng, Sima
Hu, Weiwei
Jun, Bo
Lacerda, Marcelo
Lee, Jie
Li, Yexing
Ling, Haifeng
Menéndez, Héctor
Pei, Yan
Rakitianskaia, Anna
Singh, Garima
Singh, Pramod
Wang, Aihui
Wen, Shengjun
Wenbo, Wan
Wu, Peng
Wu, Zhigang
Xiao, Xiao
Xin, Cheng
Yang, Wankou
Yang, Zhixiang
Yu, Czyujian
Zheng, Zhongyang

Table of Contents – Part II

Classification Methods

GPU-Based Methods

Scheduling and Path Planning

Wireless Sensor Network

Power System Optimization

Other Applications

Special Session on Swarm Intelligence in Image and Video Processing

Special Session on Applications of Swarm Intelligence to Management Problems

Special Session on Swarm Intelligence for Real-World Application

Special Session on ICSI 2014 Competition on Single Objective Optimization

Erratum

Table of Contents – Part I

Novel Swarm-Based Search Methods

Novel Optimization Algorithm

Particle Swarm Optimization

Ant Colony Optimization for Travelling Salesman Problem

Artificial Bee Colony Algorithms

Artificial Immune System

Evolutionary Algorithms

Neural Networks and Fuzzy Methods

Hybrid Methods

Multi-objective Optimization

Multi-agent Systems

Evolutionary Clustering Algorithms

Semi-supervised Ant Evolutionary Classification

Ping He*, Xiaohua Xu*, Lin Lu, Heng Qian, Wei Zhang, and Kanwen Li

Department of Computer Science, Yangzhou University, Yangzhou 225009, China
{angeletx,arterx,linlu60}@gmail.com

Abstract. In this paper, we propose an ant evolutionary classification model, which treats different classes as ant colonies to classify the unlabeled instances. In our model, each ant colony sends its members to propagate its unique pheromone on the unlabeled instances. The unlabeled instances are treated as unlabeled ants. They are assigned to different ant colonies according to the pheromone that different colonies leave on it. Next, the natural selection is carried out to maintain the history colony information as well as the scale of swarms. Theoretical analysis and experimental results show the effectiveness of our proposed model for evolutionary data classification.

Keywords: Evolutionary classification, Ant colony, K-nearest neighbor.

1 Introduction

Evolutionary data comes from many application fields, such as topics in weblogs and locations in GPS sensors. Evolutionary data mining can be classified into the two categories, evolutionary clustering and evolutionary classification. Among them, evolutionary classification refers to the situation where some instances in the data flow are attached with known labels, and the target is to classify the unlabeled data in the real-time.

Various evolutionary classification methods [3]–[13] have been proposed from different aspects, including concept drifts, class distribution and temporal smoothness. However, the assumption of entire labeled data availability is often violated in the real-world problems, because labels may be scare or not readily available. As a result, semi-supervised evolutionary learning methods have been recently put forward. Yangging Jia et al. [14] proposed a semi-supervised classification algorithm for dynamic mail post categorization. They carried out temporal smoothness assumption using temporal regularizers defined in the Hilbert space, and then derived the online algorithm that efficiently finds the closed-form solution to the target function. Later, H. Borchani et al. [15] proposed a new semi-supervised learning approach for concept-drifting data streams. They aim to take advantage of unlabeled data to detect possible concept drifts and, if necessary, update the classifier over time even if only a few labeled data are available. However, both the previous works assumes that at any time stamp, at least one labeled instance for each class should be provided, which can be easily violated in the real-world applications.

* Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 1–7, 2014.
© Springer International Publishing Switzerland 2014

In this paper, we propose a semi-supervised ant evolutionary classification model, which only require users to specify the number of labels and provide at least one labeled sample in the beginning. In our work, we treat each data instance as an ant and each class of labeled instances as an ant colony. The whole swarm, i.e., the whole dataset, is composed of all the different colonies and the unlabeled ants. They evolve with time based on the simulation of natural selection. Therefore, our proposed algorithm is 'self-training' in nature. Compared to the previous research, our method can be applied to a more generalized scenario, where the class distribution is arbitrary and the number of labeled instances is unfixed (even down to 0) at each time step.

The rest of this paper is organized as follows: Section 2 describes our ant evolutionary classification model in detail. Section 3 presents some simulation results to demonstrate its classification performance. Finally, Section 4 concludes the paper.

2 Ant Evolutionary Classification Model

In semi-supervised evolutionary classification, each data is associated with not only a label y but also a time stamp $t \in \{1, \ldots, T\}$. Given a set of data subsets $X = \{X^1, X^2, \ldots, X^T\}$, where X^t represents the data at time step t, $X^t = X^t_m \bigcup X^t_u$, $X^t_m = (x^t_i)_{i=1\ldots|X^t_m|}$ is labeled, the corresponding label subset is $Y^t_m = (y^t_i)_{i=1\ldots|X^t_m|}$, and $X^t_u = (x^t_i)_{|X^t_m|+1\ldots|X^t|}$ is unlabeled, the goal is to predict the label of X^t_u, i.e., $Y^t_u = (y^t_i)_{|X^t_m|+1\ldots|X^t|}$, in the real time.

To solve this problem, we propose an Ant Evolutionary Classification (AEC) model. It treats each class as an ant colony, respectively denoted as $A_{l=1\ldots c}$, where c is the number of classes or labels. Particularly, we let the unlabeled dataset X_u form a special colony with unknown class, represented by $A_0 = X_u$. The l^{th} colony at time step t is denoted as A^t_l. The i^{th} member of A^t_l is denoted by a^t_{li}, a^t_{li} is labeled if $l > 0$ and unlabeled otherwise. Therefore, the swarm at time t is composed by $c+1$ ant colonies, i.e., $A^t = \{A^t_0, A^t_1, \ldots, A^t_c\}$.

We assume that each ant colony possesses a unique pheromone. In order to expand the territory, each colony has to recruit new ants by spreading its pheromone onto the unlabeled ones. The new members joining the ant colony $A^t_{l>0}$ is composed of two groups: 1) the labeled data provided at time step t, 2) the unlabeled data assigned to $A^t_{l>0}$ at the time step t.

Instead of recording the pheromone left by each ant individual, we record the pheromone left each ant colony. We define the pheromone matrix at time step t as a $|X^t| \times c$ matrix τ^t. Each column of τ^t records the pheromone left by one colony on all the ants. The element $\tau^t_{(j+|X^t_m|)l}$ indicates the pheromone left by ants from colony A^t_l on the ant a_{0j} at time step t. The matrix τ^t is divided into two blocks. The first block with size $|X^t_m| \times c$ records the pheromone left on the labeled ants, the second block with size $|X^t_u| \times c$ records the pheromone left on the unlabeled ants. In this paper, we fix the first block of τ^t unchanged and only update its second block, whose element is τ_{ij}. The initial value of τ at time step 0 is set $\tau^0_{ij} = 1$ if and only if y_i belongs to the j^{th} class, otherwise $\tau^0_{ij} = 0$.

Since labeled and unlabeled data is provided at each time, the pheromone matrix needs to be updated accordingly. We define τ^{t_s} as the pheromone matrix in the s^{th} iteration of the pheromone update at time step t. Without loss of generality, given an ant a^t_{0i} and a nest $A^t_l (l > 0)$, the updated pheromone intensity on a^t_{0i} is

$$\tau^{t_{s+1}}_{(i+|X^t_m|)l} \leftarrow \sum_{j=1}^{|X^t_u|} \eta^t_{i(j+|X^t_m|)} \tau^{t_s}_{(j+|X^t_m|)l} + \sum_{k=1}^{|X^t_m|} \eta^t_{ik} \tau^{t_0}_{kl} \tag{1}$$

$\eta^t = (\eta^t_{ij})_{|X^t_u| \times |X^t|}$ is the heuristic value matrix (or similarty matrix),

$$\eta^t_{ij} = e^{-d^t_{ij}} \tag{2}$$

d^t_{ij} is the distance between the i^{th} and j^{th} ants at time step t,

$$d^t_{ij} = \begin{cases} (a^t_{li} - a^t_{0j})^T (\Sigma^t_l)^{-1} (a^t_{li} - a^t_{0j}) & \text{if } l > 0 \\ \frac{||a^t_{0i} - a^t_{0j}||^2}{2\sigma^2} & \text{if } a^t_{0i}, a^t_{0j} \in A^t_0 \\ \infty & \text{otherwise} \end{cases} \tag{3}$$

Σ^t_l is the covariance matrix of the l^{th} colony at time step t, and σ is a spread parameter. Note that we define the distance between labeled and unlabeled ants as Mahalanobis distance so as to utilize the prior class distribution of labeled data, and define the distance among unlabeled ants as Euclidean distance due to the lack of class information. Similar to the partition of τ^t, we also divide η^t into two parts. The first block with size $|X^t_u| \times |X^t_m|$ records the similarity between unlabeled ants and labeled ants, and the second block with size $|X^t_u| \times |X^t_u|$ records the similarity among unlabeled ants.

To interprete eq. (1), we view as two parts, corresponding to the two blocks of η^t and τ^t. In the first term, $\eta^t_{i(j+|X^t_m|)}$ represents the similarity between the i^{th} and j^{th} unlabeled ant, $\tau^{t_s}_{(j+|X^t_m|)l}$ is the pheromone on the j^{th} unlabeled ant left by the l^{th} colony in the s^{th} iteration at time step t. Therefore, the first term computes the sum of pheromone indirectly propagated from the labeled ants via the unlabeled ants. In the second term, η^t_{ik} is the similarity between the i^{th} ant and the k^{th} labeled ant, τ^0_{kl} is the initial pheromone on the k^{th} labeled ant. Hence the second term computes the sum of the pheromone directly propagated from the labeled ants. The reason for using $\tau^{t_0}_{kl}$ instead of $\tau^{t_s}_{kl}$ is because we keep the pheromone on the labeled ants unchanged to avoid concept drifting.

After the convergence of the pheromone matrix $\tau^t = \tau^{t_\infty}$, we predict the label of each unlabeled ant a^t_{0i} according to the amount of pheromone that different ant colonies leave on it.

$$y^t_i = \arg\max_l \tau^t_{il} \tag{4}$$

To determine whether an unlabeled data should be included in its predicted ant colony, we need to further evaluate its fitness to the colony. Given a colony A^t_l at time step t, we define the fitness of $a^t_{l_i} \in A^t_l$ as

$$fitness(a^t_{li}) = \frac{1}{|A^t_l|} \sum_{a^t_{lj} \in A^t_l, i \neq j} e^{-(a^t_{li} - a^t_{lj})^T (\Sigma^t_l)^{-1} (a^t_{li} - a^t_{lj})} \tag{5}$$

where $|A_l^t|$ is the size of A_l^t, Σ_l^t is the covariance matrix of A_l^t. Based on the fitness evaluation, the evolution of ant colonies are composed of two steps. 1) Member Addition. For each unlabeled ant, if its fitness to its predicted class is higher than a threshold $\beta \in (0,1]$, then it will be included in the target colony, used as the training set for the label prediction at next time step. 2) Member Deletion. To avoid class imbalance and allow member change, we set a maximum for the size of an ant colony $(MaxColonySize)$. Once this maximum is reached, the members with the lowest fitness in that colony will be removed.

3 Experiments

We test our algorithm on three datasets, whose details are summarized in Table 1. Twomoons is a synthetic dataset including two classes of intertwining moons. Mushroom and Hyperplane datasets from the UCI repository are used to simulate the concept drift problem.

Table 1. Summarization of the test datasets

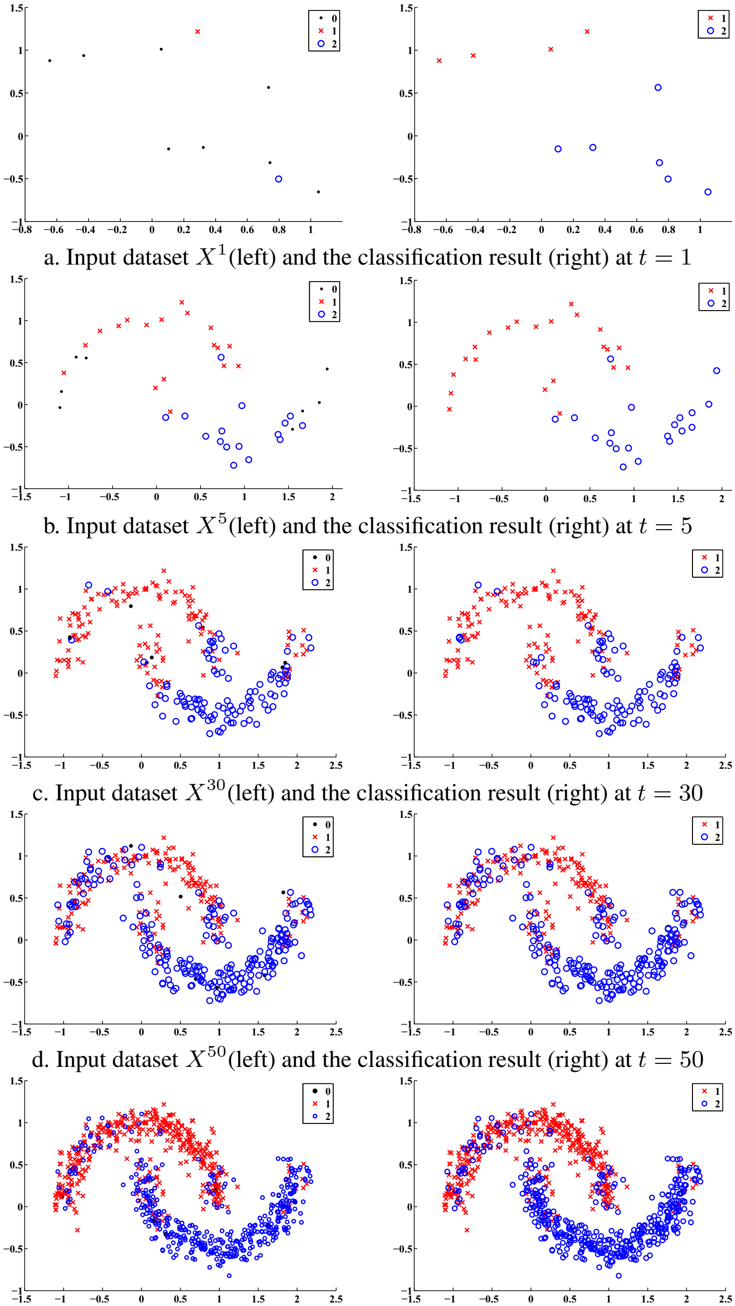
Dataset	#Size	#Attributes	#Classes
Twomoons	2000	2	2
Mushroom	8124	22	2
Hyperplane	10000	10	2

3.1 Synthetic Dataset

We first test our algorithm on the synthetic dataset Twomoons dataset for illustration. At first, the dataset is divided into $T = 100$ time intervals. Fig. 1 shows the the evolutionary classification process in five ascending time steps. The red crosses and blue circles respectively denote the two different classes of data, and the black dots represent the unlabeled instance. The subfigures at the left side depict the input data including the previous classification result, while the subfigures at the right side depict the predicted labels of those black dots in the left subfigures. As we can see, when $t = 1$, only two labeled instances are provided and we cannot recognize the intrinsic structure of the input data. Later, after more instances are provided, our algorithm gradually assigns labels to the unlabeled data points and then discovers the manifold structure of two moons.

3.2 Real-World Dataset

We use the 1% labeled ratio to generate the training data and randomly distribute them into $T = 100$ time blocks. Therefore, the provided labeled data may vary with different times blocks, and even maybe absent. For the evaluation of evolutionary classification



a. Input dataset X^1(left) and the classification result (right) at $t = 1$

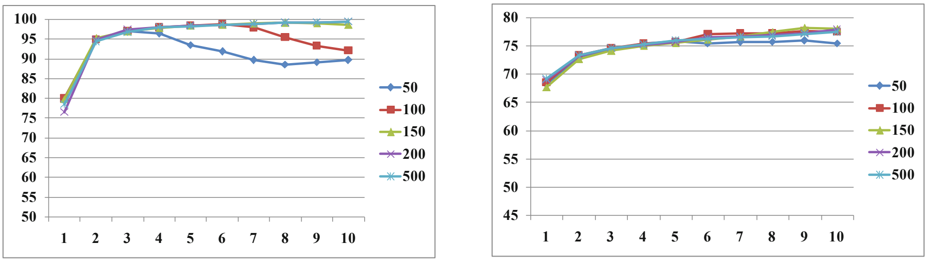
b. Input dataset X^5(left) and the classification result (right) at $t = 5$

c. Input dataset X^{30}(left) and the classification result (right) at $t = 30$

d. Input dataset X^{50}(left) and the classification result (right) at $t = 50$

e. Input dataset X^{100}(left) and the classification result (right) at $t = 100$

Fig. 1. Illustration of AEC Model on Twomoons dataset



(a) Mushroom

(b) Hyperplane

Fig. 2. Average block accuracy with different $MaxColonySize$ on two real-world datasets

Table 2. Average overall accuracy with different $MaxColonySize$

	50	100	150	200	500
Mushroom	92.76	95.01	96.26	**96.7**	96.6
Hyperplane	74.41	75.50	**75.65**	74.86	75.19

performance, we adopt both overall classification accuracy and local classification accuracy, which refers to the classification accuracy within each time block. To give a reliable result, 50 runs of random simulation are carried out to produce an average overall classification accuracy.

In our algorithm, we set a ceiling for the size of ant colonies, i.e., $MaxColonySize$. We adopt five values (50, 100, 150, 200, 500) as the max colony size. Fig. 2 illustrates the relationship between the block accuracy and parameter $MaxColonySize$ on the two real-world datasets. We can see that $MaxColonySize$ exerts obvious influence on Mushroom dataset.

In Table 2, each row shows the average overall accuracy on one dataset with five different $MaxColonySize$ values. We note that Twomoons and Hyperplane datasets perform best at size 150. It indicates that 150 might be a good choice for $MaxColonySize$. In addition, the setting of this parameter should also take into account of the physical memory and the runtime cost.

4 Conclusion

In this paper, we present an ant classification model for dynamic semi-supervised classification. It simulates a swarm containing varied ant colonies that will evolve with time under the rule of natural selection. Meanwhile, each generation of unlabeled instances are classified into these colonies using our proposed swarm classification method. Experimental results on a synthetic dataset demonstrate the effectiveness of our method. In the future work, we will investigate AEC and compare it with other classifiers on real-world datasets. Another interesting future line of research is to consider the scenario where labeled and unlabeled data possibly come from different distributions.

Acknowledgment. This work was supported in part by the Chinese National Natural Science Foundation under Grant nos. 61402395, 61003180, 61379066 and 61103018, Natural Science Foundation of Education Department of Jiangsu Province under contracts 13KJB520026 and 09KJB20013, Natural Science Foundation of Jiangsu Province under contracts BK2010318 and BK20140492, and the New Century Talent Project of Yangzhou University.

References

1. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary clustering. In: The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 554–560 (2006)
2. Chi, Y., Song, X., Zhou, D., Hino, K., et al.: Evolutionary spectral clustering by incorporating temporal smoothness. In: The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 53–62 (2007)
3. Wang, P., Wang, H.X., Wu, X.C., et al.: A low-granularity classifier for data streams with concept drifts and biased class distribution. IEEE Transactions on Knowledge and Data Engineering 19(9), 1202–1213 (2007)
4. Gao, J., Ding, B.L., Han, J.W., et al.: Classifying Data Streams with Skewed Class Distributions and Concept Drifts. IEEE Internet Computing 12(6), 37–49 (2008)
5. Anagnostopoulos, C., Tasoulis, D.K., Adams, N.M., et al.: Temporally adaptive estimation of logistic classifiers on data streams. Advances in Data Analysis and Classification 3(3), 243–261 (2009)
6. Kuncheva, L.I., Zliobaite, I.: On the window size for classification in changing environments. Intelligent Data Analysis 13(6), 861–872 (2009)
7. Peng, Z., Xingquan, Z., Jianlong, T., et al.: Classifier and Cluster Ensembles for Mining Concept Drifting Data Streams. In: IEEE 10th International Conference on Data Mining (ICDM), pp. 1175–1180 (2010)
8. Zhang, P., Li, J., Wang, P., et al.: Enabling fast prediction for ensemble models on data streams. In: The 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, pp. 177–185 (2011)
9. Peng, Z., Gao, B.J., Xingquan, Z., et al.: Enabling Fast Lazy Learning for Data Streams. In: IEEE 11th International Conference on Data Mining (ICDM), Vancouver, Canada, pp. 932–941 (2011)
10. Zhang, P., Gao, B.J., Liu, P., et al.: A framework for application-driven classification of data streams. Neurocomput. 92, 170–182 (2012)
11. Lines, J., Davis, L.M., Hills, J., et al.: A shapelet transform for time series classification. In: The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, pp. 289–297 (2012)
12. Masud, M.M., Woolam, C., Gao, J., et al.: Facing the reality of data stream classification: coping with scarcity of labeled data. Knowledge and Information Systems 33(1), 213–244 (2012)
13. Li, L.J., Zou, B., Hu, Q.H., et al.: Dynamic classifier ensemble using classification confidence. Neurocomput. 99(1), 581–591 (2013)
14. Jia, Y., Yan, S., Zhang, C.: Semi-Supervised Classification on Evolutionary Data. In: The 21st International Joint Conference on Artifical Intelligence, pp. 1083–1088 (2009)
15. Borchani, H., Larranaga, P., Bielza, C.: Classifying evolving data streams with partially labeled data. Intelligent Data Analysis 15(5), 655–670 (2011)

Evolutionary Ensemble Model for Breast Cancer Classification

R.R. Janghel[1,*], Anupam Shukla[2], Sanjeev Sharma[2], and A.V. Gnaneswar[2]

[1] Sagar Institute of Research Technology and Science, Bhopal, India
rrj.iiitm@gmail.com
[2] ABV- Indian Institute of Information Technology and Management Gwalior, India
{dranupamshukla,sanjeev.sharma1868,gnani0826}@gmail.com

Abstract. A major problem in medical science is attaining the correct diagnosis of disease in precedence of its treatment. For the ultimate diagnosis, many tests are generally involved. Too many tests could complicate the main diagnosis process so that even the medical experts might have difficulty in obtaining the end results from those tests. A well-designed computerized diagnosis system could be used to directly attain the ultimate diagnosis with the aid of artificial intelligent algorithms and hybrid system which perform roles as classifiers. In this paper, we describe a Ensemble model which uses MLP, RBF, LVQ models that could be efficiently solve the above stated problem. The use of the approach has fast learning time, smaller requirement for storage space during classification and faster classification with added possibility of incremental learning. The system was comparatively evaluated using different ensemble integration methods for breast cancer diagnosis namely weighted averaging, product, minimum and maximum integration techniques which integrate the results obtained by modules of ensemble, in this case MLP, RBF and LVQ. These models run in parallel and results obtained will be integrated to give final output. The best accuracy, sensitivity and specificity measures are achieved while using minimum integration technique.

Keywords: Breast Cancer, Medical Diagnostics, Pattern Recognition, Ensemble Approach, Neural Networks, MLP, Multilayer perceptron, RBF, Radial Basis Function Network, LVQ, Learning Vector Quantization.

1 Introduction

Many real life applications are so complex that they cannot be solved by the application of a single algorithm. This necessitated the need for development of algorithms by mixing two or more of the studied algorithms. The choice of algorithms depends upon the needs and characteristics of the problem. This further helps in solving the problem to a reasonably good extent and achieving higher performances.

[*] Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 8–16, 2014.
© Springer International Publishing Switzerland 2014

In this paper, we have concentrated our efforts towards solving the problem of breast cancer diagnosis. Every year in many countries, number of woman died from breast cancer is increasing. Breast cancer is the most common cancer in women in many countries in the world. One out of eight women wills diagnosis and prognosis of breast cancer in this country. Early detection is one of the best defenses against cancer [1].

The database used in analysis of the system has been taken from Wisconsin Diagnostic Breast Cancer (WDBC) from UCI Machine Learning Repository, which comprises of data vectors from 569 patients. Then, this data is divided into training and testing data by taking 398 vectors as training data set (about 70% of the total data set) and rest as testing data set (about 30% of the total data set).

This paper is organized as follows. Section 2 reviews related work done in the concerned field. Section 3 gives the methodology used in tackling the problem. Experimental results are presented in Section 4. Conclusion and future work are given in the last section.

2 Related Work

A classification system is one that actually maps input vectors to a specific class. Hence, classification is basically the job of learning the procedure that maps the input data [2]. This has, in turn, has enthused researchers to replicate this success in the field of medical diagnostics. Their efforts have bore significant gains through the application of several standards and techniques of pattern recognition to the said problem [3]. Also it is the most widespread form of cancer among women in the world. Early detection is one of the best defenses against cancer. According to the American Cancer Society (ACS), after every thirteen minutes, four American women develop breast cancer, and one woman dies from breast cancer [1, 4-6].

Yao and Liu et.al described neural network based approaches to breast cancer diagnosis, which had displayed good generalization. The approach was based on artificial neural networks. In this approach, a feed forward neural network was evolved using BP algorithm [12]. Fogel et al. were first to derive technique to model neural networks for solving breast cancer classification [13].

Rahul et al. used multilayer perceptron neural networks (MLPNNs), radial basis function network (RBFN), competitive learning network (CL), learning vector quantization network(LVQ), combined neural networks (CNNs), probabilistic neural networks(PNNs), and recurrent neural networks (RNNs) for breast cancer diagnosis [14].

The artificial immune system with the GA in one hybrid algorithm which is the clonal selection algorithm was inspired from the clonal selection principle and affinity maturation of the human immune responses by hybridizing it with the crossover operator, which is imported from GAs to increase the exploration of the search space. [13].

Contrary to neural networks, clustering, rule induction and many other machine learning approaches, Genetic Algorithms (GAs) provide a means to encode and evolve rule antecedent aggregation operators, different rule semantics, rule base aggregation operators and defuzzification methods. Therefore, GAs remain today as one of the few and, in some sense, optimize fuzzy systems with respect to the design decisions, allowing decision makers to decide what components are fixed and which ones evolve according to the performance measures [14]. Carlos Andres Pena-Reyes et.al proposed a fuzzy-genetic approach produces systems exhibiting two prime characteristics: first they attain high classification performance and second the resulting systems involve a few simple rules and gave 97.50 % classification accuracy [15]. The goal of Fuzzy CoCo model was to evolve a fuzzy model that describes the diagnostic decision and the classification performance was 98.98%. [16].

F A good collection of methods and applications can be found in the books by Mellin and Castillo [10], and Bunke and Kendel [11, 12].

3 Methodology

Pattern Recognition and Machine Learning field have established research work on the combination of multiple classifiers (also known as ensemble of classifiers, Mixture of experts). Overall predictive accuracy can be increased by the use of multiple classifiers instead of a single classifier. The ensemble procedure constitute two steps mainly module formation and then integration of results of modules. Firstly we need to formulate the number of modules to be used, that constitute the entire ensemble architecture. Decision towards the model and architectural parameters of each of the module is made. All the networks may be initialized in this mechanism. Next the entire ensemble needs to be trained, which means the training up of the individual models making up the ensemble.

Each of the modules is trained independently and in-parallel by all the training data present in the system.

The ANNs with BPA still have some shortcomings. It is quite likely that BPA results in some local minima in place of global minima. Also we need to specify the initial parameters before the learning starts. These pose restrictions on the use of ANNs. The GA on the other hand is known for its ability of optimization. In this section we will fuse this capability of the GA along with the ANNs to train the ANN.

This solution overcomes much of the problems with the ANN training.

The block diagram of the proposed system for breast cancer diagnosis is shown in Figure 1.

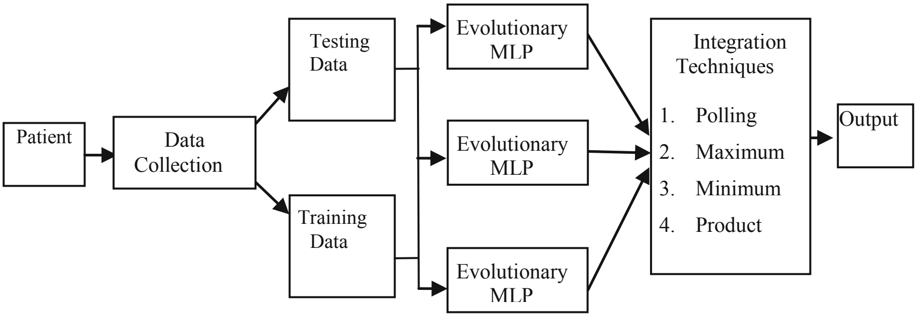


Fig. 1. Block Diagram of the Proposed System for Breast Cancer Diagnosis

In this paper, we have used the ensemble approach for classifying the inputs as malignant or benign which has 3 modules. Here, each module has evolutionary ANN and the difference between them is change in hidden neurons.

Here the GA is supposed to fix the values and the various weights as well as biases that exist in the neural networks. The GA in other words optimizes the network parameters for better performance. An ANN is a collection of various neurons. These neurons are arranged in a layered manner. Any ANN model being used in real life application normally uses a single hidden layer. The hidden layer has a specified number of neurons in it. In a fully connectionist approach, every neuron of a forward layer is connected to every neuron of the forward layer i+1 by some weight. The hidden and output neuron further have weight is adjusted during training. Besides every neuron has some bias associated with it. Now we would study the application of GA in this problem for training. Some biases that need to be optimally set.

The first task is problem encoding. The problem encoding consists of these parameters in a linear array. This is the phenotype problem representation. The population may be represented using any of double vector or a bit string representation. The Genetic Operators include Selection, Crossover, Eliticism, Mutation, etc. The Genetic Operators ensured creation of good individuals from one population to the other. Let us assume that there was a single hidden layer consisting of H neurons. The input and output layers have I an O neurons respectively. In this system, it may easily be seen that there are I x H weights between the input layer and the hidden layer and H x O weights between the hidden layer and the output layer, this makes the total number of weights as W=I x H + H x O. Further the number of biases is equal to the number of neurons. The total number of biases is H + O. This means that for a single layer ANN there would be I x H + H x O + H + O parameters to be optimized.

The fitness of any individual in the population is measured with the help of fitness function. The fitness function consists of the ANN along with its training data set. In the fitness function we initialize the ANN by the various parameters that are generated by GA. These parameters were extracted from the individual and used to set the weights and biases of the ANN. Then the training data set is passed through the ANN. The performance of the ANN against this data set is measured. This performance is the net fitness value of the GA that needs to be maximized (or the negative performance need to be minimized).Hence every time that the GA demands the measurement of fitness value of some individual, the ANN is created and the value is measured by the performance. This interfaces the GA and the ANN while training.

The neural training by GA possesses a very complex fitness landscape. Hence it is wise to use a local search strategy that places any ANN or genetic individual at the closest minima, before its fitness value is reported. This local search strategy assists the GA in the search or optimization process. In this algorithm we use Back Propagation Algorithm as the local search method. The epochs, momentum, and learning rate are kept low as per the requirements of local search.

Once the GA reaches its optimal state and terminates as per the stopping criterion, we get the final values of the weights and the biases. Then we create the ANN with these weights and bias values and this is regarded as the most optimal ANN as a result of the ANN training. We can then use this for the testing purposes. It may be seen here that validation data is not necessarily required in this type of training.

The net fitness may hence be given by equation

$$\text{Fit}(N) = P(N) - \alpha C(N)$$

Here N is the genetic individual or ANN, α is the penalty constant, Fit() is the fitness function, P() is the performance function, C() is the number of connections.

Methods for Response Integration

Here, we use different integration schemes including polling, maximum, minimum, weighed average. These schemes are used to integrate the outputs from each of the four networks separately and the resulting detection accuracies measured. In polling scheme, each network returns the class that it considers the one to which the input belongs. After taking these classes, voting takes place between the network modules. The class with the highest votes is taken as the winner. In weighted average scheme, mean of matching scores of all the networks is taken.

The integrator receives all the probability vectors and does the task of deciding the final output of the system. For this if probability is greater than 0.5 it is marked as malignant, otherwise as benign.

4 Simulation Results

Wisconsin Diagnostic Breast Cancer (WDBC) database of UCI Machine Learning Repository is used for the experimentation of our model. The goal is to classify a tumor as either benign or malignant based on cell descriptions gathered by FNA image test. The Breast Cancer data set has vectors with a total of 30 input attributes. This database contains information about 569 patients with 212 out of 569 having malignant tumors. Attributes used here are radius mean of distances from center to points on the perimeter, texture means standard deviation of gray-scale values, smoothness means local variation in radius lengths, perimeter, area, smoothness (local variation in radius lengths), compactness (perimeter2 / area - 1.0), concavity (severity of concave portions of the contour), concave points (number of concave portions of the contour), symmetry and fractal dimension (coastline approximation - 1).They are measured for a total of 3 cells. The various integration methods are compared with respect to their ability to train, learn and generalize the data. One with the best generalizing capacity will give the best detection efficiency.

First we divide the data set into training and testing sets at 70% and 30% by taking 398 vectors as training data set and rest as testing data set. Then data set is used to train and the test the ensemble model.

The results are measured against the TP (true positive), TN(true negative), FP(false positive) and FN (false negative). The various performance measures are summarized in the table 1.

Table 1. Diagnostic performance measures Breast cancer

Cancer Test	**Present**	**Absent**	**Total**
Positive	True Positive [TP]	False Positive [FP]	[TP +FP]
Negative	False Negative [FN]	True negative [TN]	[FN+TN]
Total	(TP + FN)	(TN + FP)	(TP + FN+ TN + FP)
Sensitivity		TP / (TP + FN)	
Specificity		TN / (TN + FP)	
Accuracy		(TP + TN) / (TP + TN + FP + FN)	

We run the Evolutionary ANN modules to obtain optimum weights for ANN. We applied GA for the parameter optimization. The weight matrix consisted of 30*x weights between input and hidden layer, x*1 between the hidden and the output layer and a total of 18, 20, 25 hidden layer biases and 1 output layer bias. This made the total number of variables for the GA as 30*x + x*1 + x + 1. We use 18, 20, 25 hidden neurons for each module respectively.

In GA, the double vector method of population representation was used. The total number of individuals in the population was 50. A uniform creation function, rank based scaling function and stochastic uniform selection methods were used. The elite count was 2. Single point crossover was used. The program was executed till 100 generations. The crossover rate was 0.7, Best fitness is 98.78 and mean fitness is 96.61.The best performance in terms of sensitivity, specificity, accuracy, false negative and false positive are 98.70% 97.42%, 98.24%, 4.6% and 0.65% for testing respectively.

Here the results of GA are exported which are optimized weights and bias of ANN and ANN is run for 10,000 epochs. The result of this is passed through various integrators.

We then experiment ensemble model with various integration methods to find an optimized parameter which gives best performance. After getting the optimized parameter, the detection procedure is run 20 times for the same configuration. After this, the mean and standard deviation are computed. The mean is taken as the performance accuracy of the system for training and testing dataset.

The results show that the maximum accuracy was achieved when using maximum integrator with Accuracy of 99.07% along with sensitivity, specificity, FPR and FNR values as 98.79, 99.01%, 1.23%, 0.65% respectively. Figure 2 shows the spread of values of Evolutionary ANN module.

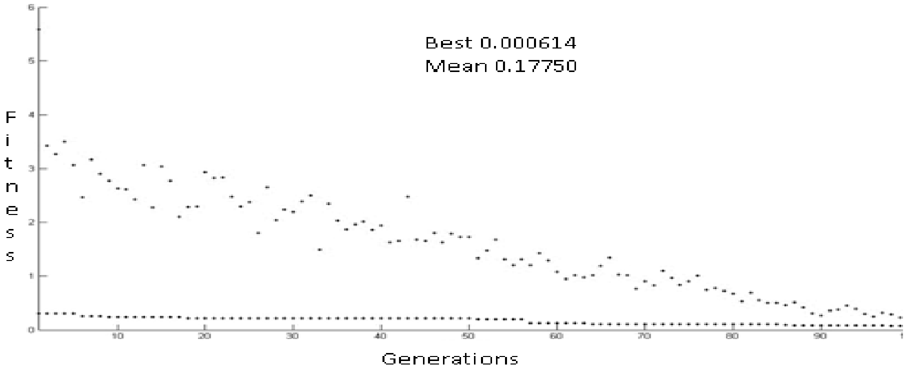


Fig. 2. Performance of Evolutionary ANN module

Table 2. Diagnostic performance of various integration techniques

Integration Techniques	S (%)	Sp (%)	A (%)	FPR (%)	FNR (%)
Maximum	**98.79**	**99.01**	**99.07**	**1.23**	**0.65**
Minimum	98.46	99.01	98.79	1.53	0.98
Polling	99.00	96.47	97.36	4.9	1.56
Sum	98.70	97.42	98.24	5.0	0.65

Now we compare our model performance with Ensemble architecture with same ANN model in all the modules. The only difference in the modules is number of hidden layers. Table 3 is given with the best results of ensemble with same modules on various integration techniques.

Here we used hidden layer of 18, 20, 25 for the 3 modules of ensemble while keeping other parameters same as that in our proposed model. Figure 3 shows the comparative analysis.

Table 3. Diagnostic performance of various integration techniques

ANN Models	Integration Techniques	Testing performances (%)		
		S (%)	Sp (%)	A (%)
MLP	Minimum	99.00	96.47	97.36
RBF	Maximum	97.01	98.05	97.17
LVQ	Polling	93.22	96.04	95.00

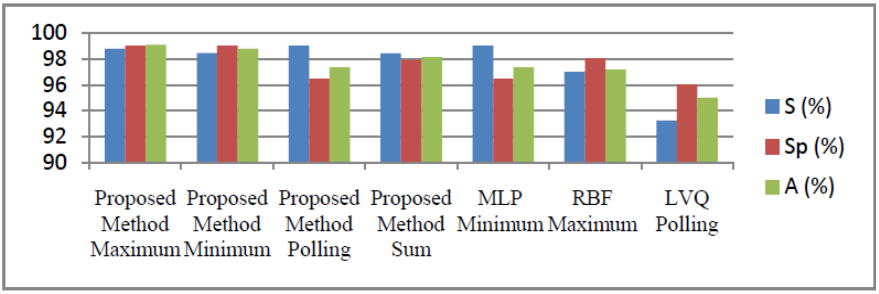


Fig. 3. Comparison of Multi Model ANN with Same ANN ensemble with different integrators

From the experimental results we can show that our proposed method performs better than that of ensemble of ANN with same ANN modules.

5 Conclusion and Future Work

In this paper we saw the working of different ensemble integration methods with three modules of Evolutionary ANN model using MLP model for detection of Breast Cancer. In all the cases we were able to solve the problem with fine accuracies using the different ensemble integration methods. The results show that, Maximum integration gives the most favorable performance when compared with other integration methods.

References

1. American Cancer Society, Cancer Facts and Figures (2011-2012)
2. Weiss, S.I., Kulikowski, C.: Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Networks, Machine Learning and Expert Systems. Morgan Kaufmann Publishers (1991)
3. Coleman, T.F., Li, Y.: Large Scale Numerical optimization. In: Proceedings of Workshop on Large Scale Numerical Optimization, Cornell University, New York (1989)
4. Andolina, V.F., Lille, S.L., Willison, K.M.: Mammographic Imaging: A Practical Guide, New York (1992)
5. Antani, S., Lee, D.J., Long, L.R., Thoma, G.R.: Evaluation of shape similarity measurement methods for spine X-ray images. Journal of Visual Communication & Image Representation 15, 285–302 (2004)
6. Clemen, R.: Combining forecasts: A review and annotated bibliography. International Journal of Forecasting, 559–583 (1989)
7. Yao, X., Liu, Y.: Neural networks for breast cancer diagnosis. In: Proceedings of the Congress on Evolutionary Computation, vol. 3, pp. 1767–1773 (August 2002)
8. Fogel, D.B., Wasson, E.C., Boughton, E.M., Porto, V.W., Angeline, P.J.: Linear and neural models for classifying breast masses. IEEE Transactions on Medical Imaging 17(3), 485–488 (1998)

9. Rahul, K., Anupam, S., Ritu, T., Janghel, R.R.: Breast cancer diagnostic system using artificial neural networks model. In: International Conference on Information Sciences and Interaction Sciences (ICIS), pp. 89–94 (2010)
10. Melin, P., Castillo, O.: Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing. STUDFUZZ, vol. 172. Springer, Heidelberg (2005)
11. Bunke, H., Kandel, A. (eds.): Hybrid Methods in Pattern Recognition. World Scientific (2002)
12. Sivanandan, S.N., Deepa, S.N.: Principle of soft computing. Wiley India Private Limited (2007)
13. Nabil, E., Badr, A., Farag, I.: An Immuno-Genetic Hybrid Algorithm. Int. Journal of Computers, Communications & Control IV(4), 374–385 (2009)
14. Alcala, R., Nojima, Y.: Special issue on genetic fuzzy systems: new advances. Evolutionary Intelligence 2, 1–3 (2009)
15. Pena-Reyes, C.A., Sipper, M.: A fuzzy-genetic approach to breast cancer diagnosis. Artificial Intelligence in Medicine, 131–155 (1999)
16. Peña Reyes, C.A.: Breast Cancer Diagnosis by Fuzzy CoCo. In: Peña Reyes, C.A. (ed.) Coevolutionary Fuzzy Modeling. LNCS, vol. 3204, pp. 71–87. Springer, Heidelberg (2004)

Empirical Analysis of Assessments Metrics for Multi-class Imbalance Learning on the Back-Propagation Context[*]

Juan Pablo Sánchez-Crisostomo[1], Roberto Alejo[1], Erika López-González[1],
Rosa María Valdovinos[2], and J. Horacio Pacheco-Sánchez[3]

[1] Tecnológico de Estudios Superiores de Jocotitlán, Carretera Toluca-Atlacomulco KM. 44.8,
Col. Ejido de San Juan y San Agustín, 50700 Jocotitlán, México
[2] Faculty of Engineering, Universidad Autónoma del Estado de México Cerro de Coatepec s/n,
Ciudad Universitaria C.P. 50100, Toluca, Estado de México
[3] Instituto Tecnológico de Toluca
Av. Tecnológico s/n Ex-Rancho La Virgen, 52140, Metepec, México

Abstract. In this paper we study some of the most common assessment metrics employed to measure the classifier performance on the multi-class imbalanced problems. The goal of this paper is empirically analyzing the behavior of these metrics on scenarios where the dataset contains multiple minority and multiple majority classes. The experimental results presented in this paper indicate that the studied metrics might be not appropriate in situations where multiple minority and multiple majority classes exist.

Keywords: Metrics, Multi-class Imbalance, Multiple Minority and Majority Classes.

1 Introduction

Class imbalance problems have drawn growing interest recently because of their classification difficulty caused by the imbalanced class distributions [8]. So, it has been into the 10 challenging problems identified in data mining research [9]. The class imbalance problem appears when in a training data set the number of instances in at least one class is much less than the samples in another class or classes [6]. Much work has been done in addressing the class imbalance problem [4] but, while two-class imbalance problem has been widely studied the *multi*-class imbalance problem has been relatively less investigated [8].

The multi-class imbalance problems pose new challenges, for instance, in task of assessments classifier performance it is necessary to apply different metrics than those used in traditional two-class classification problems [7]. On two-class imbalance problems, sometimes it is possible to provide more appropriate assessments metrics to measure the classifier performance, but in the multi-class imbalance problems, it is extremely difficult to provide realistic assessments of the relative severity of the classification performance [3].

[*] This work has been partially supported under grants of: PROMEP/103.5/12/4783 from the Mexican SEP and SDMAIA-010 of the TESJO.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 17–23, 2014.
© Springer International Publishing Switzerland 2014

Often in research where the multi-class imbalance is the focus of study, the authors use metrics that have been extended from two-class imbalance scenarios to compare the classifier performance, for example the geometric mean, F-measure or measures of the area under curve family [2]. However, this situation presents an interesting research question: The proposed metrics to assessment the classifier performance are appropriate on scenarios where exist multiple minority and multiple majority classes?.

We are interested in this question, so, in this paper we study the behavior of seven the most common multi-class assessment metrics on real multi-class databases with minority and majority multiple classes.

2 Assessments Metrics for Multi-class Imbalance Learning

The most studied metrics for assessment the classifier performing in class imbalance domains have been focused a two class imbalance problems and some of them have been modified to accommodate them at the multi-class imbalanced learning problems [4]. In this section we present some of the most common two-class imbalance metrics adapted at multi-class imbalance scenarios.

Macro Average Geometric (MAvG): This is defined as the geometric average of the partial accuracy of each class.

$$MAvG = (\prod_{i=1}^{J} ACC_i)^{\frac{1}{J}} , \tag{1}$$

where $ACC_j = (\ correctly\ classified\ of\ class\ j)/(\ total\ of\ samples\ of\ classj)$, i.e., the accuracy on the class j. J is the number of classes.

Mean F-measure (MFM): This measure has been widely employed in information retrieval

$$F - \text{measure}(j) = \frac{2 \cdot recall(j) \cdot precision(j)}{recall(j) + precision(j)} , \tag{2}$$

where $recall(j) = (correctly\ classified positives)/(total\ positives)$ and $precision\ (j) = (correctly\ classified\ positives)/(total\ predicted\ as\ positives)$; j is the index of the class considered as $positive$. Finally, mean F-measure is defined for multi-class in Reference [2] as follow:

$$MFM = \sum_{j=1}^{J} \frac{FMeasure(j)}{J} . \tag{3}$$

Macro Average Arithmetic (MAvA): This is defined as the arithmetic average of the partial accuracies of each class.

$$MAvA = \frac{\sum_{i=1}^{J} ACC_i}{J} . \tag{4}$$

One the most widely used techniques for the evaluation of binary classifiers in imbalanced domains is the Receiver Operating Characteristic curve (ROC), which is a tool for visualizing, organizing and selecting classifiers based on their trade-offs between true positive rates and false positive rates. Furthermore, a quantitative representation of a ROC curve is the area under it, which is known as AUC [1]. The AUC measure has been adapted at multi-class problems [2] and can be defined as follow.

AUC of each class against each other, using the uniform class distribution (AU1U):

$$AU1U = \frac{2}{\|J\|(\|J\|-1)} \sum_{j_i, j_k \epsilon J} AUC_R(j_i, j_k) \ , \tag{5}$$

where AUC$_R(j_i, j_k)$ is the AUC for each pair of classes j_i and j_k.

AUC of each class against each other, using the a priori class distribution (AU1P):

$$AU1P = \frac{2}{\|J\|(\|J\|-1)} \sum_{j_i, j_k \epsilon J} p(j) AUC_R(j_i, j_k) \ , \tag{6}$$

where $p(j)$ is a priori class distribution.

AUC of each class against the rest, using the uniform class distribution (AUNU):

$$AUNU = \frac{1}{J} \sum_{j \epsilon J} AUC_R(j, rest_j) \ , \tag{7}$$

where $rest_j$ gathers together all classes different from class j, i.e., the area under the ROC curve is computed in the approach one against all.

AUC of each class against the rest, using the a priori class distribution (AUNP):

$$AUNP = \frac{1}{J} \sum_{j \epsilon J} p(j) AUC_R(j, rest_j) \ , \tag{8}$$

this measure takes into account the prior probability of each class $(p(j))$.

3 Experimental Protocols

3.1 Database Description

In this section we describe briefly the two databases (92AV3C and ALL-DATA) used in our experimentation. 92AV3C corresponds to a hyperspectral image (145 x 145 pixels) taken over Northwestern Indianas Indian Pines by the AVIRIS sensor[1]. For simplicity, in this paper we use only 38 attributes from the 220 attributes of the original dataset. The attributes were selected using a common features selection algorithm (Best-First Search [5] implemented in WEKA.[2]

ALL-DATA consists of the reflectance values of image pixels that were taken by the Compact Airborne Spectrographic Imager (CASI) and the Airborne Hyper-spectral Scanner (AHS) sensors. Corresponding chlorophyll measurements for these pixels were

[1] engineering.purdue.edu/biehl/MultiSpec/hyperspectral.html

[2] www.cs.waikato.ac.nz/ml/weka/

also performed. CASI set consists of the reflectance values of image pixels that were taken by the CASI sensor. Corresponding thermal measurements for these pixels were also made. The CASI sensor reflectance curves are formed by 144 bands between 370 and 1049 nm. AHS images consist of 63 bands between 455 and 2492 nm. Therefore, the input dimensionality of this dataset is 207 (the sum of the bands corresponding to the CASI and AHS sensors).

Table 1. The class distribution of the 92AV3C and ALL-DATA datasets is presented in this table. Size represents the number of samples on each class.

Database	Class	Size	Database	Class	Size
92AV3C	0	95	ALL-DATA	1	8258
	1	489		2	4588
	2	834		3	11346
	3	968		4	4751
	4	54		5	1123
	5	614		6	5762
	6	497		7	3020
	7	1294		8	7013
	8	380		9	15
	9	26		10	79
	10	234		12	82
	11	20		13	222
	12	1434		14	1733
	13	2468		15	3628
	14	747		–	–
	15	212		–	–
	16	10659		–	–

In order to study the multi-class assessment metrics behavior in domains of multiple minority classes and multiple majority classes we split the original datasets (92AV3C and ALL-DATA) in subsets (Si, Ai and Bi).

For 92AV3C dataset the Si subsets were integrated in the following way: the first subset (S1) contains the four more minority classes (3, 7, 12 and 13) and the most majority class (16) of the original dataset. The next subset (S2) was integrated with the union of S1 with the next minority class (not used before, class 2). This process finishes when all minority have been integrated at one subset.

The ALL-DATA was split in subsets as follow: the first subset A1 was integrated by the more minority classes (9, 10, 12 and 13) of ALL-DATA dataset, the next subset (A2) corresponds to join of A1 with the class 1, i.e., A1$\cup$1. The rest of subsets were made of the same way.

The Bi subsets were integrated as follow: B1 contains eight majority classes (1–8) of ALL-DATA dataset, B2 $=$ B1 $\cup$ with the next minority class (class 9), and the rest of Bi subsets were integrated using this process. The Table 2 shows a brief summary of the integration of Si, Ai and Bi subsets.

In 92AV3C subsets (Si) the 10–fold cross–validation was applied. The datasets were divided into ten equal parts, using nine folds as training set and the remaining block test set. ALL-DATA subsets (Ai and Bi) were split in disjoints subsets: training (50% of the samples) and test (50% of the samples).

Table 2. Classes that ingrate the Si, Ai and Bi subsets

Subset	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
Classes	3,7,12,13,16	S1∪ 2	S2∪14	S3∪5	S4∪6	S5∪1	S6∪8	S7∪10	S8∪15	S9∪0	S10∪4	S11∪9	S12∪11
Subset	A1	A2	A3	A4	A5	A6	A7	A8	A9	–	–	–	–
Classes	9,10,12,13	A1∪ 1	A2∪2	A3∪3	A4∪4	A5∪5	A6∪6	A7∪7	A8∪8	–	–	–	–
Subset	B1	B2	B3	B4	B5	–	–	–	–	–	–	–	–
Classes	1,2,3,4,5,6,7,8	B1∪9	B2∪10	B3∪12	B4∪13	–	–	–	–	–	–	–	–

3.2 Neural Network Configuration

In the experimental phase we use the MLP trained with the standard back-propagation in sequential mode. For each training data set, MLP was initialized ten times with different weights, i.e., the MLP was run ten times with the same training dataset. The results here included correspond to the average of those accomplished ten different initialization and of ten partitions for 92AV3C, and only the average of ten different initializations for ALL-DATA. The learning rate (η) was set at 0.1 and only one hidden layer was used. The stop criterion was established at 5000 epoch or an MSE below to 0.001. The number of neurons (n) for the hidden layer was fixed as $n =$ number of classes $+1$, because our goal it is not to find the optimal MLP configuration but to study the assessment metrics behavior.

4 Experimental Results

In order to assessment the multi-class imbalance metrics: $MAvG$, $AUNP$, $AU1P$, MFM, $AUNU$, $MAvA$ and $AU1U$ (see section 2), we have carried out an experimental comparison over twenty seven datasets with multiple minority and multiple majority classes (see Table 2).

The tables 3 and 4 present the experimental results. The first column represents the dataset used (Si, Ai or Bi), the next columns exhibit the metrics used and at least one the number of minority classes do not classified, i.e., ignored by the classifier. The rows show the values obtained from different metrics on each dataset.

Tables 3 and 4 show an interesting behavior. Observe that in some datasets all metrics (except $MAvG$) present very similar results but, in these datasets the classifier does not classify or ignored different minority classes in each dataset. For example, in Table 3 the values the $AUNP$ for the S10 and S11 datasets are 0.717557 and 0.718413 respectively, i.e., they are very similar values. However, in S10 the classifier does not classify two minority classes and S11 does not classify one class. In other words, the values the $AUNP$ for S10 and S11 are very similar but in S10 the classifier ignored more classes

Table 3. Classification performance of the subsets (Si) obtained from 92AV3C dataset (see Table 2) measured by the metrics: $MAvG$, $MAvA$, $AU1U$, $AU1P$, $AUNU$, MFM and $AUNP$

Metric	$MAvG$	$AUNP$	$AU1P$	MFM	$AUNU$	$MAvA$	$AU1U$	No. of classes ignored by the classifier
S1	0.000000	0.680849	0.680849	**0.296549**	0.606591	0.349023	0.349023	3
S2	0.000000	0.864575	0.864575	**0.298384**	0.637522	0.373081	0.373082	2
S3	0.000000	0.843382	0.843382	0.326805	0.700919	0.524812	0.524812	1
S4	0.000000	0.775310	0.775310	0.272290	**0.627219**	0.442667	0.442667	1
S5	0.000000	0.763563	0.763564	0.260260	0.624413	0.448988	0.448988	1
S6	0.000000	0.755563	0.755563	0.245079	0.645932	0.500241	0.500241	2
S7	0.000000	0.745715	0.745714	0.227627	0.628405	0.478391	0.478391	2
S8	0.000000	0.738600	**0.738599**	0.212890	**0.626571**	**0.486770**	**0.486770**	3
S9	0.000000	0.731669	0.731668	0.197799	0.628791	0.499175	0.499175	1
S10	0.000000	0.737835	**0.737835**	0.193418	0.655136	0.549927	0.549927	2
S11	0.000000	**0.717557**	0.717558	0.170630	0.615831	**0.487472**	**0.487472**	2
S12	0.000000	**0.718413**	0.718413	0.162070	0.626332	0.509208	0.509208	1

Table 4. Classification performance of ALL-DATA dataset measured by the metrics: $MAvG$, $MAvA$, $AU1U$, $AU1P$, $AUNU$ and $AUNP$

Metric	$MAvG$	$AUNP$	$AU1P$	MFM	$AUNU$	$MAvA$	$AU1U$	No. of classes ignored by the classifier
A1	0.995583	0.995543	0.995543	0.984598	0.995823	0.995625	0.995625	0
A2	0.601254	0.915319	0.915319	0.300290	0.844969	0.796271	0.796271	0
A3	0.408697	**0.775386**	**0.775386**	**0.271673**	**0.752187**	0.760755	0.760755	0
A4	0.000000	0.747106	0.747106	0.289039	0.716155	0.682684	0.682684	1
A5	0.000000	0.803038	0.803038	0.336305	**0.751786**	0.699337	0.699337	1
A6	0.000000	**0.833518**	**0.833518**	**0.350768**	0.741196	0.649088	0.649088	1
A7	0.000000	0.734872	0.734872	0.263121	0.659345	0.580251	0.580251	2
A8	0.000000	0.708736	0.708736	0.234921	0.629486	0.546548	0.546549	2
A9	0.000000	0.706589	0.706589	0.230379	0.596762	0.483704	0.483704	3
B1	0.648743	0.750903	0.750903	0.391792	0.748725	0.744061	0.744061	0
B2	0.000000	0.730440	0.730440	0.320462	0.676222	0.618496	0.618496	1
B3	0.000000	0.735042	0.735042	0.301764	0.667699	0.598593	0.598593	2
B4	0.000000	**0.708683**	**0.708683**	0.246963	0.609454	0.506425	0.506425	2
B5	0.000000	**0.706589**	**0.706589**	0.230379	0.596762	0.483704	0.483704	3

than S11. Similar situations were observed in $AU1P$ with S8 and S10, MFM with S1 and S2, $AUNU$ with S4 and S8, $MAvA$ with S8 and S11, and $AU1U$ with S8 and S11 (see Table 3). On Table 4 this behavior was observed in $AUNU$ and $AU1P$ with B4 and B5. $AUNU$ with A3 and A5.

A dramatic situation was noticed in Table 4, we observe that in some datasets the classifier presents better results when does not classify one or more classes that when it classify all classes. For example, the values for A3 and A6 with $AUNP$ are 0.775386 and 0.833518, respectively, i.e., the result the A6 is better than the A3 result, but in A6 one class is ignored for the classifier meanwhile that in A3 all classes are identified for it. This behavior was adviced too in the metrics $AU1P$ and MFM for these datasets (A3 and A6).

On the other hand, the $MAvG$ could be more appropriate in classification problems with multiple majority classes and multiple minority classes, because it notice when the classifier ignores any class (see Table 4).

5 Conclusions

In this paper we study some of the most common metrics employed to measure the classifier performance on the multi-class imbalanced problems. We focused in problems with multiple minority classes and multiple majority classes. So, some experiments have been carried out over twenty seven real data sets using a multilayer perceptron trained with the back-propagation algorithm.

From the analysis of the experimental results in this work, we might suggest that the main problem of the assessment metrics studied in this paper (except $MAvG$), is that they were designed to provide an *average* performance of the pairs of classes, so this metrics, in some cases, do not provide information when one or more classes are ignored for the classifier.

We think, therefore, that they might not be appropriate when the dataset contains multiple minority classes and multiple majority classes, in other words these metrics might not be appropriate in muti-class imbalance context as the *accuracy* was in two-class imbalance problems. However, the $MAvG$ could be more appropriate in this scenario because it notice when the classifier ignores any class.

The assessment metrics were developed with different proposes and goals, nevertheless, in the literature the researchers use they to compare the classifier performance, for this reason we consider is necessary a deeper study about of this problem than the previous one.

References

1. Fawcett, T.: An introduction to roc analysis. Pattern Recogn. Lett. 27, 861–874 (2006)
2. Ferri, C., Hernández-Orallo, J., Modroiu, R.: An experimental comparison of performance measures for classification. Pattern Recognition Letter 30(1), 27–38 (2009)
3. Hand, D.J., Till, R.J.: A simple generalisation of the area under the roc curve for multiple class classification problems. Machine Learning 45, 171–186 (2001)
4. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263–1284 (2009)
5. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. 97(1-2), 273–324 (1997)
6. Ou, G., Murphey, Y.L.: Multi-class pattern classification using neural networks. Pattern Recognition 40(1), 4–18 (2007)
7. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. Int. J. Data Warehousing and Mining, 1–13 (2007)
8. Wang, S., Yao, X.: Multi-class imbalance problems: Analysis and potential solutions. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 42(4), 1–12 (2012)
9. Yang, Q., Wu, X.: 10 challenging problems in data mining research. International Journal of Information Technology and Decision Making 5(4), 597–604 (2006)

A Novel Rough Set Reduct Algorithm to Feature Selection Based on Artificial Fish Swarm Algorithm

Fei Wang, Jiao Xu, and Lian Li

School of Information Science & Engineering, Lanzhou University,
730000 Lanzhou, China
{wangf12,xujiao12}@lzu.edu.cn

Abstract. With the purpose of finding the minimal reduct, this paper proposes a novel feature selection algorithm based on artificial fish swarm algorithm (AFSA) hybrid with rough set (AFSARS). The proposed algorithm searches the minimal reduct in an efficient way to observe the change of the significance of feature subsets and the number of selected features, which is experimentally compared with the quick reduct and other hybrid rough set methods such as genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO) and chaotic binary particle swarm optimization (CBPSO). Experiments demonstrate that the proposed algorithm could achieve the minimal reduct more efficiently than the other methods.

Keywords: feature selection, rough set, fish swarm algorithm, ant colony optimization, chaotic binary particle swarm optimization.

1 Introduction

Feature selection is the process of choosing a good subset of relevant features and eliminating redundant ones from an original feature set, which can be perceived as a principal pre-processing tool for solving the classification problem [1]. The main objective of feature selection is to find a minimal feature subset from a set of features with high performance in representing the original features [2]. In classification problems, feature selection is a necessary step due to lots of irrelevant or redundancy features. By eliminating these features, the dimensionality of feature can be reduced and the predictive performance can be improved for classification. Feature selection methods are dimensionality reduction methods often associated to data mining tasks of classification [3], which provide a reduced subset of the original features while preserving the representative power of the original features.

Rough set (RS) was proposed by Pawlak, which provides a valid tool that can be applied for both feature selection and knowledge discovery. It has been proved to be an effective feature selection approach, which can select a subset of features while preserving the meaning of the features, therefore it can predict the classification accuracy as well as the original feature set. The essence of rough set to feature selection are to find a minimal subset of the original features with the most

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 24–33, 2014.
© Springer International Publishing Switzerland 2014

informative features and remove all other attributes from the feature set with minimal information loss [4]. Rough set is a powerful mathematical tool to reduce the number of features based on the degree of dependency between condition attributes and decision attributes, which has been widely applied in many fields such as machine learning and data mining. Though rough set has been used as a feature selection method with much success, it is inadequate at finding optimal reduct because of no perfect search techniques.

In order to find the optimal reduct and improve the performance, a variety of search techniques hybrid with rough set are introduced to address feature selection problems such as genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO). These swarm intelligence based algorithms such as particle swarm and ant colony optimization have been proved to be competitive in rough set attribute reduction fields. However, these algorithms have some disadvantages such as premature convergence in PSO and the performance of the reduct depending on initial parameters in ACO. In this paper, we propose a novel feature selection algorithm based on artificial fish swarm algorithm hybrid with rough set, which is not sensitive to initial parameters, has a strong robustness and has the faster convergence speed to find the minimal reduct subset.

2 Rough Set Theory

Rough set theory is an extension of traditional set theory that provides approximations in decision making, in which attribute reduction provides a valid method to extract knowledge from feature set in a concise way. In this paper, we adopt some relevant concepts of rough set theory related to our attribute reduction approach in [5] such as equivalence relation, lower approximation, positive region, and degree of dependency.

Definition of Core. The elements of feature core are those features that cannot be eliminated. In this paper, the algorithm for finding feature core is as follows: initialize $Core = \varnothing$; for every attribute $a \in C$, if $\mu_{C-\{a\}}(D) < \mu_C(D)$, then attribute a is one element of feature core, namely $Core = Core \cup \{a\}$. Where $\mu_C(D)$ represents the degree of dependency between condition attributes *C* and decision attribute *D*.

The quick reduct (QR) algorithm proposed in [6], attempts to obtain a reduct without exhaustively generating all possible subsets. It starts from an empty set, adds one attribute at a time until it generates its maximum value for the dataset.

3 Swarm Intelligence Based Rough Set Reduct Algorithm

3.1 Ant Colony Optimization Based Reduct Algorithm (ACORS)

ACO is considered a new meta-heuristic algorithm that is used successfully to solve many NP-hard combinatorial optimization problems [7]. In ACO, a swarm of artificial ants cooperate for finding good solutions to optimization problems. Every ant searches for optimal solutions in the problem space, which has a start state and one or more end conditions. The next move is determined by a probabilistic transition rule that is a function of locally available pheromone trails. Once ant has constructed a solution, then it updates the pheromone trial values which depend on the quality of solutions constructed by the ants. Finally, the ant constructs the optimal solution with the higher amount of pheromone trails. The algorithm stops iterating when an end condition is satisfied. The search for the optimal feature subset is a traversal through the graph where a minimal number of nodes are visited and the end conditions are satisfied [8]. This algorithm performs as follow: All ants start from feature core, each ant builds a solution and then the pheromone trials for every ant are updated.

Pheromone Trials and Heuristic Information. In the step, each edge is assigned a pheromone trail and heuristic information. Firstly, the initial pheromone trial on each edge is initialized to equal amount of pheromone. Secondly, each ant constructs a solution; after that, the pheromone of each edge in this solution is updated. In ACORS, the heuristic information is on the basis of the degree of dependency between the two attributes and decision attribute. The value of heuristic information η is limited in this paper, If $\eta(a,b) < \varepsilon$, then $\eta(a,b) = \varepsilon$, where ε is set to 0.001. Formally, for any two attributes $a,b \in C$, the heuristic information is defined as

$$\eta(a,b) = \frac{\left|POS_{\{a,b\}}(D)\right|}{|U|} \tag{1}$$

Where $|U|$ is the cardinality of set U and the $POS_{\{a,b\}}(D)$, called positive region, is defined in [5].

Construction of Feasible Solution. When constructing a solution, each ant should start from the feature core. Firstly, the ant selects randomly a feature, after that, it probabilistically selects the second attribute from those unselected attributes. That probability is calculated by

$$P_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{l \in J}\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}} \tag{2}$$

Where *t* and *k* represent the number of iterations and ants, respectively, *J* represents the set of unvisited features of ant *k*, η_{ij} is heuristic information of choosing feature *j* when at feature *i*, $\tau_{ij}(t)$ is the amount of pheromone between feature *i* and feature *j* at iteration *t*. In addition, α and β are two parameters corresponding to the importance of the pheromone trail and heuristic information. When $\mu_R(D) = \mu_C(D)$, the construction process stops, where *R* is the current solution constructed by an ant.

Pheromone Update. After each ant has constructed its own solution, the pheromone of only edges along the path visited by the ant is updated as

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + q / L_{\min} \tag{3}$$

While for other edges, the pheromone trails are updated according to the following equation.

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) \tag{4}$$

Where ρ is a decay constant used to simulate the evaporation of pheromone, q is a given constant and L_{min} is the minimal feature reduct at iteration *t*. In ACORS, if the maximum iteration is reached, then the algorithm terminates and outputs the minimal reduct encountered. If not, then the pheromone is updated, a new colony of ants are created and the process iterates once more.

3.2 Particle Swarm Optimization Based Reduct Algorithm (PSORS)

PSO is an efficient evolutionary computation technique based on swarm intelligence and originates from the simulation of social behaviors such as birds in a flock or fishes in a school. In PSO, a particle represents a candidate solution to the problem, which has its own velocity and position in a given search space. PSO starts with the stochastic initialization of a population of particles which move in the search space to find the optimal solution by updating the position of each particle by using its own experience and its companion's experience [9]. Assume a swarm includes N particles which move around in a D-dimensional search space. The velocity of the *i*th particle in different space can be represented by $v_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$, and the position for the *i*th particle in different space can be noted as $x_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$. The positions and velocities of the particles are restricted to a predefined range, respectively. The personal best position recording the previous best position of the particle is called *pbest* and the best position achieved by all individual is denoted the global best position and called *gbest* . Based on *pbest* and *gbest* , the velocity and position of each particle are updated to search for the optimal solutions. When BPSO is applied to solve the feature selection problem, a binary digit is employed to stand for a feature, where the bit values 1 and 0 stand for selected and non-selected features, respectively. The velocity of each particle is updated using (5), while the position of each particle is updated using (6). The position and velocity of each particle are updated according to the following equations:

$$v_{id}^{t+1} = w \times v_{id}^{t} + c_1 \times r1 \times (pbest - x_{id}^{t}) + c_2 \times r2 \times (gbest - x_{id}^{t}) \tag{5}$$

$$x_{id} = \begin{cases} 1, & if\ r < S(v_{id}) \\ 0, & otherwise \end{cases} \qquad S(v_{id}) = \frac{1}{1+e^{-v_{id}}} \tag{6}$$

Where *t* represents the iteration counter, *r1* and *r2* are random numbers between 0 and 1, c_1 and c_2 are learning factors that control how far a particle moves in a single generation, *w* is called the inertia weight, and the function $S(v_{id})$ is a sigmoid limiting transformation which is introduced to transform v_{id} to the range of (0, 1), r is random number selected from a uniform distribution between 0 and 1.

In BPSO, the inertia weight w is the modulus that controls the influence of previous velocity on the present one, thus balancing the global exploration and local search ability. It means the appropriate control of inertia weight value is imperative to search for the optimum solution efficiently and precisely. In this paper, chaos theory and BPSO are combined into a method called CBPSO to avoid this early convergence, then CBPSO based RS reduct algorithm (CBPSORS) could be achieved to find superior reduct. Since logistic maps are the most frequently used chaotic behavior maps and chaotic sequences have been proven easy and fast to generate and store, as there is no need for storage of long sequences [10], so logistic map is used to determine the inertia weight value. The inertia weight value is substituted by sequences generated by the logistic map according to the following (7).

$$w(t+1) = \mu \times w(t) \times (1 - w(t)) \qquad w(t) \in (0,1) \tag{7}$$

Where μ is a control parameter, which cannot be bigger than 4. When the inertia weight value is close to 0, CBPSO promotes the local search ability. For inertia weight values near 1, CBPSO strengthens the global search ability.

During the search process, each individual is evaluated using the fitness. According to the definition of RS reduct, the reduction solution must ensure that the decision ability is the same as the original decision table and the number of features in the feasible solution is kept as less as possible. Therefore, classification quality and the number of selected features are the two pivotal factors used to design a fitness function which is used to evaluate each individual. The fitness function is defined as

$$Fitness = \lambda * \mu_R(D) + \xi * \frac{|C| - |R|}{|C|} \tag{8}$$

Where $\mu_R(D)$ represents the classification quality of selected condition attributes R relative to decision D; $|R|$ denotes the number of selected feature subset; $|C|$ denotes the number of the original feature set; λ and ξ are two parameters which determine the relative importance of classification quality and the number of features, $\lambda \in [0,1]$ and $\lambda + \xi = 1$.

4 Artificial Fish Swarm Based Reduct Algorithm (AFSARS)

AFSA is a swarm-intelligence based optimization algorithm that simulates the fish swarm behaviors such as praying, swarming and following with local search of fish individual for obtaining the global optimum, which was successfully applied to solve several combinatorial problems. It is a stochastic and parallel search algorithm. What's more, it does not need to know the concrete information of problems; instead it only needs to compare disadvantages and advantages of the solutions of the problems [11], then the final global optimum will be displayed in the population through artificial fish individual behaviors of local optimization, which has strong robustness, fast speed of convergence and being non-sensitive to initial parameters. The AFSA has been proved to be an effective global optimization algorithm using the swarm intelligence in the solution of the combinatorial problem [12].

Due to these characteristics of the AFSA, it is introduced to solve feature selection problems. Assume a fish swarm includes n particles which move around in a D-dimensional search space. The artificial fish swarm is represented as $F=\{f_1,f_2,\cdots,f_n\}$, where f_i is an artificial fish (AF). An AF can represent a subset of features, and a subset of features can be a binary vector: $X=\{x_1,x_2,\cdots,x_D\}$, $x_i\in\{0,1\}, i=1,2,\cdots,D$, where X is the current state of AF, D is the number of features and the bit values 1 and 0 stand for selected and non-selected features respectively. Let Y stand for the food concentration, namely the objective function value; the visual scope of AF is represented as visual distance. The Hamming distance is used to calculate the visual distance in AFSARS. The Hamming distance of two points of equal bits length is the number of positions at which the corresponding bits are different. S_m is the moving step length, trynumber is the try number and δ is the crowd factor. The representative behavior is described as follows:

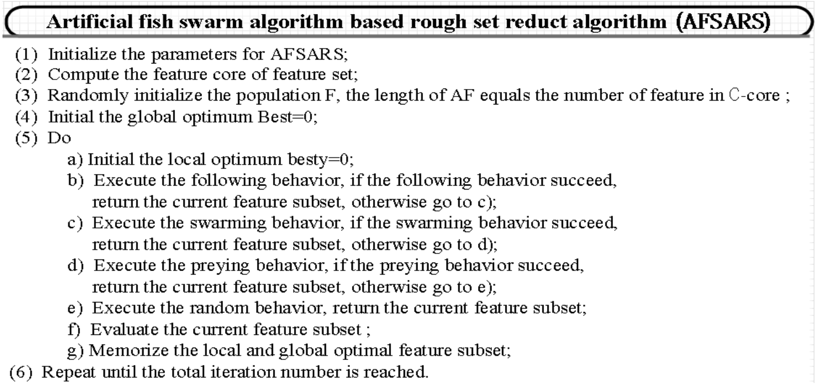
Following Behavior. In the following behavior, when the AF current state is X_i, it will judge the food concentration of all its neighborhood partners. Then it will find the state X_j in the current neighborhood, which has the greatest food concentration Y_j. Let n_f represent the number of its neighbors in the current neighborhood and n represent the total number of AF. If $Y_i<Y_j$ and $\frac{n_f}{n}<\delta$, it denotes the state X_j has more food and is not crowded, it will moves a step toward the state X_j. Otherwise, it performs the swarming behavior.

Swarming Behavior. In the swarming behavior, when the AF current state is X_i, it will assemble in groups naturally in the moving process. Let X_c represent the center position in its visual scope. If $Y_i<Y_c$ and $\frac{n_f}{n}<\delta$, it denotes the center position has higher food concentration and is not crowded. It moves a step toward the center position. Otherwise, it performs the preying behavior. The center position X_c of m fishes is defined as

$$X_c(i)=\begin{cases}1, & \sum_{k=1}^{m}X_k(i)\geq\frac{m}{2}\\ 0, & \sum_{k=1}^{m}X_k(i)\leq\frac{m}{2}\end{cases}\qquad i=1,2,3,\cdots,D \tag{9}$$

Preying Behavior. In the preying behavior, when the AF current state is X_i, it needs to select a state Y_j randomly in its visual scope. If $Y_i<Y_j$, it moves forward a step in

this direction. Otherwise, it selects randomly a state X_j again in its visual distance, and it judges whether the forward condition is satisfied. If it can satisfy before trynumber times, it moves a step toward the state X_j, otherwise, it moves a step randomly. When the AF selects to go forward a step in this direction, the mutation operation of genetic algorithm is adopted in the proposed AFSARS. One position mutation is used to create a trial point. If the AF will go forward a step from the state X_i to the state X_j, then the number of different bits n_b is calculated. Here, if $n_b > S_m$, then $S_m = 3$, otherwise, $S_m = n_b$. Randomly generate a digit n_r which represents the number of mutations, where n_r is between 1 and S_m. Here, some indexes of the positions of mutation are selected and then the bits of selected positions are changed from 0 to 1 or vice versa.



```
Artificial fish swarm algorithm based rough set reduct algorithm (AFSARS)
(1) Initialize the parameters for AFSARS;
(2) Compute the feature core of feature set;
(3) Randomly initialize the population F, the length of AF equals the number of feature in C-core ;
(4) Initial the global optimum Best=0;
(5) Do
      a) Initial the local optimum besty=0;
      b) Execute the following behavior, if the following behavior succeed,
         return the current feature subset, otherwise go to c);
      c) Execute the swarming behavior, if the swarming behavior succeed,
         return the current feature subset, otherwise go to d);
      d) Execute the preying behavior, if the preying behavior succeed,
         return the current feature subset, otherwise go to e);
      e) Execute the random behavior, return the current feature subset;
      f) Evaluate the current feature subset ;
      g) Memorize the local and global optimal feature subset;
(6) Repeat until the total iteration number is reached.
```

Fig. 1. Artificial fish swarm algorithm based rough set reduct algorithm

Random Behavior. If the other fish behaviors are not executed, the AF performs the random behavior. This behavior is related with a random movement for a better position. The behavior is similar to preying behavior, but the different point is the position of mutation which can be any position of the state X_i. The pseudo-code of our proposed method is illustrated in Fig.1.

5 Experiments and Results

5.1 Datasets and Parameters Setting

To evaluate the usefulness of the proposed algorithms, we carry out experiments on six datasets of the UCI machine learning repository. In Dermatology (Der) dataset, some samples are missed in age feature, so it is removed. In the experiments, the five algorithms require additional parameter settings for their operations. The parameters of GA are set as follows: population size P = 20, maximum iteration T = 500, the default parameters of crossover and mutation are adopted in matlab 7.0. The parameters of ACORS are set as follows: $\alpha = 1$, $\beta = 0.01$, $\rho = 0.9$, $q = 0.1$ and the initial pheromone is set to 0.5, the number of ants is half the number of features and the maximum iteration equals 50. The parameters of PSORS are set as follows: the inertia weight decreases along with the iterations, varying from 1.4 to 0.4 according to the reference [9], acceleration constants $c_1 = c_2 = 2.0$, population size P = 20, maximum iteration T = 500, velocity $V_{\max} = 4, V_{\min} = -4$. The parameters of CBPSORS are set as follows: the inertia weight $w(0) = 0.48, \mu = 4$, acceleration constants $c_1 = c_2 = 2.0$, population size P = 20, maximum iteration T = 500, velocity $V_{\max} = 4, V_{\min} = -4$. These parameters are chosen based on the literature [10]. The parameters of AFSARS are set as follows: population size P = 50, maximum iteration T = 50, trynumber=20, maximum step S_m =3, the visual distance of fish is half the number of features, crowd factor δ =0.618. The parameter λ of the fitness is set to 0.9 and $\xi = 0.1$ according to the reference [5]. The fitness function of GARS, PSORS, CBPSORS and AFSARS are defined as the equation (9). In CBPSORS, the core of feature set needs to compute, after that the population is initialized, and the operation is the same as AFSARS. The results achieved from 3 independent runs are employed in terms of the number of the evolved feature subsets in this paper.

5.2 Results and Analysis

Table 1 shows the reduct results of the various methods on the 6 UCI datasets. According to the experimental results, we find that AFSARS, CBPSORS and PSORS have similar efficiency and they are more effective than ACORS and GARS when dealing with datasets having less than 30 features. However, when dealing with datasets with over 30 features, PSORS is easy to fall into premature convergence, which means PSORS is not suitable to find the optimal reduct in most cases. Comparing with PSORS, AFSARS and ACORS become much effective and find successfully the global optimum in limited number of iterations on datasets with over 30 features. For those datasets having many features such as Dermatology and Lung, AFSARS and ACORS are more effective than PSORS and CBPSORS. Furthermore, PSORS hardly finds the optimal reduction until the maximum iteration is reached when it deals with datasets with many features. The performances of PSORS and CBPSORS are not improved after we change their generations to 1000. Apart from these, we find that the performance of CBPSORS is similar to the performance of PSORS when the maximum generation is 500, but when we run the two methods many times, we find that the result of CBPSO is more stable and better. On the whole, it seems to be the case that AFSARS outperforms the other methods in terms of the number of the minimal reducts. But compared to the other methods, AFSARS spends more time to find the optimum reducts.

Table 1. The experimental results of the different algorithms

Dataset	#Features	QR	GARS	ACORS	PSORS	CBPSORS	AFSARS
Momk1	6	4	3	3	3	3	3
Tic-tac	9	8	8	8	7	7	7
Zoo	16	5	6-7	5	5	5	5
Vote	16	10	8-10	9	8	8	8
Der	33	10	11-12	9-10	10	9-10	9
Lung	56	5	12-13	5	9-10	9	5

6 Conclusion

This paper starts with the concepts of rough set theory and the QR algorithm, but this technique often fails to find optimal reducts because of no perfect search strategy. Therefore, the swarm intelligence methods have been introduced to guide RS method to find the minimal reducts. Here, we have discussed four different computational intelligence based reducts: GARS, ACORS, PSORS and CBPSORS. These methods perform well on some datasets, but sometimes they cannot find the optimal solution in the limited number of iteration. In this paper, we propose a novel feature selection algorithm based on artificial fish swarm algorithm hybrid with rough set (AFSARS), which is non-sensitive to initial values, has a strong robustness and has the faster convergence speed to find the minimal reducts. Experimental results on real datasets have demonstrate our proposed method can provide competitive solutions in generating short reducts more efficiently than the other methods.

Acknowledgments. The authors would like to thank to the Natural Science Foundation of the People Republic of China (61073193, 61300230), the Key science and technology Foundation of Gansu Province (1102FKDA010), Natural Science Foundation of Gansu Province (1107RJZA188), science and technology support program of Gansu Province (1104GKCA037) for supporting this research.

References

1. Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R.: Feature Selection Based on Rough Sets and Particle Swarm Optimization. Pattern Recogn. Lett. 28, 459–471 (2007)
2. Suguna, N., Thanushkodi, D.K.: A Novel Rough Set Reduct Algorithm for Medical Domain Based on Bee Colony Optimization. J. Comput. 6, 49–54 (2010)
3. Inbarani, H.H., Azar, A.T., Jothi, G.: Supervised Hybrid Feature Selection Based on PSO and Rough Sets for Medical Diagnosis. Comput. Meth. Prog. Bio. 113, 175–185 (2014)
4. Arafat, H., Elawady, R.M., Barakat, S., Elrashidy, N.M.: Using Rough Set and Ant Colony Optimization in Feature Selection. Int. J. Emerg. Trends Technol. Comput. Sci. 2, 148–155 (2013)
5. Bae, C., Yeh, W., Chung, Y.Y., Liu, S.: Feature Selection with Intelligent Dynamic Swarm and Rough Set. Expert Syst. Appl. 37, 7026–7032 (2010)

6. Velayutham, C., Thangavel, K.: Unsupervised Quick Reduct Algorithm using Rough Set Theory. J. Electron. Sci. Technol. 9, 193–201 (2011)
7. Qablan, T., Al-Radaideh, Q.A., Shuqeir, S.A.: A Reduct Computation Approach Based on Ant Colony Optimization. Basic Sci. Eng. 21, 29–40 (2012)
8. Chen, Y., Miao, D., Wang, R.: A Rough Set Approach to Feature Selection Based on Ant Colony Optimization. Pattern Recogn. Lett. 31, 226–233 (2010)
9. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. Appl. Soft Comput. 18, 261–276 (2014)
10. Chuang, L., Yang, C., Li, J.: Chaotic Maps Based on Binary Particle Swarm Optimization for Feature Selection. Appl. Soft Comput. 11, 239–248 (2011)
11. Farzi, S.: Efficient Job Scheduling in Grid Computing with Modified Artificial Fish Swarm Algorithm. Int. J. Comput. Theor. Eng. 1, 13–18 (2009)
12. Liu, T., Qi, A., Hou, Y., Chang, X.: Feature Optimization Based on Artificial Fish-swarm Algorithm in Intrusion Detections. In: International Conference on Networks Security, Wireless Communications and Trusted Computing, vol. 1, pp. 542–545. IEEE Press, Wuhan (2009)

Hand Gesture Shape Descriptor Based on Energy-Ratio and Normalized Fourier Transform Coefficients

Wenjun Tan[1,2,*], Zijiang Bian[1], Jinzhu Yang[1,2],
Huang Geng[1], Zhaoxuan Gong[1], and Dazhe Zhao[1,2]

[1] Medical Image Computing Laboratory of Ministry of Education,
Northeastern University, 110819, Shenyang, China
[2] College of Information Science and Engineering,
Northeastern University, 110819, Shenyang, China
{tanwenjun,bianzijian,yangjinzhu,genghuan,
gongzhaoxuan,zhaodzh}@mail.neu.edu.cn

Abstract. The hand gesture shape is the most remarkable feature for gesture recognition system. Since hand gesture is diversity, polysemy, complex deformation and spatio-temporal difference, the hand gesture shape descriptor is a challenging problem for gesture recognition. This paper presents a hand gesture shape describing method based on energy-ratio and normalized Fourier descriptors. Firstly, the hand gesture contour of the input image is extracted by YCb'Cr' ellipse skin color model. Secondly, the Fourier coefficients of the contour are calculated to transform the point sequence of the contour to frequency domain. Then the Fourier coefficients are normalized to meet the rotation, translation, scaling and curve origin point invariance. Finally, the items of normalized Fourier coefficients are selected by calculating energy-ratio information as the hand shape descriptors. For validating the shape descriptors performance, the hand gestures 1-10 are recognized with the template matching method and the shape descriptor method, respectively. The experiment results show that the method can well describe the hand shape information and are higher recognition rate.

Keywords: Hand gesture, Shape descriptor, Fourier descriptors, Skin color.

1 Introduction

Since hand gesture has the characteristics of diversity, polysemy, complex deformation and spatio-temporal difference, hand gesture recognition is one of the current topics of new generation human-computer interaction techniques. Hand gesture recognition system based machine vision recognizes gestures by segmentation and feature extraction from 2D image sequences from camera. This hand gesture recognition system has many good performances such as simple inputs, low device requirement, freedom from interference and so on[1]. The goal of hand gesture

[*] Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 34–41, 2014.
© Springer International Publishing Switzerland 2014

segmentation is to seperate hand region from complex background and to retain gestures in foregrounds. Because the independent from data gloves or color landmarks, naturally interact and fast detect, the skin color-based model has become the most mature method at present[2-3]. Gesture feature extraction is the key recognition procedure, in which shape information is the most remarkable features and proper information of gesture recognition system[4-5].

Shape is defined as a function describing position, direction, and surrounded region of a closed curve in 2D image space. Regular descriptors originate from point coordinates on closed curve of corresponding target outline. According to the expression methods, shape descriptors are commonly divided into two classes: region-expression and outline-expression. Region descriptor focuses on global geometry features, including area, perimeter, axis directions, compactness, solid degrees and so on[6]. Usually several global geometry features are adopted in shape matching and recognition, which is slow in descripting calculating complex shapes and loses information seriously in feature extraction. These defects lead to low resolution to express shape inaccurately. Besides, the region descriptors use inter-target texture distribution statistics, such as the 7 invariant moment[7]. Contour descriptor includes skeleton-based morphology[8], neural network[9], fractal method[10], which usually requires complex geometry relevant function to meet the invariance limits in translation, rotation and zoom, leading to complex calculation. Fourier descriptor is proposed by Zhan C T and Roskies R Z[11], and improved by Persoon[12]. The method regards contour as a closed curve formed by a sequence of end-to-end discrete points, and transforms the point sequence to frequency domain. The Fourier coefficients are defined as Fourier descriptor. The high-frequency components of Fourier transformation corresponds to detail information and the low-frequency components corresponds to overall shape, thus the low-frequency components could be selected to descript object's shape information. Meanwhile, Fourier inverse transform is able to restore the shape information expressed by the descriptor, which is unable by other descriptors. However, the Fourier transformation coefficients are concerned with target's scale, direction curve origin. So the classical Fourier transformation method is difficult to express object's shape invariance accurately. Besides, Fourier coefficients corresponds with contour sequence point number, which is usually numerous. Thus, if the whole coefficients are calculated, the heavy burden for gesture classifiers and low recognition rate would be lead to.

On the basis of the gesture region and contour extraction method in the our previous work, this paper presents an accurate and effective gesture shape expression method based on energy-ratio and normalized Fourier descriptors, focusing on shape invariance and effective coefficients selection of the descriptors.

2 Contour of Hand Gesture Extraction

Hand gesture segmentation is the first step of the hand contour extraction. This means segmenting the region of gestures from a complex background, leaving them alone in the foreground. Skin color is so good clustering property to be able to separate ‘complexion’ and ‘non-complexion’ region.

Hsu R L proposed a way to use ellipse model to describe the skin color distribution nonlinearly transformed to YCb'Cr' in the Cb'Cr' region, and apply it to face detection, obtaining better result[14]. Hsu R L put forward that color values always have nonlinear dependence relation to the luminance value Y in YCbCr color space.

Just through the calculation that whether the pixel is in the ellipse can we detect whether it belongs to the skin color. So it has a fast computing speed and high detection accuracy. But it may not describe accurately for the particular imaging equipment. This skin model easily leads to some non-skin color point being included to cause skin point over detection. Meanwhile, the model may not include all color regions, causing the skin color detection incomplete.

Aim at this problem, we presents a YCb'Cr' space ellipse fitting under the skin modeling method based on the specific statistical properties of the color distribution to segment hand gesture[15]. So the hand regions are segmented by the method in this work. Then the contours of hand gestures are easily extracted by 8-neighbourhood tracing algorithm.

3 Shape Descriptor of Hand Gesture

3.1 Normalized Fourier Transform Coefficients

Let gesture contour be a closed curve represented by coordinate sequence $s(k)=[x(k),y(k)]$, in which $[x(k),y(k)]$ donates the coordinate pairs starting from (x_0,y_0) and going contour clockwise along the curve. The complex number is $p(l)=x(l)+jy(l)$, $(l=0,1,\cdots,n-1), j=\sqrt{-1}$, whose discrete Fourier coefficient is expressed as:

$$z(k)=\frac{1}{n}\sum_{l=0}^{n-1}p(l)e^{-j\frac{2\pi lk}{n}} \tag{1}$$

where $(k=0,1\cdots,n-1)$, $z(k)$ is the Fourier transformation of $p(l)$ and the expression of point sequence in frequency domain. $z(k)$ correlates with object's shape. After the transformation in Eq(1), the object's contour is simplified from 2D to 1D space. The high-frequency Fourier coefficients are able to describe contour details and low-frequency ones identify overall shape information. As a result, the low-frequency Fourier coefficients could be selected to express object's shape information. The inverse Fourier transformation $z(k)$ is defined as:

$$p(l)=\sum_{k=0}^{n-1}z(k)\mathrm{e}^{j\frac{2\pi lk}{n}} \tag{2}$$

Object's shape information could be restored with the inverse transform from Eq (2). Table.1 is the properties of Fourier descriptor of contour sequence $p(l)$ for rotation, translation, zooming and origin moving procedures[16], where Δ_{xy} is defined to be $\Delta_{xy}=(\Delta x+j\Delta y)=(x_0+jy_0)$.

The Fourier descriptors got by Eq(1) transformed were concerned with shape's scale, direction and curve origin point, thus the descriptor should be processed to meet the requirement of shape characteristic invariance. For the transformation property of translation, only the k=0 becomes impulse function and other properties are no changes. So the $z(0)$ only change the centroid position of the object and don't change the object's shape. The $z(0)$ can be set as 0 for the shape descriptors. The Eq(9) could be derived from Eq(1) expressed as[17]:

$$d(k) = \frac{\|z'(k)\|}{\|z'(1)\|} = \frac{\alpha\left\|e^{j\theta}e^{j\frac{2\pi k l_0}{n}}z(k)\right\|}{\alpha\left\|e^{j\theta}e^{j\frac{2\pi l_0}{n}}z(1)\right\|} = \frac{\|z(k)\|}{\|z(1)\|} \tag{3}$$

The Eq(3) shows the change of Fourier coefficients of module and phase of object's rotation, scaling and origin position. The $d(k)$ in Eq(3) is called normalized Fourier descriptor, $k = (2,\cdots,n-1)$, which conforms the rotation, scaling, translation and origin position invariance.

3.2 Fourier Coefficients Selection Based on Energy-Ratio

Usually the high-frequency Fourier coefficients could explain shape details well, but the low-frequency ones decide object's overall shape. Thus partial low-frequency Fourier coefficients could be selected to express gesture shape information. However, the coefficients are few selected, the corresponding details will be lost so that the shape information will be difficult to express accurately. So it is important to select proper descriptor numbers for the gesture shape description.

Assume the first p Fourier coefficients express gesture shape information but not all the ones, so $k > p-1, z(k) = 0$ in Eq(1), and the other ones remain unchanged, the curve $p(l)$ will be defined as:

$$\hat{p}(l) = \sum_{k=0}^{p-1} z(k)\mathrm{e}^{j\frac{2\pi lk}{n}} \tag{4}$$

where, $l = (0,1\cdots,n-1)$. Though p Fourier coefficients could obtain each components $\hat{p}(l)$, the range of l is still from 0 to n-1. That is to say, similar boundaries have the same numbers of points, but these needn't to be so much Fourier coefficients.

Thus the Fourier descriptors are defined as each item coefficient, and the descriptor numbers selection corresponds with Fourier coefficient items. Fourier transformation could translate gesture shape from spatial to frequency domain, and the frequency corresponds to energy information with amplitude values. The energy information shows corresponding coefficient ratio in shape expression. Thus the Fourier coefficients are selected by calculating Fourier coefficients energy-ratio information. The Fourier coefficient energy $E(l)$ of curve $p(l)$ could be expressed as follows:

$$E(l)=\sum_{k=0}^{l}\left\|z(k)\right\| \tag{5}$$

The energy ratio of the first p Fourier coefficient is:

$$e(p)=\frac{E(p)}{E(l)}=\frac{\sum_{k=0}^{p}\left\|z(k)\right\|}{\sum_{k=0}^{l}\left\|z(k)\right\|} \tag{6}$$

From Eq(1), the Fourier transformation in translation is converted to impulse function when k=0, that is, the value of $z(0)$ varies a lot, which make significant influence of the calculation of $e(p)$. Thus, the coefficient energy of $z(0)$ is ignored in this paper and only the energy ration information is calculated of $l=(1\cdots,n-1)$. $e(p)$ is iteratively calculated by increasing the value of p, when $e(p)$ is greater than a threshold or the difference between $e(p)$ and $e(p+1)$ is relatively small, the p is regarded as the final Fourier coefficient number. The normalized Fourier descriptor $d(k-1)$, $k=(2\cdots,p)$, is the descriptor of gesture shape.

4 Experiments and Discussion

To verify the shape descriptor and the items selecting method of the Fourier coefficient, the four gesture images are adopted in this work(Fig.1). The contour of gestures is extracted by the method in section 2 and is expressed with red line in Fig.1. Fig.2(a-d) shows the amplitude scattergram of the Fourier transformation coefficient corresponding of these gesture contours. To analyze accurately, Fig.2(a-d) retained the coefficient of $z(0)$. It is shown that the amplitude of $z(0)$ had a greater difference with other coefficients, which is as same as the analysis in section 3.2. When the coefficient number was bigger than 15, the amplitude varied little. The energy ratio information of different Fourier items were calculated as Eq(6) and shown in Fig.2(e).

To verify the Fourier coefficients selection method in this paper, the results of the coefficient numbers from 9 to 15 are compared. Fig.3 is the experiment results of hand gesture 1-8, in which the green lines express the contour curves through Fourier inverse transform. The Fourier descriptor details lose seriously of gesture 3,4,5,6,8 from the figure and could not express the whole shape information with 9 coefficient numbers. And the contour boundary information could be basically expressed by 13 items; and the contour boundary information of 15 descriptors is as same as 13 descriptors. According to the energy-ratio of the Fourier descriptors and analyzing dimensions and detail lose situations, the 13 item Fourier coefficients were adopted to express shape characters in this paper. The shape descriptors $d(k\text{-}1)$ is calculated as Eq(3), where $k=12$. Because the z(0)=0, so there are 11 items of Fourier coefficient as hand gesture shape descriptors in this work.

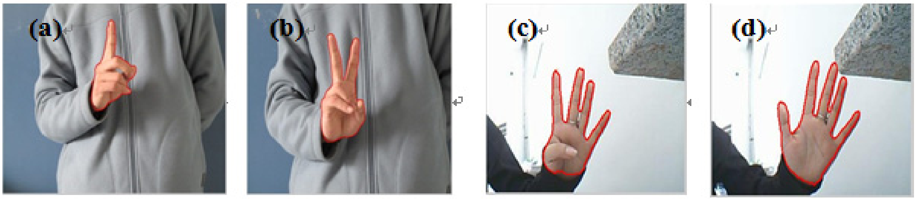


Fig. 1. The test hand gesture: (a) hand gesture 1; (b) hand gesture 2; (c) hand gesture 4; (d) hand gesture 5

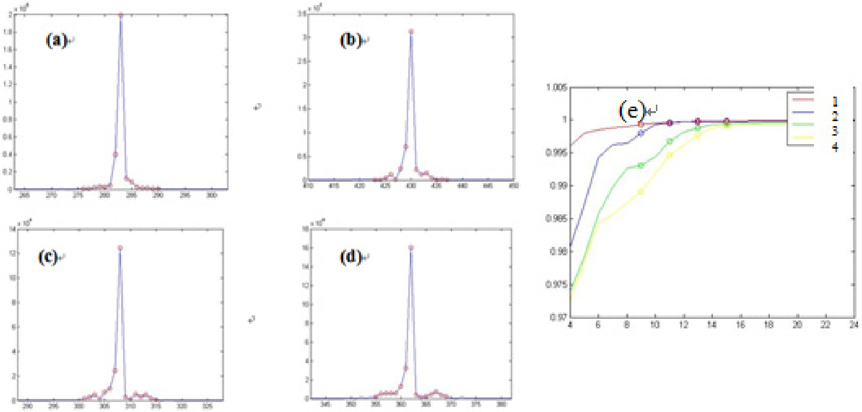


Fig. 2. Amplitude of Fourier coefficient of hand gesture contour: (a) hand gesture 1; (b) hand gesture 2; (c) hand gesture 4; (d) hand gesture 5; (e) energy ratio map of Fourier coefficient

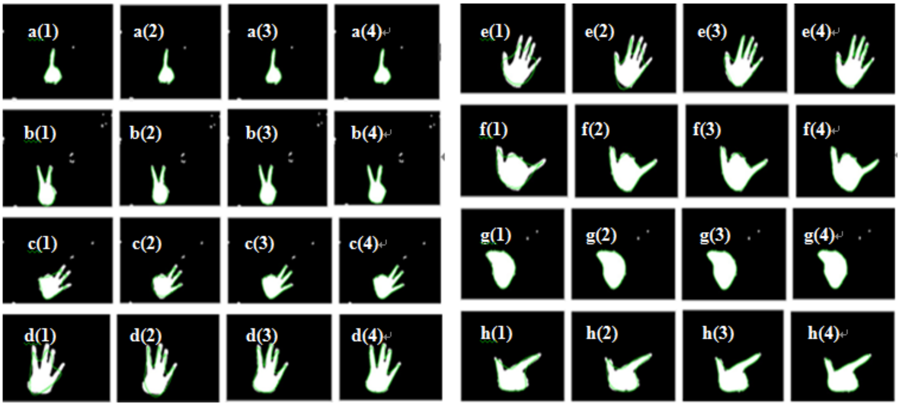


Fig.3. Comparison of the reconstruction outlines with Fourier descriptors: (a1-h1) items 9; (a2-h2) items 11; (a3-h3) items 13; (a4-h4) items 15.

The hand gestures 1-10 are recognized with the shape descriptor method in this paper to verify the descriptor validity. The template matching method is a common target recognition method, which is widely applied in pattern recognition system. For identification accuracy-ratio and time performance, the template matching method and this work are compared in this paper. In order to compare the consistency of the results of two methods, the Euclidean distance method is adopted to recognize the gesture categories. Table 1 is the recognition results of the two methods. The results show that the average recognition rate of the template matching method and our method is 79% and 90%, respectively.

Table 1. Recognition results of hand gesture with template matching method and this method

Hand gesture	Template matching method		This method	
	Correct number	Recognition rate	Correct number	Recognition rate
Gesture 1	35	87.50%	37	92.50%
Gesture 2	36	90.00%	34	85.00%
Gesture 3	32	80.00%	35	87.50%
Gesture 4	30	75.00%	36	90.00%
Gesture 5	29	72.50%	38	95.00%
Gesture 6	35	87.50%	39	97.50%
Gesture 7	27	67.50%	34	85.00%
Gesture 8	34	85.00%	38	95.00%
Gesture 9	28	70.00%	33	82.50%
Gesture 10	30	75.00%	36	90.00%

5 Conclusions

The hand gesture shape expression methods based on energy-ratio and normalized Fourier descriptor is presented in this work. The hand gesture contour is the input data for the shape descriptor, which is extracted by our previous work. Then the Fourier coefficients of the hand contour are transformed to express the point sequence of hand gesture contour in frequency domain. For meeting the rotation, translation, zooming and curve origin point invariance, the Fourier coefficients are normalized. The items of Fourier coefficients are selected by calculating energy-ratio information. Finally, to verify the shape descriptors and selection items method of the Fourier coefficients, the hand gestures 1-10 are recognized with the shape descriptor method in this paper. The experiment results show that the method can well describe the hand shape information and access higher recognition rate.

Acknowledgment. This research was partly supported by National Natural Science Foundation of China (NSFC) under Grant No. 61302012 and No. 61172002, the Fundamental Research Funds for the Central Universities under Grant N130418002 and N120518001, and Liaoning Natural Science Foundation under Grant No. 2013020021.

References

1. Elli, A.: Understanding The Color of Human Skin. In: Proceeding of the SPIE Conference on Human Vision and Electronic Imaging VI (SPIE), pp. 243–251. SPIE Press (2001)
2. Yang, J., Lu, W., Waibel, A.: Skin-Color Modeling and Adaptation. In: Chin, R., Pong, T.-C. (eds.) ACCV 1998. LNCS, vol. 1352, pp. 687–694. Springer, Heidelberg (1997)
3. Bradski, G., Yeo, B., Minerva, M.Y.: Gesture for Video Current Navigation. In: SPIE IS&T/SPIE, vol. 36, pp. 230–242 (1999)
4. Kawulok, M.: Fast Propagation-based Skin Regions Segmentation in Color Images. In: Proc. IEEE FG, pp. 1–7. IEEE Press (2013)
5. Nalepa, J., Grzejszczak, T., Kawulok, M.: Wrist Localization in Color Images for Hand Gesture Recognition. In: Gruca, A., Czachórski, T., Kozielski, S. (eds.) Man-Machine Interactions 3. AISC, vol. 242, pp. 81–90. Springer, Heidelberg (2014)
6. Ang, Y.H., Li, Z., Ong, S.H.: Image Retrieval Based on Multidimensional Feature Properties. In: SPIE, vol. 24, pp. 47–57 (1995)
7. Hu, M.K.: Visual Pattern Recognition by Moment Invariants. IEEE Trans. Informaiton Theory 8, 179–187 (1962)
8. Xu, J.: Morphological Decomposition of 2-D Binary Shapes Into Modestly Overlapped Octagonal and Disk Components. IEEE Transactions on Image Processing 16, 337–348 (2007)
9. Gupta, L., Sayeh, M.R., Tammana, R.: Neural Network Approach to Robust Shape Classification. Pattern Recognition 23, 563–568 (1990)
10. Taylor, R., Lewis, P.H.: 2D Shape Signature Based on Fractal Measurements. IEEE Proceedings -Vision, Image and Signal Processing 4, 422–430 (1994)
11. Zhan, C.T., Roskies, R.Z.: Fourier Descriptors for Plane Closed Curves. IEEE Trans. Computer 21, 269–281 (1972)
12. Persoon, E., Fu, K.S.: Shape Discrimination Using Fourier Descripters. IEEE Trans. System, Man, Cybernetics 7, 170–179 (1997)
13. Drolon, H., Druaux, F., Faure, A.: Particles Shape Analysis and Classification Using The Wavelet Transform. Pattern Recognition Letters 21, 473–482 (2000)
14. Hsu, R.L., Jain, A.K.: Face Detection in Color Images. IEEE Transactions Pattern Analysis and Machine Intelligence 24, 696–706 (2002)
15. Tan, W.J., Dai, G.Y., Su, H., Feng, Z.Y.: Gesture Segmentation Based on YCb'Cr' Color Space Ellipse Fitting Skin Color Modeling. In: The 24th Chinese Control and Decision Conference (CCDC), pp. 1905–1908. IEEE Press, Taiyuan (2012)
16. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital image processing using MATLAB. Publishing House of Electronics Industry, Beijing (2005)
17. Wang, T., Liu, W.Y., Sun, J.G., Zhang, H.J.: Using Fourier Descriptors to Recognize Object's Shape. Journal of Computer Research and Development 39, 1714–1719 (2002)

A New Evolutionary Support Vector Machine with Application to Parkinson's Disease Diagnosis

Yao-Wei Fu, Hui-Ling Chen*, Su-Jie Chen, LiMing Shen, and QiuQuan Li

College of Physics and Electronic Information,
Wenzhou University, China
chenhuiling.jlu@gmail.com

Abstract. In this paper, we present a bacterial foraging optimization (BFO) based support vector machine (SVM) classifier, termed as BFO_SVM, and it is applied successfully to Parkinson's disease (PD) diagnosis. In the proposed BFO-SVM, the issue of parameter optimization in SVM is tackled using the BFO technique. The effectiveness of BFO-SVM has been rigorously evaluated against the PD Dataset. The experimental results demonstrate that the proposed approach outperforms the other two counterparts via 10-fold cross validation analysis. In addition, compared to the existing methods in previous studies, the proposed system can also be regarded as a promising success with the excellent classification accuracy of 96.89%.

Keywords: Support vector machines, Parameter optimization, Bacterial foraging optimization, Parkinson's disease diagnosis, Medical diagnosis.

1 Introduction

As a primary machine learning technique, support vector machines (SVM) [1] is rooted in the Vapnik-Chervonenkis theory and structural risk minimization principle. Thanks to its good properties, SVM has found it's applications in a wide range of classification tasks. In particular, SVM has demonstrated excellent performance on many medical diagnosis tasks. However, there is still much room for improvement of the SVM classifier. Because it has been proved that proper model parameters setting can improve the SVM classification accuracy substantially [2]. Values of parameters such as penalty parameter and the kernel parameter of the kernel function should be carefully chosen in advance when SVM is applied to the practical problems. Traditionally, these parameters were handled by the grid-search method and the gradient descent method. However, one common drawback of theses methods is that they are vulnerable to local optimum. Recently, biologically inspired metaheuristics such as genetic algorithm and particle swarm optimization (PSO) have been considered to have a better chance of finding the global optimum solution than the traditional aforementioned methods. As a relatively new member of the swarm-intelligence algorithms, BFO has been found to be a promising technique for real-world optimization problems such as optimal controller design [3], learning of artificial neural networks [4] and active power filter design [5].

* Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 42–49, 2014.
© Springer International Publishing Switzerland 2014

This study attempts to employ BFO to handle the parameter optimization of SVM and applied the resultant effective model BFO-SVM for effective detection of Parkinson's disease (PD). The main objective of this study is to explore the maximum generalization capability of SVM and apply it to PD diagnosis to distinguish patients with PD from the healthy ones.

The remainder of this paper is organized as follows. The related works on detection of PD is presented in Section 2. In section 3 the detailed implementation of the BFO-SVM diagnostic system is presented. Section 4 describes the experimental design. The experimental results and discussion of the proposed approach are presented in Section 5. Finally, Conclusions and recommendations for future work are summarized in Section 6.

2 Related Works on Detection of PD

PD is one kind of degenerative diseases of the nervous system, which is characterized by a group of conditions called motor system disorders because of the loss of dopamine-producing brain cells. Till now, the cause of PD is still unknown, however, it is possible to alleviate symptoms significantly at the onset of the illness in the early stage [6]. It is claimed that approximately 90% of the patients with PD show vocal impairment [7], the patients with PD typically exhibit a group of vocal impairment symptoms, which is known as dysphonia. The dysphonic indicators of PD make speech measurements an important part of diagnosis. Recently, dysphonic measures have been proposed as a reliable tool to detect and monitor PD [8].

Various researchers have studied the PD diagnosis problem. Little et al. [8] conducted a remarkable study about PD identification, they employed an SVM classifier with Gaussian radial basis kernel functions to predict PD, and also performed feature selection to select the optimal subset of features from the whole feature space, and the best accuracy rate of 91.4% was obtained by the best model. Das [9] presented a comparative study of using Neural Networks (ANN), DMneural, Regression and Decision Tree for effective diagnosis of PD, the experimental results have shown that the ANN classifier yielded the best results, the overall classification score of 92.9% was achieved. Recently, Ozcift et al. [10] combined the correlation based feature selection (CFS) algorithm with the rotation forest (RF) ensemble classifiers of 30 machine learning algorithms to identify PD, and the best classification accuracy of 87.13% was achieved by the proposed CFS-RF model. Chen et al. [11] employed the fuzzy *k*-nearest neighbor (FKNN) approach in combination with the principle component analysis (PCA-FKNN) to diagnose PD, and the best classification accuracy of 96.07% was obtained by the proposed diagnosis system.

3 The Proposed BFO-SVM Model

This study proposes a novel BFO-SVM model for parameter optimization problem of SVM. In the proposed model, the parameter optimization for SVM are dynamically conducted by implementing BFO algorithm, then the obtained optimal parameters are taken by the SVM model to perform the classification task. The proposed model is comprised of two main evaluation procedures:

1) Inner_Parameter_Optimization procedure: Evaluate the performance of each candidate parameters;
2) Outer_Performance_Estimation procedure: Evaluate the overall performance of the SVM classifier with the optimal parameter values obtained;

In the Inner_Parameter_Optimization procedure, the parameters *C* and *γ* are dynamically optimized by implementing BFO algorithm. The classification accuracy is taken into account in designing the fitness:

$$f = avgACC = (\Sigma_{i=1}^{K} testACC_i)/k \ . \qquad (1)$$

where variable *avgACC* in the function *f* represents the average test accuracy achieved by the SVM classifier via *k*-fold CV, where *k* = 5. Note that here the 5-fold CV is employed to do the model selection that is different from the outer loop of 10-fold CV, which is used to do the performance estimation. The pseudo-code of this procedure is given bellow:

Pseudo-code for the Inner_Parameter_Optimization procedure

```
step 1.  Initialize parameters $p, S, Nc, Ns, Nre, Ned, Ped, \theta^i$
           where
           $p$: number of dimension of the search space,
           $S$: swarm size of the population,
           $Nc$: number of chemotactic steps,
           $Ns$: swimming length,
           $Nre$: the number of reproduction steps,
           $Ned$: the number of elimination-dispersal events,
           $Ped$: elimination-dispersal probability, and
           $C(i)$: the size of step taken in the random direction specified by the tumble.
step 2.  Elimination-dispersal loop: $l=l+1$.
step 3.  Reproduction loop: $k=k+1$.
step 4.  Chemotaxis loop: $j=j+1$.
           (a) For $i=1,2,\ldots,S$, take a chemotactic step for bacterium $i$ as follows.
           (b) Train SVM and compute the fitness $J(i, j, k, l)$
                 Let, $J(i, j, k, l)=J(i, j, k, l)+J_{ar}(\theta)$ where $J_{ar}$ is defined in Eq. (8).
           (c) Let $Jlast=J(i, j, k, l)$ to save this value since we may find a better cost
        via a run.
           (d) Tumble: generate a random vector $\Delta(i) \in R^p$ with each element
                 $\Delta_m(i), m=1,2,\ldots,p$, a uniformly distributed random number on [-1, 1].
           (f) Move: let
                 $\theta^i(j+1,k,l,di) = \theta^i(j,k,l,di) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$
           (g) Train SVM and compute the fitness $J(i,j+1,k,l)$, and let
```

```
                $J(i,j+1,k,l)=J(i, j, k, l)+Jar(\theta)$.
        (h) Swim.
            i) Let $n=0$;
              ii) While $n<Ns$
              iii) Let $n=n+1$;
              iv) If $J(i,j+1,k,l)<Jlast$, let $Jlast=J(i,j+1,k,l)$ and let
                  $\theta^i(j+1,k,l,di)=\theta^i(j,k,l,di)+C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$
                    and use this $\theta^i(j+1,k,l)$ to train SVM, and then compute the
                  new fitness
                    $J(i, j+1,k, l)$ as did in (g);
              v) Else, let $n=Ns$.
        (i) Go to next bacterium ($i$+1) if $i \neq S$.
step 5. If $j<Nc$, go to step 4.
step 6. Reproduction:
            Rank all of the individuals according to the sum of the evaluation results
        in this period, and then removes out the last half individuals and duplicates
        one copy for each of the rest half.
step 7. If $k<Nre$, go to step 3.
step 8. Elimination-dispersal:
            For $i=1,2,...,S$ with probability $Ped$, eliminate and disperse each
        bacterium.
            If $l<Ned$, then go to step 2; otherwise end.
```

In the Outer_Performance_Estimation procedure, SVM model performs the classification tasks using the obtained optimal parameters via 10-fold CV analysis. The pseudo-code of this procedure is given bellow:

Pseudo-code for the Outer_Performance_Estimation procedure

```
/*performance estimation by using k-fold CV where k = 10*/
Begin
 For j = 1:k
 Training set = k-1 subsets;
 Testing set = remaining subset;
    Train the SVM classifier on the training set using the parameters and feature
 subsets obtained from Inner_Parameter_Optimization procedure;
 Test it on the testing set;
 End For;
 Return the average classification accuracy rates of SVM over j testing set;
End.
```

4 Experimental Setup

In this study, we have performed our conduction on the Parkinson's data set taken from UCI machine learning repository.

The BFO-SVM, PSO-SVM and Grid-SVM classification models were implemented using MATLAB platform. For SVM, LIBSVM implementation was utilized, which was originally developed by Chang and Lin [12]. We implemented the BFO, PSO and grid search algorithm from scratch. The computational analysis was conducted on Windows 7 operating system with AMD Athlon 64 X2 Dual Core Processor 5000+ (2.6 GHz) and 4GB of RAM. Normalization is firstly employed before classification, in order to avoid feature values in greater numerical ranges dominating those in smaller numerical ranges, as well as to avoid the numerical difficulties during the calculation. In order to guarantee the valid results, the *k*-fold CV was employed to evaluate the classification accuracy.

In this study, we designed our experiment using a two-loop scheme, which also was used in [13]. The detailed parameter setting for BFO-SVM is shown in Table 1. For PSO-SVM, the number of the iterations and particles are set to 250 and 8, respectively. v_{max} is set about 60% of the dynamic range of the variable on each dimension for the continuous type of dimensions, $c_1 = 2$, $c_2 = 2$, w_{max} and w_{min} are set to 0.9 and 0.4, respectively. The searching ranges of $C \in [2\,^\wedge(-5), 2\,^\wedge(15)]$ and $\gamma \in [2\,^\wedge(-5), 2\,^\wedge(15)]$ for BFO-SVM, PSO-SVM and Grid-SVM were set as the same.

Table 1. Common parameter setup for BFO

S	*Nc*	*Ns*	*Ned*	*Nre*	*ped*	*datt*	*watt*	*wrepe*	*hrepe*	*C*(*i*)
8	100	20	1	4	0.25	0.1	0.2	10	0.1	0.1

Classification accuracy, sensitivity and specificity were used to test the performance of the proposed BFO-SVM model.

5 Experimental Results and Discussions

In BFO, the parameter chemotaxis step size *C*(*i*) plays an important role in controlling the search ability of BFO. Thus, we firstly present results from our investigations on the impacts of *C*(*i*) and assign initial values for it. *C*(*i*) can be initialized with biologically motivated values, but a biologically motivated value may not be the best for specific application [3]. In Table 2, we illustrate the relationship between the different values of *C*(*i*) and the performance of BFO-SVM. The average results are presented with the standard deviation described in the parenthesis. From the table we can see that BFO-SVM reaches the best performance at *C*(*i*) = 0.1 in terms of accuracy, sensitivity and specificity. Therefore, we select 0.1 as the parameter value of *C*(*i*) for the proposed BFO-SVM to implement the coming tasks.

Table 2. The detailed results of BFO-SVM with different C(i) on the PD data set

Chemotactic step size parameter *C(i)*	BFO-SVM		
	Accuracy (%)	Sensitivity (%)	Specificity (%)
0.05	94.84(6.43)	97.41(6.11)	88.00(17.79)
0.1	**96.90(4.34)**	**98.75(3.95)**	**90.83(16.87)**
0.15	95.39(4.53)	97.44(4.33)	90.42(10.77)
0.2	94.42(4.97)	97.57(4.19)	85.50(13.17)
0.25	93.40(6.28)	96.08(5.48)	87.71(17.36)
0.3	94.47(4.97)	96.02(4.82)	90.64(12.32)

To evaluate the effectiveness of the proposed BFO-SVM system for PD, we conducted experiments on the PD database. Table 3 shows the classification accuracy, sensitivity, specificity, and optimal pairs of (C, γ) for each fold obtained by BFO-SVM. The comparison among PSO-SVM, Grid-SVM and BFO-SVM is shown in Figure 1.

Table 3. The detailed results of BFO-SVM on the PD data set

Fold No.	**BFO-SVM**				
	Accuracy	Sensitivity	Specificity	C ($\times 10^4$)	γ
1	0.8947	0.8750	1.0000	1.4250	3.7726
2	1.0000	1.0000	0.8333	2.2658	3.3197
3	0.9474	1.0000	1.0000	2.2930	3.1821
4	1.0000	1.0000	0.7500	1.2372	4.6147
5	1.0000	1.0000	1.0000	2.8519	3.5251
6	0.9000	1.0000	1.0000	1.4252	3.8452
7	1.0000	1.0000	0.5000	2.0478	3.2553
8	1.0000	1.0000	1.0000	0.1331	3.7350
9	1.0000	1.0000	1.0000	2.2390	3.7168
10	0.9689	1.0000	1.0000	0.8046	4.9414
Avg.	0.0434	0.9875	0.9083	1.6723	3.7908

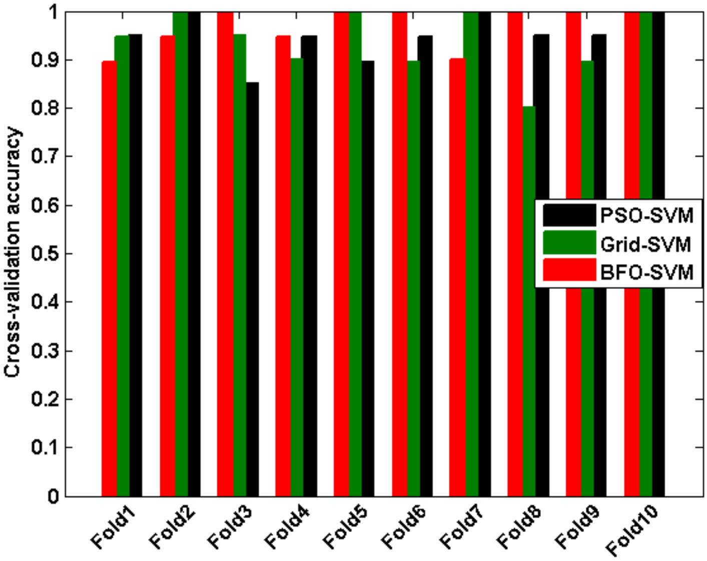


Fig. 1. The cross-validation accuracy obtained for each fold by PSO-SVM, Grid-SVM and BFO-SVM

As shown in Figure 1, we can see that BFO-SVM has dominated PSO-SVM and Grid-SVM in most folds in the process of the 10-fold CV, namely, BFO-SVM has achieved the high classification accuracy equal to or better than that of the other two models obtained for 7 folds in the whole 10 folds. The average classification accuracy of BFO-SVM is 96.89%, while the average classification accuracy of PSO-SVM and Grid-SVM are 94.89% and 93.87%, respectively.

6 Conclusions and Future Work

This work has explored a new diagnostic system, BFO-SVM, for detection of PD. The main novelty of this paper lies in the proposed BFO-based approach, which aims at maximizing the generalization capability of the SVM classifier by exploring the new swarm intelligence technique for optimal parameter tuning for PD diagnosis. The empirical experiments on the PD database have demonstrated the superiority of the proposed BFO-SVM over PSO-SVM and Grid-SVM in terms of classification accuracy. It indicates that the proposed BFO-SVM system can be used as a viable alternative solution to PD diagnosis.

Acknowledgments. This work is supported by the open project program of Wenzhou University Laboratory under Grant No. 13SK29A.

References

1. Vapnik, V.N.: The nature of statistical learning theory. Springer, New York (1995)
2. Keerthi, S., Lin, C.: Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Computation 15(7), 1667–1689 (2003)
3. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine 22(3), 52–67 (2002)
4. Ulagammai, M., et al.: Application of bacterial foraging technique trained artificial and wavelet neural networks in load forecasting. Neurocomputing 70(16-18), 2659–2667 (2007)
5. Mishra, S., Bhende, C.N.: Bacterial foraging technique-based optimized active power filter for load compensation. IEEE Transactions on Power Delivery 22(1), 457–465 (2007)
6. Singh, N., Pillay, V., Choonara, Y.E.: Advances in the treatment of Parkinson's disease. Progress in Neurobiology 81(1), 29–44 (2007)
7. Ho, A.K., Iansek, R., Marigliani, C., Bradshaw, J.L., Gates, S.: Speech impairment in a large sample of patients with Parkinson's disease. Behavioural Neurology 11(3), 131–137 (1999)
8. Little, M.A., McSharry, P.E., Hunter, E.J., Spielman, J., Ramig, L.O.: Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. IEEE Transactions on Biomedical Engineering 56(4), 1015–1022 (2009)
9. Das, R.: A comparison of multiple classification methods for diagnosis of Parkinson disease. Expert Systems with Applications 37(2), 1568–1572 (2010)
10. Ozcift, A., Gulten, A.: Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms. Computer Methods and Programs in Biomedicine 104(3), 443–451 (2011)
11. Chen, H.L., Huang, C.C., Yu, X.G., Xu, X., Sun, X., Wang, G., Wang, S.J.: An efficient diagnosis system for detection of Parkinson's disease using fuzzy< i> k</i>-nearest neighbor approach. Expert Systems with Applications 40(1), 263–271 (2013)
12. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001), `http://www.csie.ntu.edu.tw/cjlin/libsvm`
13. Chen, H.L., Yang, B., Wang, G., Liu, J., Xu, X., Wang, S.J., Liu, D.Y.: A novel bankruptcy prediction model based on an adaptive fuzzy< i> k</i>-nearest neighbor method. Knowledge-Based Systems 24(8), 1348–1359 (2011)

Parallel Bees Swarm Optimization for Association Rules Mining Using GPU Architecture

Youcef Djenouri and Habiba Drias

USTHB LRIA, Algiers, Algeria
y.djenouri@gmail.com, hdrias@hotmail.fr

Abstract. This paper addresses the problem of association rules mining with large scale data sets using bees behaviors. The bees swarm optimization method have been successfully running on small and medium data size. Nevertheless, when dealing with large benchmark, it is bluntly blocked. Additionally, graphic processor units are massively threaded providing highly intensive computing and very usable by the optimization research community. The parallelization of such method on GPU architecture can be deal large data sets as the case of WebDocs in real time. In this paper, the evaluation process of the solutions is parallelized. Experimental results reveal that the suggested method outperforms the sequential version at the order of ×70 in most data sets, furthermore, the WebDocs benchmark is handled with less than forty hours.

Keywords: bees swarm optimization, association rule mining, parallel algorithms, GPU architecture.

1 Introduction

Association Rules Mining (ARM) is one of the most important and well studied techniques of data mining tasks [1]. It aims at extracting frequent patterns, associations or causal structures among sets of items from a given transactional database. Formally, the association rule problem is as follows: let T be a set of transactions $\{t_1, t_2, \ldots, t_m\}$ representing a transactional database, and I be a set of m different items or attributes $\{i_1, i_2, \ldots, i_m\}$, an association rule is an implication of the form $X \rightarrow Y$ where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The itemset X is called antecedent while the itemset Y is called consequent and the rule means X implies Y.

Many exacts algorithms have been developed for solving ARM problem. Apriori [2] and FPgrowth [3] are the most used algorithms, Nonetheless, the exacts algorithms are high time consuming for large data sets.

Swarm intelligence algorithms have been successfully applied for association rules mining problem like: PSOARM [4], ACO_R [7], HBSO-TS [6], and BSO-ARM [5]. The experiments reveal that the bees swarm optimization outperforms

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 50–57, 2014.
© Springer International Publishing Switzerland 2014

ACO and PSO in terms of rules quality. However, when dealing with large instances like WebDocs (the huge benchmark among the web), the complexity time still expensive. In fact, the main challenge in sequential ARM algorithms is to handle massive data-sets. ARM problem has been parallelized in different ways. On each approach, the authors took advantage of the used parallel hardware like: CD, DD algorithms for distributed memory system [8], CCPD and PCCD for shared memory system [9]. However, All these algorithms have been implemented for traditional parallel architectures (supercomputers, clusters ..) which are still expensive and not always accessible for every one.

Motivating by the forcefully of graphic processors units, in this paper, we propose a parallel GPU-based approach for association rules mining problem. It is an extended version of the bees swarm optimization algorithm. The generation of solutions and the search process are done on CPU. However, to benefit from the massive GPU threaded, the evaluation process of the solutions have been performed concurrently on GPU. The results show the effectiveness of the parallel approach compared to the sequential one.

The rest of the paper is organized as follows: Section 2 relates the state of the art of ARM algorithms followed by a brief explanation of BSO-ARM algorithm. In section 4, we present the proposed algorithm PMES. Section 5 shows the results of our algorithm using several data sets. In section 6, we conclude by some remarks and futures perspective of the present work.

2 Related Works

ARM community is many investigating on GPU architecture, for this many algorithms based on GPU have been developed. The parallel ARM on GPU was first introduced by Wenbin *et al.* in [10]. They proposed two parallel versions of Apriori called PBI (Pure Bitmap Implementation) and TBI (Trie Bitmap Implementation) respectively. In PBI, the transactions datasets and the itemsets are represented by a bitmap data structure. Indeed, the itemset structure is a bitmap $(n * m)$ where n is the number of k itemsets and m is the number of its items. In this representation, bit $(i, j) = 1$ if itemsets i contains item j otherwise 0. Similarly, a transaction structure is a bitmap $(n * m)$ where n is the number of itemsets and m is the number of transactions. Here bit $(i, j) = 1$ if transaction j contains itemsets i, 0 otherwise.

In [11], a new algorithm called GPU-FPM is developed. It is based on Apriori-Like algorithm and uses a vertical representation of the data set because of the limitation of the memory size in the GPU. The speed up reached during the reported experimentations varies from ×10 to ×15.

Syed *et al.* in [12] proposed a new Apriori algorithm for GPU architecture which performs on two steps. In the first step, the generation of itemsets is done on GPU host where each thread block computes the support of a set of itemsets. In the second step, the generated itemsets are sent back to the CPU in order to generate the rules corresponding to each itemset and to determine the confidence of each rule. The main drawback of this algorithm is the cost of the CPU/GPU communications. The speed up of this algorithm reaches ×20 with a large data set.

Another Cuda-Apriori algorithm is proposed in [13]. First, the transactional dataset is divided among different threads. Then, k-candidate itemsets are generated and allocated on global memory. Each thread handles one candidate using only the portion of the dataset assigned to its block. In each iteration, a synchronization between blocks is done in order to compute the global support of each candidate. In [14], a GPApriori algorithm is proposed using two data structures in order to accelerate the itemsets counting. For small size datasets this algorithm reaches a speed up of (×100). Nevertheless, the speed up is reduced when the number of transactions increases du to thread divergence.

In [15] another work for maximal frequent itemsets mining on GPU is reported. This work is based on the main proprieties of GPU which are data parallelism and data independence. A tree structure is used to store the frequent itemsets where each node contains an itemsets, its support and a bitmap of each itemset. A bitmap is a boolean structure that represents all transactions containing a given itemset. This algorithm works well on large datasets. Nevertheless, it requires a high space memory and many pointer links which are difficult to manage on GPU architecture.

3 BSO-ARM Algorithm

In [5], we proposed BSO-ARM algorithm. The aim of this algorithm is to find one part of association rules respecting minimum support and minimum confidence constraints with reasonable time. Each rule is considered as one solution in the search space, each of which is represented by a vector S of N bits and their positions are defined as follows:

1. $S[i] = 0$ *if* the item i is not in the solution S.
2. $S[i] = 1$ *if* the item i belongs to the antecedent part of the solution S.
3. $S[i] = 2$ *if* the item i belongs to the consequent part of the solution S.

The algorithm can be decomposed into four steps (Neighborhood Search Computation, Search Area Determination, Fitness computing and Dancing Step).

Neighborhood Search Computation. The neighborhood search is obtained by changing from a given solution S one bit in random way. Based on this simple operation, N neighborhoods are created. Notice that this operation does not generate non admissible solutions.

Example 1

Consider the given solution: $S = \{1, 0, 0, 1, 2\}$

1. change the first bit in S : $S1 = \{0, 0, 0, 1, 2\}$
2. change the second bit in S: $S2 = \{1, 2, 0, 1, 2\}$
3. change the third bit in S:$S3 = \{1, 0, 1, 1, 2\}$
4. change the fourth bit S: $S4 = \{1, 0, 0, 0, 2\}$
5. change the fifth bit S: $S5 = \{1, 0, 0, 1, 0\}$

Search Area Determination. It determines K search spaces, each one is associated to a bee. The j^{th} bee builds its own search area by changing successively in the solution *Sref* the bits $j + i \times Flip$ where $i \in [0..N-1]$, $j \in [1..K]$ and *Flip* is a given parameter. This strategy can be used if and only if $K \leq \frac{N}{Flip}$. If the distance between solutions is the number of different bits, then the distance between the bees and the solution reference is equal to $\frac{N}{Flip}$.

Fitness Computing. In this step for each generated solution (rule), the entire transactional database should be scanned. The solution fitness is based on the support and the confidence measures as:

$$Fitness(s) = \alpha \times conf(s) + \beta \times sup(s) \tag{1}$$

This function should be maximized. For each invalid solution s where $Sup(s) < Minsup$ or $Conf(s) < MinConf$, the *Fitness(s)* is set to -1 and the solution is rejected.

Dancing Step. Each bee puts in the dance table the best rule found among its search. The communication between bees is done in order to find the best dance (the best rule) which becomes the reference solution for the next pass.

The general functioning of the algorithm is as follows: First, the solution reference (*Sref*) is initialized randomly so that each element of *Sref* belongs to {0,1,2}.

After that, except the Fitness Computing which is applied for each generated solution, the other steps are repeated in the order until *IMAX* is reached.

4 Parallel Single Evaluation of Solution Algorithm

The proposed algorithm parallel single evaluation of solution (PSES for short) is based on the master/slave paradigm. The master is executed on CPU and the slave is offloaded to the GPU. First, The master initializes randomly the solution reference. After that, it determines regions of the whole bees by generating the neighbors of each bee. Unlike BSO-ARM, single solution is evaluated on GPU in parallel. After, the master receives back the fitness of all rules, each bee calculates sequentially the best rule and puts it in the table dance. The best rule of the dance table become the solution reference for the next iteration. This combined CPU/GPU process is repeated until the maximum number of iterations is reached. We opted for a mapping in which all threads are mapped to one rule. Threads of the same block are launched to calculate collaboratively the fitness of a single rule with one packet of transactions. Therefore, we have as many threads as transactions. The transactions are subdivided into subsets and each subset set_i is assigned to one bloc so that each thread calculates only one transaction. After that a sum reduction is applied to aggregate the fitness value.

Such a strategy allows us to benefit from the massively parallel power of GPU by launching a large number of threads per rule. Indeed, in PSES, only the threads must synchronize after each iteration to process sum reduction technique, thus $N \times K \times IMAX$ synchronization during all the lifetime of the algorithm. The general algorithm of the GPU kernel is given in Algorithm 1.

Algorithm 1. The GPU kernel

```
Input: Sol: Single solution
Freq[] Array of integer
recuperate Sol from CPU
initialize Freq to zero
$idt$ = blockIdx.x × blockDim.x + threadIdx.x.
for i=0 to $nb$ transactions do
    if Sol$\in t_{idt}$ then
        freq[idt][i]=1.
    else
        freq[idt][i]=0.
    end if
end for
Evaluation(Sol)=Sum_Reduction(freq).
cudaMemcpy(Evaluation(Sol), cudaMemcpyDeviceToHost).
```

First, GPU recuperates the single sol containing the set of solutions generated on CPU. It initializes freq by zero. Then, each thread evaluates one solution with one transaction.

5 Performance Evaluation

To evaluate the performance of the proposed approach PSES, several data sets of different size are considered. The data sets are the well-known scientific databases that are frequently used in data mining community (Frequent and Mining Dataset Repository [17] and Bilkent University Function Approximation Repository[16].

From the smallest benchmark (*Bolts* data set) to the largest one (*WebDocs* data set), the used data sets are divided according to the number of transactions into three categories (small, average, large). The description of the different used data sets are presented in Table 1. Notice that the data sets differ according to the average size of items. On one hand, there are big data sets with a few number of items per transaction. On the other hand, there are other small data sets with a significant number of items per transaction. For instance, the number of transactions on the *Connect* data set is 100000 and the average items per transaction is only 10. Whereas, in the IBM data set the number of transactions is only 1000 and the number of items per transaction is 20.

Table 1. Data sets description

Data Set Type	Data set Name	Transactions Size	Item Size	Average Size
Small	Bolts	40	8	8
	Sleep	56	8	8
	Basket ball	96	5	5
	IBM Quest Std.	1000	40	20
	Quake	2178	4	4
	Chess	3196	75	37
	Mushroom	8124	119	23
Average	Pumbs_star	40385	7116	50
	BMS-WebView-1	59602	497	2.5
	BMS-WebView-2	77512	3340	5
	Korasak	80769	7116	50
	retail	88162	16469	10
	Connect	100000	999	10
Large	BMP POS	515597	1657	2.5
	WebDocs	1692082	526765	-

Table 2. Runtime of the proposed approach with different data sets (in Sec)

Data Set Type	Data set Name	BSO-ARM	PSES
Small	Bolts	9	2.5
	Sleep	15	5
	Pollution	22	6
	Basket ball	35	12
	IBM Q.S.	618	50
	Quake	80	20
	Chess	149956	125
	Mushroom	28815	350
Average	Pumbs_star	144120	1550
	BMS-W.V.-1	180524	6200
	BMS-W.V.-2	249985	1850
	Korasak	258451	1948
	Retail	299658	2185
	Connect	300985	4485
Large	BMP POS	Stopped After 15 Days	22565
	WebDocs	Stopped After 15 Days	45965

The suggested approach has been implemented using C-CUDA 4.0 and the experiments have been carried out using a CPU host coupled with a GPU device. The CPU host is a 64-bit quad-core Intel Xeon E5520 having a clock speed of 2.27GHz. The GPU device is an Nvidia Tesla C2075 with 448 CUDA cores (14 multiprocessors with 32 cores each), a clock speed of 1.15GHz, a 2.8GB global memory, a 49.15KB shared memory, and a warp size of 32. Both the CPU and GPU are used in single precision.

Table 2 presents the execution time of the sequential and parallel version of BSO-ARM. In order to well exploring the search space, the number of bees K, respectively the number of iterations $IMAX$ are set to 20, respectively 100.

The parallel version outperforms the sequential one in all cases. Furthermore, the GPU-based parallelization allowed us to solve two challenging large data sets (BMP POS and Web Docs) containing more than 1.5 millions of transactions and more than 0.5 million of items. To the best of our knowledge, these these two data sets have newer been solved before in the literature. Indeed, BSO-ARM blocked after 12 days whereas it takes only few hours using PSES.

6 Conclusion

In this paper, we proposed a new algorithm for association rules mining on GPU architecture. It is based on the bees behaviors, we first generate the solutions on CPU, then, the evaluation of each solution is performed in parallel using GPU threaded. The intensive multi-threaded provided on GPU conduct us to perform the single evaluation of solution at the same time. In fact, each thread is mapped with one transaction, this permits to accelerate the process of the evaluation. The experiments show that the parallel approach outperforms the sequential one in terms of the execution time. The results also reveal that using the massive threaded in GPU and the intelligent bees, the largest transactions base on the web is mined in real time.

References

1. Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques. Morgan Kaufmann (2006)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. ACM SIGMOD Record 22(2) (1993)
3. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. ACM SIGMOD Record 29(2) (2000)
4. Kuo, R.J., Chao, C.M., Chiu, Y.T.: Application of particle swarm optimization to association rule mining. Applied Soft Computing 11(1), 326–336 (2011)
5. Djenouri, Y., Drias, H., Habbas, Z., Mosteghanemi, H.: Bees Swarm Optimization for Web Association Rule Mining. In: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 3, pp. 142–146. IEEE (2012)
6. Djenouri, Y., Drias, H., Chemchem, A.: A hybrid Bees Swarm Optimization and Tabu Search algorithm for Association rule mining. In: 2013 World Congress on Nature and Biologically Inspired Computing (NaBIC). IEEE (2013)

7. Moslehi, P., et al.: Multi-objective Numeric Association Rules Mining via Ant Colony Optimization for Continuous Domains without Specifying Minimum Support and Minimum Confidence. International Journal of Computer Science (2008)
8. Agrawal, R., Shafer, J.C.: Parallel mining of association rules. IEEE Transactions on Knowledge and Data Engineering 8(6) (1996)
9. Parthasarathy, S., et al.: Parallel data mining for association rules on shared-memory systems. Knowledge and Information Systems 3(1) (2001)
10. Fang, W., et al.: Frequent itemset mining on graphics processors. In: Proceedings of the Fifth International Workshop on Data Management on New Hardware. ACM (2009)
11. Zhou, J., Yu, K.-M., Wu, B.-C.: Parallel frequent patterns mining algorithm on GPU. In: IEEE International Conference on Systems Man and Cybernetics (SMC). IEEE (2010)
12. Adil, S.H., Qamar, S.: Implementation of association rule mining using CUDA. In: International Conference on Emerging Technologies, ICET 2009. IEEE (2009)
13. Cui, Q., Guo, X.: Research on Parallel Association Rules Mining on GPU. In: Yang, Y., Ma, M. (eds.) Proceedings of the 2nd International Conference on Green Communications and Networks. LNEE, vol. 224, pp. 215–222. Springer, Heidelberg (2012)
14. Zhang, F., Zhang, Y., Bakos, J.: Gpapriori: Gpu-accelerated frequent itemset mining. In: IEEE International Conference on Cluster Computing (CLUSTER). IEEE (2011)
15. Li, H., Zhang, N.: Mining maximal frequent itemsets on graphics processors. In: Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), vol. 3. IEEE (2010)
16. Guvenir, H.A., Uysal, I.: Bilkent university function approximation repository, 2012-03-12 (2000), `http://funapp.cs.bilkent.edu.tr/DataSets`
17. Goethals, B., Zaki, M.J.: Frequent itemset mining implementations repository. This site contains a wide-variety of algorithms for mining frequent, closed, and maximal itemsets (2003), `http://fimi.cs.helsinki.fi`

A Method for Ripple Simulation Based on GPU

Xianjun Chen[1,2], Yanmei Wang[3,*], and Yongsong Zhan[1]

[1] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology,
Guilin, Guangxi, 541004, PR China
[2] Information Engineering School, Haikou College of Economics,
Haikou, Hainan, 571127, PR China
[3] QiongTai Teachers College, Haikou, Hainan, 571127, PR China
{Yanmei Wang,hingini}@126.com

Abstract. To improve the simulation of ripple on a personal workstation, a novel vector algebra model based on Graphic Process Unit (GPU) is proposed. First, the data structures and rules for data operation are established to meet the needs of vector algebra model. Second, the physical equation governing ripple motion is transformed discretely for vector multiplication, which will be solved by the Conjugate Gradient Method. Finally, the simulation of ripple is achieved from the height map providing normal information used by the calculation of light reflection and refraction in real time. Experiment results show that the method is robust and efficient to achieve real-time ripple simulation by making full use of the excellent computation power of programmable GPU.

Keywords: Ripple simulation, GPU, Vector algebra operation, Conjugate Gradient Method.

1 Introduction

As one of the most intriguing problems in computer graphics, the simulation of ripple has drawn the attention of a great sum of researchers. Ripples are everywhere in the nature, ranging from the streams to the rivers, from the pools to the oceans. In the applications of computer games and virtual reality, it is necessary to immerse players into plausible virtual worlds, which shall be constructed by the photorealistic simulation of natural scenes, such as ripple, smoke, and so on. Also, animators can also benefit from ripple simulation to achieve realistic effects in real time and improve the product efficiency of cartoon. With developing computer graphics, there exist many models that attempt to fake fluid-like effects. However, it is not easy to mimic the complexities and subtleties of ripple motion in a convincing manner on a graphic workstation in real time.

In this paper, a novel GPU based vector algebra operation model is proposed to improve the simulation of ripple, which is physically described by the fluid equation. First of all, the data structures and rules for data operation are established to meet the

[*] Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 58–65, 2014.
© Springer International Publishing Switzerland 2014

needs of vector algebra operation model. Then, the fluid dynamics equation governing ripple is transformed discretely for vector multiplication, which is to be solved by Conjugate Gradient Method. Finally, ripple rendering is achieved from the height map providing normal information used by the calculation of light reflection and refraction.

The rest of the paper is organized as follows. Related work is discussed in section 2. Section 3 introduces the GPU based vector algebra operation, including the data structures and rules for data operation. Section 4 gives a description on the Conjugate Gradient Method and the dynamics equation depicting water wave. Discretization of the fluid equation for vector multiplication is also included in this section. Section 5 presents the experimental result of a running system instance. Conclusions are drawn in section 6.

2 Related Work

In the graphics community, the early work for water phenomenon modeling placed emphasis on representations of the water surface as a parametric function, which could be animated over time to simulate wave transport[1-3]. But they are unable to easily deal with complex three-dimensional behaviors such as flow around objects and dynamically changing boundaries. To obtain water models which could be used in a dynamic animation environment, researchers turned to use two-dimensional approximations to the full 3D fluid equations[4]. Kass[5] approximated the 2D shallow water equations to get a dynamic height field surface that interacted with a static ground object. A pressure defined height field formulation was used by Chen[6] in fluid simulations with moving obstacles. O'Brien[7] simulated splashing liquids by combining a particle system and height field, while Miller[8] used viscous springs between particles to achieve dynamic flow in 3D. Terzopoulos[9] simulated melting deformable solids using a molecular dynamics approach to simulate the particles in the liquid phase. The simulation of complex water effects using the full 3D Navier-Stokes equations has been based upon the large amount of research done by the computational fluid dynamics community over the past 60 years. Foster[10] developed a 3D Navier-Stokes methodology for the realistic animation of liquids. Stam[11] replaced their finite difference scheme with a semi-Lagrangian method to achieve significant performance improvements at the cost of increased rotational damping. Foster[12] made significant contributions to the simulation and control of three dimensional fluid simulations through the introduction of a hybrid liquid volume model combining implicit surfaces and massless marker particles, the formulation of plausible boundary conditions for moving objects in a liquid, the use of an efficient iterative method to solve for the pressure, and a time step subcycling scheme for the particle and implicit surface evolution equations.

On the other hand, graphics hardware has undergone a true revolution in the past ten years. It went from being a simple memory device to a configurable unit and a fully programmable parallel processor. Although designed for fast polygon rendering, graphics hardware has been extended to various applications of general-purpose computations[13-14]. Harris[15] have implemented on the GPU the coupled map lattice, and have simulated cloud dynamics using partial differential equations. Goodnight[16] have implemented the multi-grid method on the GPU, and have applied it to heat transfer and modeling of fluid mechanics. Boltz[17] have also developed a GPU-based multi-grid solver. In addition, a conjugate-gradient solver is proposed on the GPU with a sparse matrix representation, which has been applied to the Navier-Stokes equations. At the same time, Kruger[18] has presented the GPU implementation of several linear algebra operators, which have been used to solve the Navier-Stokes equations as well. In all the above works, the computation of fluid dynamics is translated from the CPU to the GPU.

3 Vector Algebra Operation

As a main problem in the field of applied mathematics, the numerical solution for differential equations has been of prime importance in many applications of physical simulation and image procession. Transformed discretely to be linear, the differential equations are now widely used by 3D graphics applications for natural phenomena simulation. To solve the linear equations on GPU, the model of vector algebra operation is proposed to be composed of data structures and rules for data operation, both of which can be implemented by object-oriented program and extended freely.

3.1 Data Structures

The proposed model consists of three data types: float, vector and matrix. With the improvement of architecture in these years, the current GPU have been good at texture access, making it proper to define a float as a texture with the size of 1×1. A vector can be considered as either a one dimension (1D) texture or a two dimension (2D) texture. The latter is preferable in our model, for it has a better GPU support than the former. In practice, most of the current GPU set a limit to the total of 1D texture, namely 4096, which is not applicable for the complex numerical solution of differential equation. Furthermore, during the course of linear equation solving, the median is usually defined as vector, which is to be rendered as 2D texture to achieve high performance. In most computer graphics applications, matrix is usually considered as 2D texture, resulting in a high texture usage. As many matrices may be composed of a great majority of zero values and few non-zero values, it is necessary to define the matrix according to its attribute. The dense matrix can be divided into a set of vectors, which shall be considered as 2D textures. In the band matrix, the non-zero values have a distribution of band, which is considered to be a vector denoted by 2D texture as well. As to the sparse matrix, the non-zero values are distributed randomly, making it improper to be described by texture. To make full use of the parallelization operation of GPU, the sparse matrix is defined by a vertex buffer as figure 1.

Following their priority order in the sparse matrix, all the elements are inputted to the vertex buffer in turn. Each vertex will be provided with 4 non-zero values. The rows of sparse matrix are indexed by the vertex coordination, which can be adjusted by program parameter input. The texture coordination of each matrix element shall be the same as the final vertex coordination acquired by the procession of model transformation, view transformation and projection. Obviously, it can be seen from the above structure that each vertex is equipped with 6 texture coordination values, where (tu_0, tv_0), (tu_1, tv_1), (tu_2, tv_2), (tu_3, tv_4) are used as indexes for the 4 elements, (val0, val1, val2, val3) for their values, and (posX, posY) for index of the output. This definition is helpful for the multiplication between matrix and vector.

```
struct SPARSEMATRIXVERTEX
{
   FLOAT x,y,z;
   FLOAT tu_0,tv_0;
   FLOAT tu_1,tv_1;
   FLOAT tu_2,tv_2;
   FLOAT tu_3,tv_3;
   FLOAT val0,val1,val2,val3;
   FLOAT posX,posY;
   static const DWORD FVF;
};
```

Fig. 1. Data structures

3.2 Data Operation

The common data operation in the proposed model includes vector operation, vector reduction and multiplication of matrix and vector. To achieve vector addition, the following sequence of steps is performed: (1) a view frustum is set to cover a number of pixels, which is equal to the number of vector elements, and the target vector is set as Render Target; (2) a quad is rendered to cover the entire view port; (3) rasterization is implemented by the vertex program, and a mapping relationship is established between the vector elements and pixels, which can be accessed by the fragment program via texture coordination; (4) parallel processing of all the pixels by GPU is accomplished by fragment program, whose output is the target texture. The other vector operations, such as vector subtraction, multiplication between vector and scalar, and etc. can be implemented in the same way.

The operation of vector reduction is to count all the element values. Given a vector defined as a texture of $n \times n$, reduction can be implemented in $\log(n)$ rendering cycles. In the fragment program, the mean value of 4 neighboring texels is calculated and written into the Render Target, which is considered as input for the next rendering cycle. The above operation will repeat till the completion of reduction, as shown in figure 2.

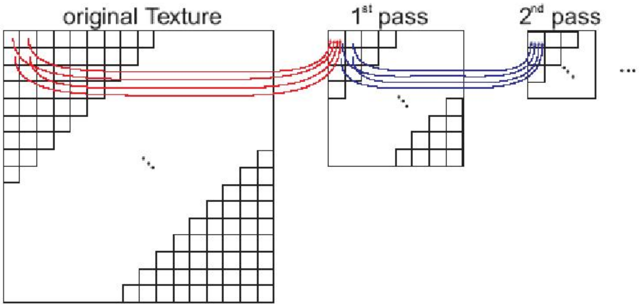


Fig. 2. Data operation

4 Ripple Rendering

The motion of ripple is controlled by 2D wave equation, which is of prime importance for the vivid simulation of wave animation. The numerical solution for this physical equation is as follows: (1) the equation is transformed to be linear by discretization; (2) the linear equation is solved by the Conjugate Gradient Method, which is completely based on the proposed vector algebra operation model accelerated by GPU.

4.1 Conjugate Gradient Method

As an applicable iteration process for solving linear equations, the Conjugate Gradient Method mainly deals with the cases with symmetric positive definite coefficient matrix. Given the linear equations as $Ax = b$, where $A \in R^{n \times n}$, the vector sequences of iteration and remainder are constructed, and the search directions are updated. These operations shall be iterated to achieve the ideal computation precision.

The procedure for Conjugate Gradient Method includes a pre-computation and a main cycle. In the course of pre-computation, the following steps shall be performed: (1) given $\forall x_0 \in R^n$; (2) $r_0 = b - Ax_0$; (3) $p_0 = r_0$; (4) a threshold constant ε is set, and $k = 0$.

4.2 Wave Equation

The inherent relationship between wave height and velocity, time and space can be described by the physical dynamics equation, which is defined as following.

$$c^2\left(\frac{\partial^2 y}{\partial x^2} + \frac{\partial^2 y}{\partial z^2}\right) = \frac{\partial^2 y}{\partial t^2}$$

where x-z is the water surface, y is height, t is time and c is wave velocity. To achieve the numerical solution for wave equation, it shall be transformed to algebraic equation by Taylor series and Centre Differentia Method, which can be defined as follows.

$$f(x)=f(x+\Delta h)+f'(x+\Delta h)\times\Delta h+\Omega(\Delta h)$$

$$\Rightarrow (\frac{\partial y}{\partial h})_{i,j}=\begin{cases}\frac{y_{i+1,j}-y_{i,j}}{\Delta h}+\mathrm{O}(\Delta h)\\ \frac{y_{i,j}-y_{i-1,j}}{\Delta h}+\mathrm{O}(\Delta h)\\ \frac{y_{i+1,j}-y_{i-1,j}}{2\Delta h}+\mathrm{O}(\Delta h)^2\end{cases}$$

$$\Rightarrow (\frac{\partial^2 y}{\partial x^2})_{i,j}=\frac{y_{i+1,j}-2y_{i,j}+y_{i-1,j}}{(\Delta h)^2}+\mathrm{O}(\Delta h)^2$$

Finally, the 2D discrete wave equation can be denoted as follows, where $\beta=\frac{c^2(\Delta t)^2}{(\Delta h)^2}$.

$$\frac{y_{i,j}^{t+1}-2y_{i,j}^{t}+y_{i,j}^{t-1}}{(\Delta t)^2}=c^2(\frac{y_{i+1,j}^{t}-2y_{i,j}^{t}+y_{i-1,j}^{t}}{(\Delta x)^2}+\frac{y_{i,j+1}^{t}-2y_{i,j}^{t}+y_{i,j-1}^{t}}{(\Delta z)^2})\Big|_{x=z=h}$$

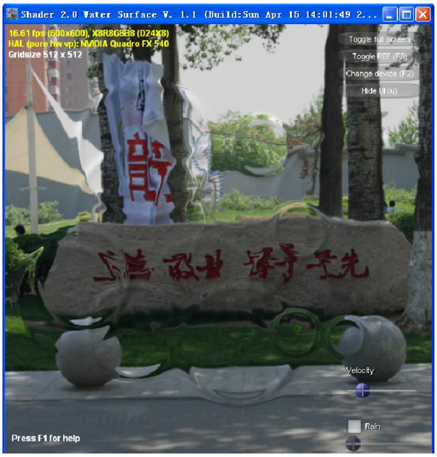
$$\Leftrightarrow \frac{y_{i,j}^{t+1}-2y_{i,j}^{t}+y_{i,j}^{t-1}}{(\Delta t)^2}=c^2(\frac{y_{i+1,j}^{t}-2y_{i,j}^{t}+y_{i-1,j}^{t}}{(\Delta h)^2}+\frac{y_{i,j+1}^{t}-2y_{i,j}^{t}+y_{i,j-1}^{t}}{(\Delta h)^2})$$

$$\Leftrightarrow y_{i,j}^{t+1}=\beta(y_{i+1,j}^{t}+y_{i-1,j}^{t}+y_{i,j+1}^{t}+y_{i,j-1}^{t})+(2-4\beta)y_{i,j}^{t}-y_{i,j}^{t-1}$$

5 Experiment

Based on the proposed method, an applicable implemented with C++ computer program, DirectX 9.0 and the hardware rendering program of HLSL is shown as figure 3. Furthermore, figure 4 gives a convincing representation of ripple on the workstation with NVIDIA Quadro Fx540. In this case, the 2D wave surface is presented discretely as meshes of 512X512, and simulated in real-time with the frame rate of 16fps.

In order to demonstrate the performance of the proposed model, two common-used GPU have been employed in the test, including NVIDIA Quadro FX 540 (with the memory of 256M) and NVIDIA FX 8800 (with the memory of 640M). The resultant data of table 1-2 show the efficiency and robustness of our method.



Shader 2.0 Water Surface V. 1.1 (Build:Sun Apr 15 14:01:49 2...

16.61 fps (600x600), X8R8G8B8 (D24X8)
HAL (pure hw vp): NVIDIA Quadro FX 540
Gridsize 512 x 512

Fig. 3. Instance of system implementation

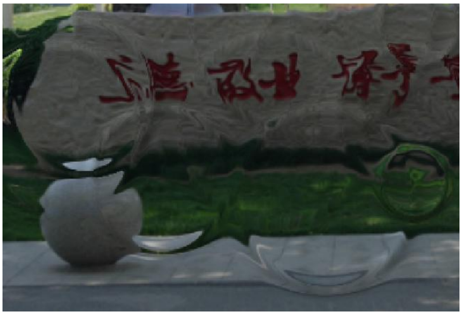


Fig. 4. The real-time ripple simulation

Table 1. Test result of NVIDIA Quadro FX 540.

	Image Resolution		
	512 x 512	512 x 256	256 x 256
Vector Reduction(ms)	1.00	0.71	0.62
Vector Addition(ms)	1.44	0.61	0.12
Frame Rate(fps)	17	32	64

Table 2. Test result of NVIDIA FX 8800.

	Image Resolution		
	512 x 512	512 x 256	256 x 256
Vector Reduction(ms)	0.10295	0.10076	0.09052
Vector Addition(ms)	0.01404	0.0139	0.0126
Frame Rate(fps)	295	423	440

6 Conclusion

In this paper, a novel GPU based vector algebra operation model is proposed to improve the simulation of water surface. The data structures and rules for data operation are established to meet the needs of vector algebra operation model, and the fluid dynamics equation governing ripple is transformed discretely for vector multiplication. Experimental results show the robustness and efficiency of the proposed method for the real-time simulation of water surface on GPU.

Acknowledgments. This research work is supported by the grant of Guangxi science and technology development project (No: 1355011-5), the grant of Guangxi Key Laboratory of Trusted Software of Guilin University of Electronic Technology (No:

kx201309), the grant of Guangxi Education Department (No: SK13LX139)，the grant of Guangxi Undergraduate Training Programs for Innovation and Entrepreneurship (No:20121059519), the grant of Hainan Natural Science Foundation (No: 613169), and the Universities and colleges Science Research Foundation of Hainan (No: Hjkj2013-48).

References

1. Fournier, A., Reeves, W.T.: A simple model of ocean waves. In: Proc. SIGGRAPH ACM, pp. 75–84 (1986)
2. Peachy, D.: Modeling waves and surf. In: Proc. ACM SIGGRAPH, pp. 65–74 (1986)
3. Schachter, B.: Long crested wave models. Computer Graphics and Image Processing 12, 187–201 (1980)
4. Chorin, A.J., Marsden, J.E.: A mathematical introduction to fluid mechanics. Texts in Applied Mathematics, vol. 4 (1990)
5. Kass, M., Miller, G.: Rapid, stable fluid dynamics for computer graphics. In: Proc. ACM SIGGRAPH, pp. 49–57 (1990)
6. Chen, J., Lobo, N.: Toward interactive-rate simulation of fluids with moving obstacles using the Navier-Stokes Equations. Graphical Models and Image Processing 57, 107–116 (1994)
7. O'Brien, J., Hodgins, J.: Dynamic simulation of splashing fluids. In: Proc. Computer Animation, pp. 198–205 (1995)
8. Miller, G., Pearce, A.: Globular dynamics: a connected particle system for animating viscous fluids. Computers and Graphics 13, 305–309 (1989)
9. Terzopoulos, D., Platt, J., Fleischer, K.: Heating and melting deformable models (from goop to glop). In: Proc. Graphics Interface, pp. 219–226 (1989)
10. Foster, N., Metaxas, D.: Realistic animation of liquids. Graphical Models and Image Processing 58, 471–483 (1996)
11. Stam, J.: Stable fluids. In: Proc. ACM SIGGRAPH, pp. 121–128 (1999)
12. Foster, N., Fedkiw, R.: Practical animation of liquids. In: Proc. ACM SIGGRAPH, pp. 23–30 (2001)
13. Hoff, K., Culver, T.: Fast computation of generalized voronoi diagrams using graphics hardware. In: Proc. ACM SIGGRAPH, pp. 277–286 (1999)
14. Trendall, C., Stewart, A.J.: General calculations using graphics hardware with applications to interactive caustics. In: Proc. Eurographics Workshop on Rendering, pp. 287–298 (2000)
15. Harris, M., Coombe, G.: Physically-based visual simulation on graphics hardware. In: Proc. SIGGRAPH/Eurographics Workshop on Graphics Hardware, pp. 109–118 (2002)
16. Goodnight, N., Woolley, C.: A multi-grid solver for boundary value problems using programmable graphics hardware. In: Proc. SIGGRAPH/Eurographics Workshop on Graphics Hardware (2003)
17. Bolz, J., Farmer, I., Grinspun, E.: Sparse matrix solvers on the GPU: conjugate gradients and multi-grid. ACM Transactions on Graphics 22, 917–924 (2003)
18. Kruger, J., Westermann, R.: Linear algebra operators for GPU implementation of numerical algorithms. ACM Transactions on Graphics 22, 908–916 (2003)

cuROB: A GPU-Based Test Suit for Real-Parameter Optimization

Ke Ding and Ying Tan*

Key Laboratory of Machine Perception and Intelligence (MOE), Peking University,
Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, China
{keding,ytan}@pku.edu.cn

Abstract. Benchmarking is key for developing and comparing optimization algorithms. In this paper, a GPU-based test suit for real-parameter optimization, dubbed cuROB, is introduced. Test functions of diverse properties are included within cuROB and implemented efficiently with CUDA. Speedup of one order of magnitude can be achieved in comparison with CPU-based benchmark of CEC'14.

Keywords: Optimization Methods, Optimization Benchmark, GPU, CUDA.

1 Introduction

Proposed algorithms are usually tested on benchmark for comparing both performance and efficiency. However, as it can be a very tedious task to select and implement test functions rigorously. Thanks to GPUs' massive parallelism, a GPU-based optimization function suit will be beneficial to test and compare optimization algorithms.

Based on the well known CPU-based benchmarks presented in [1,2,3], we proposed a CUDA-based real parameter optimization test suit, called cuROB, targeting on GPUs. We think cuROB can be helpful for assessing GPU-based optimization algorithms, and hopefully, conventional CPU-based algorithms can benefit from cuROB's fast execution.

Considering the fact that research on the single objective optimization algorithms is the basis of the research on the more complex optimization algorithms such as constrained optimization algorithms, multi-objective optimizations algorithms and so forth, in this first release of cuROB a suit of single objective real-parameter optimization function are defined and implemented.

The test functions are selected according to the following criteria: 1) the functions should be scalable in dimension so that algorithms can be tested under various complexity; 2) the expressions of the functions should be with good parallelism, thus efficient implementation is possible on GPUs; 3) the functions should be comprehensible such that algorithm behaviours can be analysed in

* To whom the correspondence should be addressed.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 66–78, 2014.
© Springer International Publishing Switzerland 2014

the topological context; 4) last but most important, the test suit should cover functions of various properties in order to get a systematic evaluation of the optimization algorithms.

The source code and a sample can be download from `code.google.com/p/curob/`.

1.1 Symbol Conventions and Definitions

Symbols and definitions used in the report are described in the following. By default, all vectors refer to column vectors, and are depicted by lowercase letter and typeset in bold.

- $[\cdot]$ indicates the nearest integer value
- $\lfloor \cdot \rfloor$ indicates the largest integer less than or equal to
- $\mathbf{x}_i$ denotes i-th element of vector $\mathbf{x}$
- $f(\cdot)$, $g(\cdot)$ and $G(\cdot)$ multi-variable functions
- f_{opt} optimal (minimal) value of function f
- $\mathbf{x}^{opt}$ optimal solution vector, such that $f(\mathbf{x}^{opt}) = f_{opt}$
- $\mathbf{R}$ normalized orthogonal matrix for rotation
- D dimension
- $\mathbf{1} = (1, \ldots, 1)^T$ all one vector

1.2 General Setup

The general setup of the test suit is presented as follows.

- **Dimensions** The test suit is scalable in terms of dimension. Within the hardware limit, any dimension $D \geq 2$ works. However, to construct a real hybrid function, D should be at least 10.
- **Search Space** All functions are defined and can be evaluated over $\mathcal{R}^D$, while the actual search domain is given as $[-100, 100]^D$.
- f_{opt} All functions, by definition, have a minimal value of 0, a bias (f_{opt}) can be added to each function. The selection can be arbitrary, f_{opt} for each function in the test suit is listed in Tab. 1.
- $\mathbf{x}^{opt}$ The optimum point of each function is located at original. $\mathbf{x}^{opt}$ which is randomly distributed in $[-70, 70]^D$, is selected as the new optimum.
- **Rotation Matrix** To derive non-separable functions from separable ones, the search space is rotated by a normalized orthogonal matrix $\mathbf{R}$. For a given function in one dimension, a different $\mathbf{R}$ is used. Variables are divided into three (almost) equal-sized subcomponents randomly. The rotation matrix for each subcomponent is generated from standard normally distributed entries by Gram-Schmidt orthonormalization. Then, these matrices consist of the $\mathbf{R}$ actually used.

1.3 CUDA Interface and Implementation

A simple description of the interface and implementation is given in the following. For detail, see the source code and the accompanied readme file.

Interface. Only benchmark.h need to be included to access the test functions, and the CUDA file benchmark.cu need be compiled and linked. Before the compiling start, two macro, DIM and MAX_CONCURRENCY should be modified accordingly. DIM defines the dimension of the test suit to used while MAX_CONCURRENCY controls the most function evaluations be invoked concurrently. As memory needed to be pre-allocated, limited by the hardware, don't set MAX_CONCURRENCY greater than actually used.

Host interface function initialize () accomplish all initialization tasks, so must be called before any test function can be evaluated. Allocated resource is released by host interface function dispose ().

Both double precision and single precision are supported through func_evaluate () and func_evaluatef () respectively. Take note that device pointers should be passed to these two functions. For the convenience of CPU code, C interfaces are provided, with h_func_evaluate for double precision and h_func_evaluatef for single precision. (In fact, they are just wrappers of the GPU interfaces.)

Efficiency Concerns. When configuration of the suit, some should be taken care for the sake of efficiency. It is better to evaluation a batch of vectors than many smaller. Dimension is a fold of 32 (the warp size) can more efficient. For example, dimension of 96 is much more efficient than 100, even though 100 is little greater than 96.

1.4 Test Suite Summary

The test functions fall into four categories: unimodal functions, basic multimodal functions, hybrid functions and composition functions. The summary of the suit is listed in Tab. 1. Detailed information of each function will given in the following sections.

2 Speedup

Under different hardware, various speedups can be achieved. 30 functions are the same as CEC'14 benchmark. We test the cuROB's speedup with these 30 functions under the following settings: Windows 7 SP1 x64 running on Intel i5-2310 CPU with NVIDIA 560 Ti, the CUDA version is 5.5. 50 evaluations were performed concurrently and repeated 1000 runs. The evaluation data were generated randomly from uniform distribution.

The speedups with respect to different dimension are listed by Tab. 2 (single precision) and Tab. 3 (double precision). Notice that the corresponding dimensions of cuROB are 10, 32, 64 and 96 respectively and the numbers are as in Tab. 1

Fig. 1 demonstrates the overall speedup for each dimension. On average, cuROB is never slower than its CPU-base CEC'14 benchmark, and speedup of one order of magnitude can be achieved when dimension is high. Single precision is more efficient than double precision as far as execution time is concerned.

Table 1. Summary of cuROB's Test Functions

	No.	Functions	ID	Description
Unimodal Functions	0	Rotated Sphere	SPHERE	Optimum easy to track
	1	Rotated Ellipsoid	ELLIPSOID	
	2	Rotated Elliptic	ELLIPTIC	Optimum hard to track
	3	Rotated Discus	DISCUS	
	4	Rotated Bent Cigar	CIGAR	
	5	Rotated Different Powers	POWERS	
	6	Rotated Sharp Valley	SHARPV	
Basic Multi-modal Functions	7	Rotated Step	STEP	With adepuate global structure
	8	Rotated Weierstrass	WEIERSTRASS	
	9	Rotated Griewank	GRIEWANK	
	10	Rastrigin	RARSTRIGIN_U	
	11	Rotated Rastrigin	RARSTRIGIN	
	12	Rotated Schaffer's F7	SCHAFFERSF7	
	13	Rotated Expanded Griewank plus Rosenbrock	GRIE_ROSEN	
	14	Rotated Rosenbrock	ROSENBROCK	With weak global structure
	15	Modified Schwefel	SCHWEFEL_U	
	16	Rotated Modified Schwefel	SCHWEFEL	
	17	Rotated Katsuura	KATSUURA	
	18	Rotated Lunacek bi-Rastrigin	LUNACEK	
	19	Rotated Ackley	ACKLEY	
	20	Rotated HappyCat	HAPPYCAT	
	21	Rotated HGBat	HGBAT	
	22	Rotated Expanded Schaffer's F6	SCHAFFERSF6	
Hybrid Functions	23	Hybrid Function 1	HYBRID1	With different properties for different variables subcomponents
	24	Hybrid Function 2	HYBRID2	
	25	Hybrid Function 3	HYBRID3	
	26	Hybrid Function 4	HYBRID4	
	27	Hybrid Function 5	HYBRID5	
	28	Hybrid Function 6	HYBRID6	
Composition Functions	29	Composition Function 1	COMPOSITION1	Properties similar to particular sub-function when approaching the corresponding optimum
	30	Composition Function 2	COMPOSITION2	
	31	Composition Function 3	COMPOSITION3	
	32	Composition Function 4	COMPOSITION4	
	33	Composition Function 5	COMPOSITION5	
	34	Composition Function 6	COMPOSITION6	
	35	Composition Function 7	COMPOSITION7	
	36	Composition Function 8	COMPOSITION8	
Search Space: $[-100, 100]^D$, $f_{opt} = 100$				

Table 2. Speedup (single Precision)

D	NO.3	NO.4	NO.5	NO.8	NO.9	NO.10	NO.11	NO.13	NO.14	NO.15
10	0.59	0.20	0.18	12.23	0.49	0.28	0.31	0.32	0.14	0.77
32	3.82	2.42	2.00	47.19	3.54	1.67	3.83	5.09	2.06	3.54
64	4.67	2.72	2.29	50.17	3.56	0.93	3.06	2.88	2.20	3.39
94	13.40	10.10	8.50	84.31	11.13	1.82	9.98	9.66	8.75	6.73

D	NO.16	NO.17	NO.19	NO.20	NO.21	NO.22	NO.23	NO.24	NO.25	NO.26
10	0.80	3.25	0.36	0.20	0.26	0.45	0.63	0.44	2.80	0.52
32	5.57	10.04	3.46	1.22	1.42	6.44	3.95	3.43	11.47	3.36
64	5.45	13.19	3.27	2.10	2.27	3.81	4.62	3.07	14.17	3.34
96	14.38	23.68	11.32	8.26	8.49	11.60	13.67	10.64	30.11	10.71

D	NO.27	NO.28	NO.29	NO.30	NO.31	NO.32	NO.33	NO.34	NO.35	NO.36
10	0.65	0.72	0.70	0.55	0.71	3.49	3.50	0.84	1.28	0.70
32	2.73	3.09	3.63	3.10	4.10	12.39	12.51	5.25	5.19	3.33
64	3.86	4.01	3.21	2.67	3.38	12.68	12.63	3.80	5.27	3.13
96	12.04	11.32	8.15	6.27	8.49	23.67	23.64	9.50	11.79	7.93

Table 3. Speedup (Double Precision)

D	NO.3	NO.4	NO.5	NO.8	NO.9	NO.10	NO.11	NO.13	NO.14	NO.15
10	0.56	0.19	0.17	9.04	0.43	0.26	0.29	0.30	0.14	0.75
32	3.78	2.43	1.80	33.37	3.09	1.59	3.52	4.81	1.97	3.53
64	4.34	2.49	1.93	30.82	3.15	0.92	2.87	2.74	2.11	3.29
96	12.27	9.24	6.95	46.01	9.72	1.78	9.62	8.74	7.87	5.92

D	NO.16	NO.17	NO.19	NO.20	NO.21	NO.22	NO.23	NO.24	NO.25	NO.26
10	0.79	2.32	0.34	0.18	0.26	0.45	0.59	0.43	1.97	0.52
32	5.10	6.79	3.28	1.13	1.29	6.10	3.63	3.14	8.15	3.23
64	4.75	8.29	3.06	1.99	2.18	3.32	4.02	2.77	9.80	2.92
96	11.91	13.81	9.75	7.37	7.78	10.24	11.55	9.57	20.81	9.40

D	NO.27	NO.28	NO.29	NO.30	NO.31	NO.32	NO.33	NO.34	NO.35	NO.36
10	0.79	2.32	0.34	0.18	0.26	0.45	0.59	0.43	1.97	0.52
32	5.10	6.79	3.28	1.13	1.29	6.10	3.63	3.14	8.15	3.23
64	4.75	8.29	3.06	1.99	2.18	3.32	4.02	2.77	9.80	2.92
96	11.91	13.81	9.75	7.37	7.78	10.24	11.55	9.57	20.81	9.40

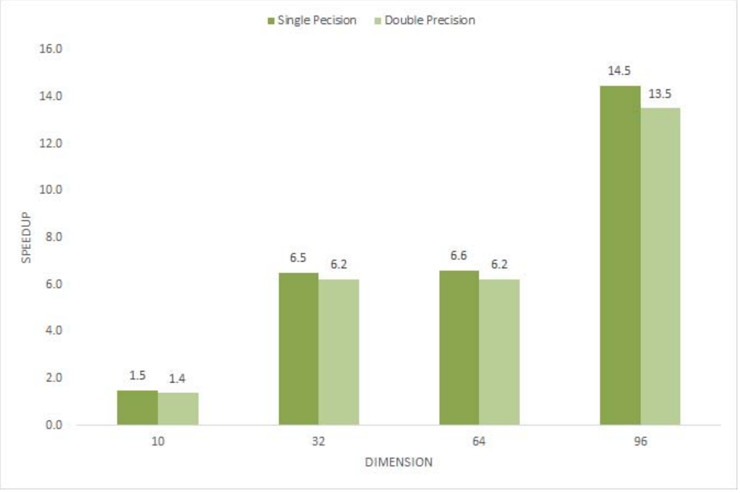


Fig. 1. Overall Speedup

3 Unimodal Functions

3.1 Shifted and Rotated Sphere Function

$$f_1(\mathbf{x}) = \sum_{i=1}^{D} \mathbf{z}_i^2 + f_{opt} \tag{1}$$

where $\mathbf{z} = \mathbf{R}(\mathbf{x} - \mathbf{x}^{opt})$.

Properties

- Unimodal
- Non-separable
- Highly symmetric, in particular rotationally invariant

3.2 Shifted and Rotated Ellipsoid Function

$$f_4(\mathbf{x}) = \sum_{i=1}^{D} i \cdot \mathbf{z}_i^2 + f_{opt} \tag{2}$$

where $\mathbf{z} = \mathbf{R}(\mathbf{x} - \mathbf{x}^{opt})$.

Properties

- Unimodal
- Non-separable

3.3 Shifted and Rotated High Conditioned Elliptic Function

$$f_2(\mathbf{x}) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} \mathbf{z}_i^2 + f_{opt} \tag{3}$$

where $\mathbf{z} = \mathbf{R}(\mathbf{x} - \mathbf{x}^{opt})$.

Properties

- Unimodal
- Non-separable
- Quadratic ill-conditioned
- Smooth local irregularities

3.4 Shifted and Rotated Discus Function

$$f_5(\mathbf{x}) = 10^6 \cdot \mathbf{z}_1^2 + \sum_{i=2}^{D} \mathbf{z}_i^2 + f_{opt} \tag{4}$$

where $\mathbf{z} = \mathbf{R}(\mathbf{x} - \mathbf{x}^{opt})$.

Properties

- Unimodal
- Non-separable
- Smooth local irregularities
- With One sensitive direction

3.5 Shifted and Rotated Bent Cigar Function

$$f_6(\mathbf{x}) = \mathbf{z}_1^2 + 10^6 \cdot \sum_{i=2}^{D} \mathbf{z}_i^2 + f_{opt} \tag{5}$$

where $\mathbf{z} = \mathbf{R}(\mathbf{x} - \mathbf{x}^{opt})$.

Properties

- Unimodal
- Non-separable
- Optimum located in a smooth but very narrow valley

3.6 Shifted and Rotated Different Powers Function

$$f_4(\mathbf{x}) = \sqrt{\sum_{i=1}^{D} |\mathbf{z}_i|^{2+4\frac{i-1}{D-1}}} + f_{opt} \tag{6}$$

where $\mathbf{z} = \mathbf{R}(0.01(\mathbf{x} - \mathbf{x}^{opt}))$.

Properties

- Unimodal
- Non-separable
- Sensitivities of the $\mathbf{z}_i$-variables are different

3.7 Shifted and Rotated Sharp Valley Function

$$f_4(\mathbf{x}) = \mathbf{z}_i^2 + 100 \cdot \sqrt{\sum_{i=2}^{D} \mathbf{z}_i^2} + f_{opt} \tag{7}$$

where $\mathbf{z} = \mathbf{R}(\mathbf{x} - \mathbf{x}^{opt})$.

Properties

- Unimodal
- Non-separable
- Global optimum located in a sharp (non-differentiable) ridge

4 Basic Multi-modal Functions

4.1 Shifted and Rotated Step Function

$$f_3(\mathbf{x}) = \sum_{i=1}^{D} \lfloor \mathbf{z}_i + 0.5 \rfloor^2 + f_{opt} \tag{8}$$

where $\mathbf{z} = \mathbf{R}(\mathbf{x} - \mathbf{x}^{opt})$

Properties

- Many Plateaus of different sizes
- Non-separable

4.2 Shifted and Rotated Weierstrass Function

$$f_9(\mathbf{x}) = \sum_{i=1}^{D} \left(\sum_{k=0}^{k_{max}} a^k \cos\left(2\pi b^k (\mathbf{z}_i + 0.5)\right) \right) - D \cdot \sum_{k=0}^{k_{max}} a^k \cos\left(2\pi b^k \cdot 0.5\right) + f_{opt} \tag{9}$$

where $a = 0.5$, $b = 3$, $k_{max} = 20$, $\mathbf{z} = \mathbf{R}(0.005 \cdot (\mathbf{x} - \mathbf{x}^{opt}))$.

Properties

- Multi-modal
- Non-separable
- Continuous everywhere but only differentiable on a set of points

4.3 Shifted and Rotated Griewank Function

$$f_{10}(\mathbf{x}) = \sum_{i=1}^{D} \frac{\mathbf{z}_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{\mathbf{z}_i}{\sqrt{i}}) + 1 + f_{opt} \tag{10}$$

where $\mathbf{z} = \mathbf{R}(6 \cdot (\mathbf{x} - \mathbf{x}^{opt}))$.

Properties

- Multi-modal
- Non-separable
- With many regularly distributed local optima

4.4 Shifted Rastrigin Function

$$f_{11}(\mathbf{x}) = \sum_{i=1}^{D} \left(\mathbf{z}_i^2 - 10\cos(2\pi\mathbf{z}_i)\right) + 10 \cdot D + f_{opt} \tag{11}$$

where $\mathbf{z} = 0.0512 \cdot (\mathbf{x} - \mathbf{x}^{opt})$.

Properties

- Multi-modal
- Separable
- With many regularly distributed local optima

4.5 Shifted and Rotated Rastrigin Function

$$f_{12}(\mathbf{x}) = \sum_{i=1}^{D} \left(\mathbf{z}_i^2 - 10\cos(2\pi\mathbf{z}_i) + 10\right) + f_{opt} \tag{12}$$

where $\mathbf{z} = \mathbf{R}(0.0512 \cdot (\mathbf{x} - \mathbf{x}^{opt}))$.

Properties

- Multi-modal
- Non-separable
- With many regularly distributed local optima

4.6 Shifted Rotated Schaffer's F7 Function

$$f_{17}(\mathbf{x}) = \left(\frac{1}{D-1} \sum_{i=1}^{D-1} \left((1 + \sin^2(50 \cdot \mathbf{w}_i^{0.2})) \cdot \sqrt{\mathbf{w}_i}\right)\right)^2 + f_{opt} \tag{13}$$

where $\mathbf{w}_i = \sqrt{\mathbf{z}_i^2 + \mathbf{z}_{i+1}^2}$, $\mathbf{z} = \mathbf{R}(\mathbf{x} - \mathbf{x}^{opt})$.

Properties

- Multi-modal
- Non-separable

4.7 Expanded Griewank plus Rosenbrock Function

$$\text{Rosenbrock Function: } g_2(x, y) = 100(x^2 - y)^2 + (x - 1)^2$$
$$\text{Griewank Function: } g_3(x) = x^2/4000 - \cos(x) + 1$$

$$f_{18}(\mathbf{x}) = \sum_{i=1}^{D-1} g_3(g_2(\mathbf{z}_i, \mathbf{z}_{i+1})) + g_3(g_2(\mathbf{z}_D, \mathbf{z}_1)) + f_{opt} \tag{14}$$

where $\mathbf{z} = \mathbf{R}(0.05 \cdot (\mathbf{x} - \mathbf{x}^{opt})) + \mathbf{1}$.

Properties

- Multi-modal
- Non-separable
-

4.8 Shifted and Rotated Rosenbrock Function

$$f_7(\mathbf{x}) = \sum_{i=1}^{D-1} \left(100 \cdot (\mathbf{z}_i^2 - \mathbf{z}_{i+1})^2 + (\mathbf{z}_i - 1)^2\right) + f_{opt} \tag{15}$$

where $\mathbf{z} = \mathbf{R}(0.02048 \cdot (\mathbf{x} - \mathbf{x}^{opt})) + \mathbf{1}$.

Properties

- Multi-modal
- Non-separable
- With a long, narrow, parabolic shaped flat valley from local optima to global optima

4.9 Shifted Modified Schwefel Function

$$f_{13}(\mathbf{x}) = 418.9829 \times D - \sum_{i=1}^{D} g_1(\mathbf{w}_i), \qquad \mathbf{w}_i = \mathbf{z}_i + 420.9687462275036 \tag{16}$$

$$g_1(\mathbf{w}_i) = \begin{cases} \mathbf{w}_i \cdot \sin(\sqrt{|\mathbf{w}_i|}) & \text{if } |\mathbf{w}_i| \leq 500 \\ (500 - \text{mod}(\mathbf{w}_i, 500)) \cdot \sin\left(\sqrt{500 - \text{mod}(\mathbf{w}_i, 500)}\right) - \frac{(\mathbf{w}_i - 500)^2}{10000D} & \text{if } \mathbf{w}_i > 500 \\ (\text{mod}(-\mathbf{w}_i, 500) - 500) \cdot \sin\left(\sqrt{500 - \text{mod}(-\mathbf{w}_i, 500)}\right) - \frac{(\mathbf{w}_i + 500)^2}{10000D} & \text{if } \mathbf{w}_i < -500 \end{cases} \tag{17}$$

where $\mathbf{z} = 10 \cdot (\mathbf{x} - \mathbf{x}^{opt})$.

Properties

- Multi-modal
- Separable
- Having many local optima with the second better local optima far from the global optima

4.10 Shifted Rotated Modified Schwefel Function

$$f_{14}(\mathbf{x}) = 418.9829 \times D - \sum_{i=1}^{D} g_1(\mathbf{w}_i), \qquad \mathbf{w}_i = \mathbf{z}_i + 420.9687462275036 \quad (18)$$

where $\mathbf{z} = \mathbf{R}(10 \cdot (\mathbf{x} - \mathbf{x}^{opt}))$ and $g_1(\cdot)$ is defined as Eq. 17.

Properties

- Multi-modal
- Non-separable
- Having many local optima with the second better local optima far from the global optima

4.11 Shifted Rotated Katsuura Function

$$f_{15}(\mathbf{x}) = \frac{10}{D^2}\prod_{i=1}^{D}(1 + i\sum_{j=1}^{32}\frac{|2^j \cdot \mathbf{z}_i - [2^j \cdot \mathbf{z}_i]|}{2^j})^{\frac{10}{D^{1.2}}} - \frac{10}{D^2} + f_{opt} \qquad (19)$$

where $\mathbf{z} = \mathbf{R}(0.05 \cdot (\mathbf{x} - \mathbf{x}^{opt}))$.

Properties

- Multi-modal
- Non-separable
- Continuous everywhere but differentiable nowhere

4.12 Shifted and Rotated Lunacek bi-Rastrigin Function

$$f_{12}(\mathbf{x}) = \min\left(\sum_{i=1}^{D}(\mathbf{z}_i - \mu_1)^2, dD + s\sum_{i=1}^{D}(\mathbf{z}_i - \mu_2)^2)\right) + 10 \cdot (D - \sum_{i=1}^{D}\cos(2\pi(\mathbf{z}_i - \mu_1))) + f_{opt} \quad (20)$$

where $\mathbf{z} = \mathbf{R}(0.1 \cdot (\mathbf{x} - \mathbf{x}^{opt}) + 2.5 * \mathbf{1})$, $\mu_1 = 2.5$, $\mu_2 = -2.5$, $d = 1$, $s = 0.9$.

Properties

- Multi-modal
- Non-separable
- With two funnel around $\mu_1\mathbf{1}$ and $\mu_2\mathbf{1}$

4.13 Shifted and Rotated Ackley Function

$$f_8(\mathbf{x}) = -20 \cdot \exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}\mathbf{x}_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi\mathbf{x}_i)\right) + 20 + e + f_{opt} \tag{21}$$

where $\mathbf{z} = \mathbf{R}(\mathbf{x} - \mathbf{x}^{opt})$.

Properties

- Multi-modal
- Non-separable
- Having many local optima with the global optima located in a very small basin

4.14 Shifted Rotated HappyCat Function

$$f_{16}(\mathbf{x}) = |\sum_{i=1}^{D}\mathbf{z}_i^2 - D|^{0.25} + (\frac{1}{2}\sum_{j=1}^{D}\mathbf{z}_j^2 + \sum_{j=1}^{D}\mathbf{z}_j)/D + 0.5 + f_{opt} \tag{22}$$

where $\mathbf{z} = \mathbf{R}(0.05 \cdot (\mathbf{x} - \mathbf{x}^{opt})) - \mathbf{1}$.

Properties

- Multi-modal
- Non-separable
- Global optima located in curved narrow valley

4.15 Shifted Rotated HGBat Function

$$f_{17}(\mathbf{x}) = |(\sum_{i=1}^{D}\mathbf{z}_i^2)^2 - (\sum_{j=1}^{D}\mathbf{z}_j)^2|^{0.5} + (\frac{1}{2}\sum_{j=1}^{D}\mathbf{z}_j^2 + \sum_{j=1}^{D}\mathbf{z}_j)/D + 0.5 + f_{opt} \tag{23}$$

where $\mathbf{z} = \mathbf{R}(0.05 \cdot (\mathbf{x} - \mathbf{x}^{opt})) - \mathbf{1}$.

Properties

- Multi-modal
- Non-separable
- Global optima located in curved narrow valley

4.16 Expanded Schaffer's F6 Function

$$\text{Schaffer's F6 Function: } g_4(x, y) = \frac{\sin^2(\sqrt{x^2 + y^2}) - 0.5}{(1 + 0.001 \cdot (x^2 + y^2))^2} + 0.5$$

$$f_{19}(\mathbf{x}) = \sum_{i=1}^{D-1} g_4(\mathbf{z}_i, \mathbf{z}_{i+1}) + g_4(\mathbf{z}_D, \mathbf{z}_1) + f_{opt} \tag{24}$$

where $\mathbf{z} = \mathbf{R}(\mathbf{x} - \mathbf{x}^{opt})$.

Properties

- Multi-modal
- Non-separable

5 Hybrid and Composition Functions

Hybrid functions are constructed according to [3]. For each hybrid function, the variables are randomly divided into subcomponents and different basic functions (unimodal and multi-modal) are used for different subcomponents.

Composition functions are constructed in the same manner as in [2,3]. The constructed functions are multi-modal and non-separable and merge the properties of the sub-functions better and maintains continuity around the global/local optima. The local optimum which has the smallest bias value is the global optimum. The optimum of the third basic function is set to the origin as a trip in order to test the algorithms' tendency to converge to the search center. Note that, the landscape is not only changes along with the selection of basic function, but the optima, σ and λ can effect it greatly.

The detailed specifications of hybrid and composition functions can be found in the extended version of this paper[1], along with illustrations for all 2-D functions except hybrid functions.

Acknowledgements. This work was supported by National Natural Science Foundation of China (NSFC), Grant No. 61375119, 61170057 and 60875080.

References

1. Finck, S., Hansen, N., Ros, R., Auger, A.: Real-parameter black-box optimization benchmarking 2010: Noiseless functions definitions. Technical Report 2009/20, Research Center PPE (2010)
2. Liang, J.J., Qu, B.Y., Suganthan, P.N., Hernández-Díaz, A.G.: Problem definitions and evaluation criteria for the cec 2013 special session and competition on real-parameter optimization. Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University and Nanyang Technological University, Singapore (2013)
3. Liang, J.J., Qu, B.Y., Suganthan, P.N.: Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University and Nanyang Technological University, Singapore (2013)

[1] Download from `http://arxiv.org/abs/1407.7737`

A Particle Swarm Optimization Based Pareto Optimal Task Scheduling in Cloud Computing

A.S. Ajeena Beegom[1] and M.S. Rajasree[2]

[1] Dept. of Computer Science and Engineering,
College of Engineering, Trivandrum, India
ajeena@cet.ac.in
[2] IIITM-K, Trivandrum, India
rajasree.ms@iiitmk.ac.in

Abstract. Task scheduling in Cloud computing is a challenging aspect due to the conflicting requirements of end users of cloud and the Cloud Service Provider (CSP). The challenge at the CSP's end is to schedule tasks submitted by the cloud users in an optimal way such that it should meet the quality of service (QoS) requirements of the user at one end and the running costs of the infrastructure to a minimum level at the other end for better profit. The focus is on two objectives, makespan and cost, to be optimized simultaneously using meta heuristic search techniques for scheduling independent tasks. A new variant of continuous Particle Swarm Optimization (PSO) algorithm, named Integer-PSO, is proposed to solve the bi-objective task scheduling problem in cloud which out performs the smallest position value (SPV) rule based PSO technique.

Keywords: Cloud Computing, Task Scheduling, Particle Swarm Optimization, Integer-PSO.

1 Introduction

Recently, cloud computing has emerged as an attractive platform for entrepreneurs as well as researchers in various domains. Cloud computing refers to leasing computing resources over the Internet. Benefits of using such a set up include reduced infrastructure cost, reduced overhead and pay only for the components he has used for the given amount of time. There are many task scheduling models available in literature for heterogeneous distributed systems but these models aim at the improvement of specific performance metrics like throughput and storage. For a cloud computing platform, apart from theses considerations, user satisfaction in terms of QoS and CSP's profit is to be considered while scheduling tasks. In this work, we consider the scheduling of large set of independent tasks of different size. The conflicting objectives of performance optimization considered are the overall execution time (makespan) of all the tasks and the cost of service. The cost include computation cost, communication cost and over all maintenance cost as well as power consumption cost.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 79–86, 2014.
© Springer International Publishing Switzerland 2014

Most of the existing task scheduling research in cloud computing addresses any one factor, namely makespan as done in [5] or cost as seen in [11] and tries to find an optimal schedule based on that factor alone. But single objective optimization solutions will try to optimize one objective making another key factor to worse. In the proposed work, we have used weighted sum approach for pareto-optimality and uses Particle swarm algorithm to solve the same.

The rest of the paper is organized as follows. In Section 2, we present Related Work in the domain and Section 3 describes the Mathematical Model and formally states the optimization problem. Section 4 details Particle Swarm Optimization Technique and the proposed Integer-PSO algorithm. Experimental Results are presented in Section 5 and Section 6 gives the Conclusion.

2 Related Work

Most of the research in computational grids and cloud systems for scheduling independent tasks to be executed in parallel tries to optimize a single objective function where the parameters are any one of makespan, profit earned, cost of service, QoS, energy consumption and average response time. Meta heuristic search techniques has been tried by many to solve the same. L.Zhu and J. Wu [15] uses PSO technique combined with Simulated annealing to solve task scheduling problem in a general scenario. M.F.Tasgetiren et al. [8] proposed Smallest Position Value (SPV) rule based PSO algorithm to solve Single Machine Total Weighted Tardiness Problem and has used a local search technique to intense the search. Lei Zhang et al. [14] used this technique to solve task scheduling problem in grid environment and has given a comparative study on the application of PSO technique with Genetic algorithm for achieving minimal completion time. PSO algorithm has also been used in solving task allocation / scheduling problems in work flows in cloud [1], [13], [7].

Multi-objective optimization for resource allocation in cloud computing has been addressed by Feng et al. [3] and uses PSO algorithm to solve the problem. They have considered total execution time, resource reservation and QoS of each task as their optimization objective and uses pareto domination mechanism to find optimal solutions. Lizheng Guo [4] addresses task assignment problem in cloud computing considering makespan and cost. They use smallest position value based PSO algorithm for finding an optimal schedule.

Our proposed method can be used to schedule tasks in pubic cloud or in private cloud for independent tasks. Our approach is unique because, to the best our knowledge, pareto optimal task scheduling using particle swarm optimization technique has not been addressed in the case of private or public cloud with independent tasks to be scheduled, but has been proposed for work flow scheduling in hybrid cloud[2]. Our work also proposes a new variant of PSO algorithm namely Integer - PSO.

3 Mathematical Model

Assume an application consists of N independent tasks, n out of N are scheduled at each time window, where the value of n is limited by the number of available VMs m and $k = \frac{N}{n}$ similar epochs are needed to complete the execution of all tasks. For each type of VM instance $I =$ *[very small, small, medium, large, extra large, super]*, the associated cost of usage and the computing power are different. Let Pf_j represent the processing power of j^{th} VM instance type where j ranges from 1 to $|I|$ and C_j represents its cost for unit time. The task length of each task $TASK_i$ is precomputed and represented as T_i, the time needed to execute each task in 'very small' type VM. The optimization objectives for N tasks are :

$$Minimize \quad Makespanfn = \sum_{p=1}^{k}\sum_{i=1}^{n} T_i * Pf_j * x_{ij} \quad for \quad some \quad j\epsilon I \tag{1}$$

$$Minimize \quad Costfn = \sum_{p=1}^{k}\sum_{i=1}^{n} C_j * T_i * Pf_j * x_{ij} \quad for \quad some \quad j\epsilon I \tag{2}$$

where x_{ij} is a decision variable, denoting $TASK_i$ is scheduled on VM_j and n tasks are there in an epoch. subject to the following constraints:

$$n \leq m \tag{3}$$

$$x_{ij} = \begin{cases} 1 & \text{if } TASK_i \quad scheduledto \quad VM_j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

and

$$\sum_{i=1}^{n} \quad x_{ij} = m \tag{5}$$

3.1 Pareto Optimality

A multi objective optimization problem consists of optimizing a vector of n_{obj} where the objective function $F(x) = (f_1(x), f_2(x), \ldots, f_{n_{obj}}(x))$. The problem here is to find an optimal task schedule considering both the objectives. i.e.,cost and makespan. We have used weighted sum approach[9], [12] for solving the bi-objective optimization problem, which in essence convert a multi-objective optimization problem to a single objective one with weights representing preferences among objectives by the decision maker. This approach is easy to solve and produce a single solution to the problem. Hence our bi-objective optimization problem can now be represented using the formula:

$$Minimize \quad \theta * Costfn + (1 - \theta) * Makespanfn \tag{6}$$

where θ represents the relative weight or preference of one objective over the other, in the range $[0, 1]$. When $\theta = 0$, the optimization problem becomes that of minimizing *Makespanfn* and when $\theta = 1$, the problem becomes minimizing *Costfn*.

4 Particle Swarm Optimization Technique

Finding an optimal schedule meeting the constraints of a bi-objective optimization problem are well-known problems in N P hard category, hence one of the heuristic techniques, Particle Swarm Optimization[6] is applied to obtain a feasible solution in reasonable time. Initially, the PSO algorithm generates a set of N solutions called *particles*, randomly in the D dimensional search space. Each *particle* is represented by a D-dimensional vector X_i where i ranges from 1 to d which stands for its location $(x_{i1}, x_{i2}, ..., x_{id})$ in space. Velocity of each particle v is constrained by v_{min} and v_{max} and its position x is updated according to the following equations:

$$v_{id}^{n+1} = w * v_{id}^{n} + c_1 * rand_1 * (pbest_{id}^{n} - x_{id}^{n}) + c_2 * rand_2 * (gbest_{d}^{n} - x_{id}^{n}) \tag{7}$$

$$x_{id}^{n+1} = x_{id}^{n} + v_{id}^{n+1} \tag{8}$$

where $i = 1, 2, \ldots, N$; $n = 1, 2, \ldots, iter_{max}$, the maximum iteration number, w, the inertia weight; c_1 and c_2 are two positive constants called acceleration coefficients and $rand_1$ and $rand_2$ are two uniformly distributed random numbers in the interval $[0, 1]$. Each particle maintains its position and its velocity. It also remembers the best fitness value it has achieved thus far during the search (*individual best fitness*) and the candidate solution that achieved this fitness (*individual best position (pbest)*). Also, the PSO algorithm maintains the best fitness value achieved among all particles in the swarm (*global best fitness*) and the candidate solution that achieved this fitness (*global best position (gbest)*). Equations (7) and (8) enable the particles to search around its individual best position *pbest* and update global best position *gbest*. This technique was initially proposed for solving problems in the continuous domain through the velocity updating rule. Since our problem work in the discrete domain, it has to be modified to suit the discrete domain. The Smallest Position Value rule based PSO (PSO-SPV) algorithm [8] is widely used for the same. This technique performs poor when there exist high variance in the length of the tasks submitted by end-users and when high variance exists in computational speed of resources[10]. We

Algorithm 1. Abstract of Particle Swarm Optimization Algorithm

```
$P \leftarrow$ Initial Population
Evaluate (P)
Initialize $pbest$ and $gbest$
while termination criterion not met do
    Update Velocity(P) as indicated in equation (7)
    Update Position (P) as indicated in equation (8)
    Evaluate (P)
    Find $pbest$ and $gbest$
end while
Output $gbest$
```

too observed that the same technique is not able to converge to near optimal solution with bi-objective optimization of task scheduling in cloud computing. Hence a new method for generating discrete permutations is proposed, namely integer-PSO. Here permutation encoding technique is used where every VM is assigned a number from 1 to n and a solution sequence $(5, 2, 1, 3, 4)$ means assign Task 1 to VM 5, Task 2 to VM 2 and so on. Initial populations are randomly generated. Each solution is evaluated to find its fitness based on equation (6) on different values of θ.

4.1 Integer – PSO

An update in the position of the particle based on equations (7) and (8) should result in new task assignment for a scheduling problem, but they produce floating point values in the continuous domain. Many discrete versions of PSO rounds-off the floating point position values and stores the discrete integer value for the particle's position. To preserve the stochastic nature of the continuous PSO, we have modified equation (8) in our algorithm, as shown below:

$$Y_{id}^{n} = ceil((x_{id}^{n} + v_{id}^{n}) * \beta) \quad where \qquad \beta = 10^{y} \tag{9}$$

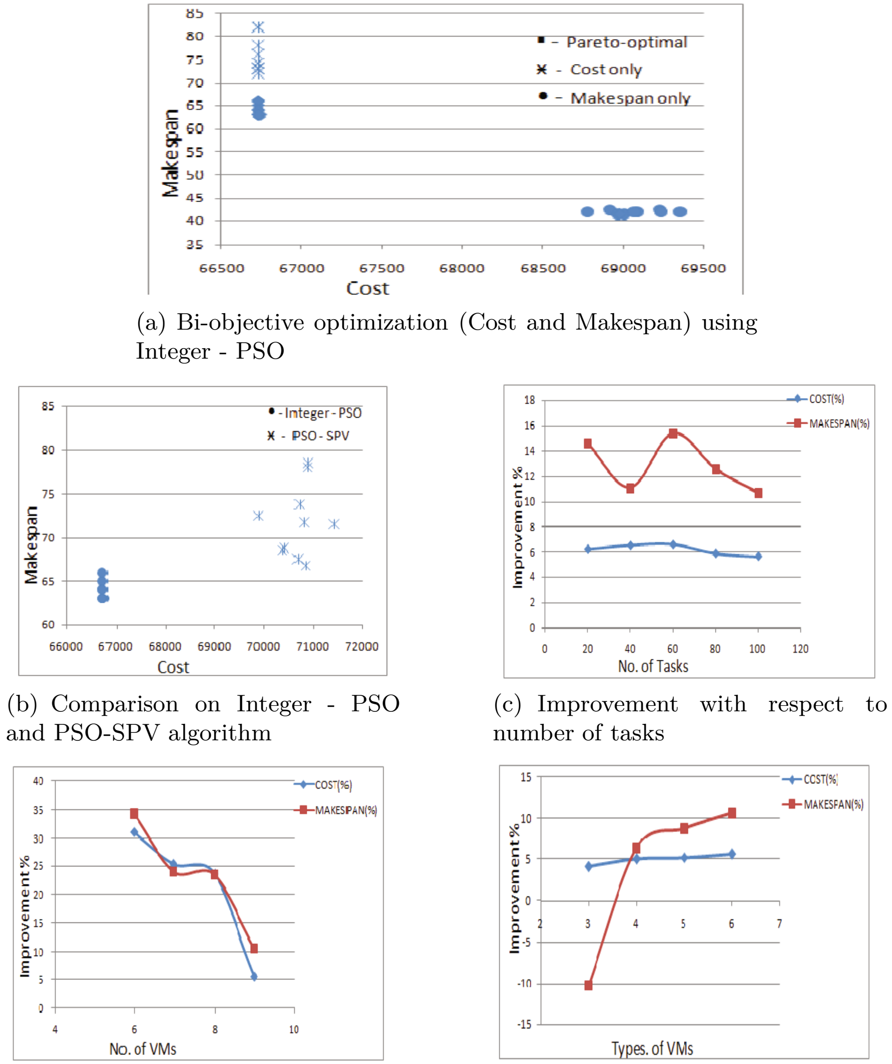
$$Pos_{id}^{n+1} = (Y_{id}^{n}) \quad mod \quad m \tag{10}$$

$$xc_{id}^{n+1} = \begin{cases} Pos_{id}^{n+1} & \text{if } Pos_{id}^{n+1} > 0 \\ m & \text{otherwise} \end{cases} \tag{11}$$

Table 1. Integer-PSO Example

Task J	1	2	3	4	5
x_{ij}^{k}	4	5	1	3	2
v_{ij}^{k+1}	-0.6015	-0.2413	0.0327	-0.0352	-0.8544
Y_{ij}^{k+1}	33985	47587	10327	29648	11456
Pos_{ij}^{k+1}	0	2	2	3	1
xc_{ij}^{k+1}	5	2	2	3	1
x_{ij}^{k+1}	5	4	2	3	1

New variables Y_{id}^{n} and Pos_{id}^{n+1} are introduced to store the continuous value as an integer of required accuracy and the temporary task assignment respectively. This method may create more than one task assignment to some of the VMs and may not assign any task to some other VMs. which need to be handled separately. The procedure is as shown in Table 1, assuming $m = 5$ and k=4.



(a) Bi-objective optimization (Cost and Makespan) using Integer - PSO

(b) Comparison on Integer - PSO and PSO-SPV algorithm

(c) Improvement with respect to number of tasks

(d) Improvement with respect to number of VMs

(e) Improvement with respect to type of VMs

Fig. 1. Performance Analysis

5 Results and Discussion

The algorithm has been simulated 10 times on the same data set with $\theta = 0.9$, $\theta = 0$ and $\theta = 1$ and the average value is taken to find an estimate on best convergences, on a laptop PC with PIV processor of 3 GHz clock frequency and 8GB RAM. We have tested the algorithm on a task set of 99 tasks, setting the population size (|P|) as 5 and other PSO parameters as $w = 0.6$, $c1 = c2 = 0.2$, $v_{max} = 4$ and $v_{min} = -4$. A suitable value of θ is found through trial and error as 0.9. Assumptions regarding the capacity of each VM in terms of speed and cost of usage per unit time and the task length are given as vectors. The algorithm runs for 500 generations per task set and is repeated 11 times. The results are shown in Figure 1(a).

For $\theta = 0.9$, the proposed algorithm (Integer-PSO) finds optimal cost in 90 percentage of time, but SPV-PSO has not converged to optimal or near optimal cost value on any value of θ, shown in Figure 1(b). When the algorithm is applied to single objective optimization scenario, for 90 percentage of time, the Integer-PSO algorithm converges to optimal value. A detailed performance analysis of both the approaches were done with different number of tasks, different number of VMs and different types of VMs and are shown in Figure 1(c) to 1(e).

6 Conclusion

Scheduling tasks in the cloud is a challenging one as the same involves many factors such as cost and profit considerations, execution time, SLAs, Quality of service parameters requested by the end user and committed by the CSP and power considerations. Also the task arrival rate is highly unpredictable and dynamic in nature. We have modelled the problem as a constraint bi-objective optimization problem, where the objectives are makespan and cost and have used Particle Swarm Optimization algorithm to solve the same, where the pareto optimality is achieved through weighted sum approach. A variant of PSO technique is proposed (Integer-PSO) whose results are promising.

References

1. Szabo, C., Kroeger, T.: Evolving multi-objective strategies for task allocation of scientific workflows on public clouds. In: Proc. of IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2012)
2. Farahabady, R.H., Lee, Y.C., Zomaya, A.Y.: Pareto optimal cloud bursting. Accepted for publication in IEEE Transactions on Parallel and Distributed Systems (2013)
3. Feng, M., Wang, X., Zhang, Y., Li, J.: Multi-objective particle swarm optimization for reseource allocation in cloud computing. In: Proc. of 2nd International Conference on Cloud Computing and Intelligent Systems (CCIS), vol. 3, pp. 1161–1165 (2012)
4. Guo, L., Shao, G., Zhao, S.: Multi-objective task assignment in cloud computing by particle swarm optimization. In: Proc. of 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), pp. 1–4 (2012)

5. Jin, J., Luo, J., Song, A., Dong, F., Xiong, R.: Bar: An efficient data locality driven task scheduling algorithm for cloud computing. In: Proc. of 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 295–304 (2011)
6. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
7. Rodrignez Sossa, M., Buyya, R.: Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. IEEE Transactions on Cloud Computing (2014)
8. Liang, Y.-C., Tasgetiren, M.F., Sevkli, M., Gencylmaz, G.: Particle swarm optimization algorithm for single machine total weighted tardiness problem. In: IEEE Congress on Evolutionary Computation (CEC2004), vol. 2, pp. 1412–1419 (2004)
9. Marler, T., Arora, J.S.: The weighted sum method for multi-objective optimization: new insights. Springer (2009)
10. Sidhu, M.S., Thulasiraman, P., Thulasiram, R.K.: A load-rebalnce pso heuristic for task matching in heterogeneous computing systems. In: IEEE Symposium on Swarm Intelligence (SIS), pp. 180–187 (2013)
11. Sadhasivam, G.S., Selvarani, S.: Improved cost-based algorithm for task scheduling in cloud computing. In: Proc. of IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–5 (2010)
12. Ivan, P.: Stanimirovic, Milan Lj. Zlatanovic, and Marko D Petkovic: On the linear weighted sum method for multi-objective optimization. FACTA UNIVERSITATIS (NIS), Ser. Math. Inform, 49–63 (2011)
13. Wu, Z., Ni, Z., Liu, X.: A revised discrete particle swarm optimization for cloud workflow scheduling. In: Proc. of International Conference on Computational Intelligence and Security (CIS), pp. 184–188 (2010)
14. Zhang, L., Chen, Y., Sun, R., Jing, S., Yang, B.: A task scheduling algorithm based on pso for grid computing. International Journal of Computational Intelligence Research 4(1), 37–43 (2008)
15. Zhu, L., Wu, J.: Hybrid particle swarm optimization algorithm for flexible task scheduling. In: Proc. of 3rd Internatinal Conference on Genetic and Evolutionary Computing (WGEC 2009), pp. 603–606 (2009)

Development on Harmony Search Hyper-heuristic Framework for Examination Timetabling Problem

Khairul Anwar[1], Ahamad Tajudin Khader[1],
Mohammed Azmi Al-Betar[1, 2], and Mohammed A. Awadallah[1]

[1] School of Computer Sciences, Universiti Sains Malaysia (USM),
11800 Pulau Pinang, Malaysia
[2] Department of Information Technology, Al-Huson University College,
Al Balqa Applied University, P.O. Box 50, Al-Huson, Irbid-Jordan
{ka10_com097,mama10_com018}@student.usm.my,
{tajudin,mohbetar}@cs.usm.my

Abstract. In this paper, a Harmony Search-based Hyper-heuristic (HSHH) approach is proposed for tackling examination timetabling problems. In this approach, the harmony search algorithm will operate as a high level of abstraction which intelligently evolves a sequence of low level heuristics. This sequence is a combination of improvement heuristics which consist of neighborhood structure strategies. The proposed approach is tested using the examination timetabling tracks in Second International Timetabling Competition (ITC-2007) benchmarks. Experimentally, the HSHH approach can achieve comparable results with the comparative methods in the literature.

Keywords: Examination Timetabling, Harmony Search, Hyper-heuristic.

1 Introduction

Examination timetabling is the process of scheduling a set of exams to a set of timeslots and rooms, subject to *hard* and *soft* constraints. It becomes extremely difficult when it involves a large number of events (could be hundreds or thousands) to be scheduled in limited resources in accordance with a wide variety of constraints, which need to be satisfied [1]. The hard constraints are mandatory to fulfill while the soft constraints are desired but not absolutely necessary. The solution that satisfied all the hard constraints is called *feasible*. Conventionally, the soft constraints play a major role in measuring the *quality* of the solution. The main target is to find an examination timetabling solution that satisfies all hard constraints and minimize the violation of soft constraints as much as possible.

Several optimization techniques have been introduced to solve examination timetabling problems such as sequential techniques, constraints based techniques, local search-based techniques, population-based techniques, and many others as surveyed in [1]. But recently, the attention of the researches in the operation research and the artificial intelligence fields has shifted to be concerned with a more general approach

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 87–95, 2014.
© Springer International Publishing Switzerland 2014

as hyper-heuristic. Hyper-heuristics (HH) have a higher level heuristic to choose from a set of heuristics that are applicable for the problem in hand. The main difference between the hyper-heuristics and meta-heuristics is that hyper-heuristics explore the heuristic search space while meta-heuristics explore the solution search space [2].

There are several hyper-heuristics to solve the examination timetabling problems. Sabar et al.[3] investigated a new graph coloring constructive hyper-heuristic (GCCHH) for solving examination timetabling problems using ITC-2007 dataset. The results demonstrate that GCCHH produces good results and outperforms other approaches on some of the benchmark instances. In another study, Pillay and Banzhaf [4] used genetic programming (GP) for the evolution on hyper-heuristics framework to solve uncapacitated examination timetabling problems. From the research, it shows that the genetic programming system was comparable to the other search algorithm, and in some cases it can produce better quality timetables.

In our previous work [5], Harmony search algorithm (HSA) has been employed in a hyper-heuristic framework as a high-level heuristic where some *move* heuristics (i.e. move and swap) have been employed as low-level heuristics. HSA is a population-based algorithm developed by Geem et al.[6]. HSA is a stochastic search mechanism, simple in concept, and no derivation information is required in the initial search [6]. It has been successfully applied to a wide range of optimization and scheduling problems such as course timetabling problem [7], nurse scheduling problem [8, 9], as well as examination timetabling problem [10] and many others as reported in [11].

Evidently, the initial employment produced an impressive result for the initial investigation, with a chance to improve the results by modifying the proposed method. In this research, the new low-level heuristics strategies and the pitch adjustment procedure are modified to enhance the proposed method. For purposes of evaluation, the dataset of examination timetabling tracks established in the Second International Timetabling Competition (ITC-2007) is used. The results produced by new HSHH outperformed those produced by previous versions of HSHH and comparable to the results of the ITC-2007 comparative methods.

This paper is organized as follows: Section 2 discusses the examination timetabling problem (ETP) and benchmark dataset of ITC-2007 for examination track. The description of harmony search hyper-heuristic (HSHH) algorithm is presented in Section 3. Section 4 discusses the experimental setup and the computational results. Finally, the conclusion and future works are provided in Section 5.

2 Examination Timetabling

In this research, we used ITC-2007 [12] examination version as a benchmark to evaluate the proposed method. ITC-2007 provides a capacitated examination timetabling dataset which contains eight instances and four hidden ones. For this research we experimented with the eight instances. This dataset consists of five hard constraints (i.e., H1-H5) and seven soft constraints (i.e., S1-S7) as shown in Table 1.

Table 1. ITC-2007 Hard and Soft constraints

Key	Constraints
H1	No student sits for more than one examination at the same time.
H2	The capacity of individual rooms is not exceeded at any time throughout the examination session.
H3	Period Lengths are not violated.
H4	Satisfaction of period related hard constraints (e.g., exam B must be scheduled after exam A).
H5	Satisfaction of room related hard constraints (e.g., exam A exclusively scheduled in room X).
S1	Two exams in a row.
S2	Two exams in a day.
S3	Specified spread of examinations.
S4	Mixed duration of examinations within individual periods.
S5	Larger examinations appearing later in the timetable.
S6	Period related soft constraints – some period has an associated penalty.
S7	Room related soft constraints – some room has an associated penalty.

The objective function to summarize the ITC-2007 dataset is formalized in equation (1) and the details' explanations is provided in [12].

$$min \quad f(\boldsymbol{x}) = \sum_{s \in S} (w^{2R} C_s^{2R} + w^{2D} C_s^{2D} + w^{PS} C_s^{PS}) + w^{NMD} C^{NMD} + w^{FL} C^{FL} + C^P + C^R \tag{1}$$

where $\boldsymbol{x}$ is a complete timetabling solution; S refers to a set of students while w refers to the institutional penalty for each constraint except for period and room related soft constraint (i.e., C^P *and* C^R). Table 2 shows the detail notation of the variables used in equation (1).

Table 2. List of abbreviations given to the ITC-2007 soft constraint [12]

Math Symbol	*Description*
$\boldsymbol{C_s^{2R}}$	"two exam in row" penalty for student *s*.
$\boldsymbol{C_s^{2D}}$	"two exam in day" penalty for student *s*.
$\boldsymbol{C_s^{PS}}$	"period spread" penalty for student *s*
$\boldsymbol{C^{NMD}}$	"No mixed duration" penalty
$\boldsymbol{C^{FL}}$	"Front load" penalty
$\boldsymbol{C^{P}}$	"Period" penalty
$\boldsymbol{C^{R}}$	"Room" penalty

3 Harmony Search Hyper-heuristic for ITC-2007

In previous work of Harmony Search Hyper-Heuristic (HSHH) [5], the pitch adjustment operator is deactivated in the improvisation step, and two neighborhood structures are utilized as low-level heuristics. In this study, the pitch adjustment operator is added during the improvisation step, and seven different neighborhood structures have been utilized as low level heuristics. They can be summarized as follows:

- ***h1***: Move Exam. Select one exam at random and move to a new randomly selected feasible timeslot.
- ***h2***: Swap Timeslot. Select two exams at random and swap their timeslots.
- ***h3***: Swap Exam. Select two timeslots randomly and exchange all exams between them.
- ***h4***: Swap Period. Select two periods and swap the exams between the periods.
- ***h5***: Select two timeslots (e.g. *t1* and *t2*) randomly and move some exams from the timeslot *t1* to *t2* and vice versa.
- ***h6***: This heuristic similar to *h3* but it only swaps the conflicting exams in two distinct timeslots. This heuristic is similar to kempe chain method in (Al-Betar et al.,[10]).
- ***h7***: do nothing.

Basically, HSHH has five main steps as follows:

Step 1: Initialization. The HSHH begins by setting the harmony search parameters: harmony memory size (HMS), harmony memory consideration rate (HMCR), number of iterations (NI) and Harmony Memory Length (HML) which represents the length of heuristic vector. Furthermore, the Pitch Adjustment Rate (PAR) parameter also will be set. Initially, the largest degree (LD) heuristic is used to construct the initial feasible solution ($\boldsymbol{x}^{feasible}$). If the solution is not feasible, then the repair procedure as used in [7] will be triggered to maintain the feasibility of the solution.

Step 2: Initializing Harmony Memory. HSHH consists of two complemented search spaces (*heuristic search space and solution search space*), each represented in a harmony memory: Heuristics Harmony Memory (HHM) and Solutions Harmony Memory (SHM). HHM contains sets of heuristic vectors determined by HMS where every vector is a heuristics sequence. The length of the vector is determined by HML. Similarly, SHM contains sets of solution vectors and the length is determined by the number of exam, *N*.

In initializing HHM and SHM, the HSHH, firstly, generates the new heuristics vector (***h'***) randomly and apply this vector to the initial feasible solution ($\boldsymbol{x}^{feasible}$) to produce the new solution (***x'***). The new solution will be evaluated using the objective function as in equation (1). If the new solution (***x'****)* is better than the initial solution ($\boldsymbol{x}^{feasible}$), then the new solution (***x'****)* will be saved in the SHM and the new heuristic vector (***h'****)* will be saved in the HHM. This process will be repeated until HHM and SHM are filled (see equations (2) and (3)). After completing the process, HSHH will retain the worst solution (x^{worst}) and the best solution (x^{best}) in SHM.

Step 3: Improvise a new heuristic sequence. In this step, a new heuristics vector ***h'*** = $(h'_1 , h'_2 , \dots h'_{HML})$ is generated from scratch, based on three HSA operators: *memory consideration, random consideration* and *pitch adjustment.*

$$HHM = \begin{bmatrix} h_1^1 & h_2^1 & \cdots & h_{HML}^1 \\ h_1^2 & h_2^2 & \cdots & h_{HML}^2 \\ \vdots & \vdots & \ddots & \vdots \\ h_1^{HMS} & h_2^{HMS} & \cdots & h_{HML}^{HMS} \end{bmatrix} \quad (2) \qquad SHM = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_N^{HMS} \end{bmatrix}$$

(3)

Note that N refers to the number of examinations. In *memory consideration operator*, the new heuristic index of h'_i in the new heuristic vector (***h'***) is randomly selected from the historical indexes (e.g. $h_i^1, h_i^2, \dots\dots h_i^{HMS}$), stored in the heuristic harmony memory with probability of HMCR, where HMCR ϵ [0, 1] .

For *Random consideration operator*, the new heuristic index is randomly assigned from the set of heuristics h'_i ϵ *{h1, h2, h3, h4, h5, h6, h7}* with probability of (1-HMCR) as in equation (4).

$$h'_i \in \begin{cases} \{h_i^1, h_i^2, \dots\dots h_i^{HMS}\} & w.p. \quad HMCR \\ \{h1, h2, h3, h4, h5, h6, h7\} & w.p. \quad (1 - HMCR) \end{cases} \qquad (4)$$

In *Pitch Adjustment operator,* a simple adjustment is used. The new index of h'_i will be added/subtracted by 1 with a probability of PAR where $0 \leq PAR \leq 1$ as in equation (5).

$$Pitch\, adjust\, for\ h'_i = \begin{cases} Yes & w.p. \quad PAR \\ No & w.p. \quad (1 - PAR) \end{cases} \qquad (5)$$

In case the decision of PAR is *yes*, the index of h'_i will be recalculated as follows:

$$h'_i = h'_i \pm 1 \qquad (6)$$

Note that if the index is out of range, it will remain the same. Then the new harmony of heuristic vector ***h'*** will be applied to a solution (e.g. $\boldsymbol{x}^{rand}$ = $x_1^{rand}, x_2^{rand}, \dots\dots x_N^{rand}$) where x^{rand} is randomly selected from the solution search space or SHM. The HSHH used random selection to select the solution from the SHM to avoid the local optima. In this process, the heuristic in ***h'*** will be executed sequentially to the selected solution ($\boldsymbol{x}^{rand}$). The process will continue until all the heuristics in ***h'*** have been executed, and a new solution (***x'***) will be produced. Pseudo-code for improvisation step is shown in the Algorithm 1.

Step 4: Update HHM and SHM. In hyper-heuristic environment, this step is called a move acceptance step. HSHH will decide either to accept or neglect the new heuristic vector ***h'***. In this process, the new solution (***x'***) will be evaluated using the objective function. The new solution must be complete and feasible. If the new solution is better than the worst solution in solution harmony memory (SHM), the new ***h'*** and ***x'*** will be saved in the memory (***h'*** in HHM and ***x'*** in SHM) and the worst heuristic vector and solution will be excluded from the memory (i.e., HHM and SHM).

Step 5: Check the stop criterion. Step 3 and step 4 in this approach are repeated until the stop criterion (i.e., NI) is met.

Algorithm 1: *Pseudo-code for selecting and generating heuristic vector during the improvisation process in step 3.*

```
$h'$= 0; //heuristic vector
 for $l$ = 0,…,HML do
     if (U(0,1) ≤ HMCR) then
         $h_i' \in \{h_i^1, h_i^2, \ldots\ldots h_i^{HMS}\}$ ; //Memory consideration;
         if (U(0,1) ≤ PAR) then
               $h_i' = h_i' \pm 1$ ; //Pitch adjustment;
     else
         $h_i' \in \{h1, h2, h3, h4, h5, h6, h7\}$; //Random consideration;
     end if
 end for
 $x^{rand} \in (x^1, x^2, \ldots, x^{HMS})$; //Select random solution from SHM;
 $x'$ = apply $h'$ to $x^{rand}$ ;
```

4 Experiments and Results

In this section, Harmony Search Hyper-heuristic is evaluated using the real world problem dataset (ITC-2007) for university examination timetabling problem. The proposed method is coded in Microsoft Visual C++ 6 under Windows 7 on Intel processor with 2G RAM. We chose to test the proposed method with each problem instances in ITC-2007.

The characteristics of the ITC-2007 dataset are provided in Table 3. This table includes information such as number of students (*Info1*), actual number of students (*Info2*), number of exams (*Info3*), number of timeslots (*info4*), and number of rooms (*Info5*). We ran each experiment 10 times for each problem due to the stochastic nature of the method [13]. The Harmony Search Hyper-Heuristic (HSHH) parameters are set as HMS=10, HML=10, PAR=0.1 HMCR=0.95, and N1=100000, where these parameter settings are used based on some experiments carried out previously.

Experimentally, the HSHH is able to find a feasible solution for seven out of eight instances in ITC-2007 dataset. Table 4 provides the comparative results of the HSHH and the other comparative methods that are working using the same dataset. The different comparative methods are provided as shown in Table 5. The numbers in table 4 referred to the penalty value of the soft constraint violations. The best results are highlighted in bold. The indicator 'x% inf' indicates that the percentage of such algorithm could not find a feasible solution.

Table 3. The Characteristics of ITC-2007 Examination Timetabling Dataset

Dataset	*Info1*	*Info2*	*Info3*	*Info4*	*Info5*
Exam1	7891	7833	607	54	7
Exam2	12743	12484	870	40	49
Exam3	16439	16365	934	36	48
Exam4	5045	4421	273	21	1
Exam5	9253	8719	1018	42	3
Exam6	7909	7909	242	16	8
Exam7	14676	13795	1096	80	15
Exam8	7718	7718	598	80	8

Table 4. Comparison with previous HSHH and other methods

Dataset	**HSHH 2**	**HSHH 1**	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**	**M9**
Exam1	9885	11823	8559	6235	6234	4775	4370	4633	6582	**4368**	12035
Exam2	393	976	830	2974	395	**385**	**385**	405	1517	390	3074
Exam3	19931	26770	11576	15832	13002	**8996**	9378	9064	11912	9830	15917
Exam4	100% inf	100% inf	21901	35106	17940	16204	**15368**	15663	19657	17251	23582
Exam5	4065	6772	3969	4873	3900	**2929**	2988	3042	17659	3022	6860
Exam6	29935	30980	28340	31756	27000	**25740**	26365	25880	26905	25995	32250
Exam7	8801	11762	8167	11562	6214	4087	4138	**4037**	6840	4067	17666
Exam8	12145	16286	12658	20994	8552	7777	7516	**7461**	11464	7519	16184

Table 5. The comparison methods for ITC-2007 Examination Timetabling Dataset

Key	*Method*
HSHH 1	Harmony Search hyper-heuristic [5].
M1	Evolutionary Algorithm hyper-heuristic [14].
M2	Hybrid Approach hyper-heuristic[15].
M3	Graph Coloring Constructive hyper-heuristic[3].
M4	An improved multi-staged algorithmic[16].
M5	A Three phase constraint-based approach[17]
M6	An extended great deluge algorithm [18].
M7	Artificial Bee Colony algorithm [19] .
M8	Hybrid approach within great deluge algorithm[20].
M9	Developmental Approach [21].

As shown in Table 4, the performance of the new HSHH (i.e. HSHH 2) is much better than the performance of the previous version of HSHH (i.e. HSHH 1). Figure 1 shows the comparison between the HSHH 1 and HSHH 2 in terms of convergence behavior. Experimental results show that HSHH is able to produce good results and one of these datasets (i.e. Exam2) has achieved comparable result as shown in Table 4. Furthermore, the proposed method has also been able to obtain better results compared to the several other approaches. As compared with hybrid approach hyper-heuristic (M2), HSHH are able to produce better results in five problem instances (i.e., *Exam2, Exam5, Exam6, Exam7 and Exam8*) and six problem instances (i.e., *Exam1, Exam2, Exam5, Exam6, Exam7 and Exam8*) compared to the developmental approach (M9).

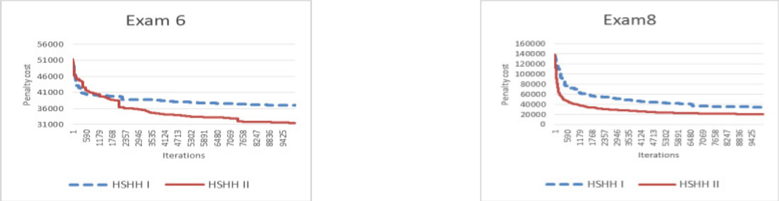


Fig. 1. Comparison of Convergence behavior between HSHH I and HSHH II

5 Conclusion and Future Work

This paper presented Harmony search-based hyper-heuristics (HSHH) for solving examination timetabling problems using the ITC-2007 dataset. The harmony search is utilized at the high-level to evolve a sequence of improvement low level heuristics. In order to evaluate HSHH, problem instances of ITC-2007 dataset have been used. The experimental result shows that HSHH is able to solve examination timetabling problems. Although, the results produced by HSHH in this study have not reached the best known results, they seem comparable or even better in some cases when compared to the previous approaches using ITC-2007. Utilizing several low-level heuristics in the HSHH framework is a very promising extension to the hyper-heuristic domain in general. This is because each low-level heuristic can deal with a region of the search and touching several regions in the search space might increase the chance of improvements.

The main objective behind proposing the HSHH is to suggest an applicable framework that is general enough to be re-implemented for other types of scheduling or combinatorial optimization problems. Therefore, we plan to apply the proposed approach to solve the nurse rostering problem using INRC2010 dataset. For future research, we plan to adapt learning mechanism within the HSHH algorithm in order to improve the heuristic selection and to enhance the speed of convergence.

References

1. Qu, R., et al.: A Survey of Search Methodologies and Automated System Development for Examination Timetabling. Journal of Scheduling 12(1), 55–89 (2009)
2. Burke, E.K., et al.: A survey of Hyper-heuristics. Computer Science Technical Report No. NOTTCS-TR-SUB-0906241418-2747, School of Computer Science and Information Technology, University of Nottingham (2009)
3. Sabar, N.R., et al.: A Graph Coloring Constructive Hyper-heuristic for Examination Timetabling Problems. Applied Intelligence, 1–11 (2011)
4. Pillay, N., Banzhaf, W.: A Genetic Programming Approach to the Generation of Hyper-Heuristics for the Uncapacitated Examination Timetabling Problem. In: Neves, J., Santos, M.F., Machado, J.M. (eds.) EPIA 2007. LNCS (LNAI), vol. 4874, pp. 223–234. Springer, Heidelberg (2007)

5. Anwar, K., et al.: Harmony Search-based Hyper-heuristic for Examination Timetabling. In: 9th International Colloquium on Signal Processing and its Applications (CSPA). IEEE (2013)
6. Geem, Z.W., Kim, J.H., Loganathan, G.: A New Heuristic Optimization Algorithm: Harmony Search. Simulation 76(2), 60–68 (2001)
7. Al-Betar, M.A., Khader, A.T.: A Harmony Search Algorithm for University Course Timetabling. Annals of Operations Research, 1–29 (2012)
8. Awadallah, M.A., et al.: Nurse Scheduling Using Harmony Search. In: Sixth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA). IEEE (2011)
9. Awadallah, M.A., et al.: Harmony Search with Novel Selection Methods in Memory Consideration for Nurse Rostering Problem. Asia-Pacific Journal of Operational Research (2013)
10. Al-Betar, M.A., Khader, A.T., Nadi, F.: Selection Mechanisms in Memory Consideration for Examination Timetabling with Harmony Search. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO 2010). ACM, New York (2010)
11. Manjarres, D., et al.: A Survey on Applications of the Harmony Search Algorithm. Engineering Applications of Artificial Intelligence 26(8), 1818–1831 (2013)
12. McCollum, B., et al.: The Second International Timetabling Competition: Examination Timetabling Track. Technical Report QUB/IEEE/Tech/ITC2007/Exam/v4. 0/17, Queens University, Belfast, UK (2007)
13. Al-Betar, M.A., Khader, A.T., Doush, I.A.: Memetic Techniques for Examination Timetabling. Annals of Operations Research, 1–28 (2013)
14. Pillay, N.: Evolving Hyper-heuristics for a Highly Constrained Examination Timetabling Problem. In: Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling, PATAT 2010 (2010)
15. Burke, E.K., Qu, R., Soghier, A.: Adaptive Selection of Heuristics for Improving Constructed Exam Timetables. In: Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling, PATAT 2010 (2010)
16. Gogos, C., Alefragis, P., Housos, E.: An Improved Multi-staged Algorithmic Process for the Solution of the Examination Timetabling Problem. Annals of Operations Research 194(1), 203–221 (2012)
17. Müller, T.: ITC2007 Solver Description: a Hybrid Approach. Annals of Operations Research 72(1), 429–446 (2009)
18. McCollum, B., et al.: An Extended Great Deluge Approach to the Examination Timetabling Problem. In: Proceedings of the 4th Multidisciplinary International Scheduling: Theory and Applications (MISTA 2009), pp. 424–434 (2009)
19. Alzaqebah, M., Abdullah, S.: Artificial Bee Colony Search Algorithm for Examination Timetabling Problems. International Journal of the Physical Sciences 6(17), 4264–4272 (2011)
20. Turabieh, H., Abdullah, S.: An Integrated Hybrid Approach to the Examination Timetabling Problem. Omega 39(6), 598–607 (2011)
21. Pillay, N.: A Developement Approach to the Examination Timetabling (2007), http://www.cs.qub.ac.uk/itc2007

Predator-Prey Pigeon-Inspired Optimization for UAV Three-Dimensional Path Planning

Bo Zhang[1] and Haibin Duan[1,2,*]

[1] Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, P. R. China
[2] Provincial Key Laboratory for Information Processing Technology, Soochow University, Suzhou, 215006, P. R. China
zhangbo0216@163.com, hbduan@buaa.edu.cn

Abstract. Pigeon-inspired optimization (PIO) is a new bio-inspired optimization algorithm. This algorithm searches for global optimum through two models: map and compass operator model is presented based on magnetic field and sun, while landmark operator model is designed based on landmarks. In this paper, a novel Predator-prey pigeon-inspired optimization (PPPIO) is proposed to solve the three-dimensional path planning problem of unmanned aerial vehicles (UAVs), which is a key aspect of UAV autonomy. To enhance the global convergence of the PIO algorithm, the concept of predator-prey is adopted to improve global best properties and enhance the convergence speed. The comparative simulation results show that our proposed PPPIO algorithm is more efficient than the basic PIO and particle swarm optimization (PSO) in solving UAV three-dimensional path planning problems.

Keywords: pigeon-inspired optimization (PIO), unmanned aerial vehicle (UAV), path planning, predator-prey.

1 Introduction

Three-dimensional path planner is an essential element of the unmanned aerial vehicle (UAV) autonomous control module [1]. It allows the UAV to compute the best path from a start point to an end point autonomously [2, 3]. Whereas commercial airlines fly constant prescribed trajectories, UAVs in operational areas have to travel constantly changing trajectories that depend on the particular terrain and conditions prevailing at the time of their flight.

Pigeon-inspired optimization (PIO), which is a new swarm intelligence optimizer based on the movement of pigeons, was firstly invented by Duan in 2014 [4]. Homing pigeons can easily find their homes by using three homing tools: magnetic field, sun and landmarks. In the optimization, map and compass model is presented based on magnetic field and sun, while landmark operator model is presented based on landmarks.

In this paper, we propose a predator-prey pigeon-inspired optimization (PPPIO) method, integrating the concept of predator-prey into PIO in order to improve its

[*] Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 96–105, 2014.
© Springer International Publishing Switzerland 2014

capability of finding satisfactory solutions and increasing the diversity of the population. We also solve the UAV three-dimensional path planning problem by PPPIO. Simulation results and comparisons verified the feasibility and effectiveness of our proposed algorithm.

The rest of the paper is organized as follows: Section 2 provides the representation and the cost function we developed to evaluate the quality of candidate trajectories. Section 3 describes the principle of basic PIO algorithm. Section 4 shows the implementation procedure of our proposed predator-prey PIO algorithm. Finally, we compare the quality of the trajectories produced by the PIO, particle swarm optimization (PSO) and the PPPIO in Section 5.

2 Problem Formulation

The first step of three-dimensional path planning is to discretize the world space into a representation that will be meaningful to the path planning algorithm. In this work, we use a formula to indicate the terrain environment. The mathematical function is of the form [5]:

$$z(x,y)=\sin(x/5+1)+\sin(y/5)+\cos(a\bullet\sqrt{x^2+y^2})+\sin(b\bullet\sqrt{x^2+y^2}) \quad (1)$$

where z indicate the altitude of a certain point, and *a, b* are constants experimentally defined. Our representation of cylindrical danger zones (or no-fly zones) to be in a separate matrix where each row represents the coordinates $(\mathrm{x}_i,\mathrm{y}_i)$ and the radius r_i of the *i*th cylinder as shown in Eq. (2). Complex no-fly zone can be built by partially juxtaposing multiple cylinders

$$\text{danger zones}=\begin{pmatrix} x_1 & y_1 & r_1 \\ x_2 & y_2 & r_2 \\ \cdots & \cdots & \cdots \\ x_n & y_n & r_n \end{pmatrix} \quad (2)$$

The three-dimensional trajectories generated by the algorithm are composed of line segments and $(\mathrm{x}_i,\mathrm{y}_i,\mathrm{z}_i)$ represents the coordinates of the ith way point. The trajectories are flown at constant speed.

In the situation of UAV path planning, the optimal path is complex and includes many different characteristics. To take into account these desired characteristics, a cost function is used and the path planning algorithm becomes a for a path that will minimize the cost function. We define our cost function as follows [6]:

$$F_{\text{cost}}=C_{\text{length}}+C_{\text{altitude}}+C_{\text{danger zones}}+C_{\text{power}}+C_{\text{collision}}+C_{\text{fuel}} \quad (3)$$

In the cost function, the term associated with the length of a path is defined as follows:

$$C_{\text{length}}=1-(\frac{L_{\text{p1p2}}}{L_{\text{traj}}}) \quad (4)$$

$$C_{\text{length}}\in[0,1] \quad (5)$$

where L_{p1p2} is the length of the straight line connecting the starting point $P1$ and the end point $P2$ and L_{traj} is the actual length of the trajectory.

The term associated with the altitude of the path is defined as follows:

$$C_{\mathrm{altitude}} = \frac{A_{\mathrm{traj}} - Z_{\min}}{Z_{\max} - Z_{\min}} \tag{6}$$

$$C_{\mathrm{altitude}} \in [0,1] \tag{7}$$

where $Z_{\max}$ is the upper limit of the elevation in our search space, $Z_{\min}$ is the lower limit and A_{traj} is the average altitude of the actual trajectory. $Z_{\max}$ and $Z_{\min}$ are respectively set to be slightly above the highest and lowest point of the terrain.

The term associated with the violation of the danger zones is defined as follows:

$$C_{\text{danger zones}} = \frac{L_{\text{inside d.z.}}}{\sum_{i=1}^{n} d_i} \tag{8}$$

$$C_{\text{danger zones}} \in [0,1] \tag{9}$$

where n is the total number of danger zones, $L_{\text{inside d.z.}}$ is the total length of the subsections of the trajectory which go through danger zones and d_i is the diameter of the danger zone i.

The term associated with a required power higher than the available power of the UAV is defined as follows:

$$C_{\mathrm{power}} = \begin{cases} 0, & L_{\text{not feasible}} = 0 \\ P + \left(\dfrac{L_{\text{not feasible}}}{L_{\mathrm{traj}}} \right), & L_{\text{not feasible}} > 0 \end{cases} \tag{10}$$

$$C_{\mathrm{power}} \in 0 \bigcup [P, P+1] \tag{11}$$

where $L_{\text{not feasible}}$ is the sum of the lengths of the line segments forming the trajectory which require more power than the available power of the UAV, L_{traj} is the total length of the trajectory and P is the penalty constant. This constant must be higher than the cost of the worst feasible trajectory which would have, based on our cost function, a cost of 3. By adding this penalty P, we separate nonfeasible solutions from the feasible ones.

The term associated with ground collisions is defined as follows:

$$C_{\mathrm{collision}} = \begin{cases} 0, & L_{\text{under terrain}} = 0 \\ P + \left(\dfrac{L_{\text{under terrain}}}{L_{\mathrm{traj}}} \right), & L_{\text{under terrain}} > 0 \end{cases} \tag{12}$$

$$C_{collision} \in 0 \bigcup [P, P+1] \tag{13}$$

where $L_{\text{under terrain}}$ is the total length of the subsections of the trajectory which travels below the ground level and L_{traj} is the total length of the trajectory.

The term associated with an insufficient quantity of fuel available is defined as follows:

$$C_{\text{fuel}} = \begin{cases} 0, & \mathrm{F}_{\text{traj}} \leq F_{\text{init}} \\ P+1-\left(\dfrac{F_{\text{P1P2}}}{F_{\text{traj}}}\right), & \mathrm{F}_{\text{traj}} > F_{\text{init}} \end{cases} \tag{14}$$

$$C_{\text{fuel}} \in 0 \bigcup [P, P+1] \tag{15}$$

where F_{P1P2} is the quantity of fuel required to fly the imaginary straight segment connection the starting point $P1$ to the end point $P2$, F_{traj} is the actual amount of fuel needed to fly the trajectory, F_{init} is the initial quantity of fuel on board the UAV.

The search engine will be adopted to find a solution, which can minimize the cost function during the optimization phase of our path planner algorithm. This can also be explained as to find a trajectory that best satisfies all the qualities represented by this cost function. Our cost function demonstrates a specific scenario where the optimal path minimizes the distance travelled, the average altitude (to increase the stealthiness of the UAV) and avoids danger zones, while respecting the UAV performance characteristics. This cost function is highly complex and demonstrates the power of our path planning algorithm. However, this cost function could easily be modified and applied to a different scenario.

3 Principle of Basic PIO

PIO is a novel swam intelligence optimizer for solving global optimization problems. It is based on natural pigeon behavior. Studies show that the species seem to have a system in which signals from magnetite particles are carried from the nose to the brain by the trigeminal nerve [4, 7]. Evidence that the sun is also involved in pigeon navigation has been interpreted, either partly or entirely, in terms of the pigeon's ability to distinguish differences in altitude between the Sun at the home base and at the point of release [8]. Recent researches on pigeons' behaviors also show that the pigeon can follow some landmarks, such as main roads, railways and rivers rather than head for their destination directly. The migration of pigeons is summarized as two mathematical models. One is map and compass operator, and the other is landmark operator.

3.1 Map and Compass Operator

In PIO model, virtual pigeons are used. In the map and compass operator, the rules are defined with the position X_i and the velocity V_i of pigeon i, and the positions and velocities in a *D*-dimension search space are updated in each iteration. The new position X_i and velocity V_i of pigeon i at the t-th iteration can be calculated with the follows [3]:

$$V_i(t) = V_i(t-1) \cdot e^{-Rt} + rand \cdot (X_g - X_i(t-1)) \tag{16}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{17}$$

where R is the map and compass factor, *rand* is a random number, and X_g is the current global best position, and which can be obtained by comparing all the positions among all the pigeons.

3.2 Landmark Operator

In the landmark operator, half of the number of pigeons is decreased by N_p in every generation. However, the pigeons are still far from the destination, and they are unfamiliar the landmarks. Let $X_c(t)$ be the center of some pigeons' position at the t-th iteration, and suppose every pigeon can fly straight to the destination. The position updating rule for pigeon i at t-th iteration can be given by:

$$N_P(t) = \frac{N_P(t-1)}{2} \tag{18}$$

$$X_c(t) = \frac{\sum X_i(t) \cdot fitness\left(X_i(t)\right)}{N_P \sum fitness\left(X_i(t)\right)} \tag{19}$$

$$X_i(t) = X_i(t-1) + rand \cdot (X_c(t) - X_i(t-1)) \tag{20}$$

where *fitness* is the quality of the pigeon individual. For the minimum optimization problems, we can choose $fitness\left(X_i(t)\right) = \frac{1}{f\left(X_i(t)\right) + \varepsilon}$ for maximum optimization problems, we can choose $fitness\left(X_i(t)\right) = f\left(X_i(t)\right)$.

4 PPPIO for Three-Dimensional Path Planning

4.1 Predator-Prey Concept

Predatory behavior is one of the most common phenomena in nature, and many optimization algorithms are inspired by the predator-prey strategy from ecology [9]. In nature, predators hunt prey to guarantee their own survival, while the preys need to be able to run away from predators. On the other hand, predators help to control the prey population while creating pressure in the prey population. In this model, an individual in predator population or prey population represents a solution, each prey in the population can expand or get killed by predators based on its fitness value, and a predator always tries to kill preys with least fitness in its neighborhood, which represents removing bad solutions in the population. In this paper, the concept of predator-prey is used to increase the diversity of the population, and the predators are modeled based on the worst solutions which are demonstrated as follows:

$$P_{\text{predator}} = P_{\text{worst}} + \rho(1 - t / t_{\max}) \tag{21}$$

where P_{predator} is the predator (a possible solution), P_{worst} is the worst solution in the population, t is the current iteration, while $t_{\max}$ is the maximum number of iterations and ρ is the hunting rate. To model the interactions between predator and prey, the solutions to maintain a distance of the prey from the predator is showed as follows:

$$\begin{cases} P_{\text{k+1}} = P_{\text{k}} + \rho e^{-|d|}, & \text{d>0} \\ P_{\text{k+1}} = P_{\text{k}} - \rho e^{-|d|}, & \text{d<0} \end{cases} \tag{22}$$

where d is the distance between the solution and the predator, and k is the current iteration.

4.2 Parallelization of the Map and Compass Operations and the Landmark Operations

In the basic model of PIO algorithm, the landmark operation is used after several iterations of map and compass operation. For example, when the number of generations N_c is larger than the maximum number of generations of the map and compass operation $N_{c\max 1}$. The map and compass operator will stop and it the landmark operation will be start. During my experiment, we found it's easy to fall into a local best solution before the number of generations got to $N_{c\max 1}$. Furthermore, half of the number of pigeons is decreased by N_p in every generation on the landmark operator. The population of pigeons is decreased too rapidly according to formula (18), which would reach to zero after a small amount of iterations. The landmark operator would make only a small impact on the pigeons' position by this way. So we make a small modification on the basic PIO algorithm. The map and compass operation and the compass operation are used parallelly at each iteration. A parameter ω is used to define the impaction of the landmark increase with a smoothly path. And a constant parameter c is used to define the number of pigeons that are in the landmark operator. Our new formula of landmark operator is as follows:

$$N_P(t) = c \cdot N_{P\max} \qquad \text{c} \in (0,1) \tag{23}$$

$$X_c(t) = \frac{\sum X_i(t) \cdot fitness\left(X_i(t)\right)}{N_P \sum fitness\left(X_i(t)\right)} \tag{24}$$

$$\omega = s + (1 - \mathrm{s}) \cdot \mathrm{t} / \mathrm{N}_{c\max} \qquad \mathrm{s} \in (0,1) \tag{25}$$

$$X_i(t) = X_i(t-1) + \omega \cdot rand \cdot (X_c(t) - X_i(t-1)) \tag{26}$$

where s is a constant experimentally defined.

4.3 Proposed Predator-Prey PIO (PPPIO) Based Path Planner

In order to overcome the disadvantages of the classical PIO algorithm, such as the tendency to converge to local best solutions, PPPIO, which integrates PIO with the concept of predator-prey, was proposed in our work. After the mutation of each generation, the predator-prey behavior is been conducted in order to choose better solutions into next generation. In this way, our proposed algorithm takes the advantage of the predator-prey concept to make the individuals of sub generations distributed ergodically in the defined space and it can avoid from the premature of the individuals, as well as to increase the speed of finding the optimal solution.

The implementation procedure of our proposed PIO approach to UAV path planning can be described as follows:

Step 1: According to the environmental modeling in Section 2, initialize the detailed information about the path planning task.

Step 2: Initialize the PIO parameters, such as solution space dimension *D*, the population size N_p, map and compass factor R, the number of iteration N_c.

Step 3: Set each pigeon with a randomized velocity and path. Compare the fitness of each pigeons, and find the current best path.

Step 4: Operate map and compass operator. Firstly, we update the velocity and path of every pigeon by using Eqs. (16) and (17).

Step 5: Rank all pigeons according their fitness values. Some of pigeons whose fitness are low will follow those pigeons with high fitness according to Eq. (23). We then find the center of all pigeons according to Eq. (24), and this center is the desirable destination. All pigeons will fly to the destination by adjusting their flying direction according to Eq. (26). Next, store the best solution parameters and the best cost value.

Step 6: Model the predators based on the worst solution as Eq. (15) demonstrates. Then, use Eq. (16) to provide the other solutions to maintain a distance between the predator and the prey.

Step 7: If $Nc > N_{c\max}$, stop the iteration, and output the results. If not, go to step 6.

5 Comparative Experimental Results

In order to evaluate the performance of our proposed PPPIO algorithm in this work, series of experiments are conducted in Matlab2012a programing environment. Coordinates of a starting point are set as (10, 16, 0), and the target point as (55, 100, 0). The initial parameters of PIO algorithm were set as: NP =150. The comparative results of PPPIO with PIO and PSO are showed as follows:

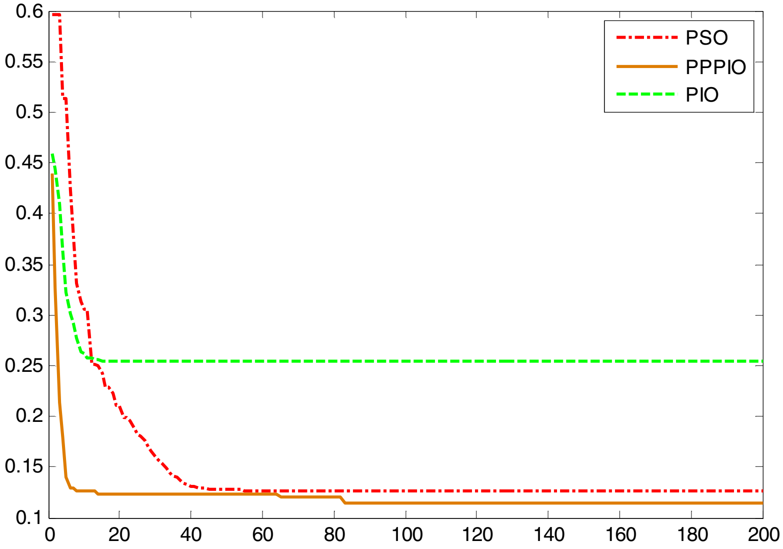


Fig. 1. Comparative evolutionary curves of PPPIO, PIO and PSO

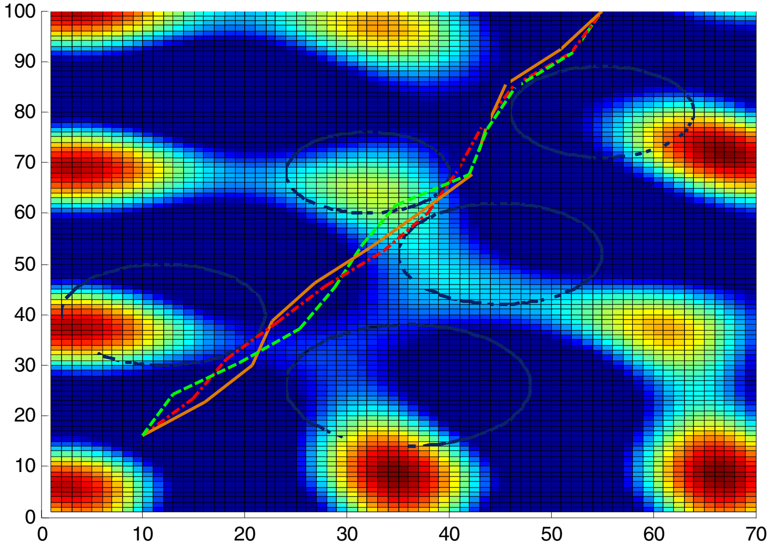


Fig. 2. Comparative path planning results of PPPIO, PIO and PSO

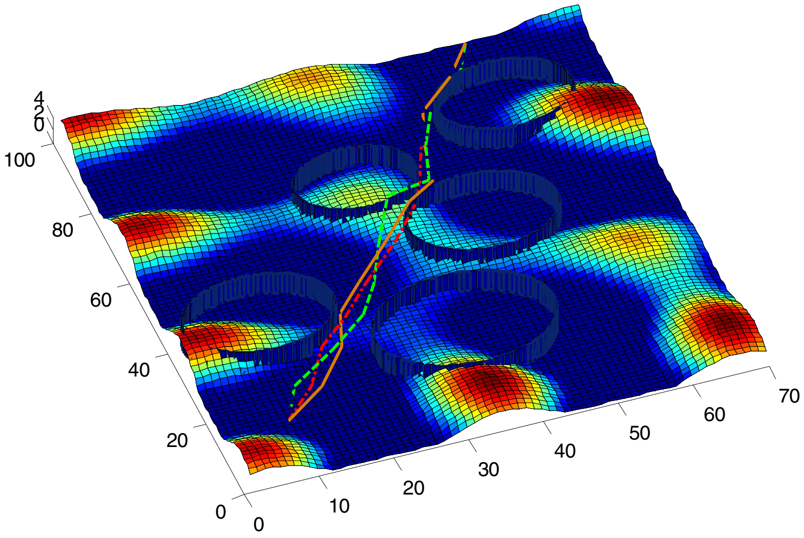


Fig. 3. Comparative path planning results of PPPIO, PIO and PSO on 3D version

6 Conclusions

This paper proposed a novel PPPIO algorithm for solving the UAV three-dimensional path planning problem in complex environments. The concept of predator-prey is adopted to improve the performance of the basic PIO algorithm. Series of comparative simulation results were given to show that our proposed PPPIO algorithm is more efficient than basic PIO and PSO in solving UAV three-dimensional path planning problems.

Acknowledgements. This work was partially supported by National Key Basic Research Program of China(973 Project) under grant #2014CB046401, Natural Science Foundation of China (NSFC) under grant # 61333004 and #61273054, National Magnetic Confinement Fusion Research Program of China under grant # 2012GB102006, and Aeronautical Foundation of China under grant #20135851042.

References

1. Chen, H., Wang, X.M., Li, Y.: A Survey of Autonomous Control for UAV. In: International Conference on Artificial Intelligence and Computational Intelligence, vol. 2, pp. 267–271 (2009)
2. Duan, H.B., Li, P.: Bio-inspired Computation in Unmanned Aerial Vehicles. Springer, Heidelberg (2014)

3. Duan, H.B., Luo, Q.N., Ma, G.J., Shi, Y.H.: Hybrid Particle Swarm Optimization and Genetic Algorithm for Multi-UAVs Formation Reconfiguration. IEEE Computational Intelligence Magazine 8(3), 16–27 (2013)
4. Duan, H.B., Qiao, P.X.: Pigeon-Inspired Optimization: A New Swarm Intelligence Optimizer for Air Robot Path Planning. International Journal of Intelligent Computing and Cybernetics 7(1), 24–37 (2014)
5. Ioannis, K.N., Athina, N.B.: Coordinated UAV Path Planning Using Differential Evolution. In: IEEE International Symposium on, Mediterrean Conference on Control and Automation, vol. 70, pp. 77–111. Springer, Heidelberg (2005)
6. Vincent, R., Mohammed, T., Gilles, L.: Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. IEEE Transactions on Industrial Informatics 9(1), 132–141 (2013)
7. Mora, C.V., Davison, M., Wild, J.M., Michael, M.W.: Magnetoreception and Its Trigeminal Mediation in the Homing Pigeon. Nature 432, 508–511 (2004)
8. Whiten, A.: Operant Study of Sun Altitude and Pigeon Navigation. Nature 237, 405–406 (1972)
9. Zhu, W.R., Duan, H.B.: Chaotic Predator-Prey Biogeography-Based Optimization Approach for UCAV Path Planning. Aerospace Science and Technology 32(1), 153–161 (2014)

Research on Route Obstacle Avoidance Task Planning Based on Differential Evolution Algorithm for AUV

Jian-Jun Li[1,3], Ru-Bo Zhang[1,2], and Yu Yang[3]

[1] College of Computer Science and Technology,
Harbin Engineering University, Harbin 150001, China
[2] College of Electromechanical & Information Engineering,
Dalian Nationalities University, Liaoning Dalian, 116600, China
[3] School of Computer and Information Engineering ,
Harbin University of Commerce, Harbin 150028, China

Abstract. AUV mission planning route avoidance purpose is to be able to successfully avoid the threat of a number of different levels of obstacles between the start and end of the route , and plan the optimal route planning to meet certain performance indicators. Through the differential evolution algorithm analysis and description , the avoidance route mission planning problem into a multi-dimensional function optimization problems, optimization problems for AUV mission planning route avoidance functions , based on differential evolution algorithm is proposed route obstacle avoidance task planning methods and after a comprehensive analysis and simulation results validate the differential evolution algorithm in high-dimensional function optimization convergence and stability demonstrated good performance.

Keywords: Differential evolution algorithm, Autonomous Underwater Vehicle, route avoidance, mission planning.

1 Introduction

Due to the complex undersea environment, (Autonomous Underwater Vehicle, AUV), Also known as underwater robots, the need for obstructions on the route, torpedoes and other potential threats to take evasive strategy. Josep et al proposed a low cost computing underwater vehicle planning program, create a static or dynamic obstacle avoidance optimization campaign mode [1]. Zouming Cheng et al use of electronic navigation and positioning collision path planning , and application of artificial intelligence ant colony optimization algorithm to construct collision model[2]. Cruz made hydrodynamics based obstacle avoidance algorithm to determine the location of obstacles and target drones by the harmonic function, thus avoiding local optimum[3]. China Shipbuilding Industry Corporation710Research Institute, Yan Gang, who proposed an improved genetic algorithm to improve search speed and path optimization AUV levels[4]. AUV route avoidance is a key component of AUV mission planning system. Differential evolution algorithm (Differential Evolution

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 106–113, 2014.
© Springer International Publishing Switzerland 2014

Algorithm, DE) by the American scholar Storn and Price propose a heuristic algorithm to solve optimization problems[5]. Differential evolution algorithm in solving global optimization problems in complex environments and continuous domain optimization problem, with outstanding advantages. Therefore, based on differential evolution algorithm AUV route avoidance, the successful completion of the task execution for AUV has important practical significance.

2 Differential Evolution Algorithm

Differential evolution algorithm remembers the evolution of individual groups and groups in the optimal solution features internal information sharing, through competition and cooperation between individuals within the group to achieve the optimal solution. Assuming a population size of NP , m to the first generation of the evolution of the population of $X(m)$,The dimension of the solution space is K .Initial population $X(0)=\left\{x_1^0, x_2^0, \cdots, x_{NP}^0\right\}$, Solutions of the i-th individual $x_i^0=\left[x_{i,1}^0, x_{i,2}^0, \cdots, x_{i,k}^0\right]$. Individual components of the formula:

$$x_{i,j}^0 = x_{j,\min} + rand(x_{j,\max} - x_{j,\min}) \tag{1}$$

$x_{j,\max}$ upper bound for the solution space, $x_{j,\min}$ lower bound for the solution space. Differential evolution algorithm including mutation , crossover and selection of three operating [6-9].

2.1 Mutation

Differential evolution algorithm mutation is the last generation of linear combinations of multiple individuals in the population , the variability of individual difference vector generation. Process variation follows the formula:

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}), i = 1, 2, \cdots, NP \tag{2}$$

From the previous generation arbitrary choice of three different populations of individuals $\left\{x_{r1}, x_{r2}, x_{r3}\right\}$, and $r_1 \neq r_2 \neq r_3$,F constant factor between [0,2] between, Also known as the scaling factor, Used to control the difference vector $\left(x_{r2}-x_{r3}\right).\left(x_{r2}-x_{r3}\right)$ smaller value of the difference vector, The smaller the disturbance, That is closer to the optimal value group , the disturbance value is automatically reduced.

2.2 Crossover

Differential evolution algorithm is a variation vector crossover target vectors v_i and x_i random reorganization , thereby increasing the diversity of the population of individuals. Process crossover following formula:

$$u_{ij} = \begin{cases} v_{i,j}, randk \leq CR \; or \; j = rand_j; \\ x_{i,j}, randk > CR \; or \; j \neq rand_j; \end{cases} \quad (3)$$
$$i = 1, \cdots, NP, j = 1, \cdots, NP$$

$randk$ Is a random variable [0,1],CRIs a constant [0,1]. CR The larger the value, the greater the probability of crossover, CR=0 Cross probability 0.

2.3 Select Options

Differential evolution algorithm selection operation is to adapt to the new vector values and objectives of the individual u_i vector of individual fitness value x_i compare, When the value of u_i is better than x_i , replacing x_i . The select operation by the following formula :

$$x_i^{t+1} = \begin{cases} u_i, f(u_i) < f(x_i^t) \\ x_i^t, \text{other} \end{cases} \quad (4)$$

3 AUV Multiple Route Avoidance Model

AUV safe navigation area, Refers to the current position of the center AUV, R_ϕ radius of the circular area , And there is no obstacle in this circular area , the AUV can be achieved without collision safe navigation. If there is an obstacle , then the AUV be single or multiple route avoidance[10-11].

AUV route avoidance of multiple models , which means that AUV underwater work space with the ability to meet multiple obstacle avoidance.

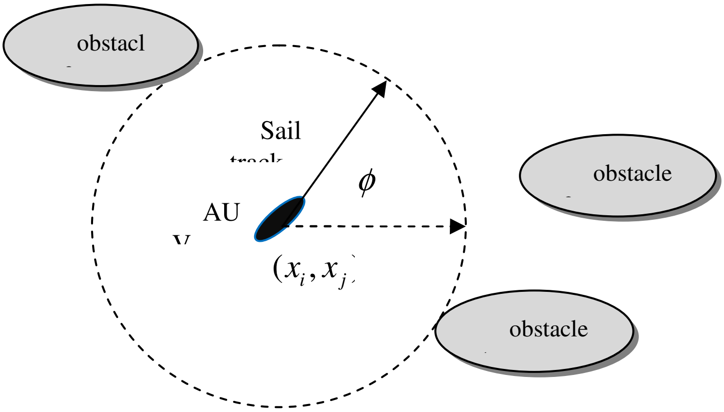


Fig. 1. AUV safe navigation area

3.1 Single Obstacle Avoidance Route Model

AUV set maximum safe radius r_a , Obstacle radius r is between $(c_x, c_y) \in R^+$,$(a_x, a_y) \in R^+$ between the starting point A and the target point B between $(b_x, b_y) \in R^+$ .If the following conditions are met.

$$\sqrt{(\Delta c_x)^2 + (\Delta c_y)^2} > \sqrt{(\Delta b_x)^2 + (\Delta b_y)^2} + r + r_a \tag{5}$$

A starting point is the presence of AUV through navigational path and destination point B without a single obstacle collision avoidance.

3.2 Route Multiple Obstacles Avoidance Model

AUV set maximum safe radius of r_a, and set between the start and end of all obstructions are located between. A starting point is between $(a_x, a_y) \in R^+$,B target point is between $(c_x, c_y) \in R^+$ and an n the obstacle underwater space. Assuming the k the obstacle $b(k)$ is located $(b_{kx}, b_{ky}) \in R^+$, the radius of the obstacle is r_k . Then , if the following conditions.

$$\sqrt{(\Delta c_x)^2 + (\Delta c_y)^2} > \sqrt{(\Delta b_{kx})^2 + (\Delta b_{ky})^2} + r_{\max} + r_a, \forall (b_{kx}, b_{ky}) \in O \tag{6}$$

A starting point is the presence of AUV through navigational path and destination point B without collision avoidance multiple obstacles.

4 Experimental Verification

4.1 Problem Description

AUV obstacle avoidance task route planning is planning to meet the optimal route planning based on certain performance indicators mission objectives. The route mission planning problem into a k dimensional function optimization problems.

$$\alpha = \arcsin \frac{y_2 - y_1}{\left|\overrightarrow{AB}\right|} \tag{7}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \tag{8}$$

According to equation (7) and Equation (8) to the original coordinate system conversion of the connection of the horizontal coordinate system for the new start and end points.

α is a coordinate rotation angle. The new coordinate system x' -axis is divided into k segments, optimizing the corresponding y' -coordinate.

After the conversion of the coordinates (x', y') connected in order to obtain a path connecting the start and end points , Thereby converting the problem into a k dimensional function route optimization problem.

4.2 Barriers Threat Level

AUV navigation path length $L_{i,j}$,The overall threat level barriers to the M_n obstacle

$$\lambda_{n,L_{ij}} = \int_0^{L_{ij}} \sum_{k=1}^{M_n} \frac{t_k}{[(x - x_k)^2 + (y - y_k)^2]^2} dl \tag{9}$$

The AUV navigation path into X segments, If the obstruction to navigation path segment from the threat within a radius of obstacles, the barrier is calculated as the threat level.

$$\lambda_{n,L_{ij}} = \frac{L_{ij}^5}{x}\sum_{k=1}^{M_n}\alpha_k\left(\frac{1}{l_{0.2,k}^4}+\frac{1}{l_{0.4,k}^4}+\frac{1}{l_{0.6,k}^4}+\frac{1}{l_{0.8,k}^4}+\frac{1}{l_{1.0,k}^4}\right) \quad (10)$$

The length of the start and end points y z edge L_{ij}, α_k obstacle to obstacle threat level, $l_{0.2,k}^4$ represents 1/5 the first pitch from the center of the k obstacle edge L_{ij}.

4.3 Program Flow

Steps are as follows :

Step1 Transformed coordinate system, the threat level obstacles to the rotating coordinate system conversion, and the horizontal axis of rotation of the coordinate system K aliquots.

Step2 Initialization K segment route , each route is calculated barriers threat level.

Step3 Iterative calculations.

Step4 For the population of feasible solutions consisting of K, perform mutation operation.

Step5 Individual against individual variation generated with the original crossover operation execution, generate new individuals.

Step6 Calculate the value of the cost function crossover operation to generate new individuals, compared to individuals with newly generated target individuals choose to perform the operation.

Step7 Iterations<maximum number of iterations, jump to Step3 iterative calculation, otherwise exit the loop.

Step8 Coordinate inverse transform, output optimal route avoidance task planning results.

4.4 Simulation

AUV starting point coordinates [10,10],End coordinates [55,100],Consideration weights0.5. Set the initial parameters , population size NP=20,Optimization dimension K=20,The maximum number of iterations $N_{c_{\max}}$ =200,Variability factor F=0.5,Cross factor CR=0.9.

Table 1. AUV route barrier parameter

Obstacle radius	Barriers threat level	Disorders Center
20	5	[45,50]
15	2	[12,40]
20	6	[32,48]
18	6	[36,26]
16	4	[22,40]
22	10	[30,45]

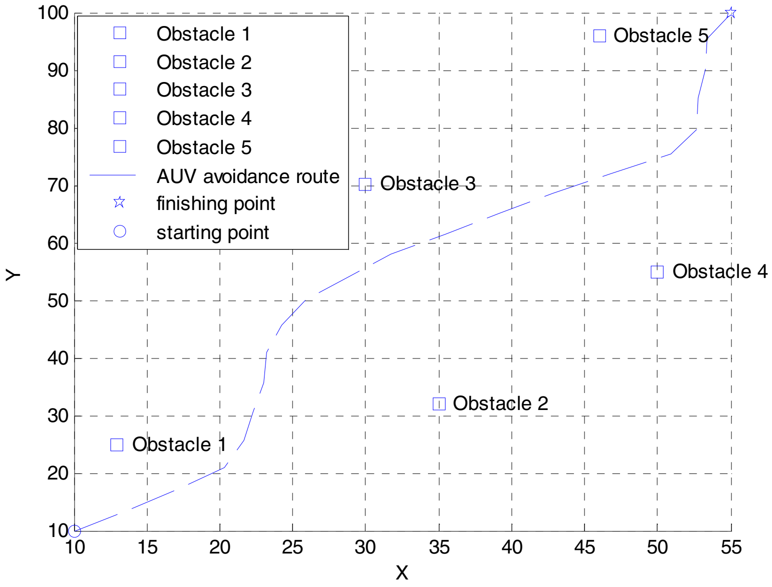


Fig. 2. AUV mission planning route avoidance

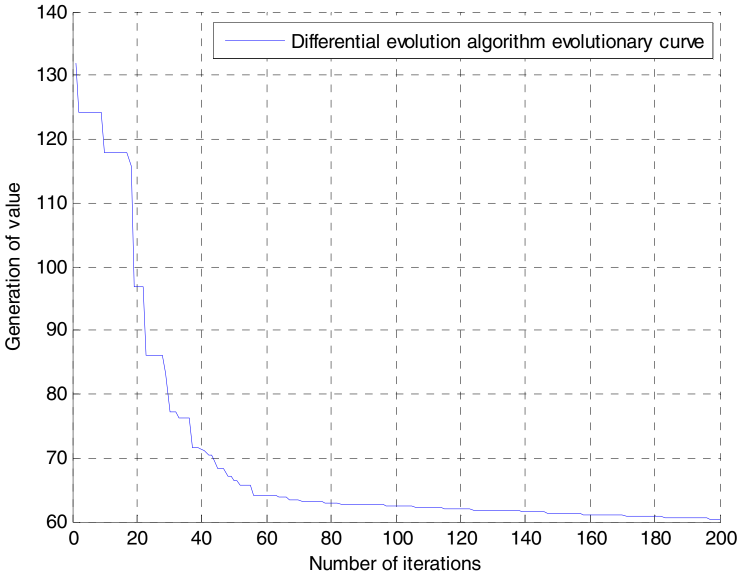


Fig. 3. Differential evolution algorithm evolutionary curve

5 Conclusion

By differential evolution algorithm route planning AUV mission planning can successfully avoid multiple obstacles, arrived in the end of the mission objectives. Simulation results show the differential evolution algorithm solves the problem of high-dimensional optimization , convergence speed and good performance.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China (60975071,61100005) , Ministry of Education, Scientific Research Project (13YJA790123).

References

1. Isern-Gonzalez, J., Hernandez-Sosa, D., et al.: Obstacle Avoidance in Underwater Glider Path Planning. Physical Agents 6(1), 11–20 (2012)
2. Tsou, M.-C., Hsueh, C.-K.: The Study of Ship Collision Avoidance Route Planningby Ant Colony Algorithm. Journal of Marine Science and Technology, 746–756 (2010)
3. Cruz, G.C.S., Encarnação, P.M.M.: Obstacle Avoidance for Unmanned Aerial Vehicles. Journal of Intelligent & Robotic Systems, 203–217 (2012)
4. Yan, G., Wang, L., Zhou, J., Zha, Z.: Path Planning Based on Improved Genetic Algorithm for AUV. Journal of Chongqing University of Technology, 115–120 (2010)
5. Storn, R., Price, K.: Differential Evolution A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, 341–359 (1997)
6. Price, K.: Differential Evolution A Fast and Simple Numerical Optimizer. In: Proceedings of Biennial Conference of the North American Fuzzy Information Processing Society, pp. 524–527 (1996)
7. Islam, S.M., Das, S.: An Adaptive Differential Evolution Algorithm With Novel Mutation and Crossover Strategies for Global Numerical Optimization. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernet ICS, 482–500 (2012)
8. Deng, Y., Beaujean, P.-P.J., An, E., Carlson, E.: Task Allocation and Path Planning for Collaborative Autonomous Underwater Vehicles Operating through an Underwater Acoustic Network. Journal of Robotics, 1–15 (2013)
9. Qu, H., Xing, K., Alexander, T.: An Improved Genetic Algorithm With Coevolution Strategy for Global Path Planning of Multiple Mobile Robots. Neuro Computing 120, 509–517 (2013)
10. Ye, W., Wang, C., Yang, M., Wang, B.: Virtual Obstacles Based Path Planning for Mobile Robots. Robot, 273–286 (2011)
11. Song, Q., Liu, L.: Mobile Robot Path Planning Based on Dynamic Fuzzy Artificial Potential Field Method. International Journal of Hybrid Information Technology 5, 85–94 (2012)

An Improved Particle Swarm Optimization-Based Coverage Control Method for Wireless Sensor Network

Huimin Du[1,2], Qingjian Ni[1,2,3,⋆], Qianqian Pan[4], Yiyun Yao[1], and Qing Lv[1]

[1] College of Software Engineering, Southeast University, Nanjing, China
[2] Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, China
[3] School of Computer Science and Engineering, Southeast University, Nanjing, China
[4] School of Information Science and Engineering, Southeast University, Nanjing, China
nqj@seu.edu.cn

Abstract. Coverage control plays a significant role in wireless sensor network (WSN) design. To meet a layout with a certain cover rate, movable nodes are maintained in deployment which accomplish self-organization through moving and changing topological structure. This paper proposes an improved discrete particle swarm optimization algorithm aimed at coverage control method of WSN, and the optimization is implemented under two processes: deployment planning and movement control. The method interpreted in this paper can be easily used solving such problems and the experiment result shows its efficiency, which will inspire new insights in this field.

Keywords: Wireless Sensor Network, Coverage Control, Discrete Particle Swarm Optimization.

1 Introduction

Wireless sensor network (WSN) in complex environment has typical characteristics like large-scale, self-organization, limited energy for nodes and inconstant topology structure, etc. Every node in the network contains a small volume, cheap, energy-saving, multifunction sensor and each sensor has the ability of signal acquiring, data handling and communicating with its neighbors. These features have made WSN topology control a challenging issue.

The quality of topology control influences directly on the lifetime and performance of networks, while a good topology scheme relies on a complete evaluation methodology. Composing those characters and system features, three following indicators are taken into major considerations[1] to evaluate the WSN topology control:

- Coverage: Coverage is a measure of WSN service quality, which is mainly focused on the coverage rate of initial nodes deployment and whether these nodes can acquire signals of the region of interest(ROI), completely and accurately.
- Connectivity: Sensor networks are usually of large scale, thus connectivity is an assurance that data information obtained by sensor can be delivered to sink nodes.

⋆ Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 114–124, 2014.
© Springer International Publishing Switzerland 2014

- Network lifetime: Network lifetime is generally defined as the time duration from the start to when the percentage of dead nodes comes to a threshold.

Coverage control, or deployment design, is the cornerstone of wireless sensor networks. Node deployment can follow two trends: structured and randomized[2]. Structured method are suitable for small scope deployment where nodes positions are predefined when planning, while randomized way are more pervasive. For supervisory region with large scope which is hard to approach for humans, nodes are initialized (airdropped usually) randomly and adjusted by topology control technology to achieve monitoring. In such case, the mobility of nodes is rather crucial.

Aleksandra *et al.* stated a hexagonal repartition-based C^2 algorithm[3]. This algorithm organizes the space to hexagonal grid, and chooses Cluster Heads (CHs) in the center for each grid cell, using them to rearrange the nodes inside and adjacent to the cells to improve the coverage ratio and connectivity. Zou *et al.* proposed a Virtual Force-based deployment algorithm(VFA), dividing the nodes in the network into clusters[4], where every cluster head node collects information of nodes inside the cluster and computes their final positions and instructs the movement of nodes. Ma *et al.* put forward an Adaptive Triangular Deployment Algorithm(ATRI) to deal with large-scope situations[5]. This process adapts node deployments to regular triangles, and divide node transmission range into six sector and thus nodes can be adjusted from view of its neighbors from each sector.

The utilization of swarm intelligence has made the control processes more effective and easier to implement. Liu[6] *et al.* introduced Easidesign algorithm for WSN coverage control based on Ant Colony Optimization(ACO), which combines greedy strategy and additional pheromone evaporation methods to satisfy network connectivity of different sink positions. A Virtual force co-evolutionary PSO(VFCPSO) was proposed by Wang *et al.*[7] In this algorithm a node is moved several times by the virtual force from other nodes, and the virtual force vectors come from the distance information, their moving direction and other factors. This can also reach a higher coverage ratio. Another situation where mixing stationary and mobile nodes is solved by Li *ed al.* using a novel particle swarm genetic optimization(PSGA), combining PSO and Genetic Algorithm(GA) to repair network holes in [8]. In this method, positions of mobile nodes (or robots) are adjusted to improve the quality-of-service(QoS). This method imports mutation and selection operators to PSO and implements some extra update methods, which are proved to be well-performed.

This paper proposed a novel discrete PSO strategy and applied it to WSN coverage control, to improve QoS stated in the following part. The rest of this paper is arranged as follows: Section 2 describes the abstraction and modeling of coverage problem, Section 3 explains the basic concepts of PSO, and a new discrete strategy with redefined operators is presented in this section. Experiments are conducted and analysed in Section 4, and conclusion shows up in Section 5.

2 Problem Analyses

2.1 Problem Statement

In WSNs, every node has a certain length of sense radius R_s and communication radius R_c. Metrics of QoS include coverage rate, uniformity, time and distance[9], and we mainly consider the coverage and distance problem in this paper.

Coverage. Measuring coverage rate is to detect the ratio of scope inside sense range to the whole object range. Coverage scope is often interpreted as the amount of area. For a node v_i, its coverage scope COV_i in the object region A equals to its sense range, and the total amount of coverage range of the network is explained in formula (1):

$$COV_A = \cup_{i=1}^{k} COV_i \tag{1}$$

thus coverage rate can be represented as (2):

$$C = COV_A / M \tag{2}$$

where M is the area of object region.

Raster coverage is a meliorative strategy, where the ROI is meshing into a grid. The grid points rather than the whole area are treated as the coverage object[10]. This strategy is expanded as region-based point covering in [6]. In such a problem, some geographical points that can show the environment situation are chosen as covering points (**CP**s), and the coverage object is to cover these CPs which are not necessarily grid distributed. Hence, coverage rate becomes the ratio of covered CPs. Typical applications of such problems are Environmental Monitoring Systems and Targets Monitoring, etc. For a node $v_i^{'}$, its coverage scope is the number of CPs inside its sense range, coverage rate is also formulated as (1)(2), where M is the total number of CPs. Research in this paper are expended based on raster coverage.

Distance. The distance a node travels in the movement process is related to the energy limitation. Therefore, optimization strategies are taken to minimize the distance of a node and the total distance of a network. Distance a node takes is regard as the moving range from its initial position to the objective.

2.2 Problem Modeling

Nodes are deployed randomly at initial stage and after the position optimization, a higher coverage rate is obtained. The goal of this process is to maximize the coverage rate and minimize moving distances as well, achieving the minimum energy cost.

In deployment design, object region are divided into a grid, and every grid point acts as a candidate position of nodes. Deployment process contains two procedures, deployment planning and movement control, which are explained as followings:

- For a determined region and CP distribution (showing in Fig.1), design the objective deployment layout for a certain amount of nodes.
- Adjust the positions of randomly-deployed nodes to meet the objective layout.

Formula (1) and (2) are detecting indicators in covering. As for moving process, whose object sketch is shown in Fig.2, consider two sets, $P = \{p_1, p_2, ..., p_N\}$ and $Q = \{q_1, q_2, ..., q_N\}$, where P is the set of stochastic positions generated preliminarily, and Q is the objective position set, N is the number of nodes. The purpose is to pair the vertexes from different set completely in an nonredundant way, that is to make a vertex q_i of Q the moving target of vertex p_j in P, and at the same time, minimizing total moving distance. This can be measured by (3),

$$F = \sum_{i=1}^{N} Distance(q_i, p_{pair\_with\_q_i}) \tag{3}$$

where F is the objective function to be minimized.

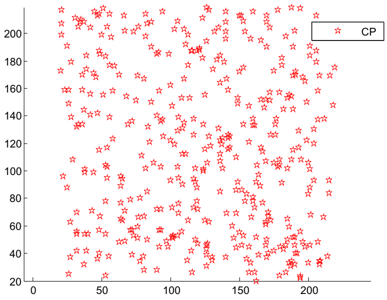


Fig. 1. Distribution of CPs

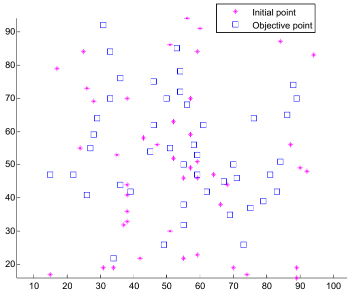


Fig. 2. Objective Sketch for Moving

3 Improved Particle Swarm Optimization for WSN Coverage Control

The section gives a brief summary of basic Particle Swarm Optimization and puts forward an improved method for discrete PSO.

3.1 Basic Principle of PSO

Particle Swarm Optimization (PSO) is a meta-heuristic algorithm that simulates the action of bird folks, proposed by J. Kennedy and R.C. Eberhart, and has been wildly used in solving NP-hard problems. Basic PSO method is:

$$v_{id}(t+1) = v_{id}(t) + c_1 \cdot r1 \cdot (p_{id}(t) - x_{id}(t)) + c_2 \cdot r2 \cdot (p_{gd}(t) - x_{id}(t)) \tag{4}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t) \tag{5}$$

Assume that in a searching space of D-dimension, a population consists of m particles. Position of particle i in the population can be represented by a D-dimension vector: $X_i = \{x_{i1}, x_{i2}, \cdots, x_{iD}\}$, its present velocity is expressed as $V_i = \{v_{i1}, v_{i2}, \cdots, v_{iD}\}$, and its particle best position is recorded as $P_i = \{p_{i1}, p_{i2}, \cdots, p_{iD}\}$, where $i = 1, 2, \cdots, m$. During optimization, a fitness function is built to judge the quality of a particle position.

3.2 Typical Discrete Strategy

Three typical discrete particle swarm optimization(DPSO) will be introduced: Binary Strategy, Integer Strategy[11] and DPSO aimed at TSP.

Binary PSO: In Binary PSO, each particle is made of binary decisions, valid values are like 0(False) and 1(True), and the probability of those values are modeled as[12]:

$$P(X_{id} = 1) = f(X_{id}(t-1), V_{id}(t-1), p_{id}, p_{gd}) \tag{6}$$

It's proved that the probability for a dimension to choose 1/True of a particle is a multivariate function, depending on its previous position X_{id} and velocity(trend), which are radically decided by the particle itself and the environment, i.e., p_{id} and p_{gd}. Use sigmoidal function (7)

$$s(V_{id}) = \frac{1}{1 + exp(-V_{id})} \tag{7}$$

can educe the corresponding threshold determined by V_{id} in the probability function P. And

$$X_{id}(t) = \begin{cases} 1, & if \quad rand() < s(V_{id}) \\ 0, & otherwise \end{cases} \tag{8}$$

gives the value conditions of X_{id}.

Integer PSO: Integer PSO comes out when solutions are constrained in only integer space but not necessarily binary. This can be seemed as a constrained problem in continuous space, where optimal values are approximated to integers. Advantage of this method is that, though solution space are truncated, algorithm performs still well[13].

DPSO for TSP: Clerc[14], Wang[15] *et al.*, Shi[16] *et al.* all proposed novel DPSOs when solving traveling salesman problem(TSP). Basic thoughts of DPSO for TSP is like: particle $X_i(t) = \{x_{i1}, x_{i2}, \cdots, x_{iD}\}$ stands for a traveling routine. Start from city x_{i1}, go through x_{i2}, x_{i3}...,reach x_{iD}. In this kind of DPSO, velocity are defined as a swap table. That's to say, (π_{ai}, π_{bi}) means to swap city π_{ai} and π_{bi} , $v_{ij} = ((\pi_{ai1}, \pi_{bi1}), ..., (\pi_{ain}, \pi_{bin}))$, for example. For the velocity above, $|v_{ij}| = n$, vector length equals to the number of swap items. Therefore, velocity here contains no repeated item.

3.3 Proposed DPSO for WSN

Based on those existed DPSOs, this paper proposed an improved discrete strategy aimed at WSN coverage problem.

For PSO basic updating formulas (4)(5), there are elements like: position information (such as X, P), velocity information (V), weighting coefficients ($\omega, c * rand()$). In this discrete algorithm, the information are defined as follows:

Position Information: A multi-dimension vector that contains no repeated element, it stands for the spatial situation of a particle and is a sequential list.

Velocity Information: A multi-dimension vector in which elements may repeat, and is the difference of two positions.

Weighting Coefficient: A float number in [0.0, 1.0] that shows the weight of a certain element among all the elements.

Updating method for velocity and position are redefined as:

$$V_i^{(t+1)} = W(\omega) \otimes V_i^{(t)} \circ W(c_1 r_1) \otimes (P_{id} \ominus X_i^{(t)}) \circ W(c_2 r_2) \otimes (P_{gd} \ominus X_i^{(t)}) \quad (9)$$

$$X^{(t+1)} = X^{(t)} \oplus V^{(t)} = \{x_1^{(t)} + ...x_N^{(t)}\} \oplus \{v_1^{(t)}, ..., v_N^{(t)}\} \quad (10)$$

where W is a normalizing function to produce weighting coefficient. Assume there are k factors in all ($k = 3$ in (9)) for a formula, then W is built as:

$$W(factor_i) = \frac{factor_i}{\sum\limits_{i=0}^{k} factor_i} \quad (11)$$

Operators are defined as follows:

$\otimes$ ***Multiplication operator for coefficient and velocity:*** The result is a vector containing **null** values. Retain a certain number (round of $coefficient \times |vector|$) of dimensions using some strategy (randomly, for example) and set other dimensions to null value and composes a resultant vector. Indexes of retained items should be unique among all the vectors to be added.

$\circ$ ***Addition operator for two velocities:*** Merge to a new vector according to the item indexes. For example, $\{v_{i1}, v_{i2}, \mathbf{null}, v_{i4}\} \circ \{\mathbf{null}, \mathbf{null}, v_{i3}, \mathbf{null}\} = \{v_{i1}, v_{i2}, v_{i3}, v_{i4}\}$. According to (11) and multiple method above, all the sub-vectors will be merged to exact one complete vector.

$\ominus$ ***Subtraction operator for two positions:*** This operation results in a velocity vector. Define operate mode for each dimension as follow:

$$p_d \ominus x_d = \begin{cases} p_d, & if \quad rand() \geq \alpha \\ x_d + \gamma(p_d - x_d), & if \quad \alpha > rand() > \beta \\ x_d, & otherwise \end{cases} \quad (12)$$

where $0 < \gamma < 1, \beta \leq \alpha < 1, rand() \in [0, 1]$, and (β, α) is the probability interval of interference factor, which is optional.

$\oplus$***Addition operator for a velocity and a position:*** This will produce a new position. Here, assume that velocity and position vectors are of the same dimension. Operation for each dimension are defined as:

$$x_d \oplus v_d = \begin{cases} v_d, & if \quad rand() \geq \alpha' \\ x_d + \gamma'(v_d - x_d), & if \quad \alpha' > rand() > \beta' \\ x_d, & otherwise \end{cases} \quad (13)$$

$$THEN \quad do \quad x_i \leftarrow x_d \quad where \quad x_i == x_d \oplus v_d$$

where $0 < \gamma' < 1, \beta' \leq \alpha' < 1, rand() \in [0, 1]$, and (β', α') is the probability interval of interference factor which is optional. This operation has two steps: calculate $x_d \oplus v_d$ and swap the result with x_d inside the vector.

4 Experiment Results and Analyses

4.1 Deployment Planning

While implementing PSO algorithm, every particle is a candidate solution maintaining the coordinates of all the nodes, i.e., a deployment layout. Thus, for N nodes in the two-dimensional space, every particle is a $2N$-dimensional vector searching those discrete vertexes in the space, i.e., the grid points. For particle i, $X_i = \{x_{i1}, x_{i2}, ..., x_{i2N}\}$.

CPs are chosen in the space as Fig.1, and N nodes are placed randomly. Appropriate decision of N can be different, but it should usually be more than needed to face unexpected conditions. Algorithm 1 and 2 deal with this situation. Use PSO process (algorithm 1) for elementary optimization. LocalBest version of PSO using ring topology [11] is adopted in this process. In particular, this is a constraint problem that the position of a node should stay in a certain scale and should have neighbor nodes near around which it can deliver message to. For a node n_i, define NC_i as the set of neighbor nodes which are inside the communication range of n_i.

$$NC_i = \{n_j | distance(n_j, n_i) \leq R_c\} \neq \emptyset \tag{14}$$

Then the position update formula (5) can be strengthened as update-method 2.

```
UPDATE-METHOD 2
$tempP_i^{(t+1)} = P_i^{(t)} + V_i^{(t+1)} = \{x_i^{(t+1)}, y_i^{(t+1)}, z_i^{(t+1)}, ...\}$
if $x_i^{(t+1)}$ > threshold
    then $x_i^{(t+1)}$=threshold
judge $y_i^{(t+1)}, z_i^{(t+1)}$...
obtain $NC_i^{(t+1)}$
if $NC_i^{(t+1)}$ ≠null
    then $P_i^{(t+1)} = tempP_i^{(t+1)}$
    else $P_i^{(t+1)} = P_i^{(t)}$
```

Result of algorithm 1 is shown in Fig.3. This algorithm is aimed at single-cover condition, and some CPs in this region are covered more than once. When multi-coverage is not required, nodes are said to be of no coverage benefit where CPs inside its coverage range are all covered by other nodes already. These nodes will be chosen as dormant

```
Algorithm 1: Pso Process
1  Initialize $D$ randomly particles, $P_1^{(0)}, \cdots, P_D^{(0)}$
2  for i=1 to D
3          Set $Pbests_i^{(0)}=P_i^{(0)}$
4          Judge quality of $P_i$ and set $Lbests^{(0)}$
5  for i=1 to D
6          Update $P_i$ value by (4) and Update-Method2
7          Judge quality of $P_i$
8          Update Pbests and Lbests
9  if a terminal condition is met
10         then go to step 5
11         else go back to step 12
12 Stop and output the best solution
```

nodes and adapted to sleep mode. Algorithm 2 using a traverse accomplished this job, reducing the number of working nodes to a smaller quantity.

Object deployment layout is like Fig.4 after the optimization, which reached a high coverage rate with a smaller number of nodes for single-cover case, and the connectivity of this network can then be achieved overtly. Flexibility is a significant advantage of this deployment method, since some extra nodes will be awaiting inside the region, a new layout will comes up quickly without external aid once environment changes occur or working nodes get problem. In such case the coverage rate $C \geq 98\%$.

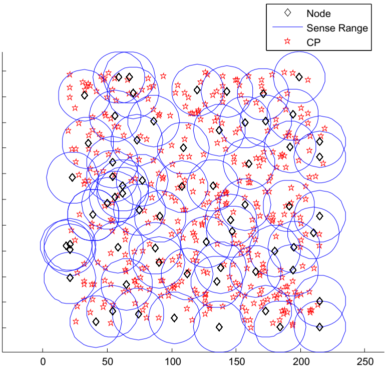


Fig. 3. Elementary Coverage Result

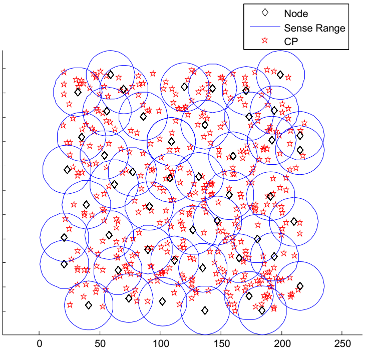


Fig. 4. Adapt Dormant Nodes

ALGORITHM 2: TRAVERSE METHOD

```
1  Nodes are represented as ${x_1, x_2, ..., x_N}$
2  **for** $i = 1$ to $N$
3        $j = i$
4        Mark all the unmarked CPs in the sense range of $x_j$,
   the amount is denoted as $n$
5        **if** $n = 0$
6              **then** pick $x_j$ into dormant set
7        Let $j = (j + 1)\%N$, **if** $j \neq i$
8              **then** go back to step 4
9              **else** go to step 10
10       Get $plan_i$
11 $plan = min\{plan_1, ..., plan_N\}$
```

4.2 Movement Control

Geographical positions of nodes are stochastically initialized and will be changed to specific ones. Since movement consumes energy and reduces life cycle, a shortest moving plan is expected. Assume that every two vertexes in the region can be connected with one straight path (no obstacle between).

Improved DPSO proposed in this paper is the solver. P is the initial position set and Q is the object set, which are taken as sequences. Purpose of this algorithm is to realign P to $P^{'} = \{p^{'}_1, p^{'}_2, ..., p^{'}_N\}$, as to let $p^{'}_i$ pair with q_i, making up the moving plan. This plan are illustrated in Fig.5.

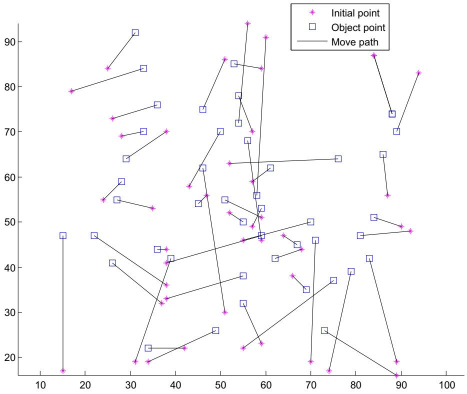


Fig. 5. Moving Plan Sketch Generated by DPSO

Parameters in formulas (9)(12)(13) are set as following: $\omega = 0.6, c_1 = 2.2, c_2 = 1.4, \alpha = 0.6, \beta = 0.4, \gamma = 0.6, \alpha' = 0.7, \beta' = 0.7, \gamma' = 0.6$.

Besides PSO, Ant Colony Optimization (ACO) also has dominant performance solving discrete problems[17]. Comparisons are made between ACO with MMAS strategy and DPSO proposed in this paper. Fig.6 shows when area scope is 200 × 200, performance difference with different node amount. Solid line indicates DPSO performance while dotted line is for ACO. Fig.7 represents the coverage quality for different object area scope with a certain amount of nodes ($N = 45$). Figures stand for the average result of decades runs of the algorithms.

As can be seen, discrete strategy put forward in this paper performs dominantly within a certain range. A lower total distance in the figures claims a better performance. While the node amount is within a threshold(around 100), a lower energy is obtained, and so is the result when the ROI scale is less than 450. This signifies that when implemented in a relatively small scale (of node amount and scale), DPSO proposed in this paper can provide a faster and easier way to a solution. What should be noticed is about the parameters adopted above. Adjustment to those parameters may be needed when situations changes, such as different sensing range and irregular shape of ROI. In such situations, actual concrete data of algorithm may also be different.

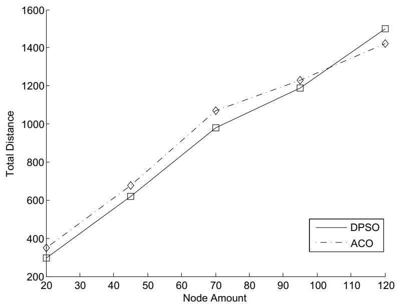


Fig. 6. Performance Comparison with Different Node Amount

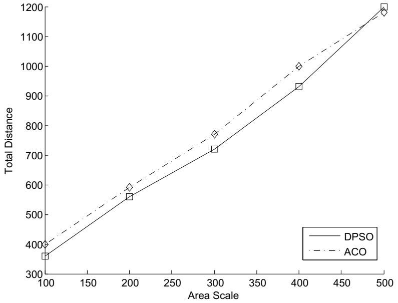


Fig. 7. Performance Comparison with Different Area Scope

5 Conclusion

Coverage control for WSN includes many aspects. Besides coverage rate and connectivity, multi-dimensional environments in real world with obstacles and change still need more considerations. This paper mainly discusses application of PSO to two-dimension coverage problem and improved traditional DPSO based on characteristics of WSN coverage control. Results illustrate that in movement control, the improved DPSO which is easy to handle, is of high performance which is no less than traditional discrete problem solver. Successive research in multi-dimension covering problem is to be conducted with further applications of improved DPSO. Multi-objective problems on network uniformity and life-cycling are also crucial points following up.

Acknowledgement. This paper is supported by Provincial Key Laboratory for Computer Information Processing Technology, Soochow University (KJS1223), Suzhou, China and NSFC (Grant No.61170164).

References

1. Ghosh, A., Das, S.: Coverage and Connectivity Issues In Wireless Sensor Networks: A Survey. Pervasive and Mobile Computing 4, 303–334 (2008)
2. Younis, M., Akkaya, K.: Strategies and Techniques For Node Placement in Wireless Sensor Networks: A survey. Ad Hoc Networks 6, 621–655 (2008)
3. Mateska, A., Gavrilovska, L.: Wsn coverage and connectivity improvement utilizing sensors mobility. European Wireless, pp. 686–693 (2011)
4. Zou, Y., Chakrabarty, K.: Sensor Deployment and Target Localization Based on Virtual Forces. In: International Conference on Computer Communications (INFOCOM), NC, USA, pp. 1293–1303 (2003)
5. Ma, M., Yang, Y.: Adaptive Triangular Deployment Algorithm For Unattended Mobile Sensor Networks. IEEE Transactions on Computers 56, 946–958 (2007)
6. Wei, L., Li, C.: Ant Based Approach to The Optimal Deployment in Wireless Sensor Networks. Journal on Communications 30, 25–33 (2009)
7. Wang, X., Wang, S., Ma, J.J.: An Improved Co-Evolutionary Particle Swarm Optimization for Wireless Sensor Networks with Dynamic Deployment. Sensors 7, 354–370 (2007)
8. Li, J., Li, K., Zhu, W.: Improving Sensing Coverage of Wireless Sensor Networks by Employing Mmobile Robots. In: Proceedings of the International Conference on Robotics and Biomimetics (ROBIO), pp. 899–903 (2007)
9. Heo, N., Varshney, P.K.: A distributed self spreading algorithm for mobile wireless sensor networks. In: IEEE Conference on Wireless Communications and Networking, March 16-20, vol. 3, pp. 1597–1602 (2003)
10. Chakrabarty, K., Iyengar, S.S., Qi, H., Cho, E.: Grid coverage for Surveillance and Target Location in Distributed Sensor Networks. IEEE Transactions on Computers 51, 1148–1153 (2002)
11. del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.-C., Harley, R.G.: Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. IEEE Transactions on Evolutionary Computation 12(2), 171–196 (2008)
12. Kennedy, J., Eberhart, R.: A Discrete Binary Version of The Particle Swarm Algorithm. In: Computational Cybernetics and Simulation (ICSMC), vol. 5, pp. 4104–4108 (1997)
13. Parsopoulos, K., Vrahatis, M.: Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization. Natural Computing 1, 235–306 (2002)
14. Clerc, M.: Discrete Particle Swarm Optimization Illustrated by The Traveling Salesman Problem. In: Onwubolu, G.C., Babu, B.V. (eds.) New Optimization Techniques in Engineering. STUDFUZZ, vol. 141, pp. 219–239. Springer, Heidelberg (2004)
15. Wang, K.P., Huang, L., Zhou, C.G., Pang, W.: Particle Swarm Optimization for Traveling Salesman Problem. In: International Conference on Machine Learning and Cybernetics, vol. 3, pp. 1583–1585 (2003)
16. Shi, X.H., Lianga, Y.C., Leeb, H.P., Lub, C., Wanga, Q.X.: Particle Swarm Optimization-Based Algorithms for TSP And Generalized TSP. Information Processing Letters 103, 169–176 (2007)
17. Dorigo, M., Stutzle, T.: Ant Colony Optimization: Overview and Recent Advances. International Series in Operations Research and Management Science, Handbook of Metaheuristics, IRIDIA/2009-013, pp. 227–263 (2009)

An Improved Energy-Aware Cluster Heads Selection Method for Wireless Sensor Networks Based on K-means and Binary Particle Swarm Optimization

Qianqian Pan[1,2], Qingjian Ni[2,3,4★], Huimin Du[3], Yiyun Yao[3], and Qing Lv[3]

[1] School of Information Science and Engineering,
Southeast University, Nanjing, China
[2] Laboratory of Military Network Technology, PLA University
of Science and Technology, Nanjing, China
[3] College of Software Engineering, Southeast University,
Nanjing, China
[4] School of Computer Science and Engineering,
Southeast University, Nanjing, China
nqj@seu.edu.cn

Abstract. The limited and non-replenishable energy supply is the main character of Wireless Sensor Networks (WSNs). Hence, maximizing the lifetime of WSNs becomes a critical issue in sensor networks. Clustering is one of the most effective means to extend the lifetime of the whole network. In this paper, an energy-aware cluster heads selection method, based on binary particle swarm optimization (BPSO) and K-means, is presented to prolong the network lifetime. We apply the BPSO and make it suitable for this issue. The selection criteria of the objective cost function are based on minimizing the intra-cluster distance as well as the distance between cluster heads and base station, and optimizing the energy consumption of the whole network. In addition, the sensor nodes are divided into several clusters based on K-means algorithm at the beginning, which can reduce the complexity of the whole algorithm. The performance of our technique is compared with the well-known cluster-based sensor network protocols, LEACH-C and PSO-C respectively. The simulation results demonstrate that our proposed work can achieve better network lifetime over its comparatives.

Keywords: Wireless Sensor Networks, Binary Particle Swarm Optimization, Energy-aware Cluster, K-means Algorithm.

1 Introduction

Wireless sensor network is a kind of wireless network composed of a large number of sensor nodes [1][2]. These tiny nodes are usually scattered in a sensor field

[★] Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 125–134, 2014.
© Springer International Publishing Switzerland 2014

and each of these scattered sensors is equipped with a capability to collect data and route data back to the sink, base station (BS). Recent advancement in wireless communication and electronics enable the development of the WSNs. The network has a wide range of applications such as health, military and home. However, sensor node, usually powered by batteries, is limited in energy supply. Therefore, energy efficiency should be considered as the critical design objective. Prolonging lifetime of the WSNs becomes a key issue.

Clustering is one of the most popular methods to prolong the lifetime of the WSNs. One of the well-known clustering protocols is called LEACH [3], which uses randomized rotation of cluster heads to evenly distribute the energy load among the sensors in the network. LEACH-C (LEACH Centralized) [4] is the extension to LEACH. In LEACH-C, BS finds the optimal cluster heads among sensor nodes whose energy are above average, using the simulated annealing algorithm [5].

Particle swarm optimization (PSO) [6][7] is a popular optimization technique, simulating the social behavior of a flock of birds flying to the food. PSO algorithm is applied to find cluster heads and produces better results [8]. A popular clustering algorithm based on PSO is PSO-C [9], which selects cluster heads considering both energy available to nodes and distances between the nodes and their cluster heads.

In this paper, we develop an energy-aware cluster heads selection method based on binary particle swarm optimization (BPSO) [10] and K-means [11] (BPSO-K). Our proposed method selects the high-energy nodes as the cluster heads and evenly distributes the energy load among nodes in the network. The main idea of our protocol is selecting cluster heads that can minimize the intra-cluster distance as well as the distance between cluster heads and BS, and optimize the energy consumption of the whole network. K-means algorithm is utilized at the beginning to divide the nodes into several initial clusters. The rest of this paper is organized as follows: the network and energy models are described in section II. The detailed description of our proposed energy-aware cluster heads selection method for WSNs based on K-means and BPSO is outlined in section III. In section IV, we discuss the simulation study of the proposed protocol. Finally, the concluding remarks appear in section V.

2 The System Model

2.1 Network Model

We assume the network model similar to those used in paper [4] and [9], with the following properties.

1. Sensor nodes are scattered in a field at random and all nodes are static.
2. All sensor nodes are energy constrained, generally powered by batteries.
3. A BS is fixed inside or outside of the sensor filed.
4. Each sensor node has capabilities of processing data and sending data to the BS.
5. Each node can compute its own location and energy level, and send them to the BS.

2.2 Energy Model

The energy model of our protocol is based on the classical model used in paper [3]. The radio hardware energy dissipation of this model is that the transmitter dissipates energy to run the radio electronics and power amplifier, and the receiver dissipates energy to run the radio electronics. Both free space and multipath fading channel model are used according to the distance between transmitter and receiver. Free space model with d^2 is used, if the distance is less than a threshold d_0. Otherwise, the multipath model with d^4 is applied. Thus, in order to transmit an l-bit message over a distance d, the radio energy extended is given in the equation (1).

$$E_{\mathrm{Tx}}(l,d) = \begin{cases} lE_{\mathrm{elec}} + l\epsilon_{\mathrm{fs}}d^2, & \text{if } d < d_0 \\ lE_{\mathrm{elec}} + l\epsilon_{\mathrm{mp}}d^4, & \text{if } d \geq d_0 \end{cases} . \tag{1}$$

And to receive an l-bit message, the energy expended by the radio is given as equation (2).

$$E_{\mathrm{Rx}}(l) = lE_{\mathrm{elec}}. \tag{2}$$

where E_{elec} denotes the electronics energy, depending on the energy dissipated per bit to run the transmitter or the receiver. The amplifier energy $\epsilon_{\mathrm{fs}}d^2$ and $\epsilon_{\mathrm{mp}}d^4$ depend on the distance between transmitter and receiver.

3 Method Description

3.1 Binary Particle Swarm Optimization

PSO [6][12] is a simple, effective, and computationally efficient optimization algorithm for continuous optimization [14][15]. Binary PSO (BPSO) [16] is an extension of PSO based on the binary coding scheme, proposed by Kennedy and Eberhart. BPSO consists of a swarm of S particles. An individual possible solution of a problem is presented by a D-dimensional particle. A particle i has a coordinates x_{id} and a velocity v_{id} in the dth dimension, $1 \leq i \leq S$ and $1 \leq d \leq D$. The velocity is defined as changes of probabilities that decide bits of coordinate in one state or the other. Thus, each dimension of a particle moves to a state restricted to 0 or 1 depending on velocity. Each bit v_{id} of velocity represents the probability of bit x_{id} taking value 1. Velocity of a particle is determined by equation (3).

$$v_{id}(t+1) = v_{id}(t) + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{\mathrm{g}d} - x_{id}). \tag{3}$$

where c_1 and c_2 are positive numbers, r_1 and r_2 are two random numbers between 0 and 1 with uniform distribution, and p_{id}, p_{gd} denote particle's and global best position respectively.

Since v_{id} is a probability, it must be constrained in the interval of [0,1]. Sigmoid function is used to normalization velocity v_{id} based on equation (4).

$$s(v_{id}) = \frac{1}{1+e^{-v_{id}}}. \tag{4}$$

where $s(v_{id})$ denotes the velocity after normalization.

The position of a particle is defined as equation (5).

$$x_{id} = \begin{cases} 1, & \text{if rand()} \le s(v_{id}) \\ 0, & \text{otherwise} \end{cases} . \tag{5}$$

3.2 Proposed Method of Initialization Using K-means

In the process of establishing clusters, the nodes of location proximity are easier to be assigned to a cluster, mainly because of lower energy for transmission among location closer nodes. Hence, we propose a method of initialization to divide sensor nodes into several initial clusters [13] according to location of nodes.

K-means [17] as a classical clustering algorithm can get different K groups which are described by their centroid of nodes. Besides, the number of clusters is determined by user. The algorithm of initialization is given as algorithm 1 in detail.

Algorithm 1. Initialization using K-means

```
function K-MEANS($Array, K, N$)
    for $j = 1 \to K$ do
        $C[j] \leftarrow \text{random}(1, N)$
    end for
    repeat
        for $j = 0 \to K$ do
            $Q[j] = \emptyset$
        end for
        for $i = 1 \to N$ do
            MinDist $\leftarrow \infty$
            for $j = 1 \to K$ do
                if $Distance(Array[i], C[j]) <$ MinDist then
                    MinDist $\leftarrow Distance(Array[i], C[j])$;
                    MinNode $\leftarrow j$
                end if
            end for
            $Q[\text{MinNode}] \leftarrow Q[\text{MinNode}] + i$
        end for
        $C \leftarrow \text{Updata}(Q)$
    until $C[j]$ not changed for $j = 1 \to K$
    return $Q$
end function
```

We divide sensor nodes into K initial clusters based on K-means algorithm before the operation of the network and ensure that $K \le M$, where M denotes the predetermined number of cluster heads. The proposed method of initialization will reduce the complexity of the whole algorithm with slight influence of lifetime.

3.3 Proposed Cluster Heads Selection Method Based on BPSO

Our proposed method is a centralized clustering algorithm to form clusters by dispersing the cluster heads throughout the network. The selection process of our method is working on the BS. The operation is divided into rounds. Each round begins with a set-up phase when the clusters are formed, followed by a steady-state phase when data are transmitted to cluster heads and to the BS [3]. During the set-up phase, each node sends information about current location and energy level to the BS. Based on these statistics, BS computes the average energy of the network. To evenly distribute energy among the whole network, only the nodes with an energy level above the average can be possible cluster heads for current round [18][19]. BS selects M cluster heads using BPSO, where M is the optimizing number of cluster heads for the network.

In the cluster heads selection method based on BPSO [14], a D-dimensional particle represents a selection of cluster heads, where D denotes the number of the possible cluster heads with sufficient energy in current round. The position x_{id} represents the state of the possible cluster head d, where $1 \leq d \leq D$. If x_{id} is restrained to 0, that means the possible cluster head d is not selected as a cluster head in this particle. Otherwise, the value of x_{id} is 1, in this particle, possible cluster head d is chosen as head node. The protocol should ensure each D-bit particle of the flock has exactly M_j bits restrained to 1 and the values of other bits are 0, where $M_j (j = 1, 2, \ldots, K)$ represents the optimizing number of cluster heads for initial cluster j and also ensure $\sum_{j=1}^{K} M_j = M$. To do this, we can adjust the value of x_i with normalized velocity $s(v_i)$. If the number of bits whose value is 1 in a particle is more than M_j, we select the nodes with larger normalized velocity as cluster heads. On the other hand, when the number is less than M_j, we obtain other cluster heads from remaining possible cluster heads according to larger $s(v_i)$.

BS determines M_j cluster heads that suit the cost function best. The main objective of cost function is to optimize the combined influences of distance and energy [20]. Thus, we define the function depending on the following factors.

(1)The longest average distance between nodes and their cluster heads, defined by equation (6).

$$dist_{j1} = \max_{i=1,2,\cdots,M_j} \left\{ \frac{\sum\limits_{k=1}^{C_{ji}} d(CM_{jik}, CH_{ji})}{C_{ji}} \right\}. \tag{6}$$

where M_j and C_{ji} denote the number of initial clusters and nodes in cluster i of the initial cluster j respectively, $d(CM_{jik}, CH_{ji})$ is distance between nodes CM_{jik} and its cluster head CH_{ji}.

(2)The longest distance between cluster heads and BS, given as equation (7).

$$dist_{j2} = \max_{i=1,2,\cdots,M_j} \{d(CH_{ji}, \mathrm{BS})\}. \tag{7}$$

(3)Energy consumption of whole network, represented as $E_{j\mathrm{sum}}$. We can get it based on the system model.

All these three factors should be considered in an integrated manner. However, data for each of them has difference in dimension and magnitude between others. To eliminate such effects, we introduce a normalized function to these three factors, given as equation (8).

$$y = \frac{2}{\pi}\tanh(x). \tag{8}$$

where y is the normalized data of x.

Combining three factors mentioned above, the cost function, represented as f_j, is specified in the equation (9).

$$f_j = \alpha dist_{\mathrm{t}j1} + \beta dist_{\mathrm{t}j2} + \gamma E_{\mathrm{t}j\mathrm{sum}}. \tag{9}$$

where $dist_{\mathrm{t}j1}$, $dist_{\mathrm{t}j2}$ and $E_{\mathrm{t}j\mathrm{sum}}$ are the normalized data of $dist_{j1}$, $dist_{j2}$ and $E_{j\mathrm{sum}}$, α, β, γ are positive factors determining the priority weighting of $dist_{\mathrm{t}j1}$, $dist_{\mathrm{t}j2}$ and $E_{\mathrm{t}j\mathrm{sum}}$, with $\alpha + \beta + \gamma = 1$. In this paper, we assume $dist_{\mathrm{t}j1}$ and $dist_{\mathrm{t}j2}$ have identical impact on the cost function and $E_{\mathrm{t}j\mathrm{sum}}$ has a bit higher influence due to energy often being considered as a main factor of WSNs. Hence, we set $\alpha = 0.3, \beta = 0.3, \gamma = 0.4$.

For a wireless sensor network with N nodes and M predetermined cluster heads, the clusters formed in each round is given as algorithm 2.

Algorithm 2. Cluster heads selection method using BPSO-K

```
$Q \leftarrow$ K-means($Arrary, K, N$)
repeat
    for $j = 1 \rightarrow K$ do
        repeat
            Select $M_j$ cluster heads among candidates of $Q_j$
            for $i = 1 \rightarrow M_j$ do
                $P[j, i] \leftarrow \emptyset$
            end for
            for $i = 1 \rightarrow NumOfNode[j]$ do
                ClosestHead $\leftarrow$ the closest cluster head from node $i$
                $P[j,$ ClosestHead$] \leftarrow P[j,$ ClosestHead$] + i$
            end for
            Calculate cost function
            Update($LocalBest$)
            Update($GlobalBest$)
            Update($v$, $x$)
        until reach the set number of iteration
    end for
until the network is dead
```

After selecting cluster heads of the network and each nodes is decided which cluster it belongs to, the cluster heads act as local control centers to coordinate the data transmissions in their cluster and send the fused data to the BS.

4 Simulations and Analysis

The proposed cluster heads selection method is simulated to evaluate its performance. We define that a wireless sensor node is dead when it runs out of energy and the network is dead at the moment the first node dies. We ran the simulation for 100 nodes in a 200m × 200m network area with both equal and unequal initial energy of nodes. Paraments used in energy model are similar to paper [4], $E_{\text{elec}} = 0.5\text{nJ/bit}, \epsilon_{\text{fs}} = 10\text{pJ/bit/m}^2, \epsilon_{\text{mp}} = 0.0013\text{pJ/bit/m}^4$. The number of clusters is set to be 5 percent [9] of the nodes $M = 5$. The initial clusters divided by K-means is set as $K = 5$. For the paraments of BPSO, we use $S = 30$ particles and $c_1 = c_2 = 2$. In addition, BS is set at location $(0, 200)$ and the data message size is fixed at $l = 4000bit$. The performance of our proposed method is compared with the well-known cluster-based sensor network protocols, LEACH-C and PSO-C.

Fig.1 illustrates the system lifetime, defined by the time of the first node died. It also shows the performance of our proposed method compared with LEACH-C and PSO-C with equal initial energy of nodes. We set the total nodes have 0.5J of initial energy. Fig.2 shows the performance with unequal initial energy of nodes set from 0.3J to 0.7J randomly. The results shown in Fig.1 and Fig.2 are selected randomly from our experiences, which are simulated 20 times for each. We can find that when the first dead node occurs, the LEACH-C and PSO-C do not run exceeding 50 rounds. Whereas the BPSO-K has run about 500 rounds until the first node die.

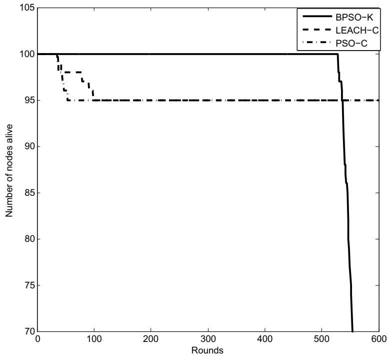


Fig. 1. Number of nodes alive with equal energy

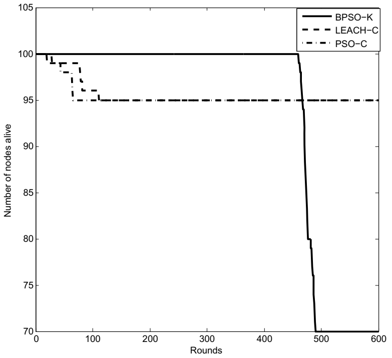


Fig. 2. Number of nodes alive with unequal energy

Clearly our proposed protocol can prolong the lifetime of network significantly compared to LEACH-C and PSO-C. It fairly assigns energy consumption to each node in the field by selecting cluster heads periodically based on BPSO-K, which is helpful to avoid some sensor nodes scattered in the field dying too early. Besides, the cost function consist of both distance and energy also plays a critical role in prolonging the lifetime of the WSNs.

Fig.3 shows the clusters divided by K-means when $K = 5$. Wireless sensor nodes are assigned to different clusters based on their location. The nodes of location proximity are assigned to same cluster.

Fig.4 and 5 show the comparison of BPSO and BPSO-K with equal and unequal initial energy nodes. The results shown in Fig.4 and Fig.5 are selected randomly from our experiences, which are simulated 20 times for each. The simulation results demonstrate K-means can reduce the complexion of the method while having slight impact on the lifetime of the network.

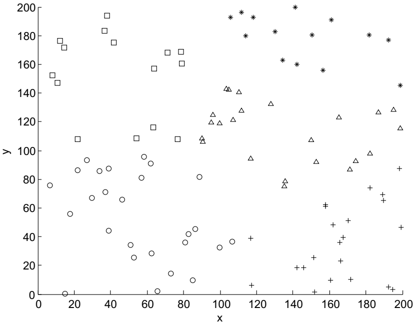


Fig. 3. Clusters divided by K-means

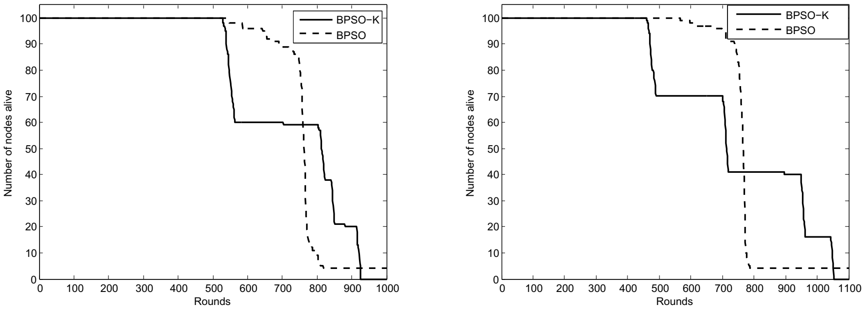


Fig. 4. Number of nodes alive with equal energy

Fig. 5. Number of nodes alive with unequal energy

5 Conclusion

In this paper, we presented an improved energy-aware cluster heads selection method for wireless sensor networks based on K-means and BPSO. We defined a new cost function that takes into account the maximum of the intra-cluster distance as well as the distance between cluster heads and BS, and the minimum of energy consumption. Results from the simulations indicate that the proposed protocol using BPSO and K-means algorithm gives a higher network lifetime compared to LEACH-C and PSO-C. Furthermore, the proposed protocol produces better clustering by evenly allocating the cluster heads throughout the network area. The extension of this work would be a further discussion of the parameters setting to the BPSO-K and to prolong the lifetime of networks consist of mobile nodes.

Acknowledgment. This paper is supported by Laboratory of Military Network Technology, PLA University of Science and Technology (LMNT2012-1), Nanjing, China and NSFC (Grant No.61170164).

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38(4), 393–422 (2002)
2. Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks. Ad Hoc Networks 3(3), 325–349 (2005)
3. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-Efficient Communication Protocol forWireless Microsensor Networks. In: 33rd Hawaii International Conference on System Sciences. IEEE Press, Hawaii (2000)
4. Wendi, B., Heinzelman, A.P.: Chandrakasan, Hari Balakrishnan: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications 1(4), 666–670 (2002)
5. Murata, T., Ishibuchi, H.: Performance Evaluation of Genetic Algorithms for Flowshop Scheduling Problems. In: 1st IEEE Conference on Computational Intelligence, pp. 812–817. IEEE Press, Orlando (1994)
6. del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.-C., Harley, R.G.: Particle Swarm Optimization:Basic Concepts,Variants and Applications in Power Systems. IEEE Transactions on Evolutionary Computation 12(2), 171–196 (2008)
7. Kulkarni, R.V., Venayagamoorthy, G.K.: Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey. IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews 41(2), 262–267 (2011)
8. Bala Krishna, M., Doja, M.N.: Swarm intelligence-based topology maintenance protocol for wireless sensor networks. IET Wireless Sensor Systems 1(4), 181–190 (2011)
9. Abdul Latiff, N.M., Tsimenidis, C.C., Sharif, B.S.: Energy-aware Clustering for Wireless Sensor Networks Using Particle Swarm Optimization. In: 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–5. IEEE Press, Athens (2007)
10. Luh, G.-C., Lin, C.-Y., Lin, Y.-S.: A Binary Particle Swarm Optimization for Continuum Structural Topology Optimization. Applied Soft Computing 11(2), 2833–2844 (2011)
11. Jain, A.K.: Data Clustering: 50 years beyond K-means. Pattern Recognition Letters 31(8), 651–666 (2010)
12. Shi, Y., Eberhart, R.: A Modified Particle Swarm Optimizer. In: The 1998 IEEE International Conference on Computational Intelligence, pp. 577–584. IEEE Press, Anchorage (1998)

13. Wagstaff, K., Cardie, C.: Constrained K-means Clustering with Background Knowledge. In: The Eighteenth International Conference on Machine Learning, pp. 577–584. Morgan Kaufmann Publishers Inc., San Francisco (2001)
14. Chuang, L.-Y., Tsai, S.-W., Yang, C.-H.: Improved Binary Particle Swarm Optimization Using Catfish Effect for Feature Selection. Expert Systems with Applications 38(10), 12699–12707 (2011)
15. Fernández-Martínez, J.L., García-Gonzalo, E.: Stochastic Stability Analysis of the Linear Continuous and Discrete PSO Models. IEEE Transactions on Evolutionary Computation 15(3), 405–423 (2011)
16. Kennedy, J., Eberhart, R.C.: A Discrete Binary Version of the Particle Swarm Algorithm. In: 1997 IEEE International Conference on Systems, Man, and Cybernetics, pp. 4104–4108. IEEE Press, Orlando (1997)
17. Na, S., Xumin, L., Yong, G.: Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm. In: 3rd International Symposium on Intelligent Information Technology and Security Informatics, pp. 63–67. IEEE Press, Jinggangshan (2010)
18. Shi, S., Liu, X., Gu, X.: An Energy-Efficiency Optimized LEACH-C for Wireless Sensor Networks. In: 7th International ICST Conference on Communications and Networking, pp. 487–492. Kun Ming (2012)
19. Karaboga, D., Okdem, S., Ozturk, C.: Cluster Based Wireless Sensor Network Routing Using Artificial Bee Colony Algorithm. Wireless Networks 18(7), 847–860 (2012)
20. Singh, B., Lobiyal, D.K.: A Novel Energy-aware Cluster Head Selection Based on Particle Swarm Optimization for Wireless Sensor Networks. Human-centric Computing and Information Sciences 2(1) (2012)

Comparison of Multi-population PBIL and Adaptive Learning Rate PBIL in Designing Power System Controller

Komla A. Folly

Department of Electrical Engineering, University of Cape Town,
Private bag., Rondebosch 7701, Cape Town, South Africa
Komla.Folly@uct.ac.za

Abstract. Population-Based Incremental Learning (PBIL) is a combination of Genetic Algorithm with competitive learning derived from Artificial Neural Network. It has recently received increasing attention due to its effectiveness, easy implementation and robustness. Despite these strengths, it has been reported recently that PBIL suffers from issues of loss of diversity in the population. To deal with the issue of premature convergence, we propose in this paper a parallel PBIL based on multi-population. In parallel PBIL, two populations are used where both probability vectors (PVs) are initialized to 0.5. The approach is used to design a power system controller for damping low-frequency oscillations. To show the effectiveness of the approach, simulations results are compared with the results obtained using standard PBIL and another diversity increasing PBIL called herein as PBIL with Adapting learning rate (APBIL). It is shown that Parallel PBIL approach performs better than the standard PBIL and is as effective as APBIL.

Keywords: Adaptive learning rate, Low frequency oscillations, Population-based incremental learning, Parallel PBIL.

1 Introduction

Recently, a novel type of Evolutionary Algorithm called Population-Based Incremental Learning (PBIL) [1]-[2] has received increasing attention [3]-[6]. Population-Based Incremental Learning (PBIL) is a combination of Genetic Algorithm with competitive learning derived from Artificial Neural Network. Like other Evolutionary Algorithms such as GAs [7]-[9], Differential Evolution (DE) [10]-[11], and variants such as Particle Swarm Optimization (PSO) [12]-[13], PBIL works with a population of individuals rather than a single individual (e.g., point) [1], [2]. Over successive generations, the population "evolves" toward an optimal solution. PBIL is simpler than Genetic Algorithms GAs, and yet more effective than GAs. In PBIL, the crossover operator of GAs is abstracted away and the role of population is

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 135–145, 2014.
© Springer International Publishing Switzerland 2014

redefined [1]. PBIL works with a probability vector (PV) which controls the random bit strings generated by PBIL and is used to create other individuals through learning. Learning in PBIL consists of using the current probability vector (PV) to create *N* individuals. The best individual is used to update the probability vector, increasing the probability of producing solutions similar to the current best individuals [2], [3]. It has been shown that PBIL outperforms standard GAs approaches on a variety of optimization problems including commonly used benchmark problems [1], [2]**.** PBIL has also been applied for controller design in power systems for small-signal stability improvement. In [4], PBIL based power system stabilizers (PSSs) were compared with GA based PSSs and were found to give better results GA based PSSs**.** In [5]-[6], it was shown that PBIL based PSS performed as effectively as BGA based PSS. However, there are still some issues related to PBIL [14]. It has been reported in [15]-[17] that PBIL suffers from diversity loss making the algorithm to converge to local optima. To cope with this problem, a PBIL with adaptive learning rate strategy was proposed in [16]-[17]. In this paper, a new approach that can improve population diversity in PBIL is presented. The idea of using parallel PBIL (PPBIL) based on multi-population to improve population diversity is explored [18]-[19]. The proposed approach is applied to a power system controller design in a multi-machine power system. The effectiveness of the proposed approach is demonstrated by comparing it to the Adaptive PBIL (APBIL) introduced in [16]-[17] and the standard PBIL (SPBIL). Simulation results show that the parallel PBIL based on multi-population performs better than the standard PBIL and is as effective as APBIL.

2 Overview of the Standard PBIL

Population–based incremental learning (PBIL) is a technique that combines aspects of Genetic Algorithms and simple competitive learning derived from Artificial Neural Networks [1], [2]. PBIL belongs to the family of Estimation of Distribution Algorithms (EDAs), which use the probability (or prototype) vector to generate sample solutions. Unlike GAs which performance depends on crossover operator, PBIL performance depends on the learning process of the probability vector. The probability vector guides the search, which produces the next sample point from which learning takes place. The learning rate determines the speed at which the probability vector is shifted to resemble the best (fittest) solution vector [3]. Initially, the values of the probability vector are set to 0.5 to ensure that the probability of generating 0 or 1 is equal. As the search progresses, these values are moved away from 0.5, towards either 0.0 or 1.0.

Like in GA, mutation is also used in PBIL presented in this paper to maintain diversity. In this paper, the mutation is performed on the probability vector; that is, a forgetting factor is used to relax the probability vector toward a neutral value of 0.5 [3], [4]. The pseudocode for the standard PBIL is shown in Fig. 1, [1]-[4].

If the learning rate is fixed during the run, it cannot provide the flexibility needed to achieve a trade-off between exploration and exploitation. To achieve a trade-off between exploration and exploitation, PBIL with adaptive learning rate strategy presented in [16]-[17] could be used. However, the approach proposed here is to use multi-population PBIL instead of a single population to achieve the same objective as discussed in section 4

```
  Begin
  g:= 0;
  //initialize probability vector
  for $i$:=1 to $l$, do $PV_i^0 = 0.5$;
  endfor;
  while not termination condition do
  generate sample $S(g)$ from (PV(g) , pop.)
  Evaluate samples $S(g)$
  Select best solution $B(g)$
  // update probability vector $PV(g)$ toward best
solution according to (1)
  //mutate $PV(g)$
  Generate a set of new samples using the new
probability vector
  g=g+1
  end while // e.g., g>$G_{max}$
```

Fig. 1. Pseudocode for standard PBIL

3 Overview of Multi-Population PBIL

For the multi-population or Parallel PBIL (PPBIL), two populations are used with two probability vectors (PV_1 and PV_2). Each probability vector is initialized to 0.5 and sampled to generate solutions independently from each other. The PVs are updated independently according to the best solution generated by each. Initially, each probability vector has equal sample solutions. That is, the total population is divided into two populations and a PV is assigned to each population. As the run progresses, the population of the probability vector (PV) that performs better is allowed to increase its share of samples. The sample sizes of the probability vectors are slightly adapted within the range [pop_{min} pop_{max}] = [0.4*pop 0.6*pop] according to their relative performances. The probability that outperforms the other is increased by a constant value $\Delta = LR$*pop, where LR is the learning rate (which was selected as 0.1 in this paper). Fig. 2 shows the pseudocode of PPBIL.

```
Begin
g:= 0;
//initialize probability vector
for $i$:=1 to $l$, do $PV_{i1}^0$ = $PV_{i2}^0$ = 0.5;
endfor;
// initialize the sizes of the probability vectors such that: pop1= pop 2= pop/2
while not termination condition do
generate sample $S_1(g)$ from ($PV_1(g)$ , pop1.)
generate sample $S_2(g)$ from ($PV_2(g)$ , pop2.)
Evaluate samples ($S_1(g), S_2(g)$)
Select best solutions $B_1(g)and\ B_2(g)$
// update probability vectors $PV_1$(g) and $PV_2(g)$ toward bests solution $B_1(g)$and $B_2(g)$ according to (1)
If $f(B_1(g))> f(B_2(g)\ )$
then   $pop_1$= min [($pop_1 + \Delta$)      $pop_{max}$]
If $f(B_1(g))< f(B_2(g)\ )$
then   $pop_1$= max [($pop_1 - \Delta$)      $pop_{min}$]
$pop_2= pop-pop_1$
//mutate $PV_1(g)$ and $PV_2(g)$
g=g+1
end while // e.g., g>$G_{max}$
```

Fig. 2. Pseudocode for parallel PBIL

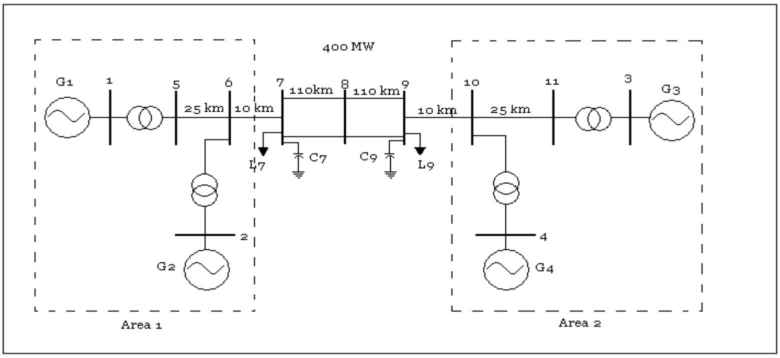


Fig. 3. Power system model

4 Problem Description and Formulation

4.1 Problem Description

The controller to be designed is also known as Power System Stabilizer (PSS) and is needed to damp low frequency oscillations ranging from 0.1 Hz to 2.5 Hz which occur in overly stressed power systems or when power is transmitted over weak transmission lines [20]-[21]. These oscillations are highly undesirable because they can lead to fatigue of machine shafts and limit the ability of the system to transfer the maximum power. It is therefore important that low-frequency oscillations are damped quickly if the security of the system is to be maintained. The power system model used in this paper is the IEEE 2-area system, 4-machine power system as shown in Fig. 3. Each machine is represented by the detailed six order differential equations. The machines are equipped with simple exciter systems. For more information on this system, the reader is referred to [18], [20]. To design the controller, several operating conditions have been considered. However, for simplicity only three operating conditions are shown in Table 1. This Table also shows the eigenvalues and damping ratios in brackets of the three operating conditions.

The system exhibits two local modes one in area 1 and the other in area 2 and one inter-area mode. For the purpose of this study, only the inter-area modes are shown in Table 1 since they are the most difficult to control.

Case 1 is the light load condition, where about 200 MW of real power is transferred from area 1 to area 2. The system is stable for this case as can be seen by the negative value of the real part of the eigenvalue. Case 2 is the nominal condition, under this operating condition, there is a transfer of 400 MW power from area 1 to area 2. The system is unstable for this case, since the real part of the eigenvalue is positive. Case three is the heavy load condition where about 500 MW of power is transferred from area 1 to area 2. This case is also unstable.

4.2 Problem Formulation and Objective Functions

The purpose of the design is to optimize the parameters of the generator excitation controls (i.e., PSSs) simultaneously and in a coordinated and decentralized manner such that adequate damping is provided to the system over a wide range of operating conditions, while keeping the structure of the PSS as simple as possible. The structure of the widely used conventional PSS was adopted here. The PBILs are applied to optimize the parameters of a fixed structure ($\Delta\omega$ input) PSS of the form:

$$K(s) = K_p\left(\frac{T_w s}{1+T_w s}\right)\left(\frac{1+T_1 s}{1+T_2 s}\right)\left(\frac{1+T_3 s}{1+T_4 s}\right). \tag{1}$$

where, K_p is the gain, T_1-T_4 represent suitable time constants. T_w is the washout time constant needed to prevent steady-state offset of the voltage. The value of T_w is not critical for the PSS and has been set to 5sec. Therefore, five parameters are required for the optimization.

Since most of oscillation modes considered in this paper are unstable and dominate the time response of the system, it is expected that by maximizing the minimum damping ratio, a set of system models could be simultaneously stabilized over a wide range of operating conditions [10]-[12]. The following objective function was used to design the PSSs.

$$J = \max\left(\min\left(\zeta_{i,j}\right)\right). \tag{2}$$

where $i = 1, 2 \ldots n$, and $j = 1, 2, \ldots m$

and $\zeta_{i,j} = \dfrac{-\sigma_{i,j}}{\sqrt{\sigma_{i,j}^{\;2} + \omega_{i,j}^{\;2}}}$ is the damping ratio of the i-th eigenvalue in the j-th operating condition. σ_{ij} is the real part of the eigenvalue and the ω_{ij} is the frequency. n denotes the total number eigenvalues and m denotes the number of operating conditions.

Table 1. Selected open-loop operating conditions including eigenvalues and damping ratios

Case	P_e [MW]	Eigenvalue (ζ)
1	200	-0.35±3.92i (0.0889)
2	400	0.0096±3.84 i (-0.0025)
3	500	0.148±3.09i (-0.0478)

4.3 Application of Standard PBIL to Controller Design

The configuration of the standard PBIL is as follows:
Length of chromosome: 15 bits
Trial solutions (population): 10
Generations: 400
Learning rate (LR): 0.1
Mutation (Forgetting factor-FF): 0.005

4.4 Application of APBIL to Controller Design

The configuration of the APBIL is as follows:
Length of chromosome: 15 bits
Trial solutions (population): 10
Generations: 400
Initial Learning rate (LR_0) = 0.0005

Final Learning rate (LR_{max}): 0.2
Mutation (Forgetting factor-FF): 0.005

4.5 Application of PPBIL to Controller Design

The configuration of the APBIL is as follows:
Length of chromosome: 15 bits
Trial solutions (population): 10
Initial population for PV_1: 5
Initial population for PV_2: 5
Generations: 400
Final Learning rate (LR): 0.1
Mutation (Forgetting factor-FF): 0.005

For all the controllers, the parameter domain is as follows:

$$0 \leq K_p \leq 30$$

$$0 \leq T_1, T_3 \leq 1$$

$$0.010 \leq T_2, T_4 \leq 0.3$$

5 Simulation Results

5.1 Convergence Rate

Figs. 4-6 show the convergence rate of SPBIL, APBIL and PPBIL, respectively. It can be seen that APBIL and PPBIL converge to higher fitness values of 0.514 and 0.502, respectively, compared to a value of 0.484 for SPBIL. From the simulation results, it can be seen that APBIL has more diversity in the population at the middle of the run between generation 100 and 200 than PPBIL and SPBIL. This can be attributed to the small value of learning rate, at these generations. Small learning rate increases the exploration of the algorithm and thereby introduces more diversity in the population. Unlike SPBIL which diversity is much more concentrated at the beginning of the run between generation 1 and generation 150, the diversity in PPBIL is somehow spread across all generations. At generations 300 to 400 for example, the SPBIL and APBIL have converged (i.e., almost no diversity). On the other hand, PPBIL still has some diversity. Therefore, it can still explore the search space although at a limited pace. Table 2 shows the comparison between the best, mean and worst, fitness values. It can be seen that on average SPBIL and PPBIL have practically the same fitness. The mean for PPBIL and SPBIL which is approximately 0.434 is higher than the mean of APBIL which is 0.383. The main reason for this is that APBIL has much more spread, with the worst fitness value at 0.09 compared to 0.124 for PPBIL and 0.166 for SPBIL.

In terms of the distance between the best and the worst fitness values, APBIL has the highest distance (0.424), followed by PPBIL (0.378) and then SPBIL (0.318). This suggests that both APBIL and PPBIL have more diversity in their populations than SPBIL. Table 3 shows the number of functions evaluations for each algorithm before the best fitness was found. It can be seen that that SPBIL has the lowest function evaluations (3810) and APBIL has the highest function evaluations (15950). PPBIL is somehow in the middle (10610). In terms of the speed in finding the best fitness value, SPBIL is better and APBIL is the worst. However, the best value found by SPBIL is lower than that found by APBIL and PPBIL. This suggests that although SPBIL converges faster, it converges to local optima, which may not be appropriate.

5.2 Eigenvalue Analysis

Table 4 shows the eigenvalues and damping ratios in brackets of the closed-loop systems with the three controllers. It can be seen that PPBIL and APBIL give better performances (i.e., better damping ratios) than SPBIL. However, PPBIL provides slightly a better damping than APBIL.

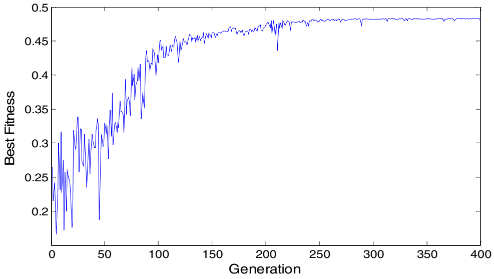


Fig. 4. SPBIL convergence rate

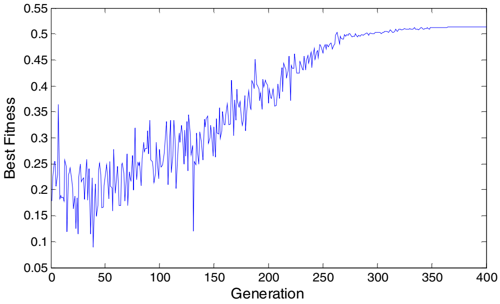


Fig. 5. APBIL convergence rate

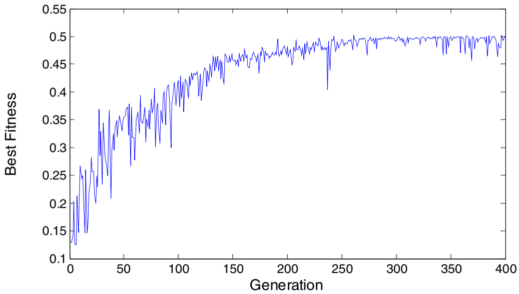


Fig. 6. PPBIL convergence rate

Table 2. Fitness Values

Fitness	SPBIL	APBIL	PPBIL
Best	0.484	0.514	0.502
Mean	0.434	0.383	0.434
Worst	0.166	0.090	0.124

Table 3. Number of Function Evaluation

Controllers	Evaluations
SPBIL	3810
PPBIL	10610
APBIL	15950

Table 4. Closed-system eigenvalues and damping ratio

Case	SPBIL	APBIL	PPBIL
1	-1.13 ± j2.04 (0. 48)	-1.54 ± j2.57 (0.51)	-1.53 ± j2.53 (0.52)
2	-0.778 ± j1.78 (0.44)	-1.26 ± j2.11 (0. 51)	-1.26 ± j2.08 (0. 52)
4	-0.582 ± j1.25 (0.42)	-1.15 ± j1.52 (0.61)	-1.14 ± j1.54 (0.60)

6 Conclusions

By using Parallel PBIL based on multi-population we have been able to increase the diversity in the population. This is important to prevent premature convergence that is inherent to the standard PBIL. The effectiveness of the proposed approach is demonstrated by comparing it to the Adaptive PBIL (APBIL) and the standard PBIL (SPBIL). Simulation results show that the performance of PPBIL in increasing the population diversity is as effective as that of APBIL. Both the PPBIL-PSS and the APBIL-PSS performed better than the standard SPBIL-PSS in terms of improving the damping of the system.

Acknowledgement. This work is based on the research supported in part by the National Research Foundation of South Africa, UID 83977 and UID 85503.

References

1. Baluja, S.: Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Technical Report, CMU-CS-94-163, Carnegie Mellon University (1994)
2. Baluja, S., Caruana, R.: Removing the Genetics from the Standard Genetic Algorithm. Technical Report CMU-CS-95-141, Carnegie Mellon University (1995)
3. Greene, J.R.: Population-Based Incremental Learning as a Simple, Versatile Tool for Engineering Optimization. In: EvCA 1996, Moscow (1996)
4. Folly, K.A.: Design of Power System Stabilizer: A Comparison Between Genetic Algorithms (GAs) and Population-Based Incremental Learning (PBIL). In: Proc. of the IEEE PES 2006 General Meeting, Montreal, Canada (2006)
5. Sheetekela, S., Folly, K.: Power System Controller Design: A Comparison Between Breeder Genetic Algorithm (BGA) and Population-Based Incremental Learning (PBIL). In: Proc. of the Int. Joint Conference on Neural Networks, IJCNN (2010)
6. Folly, K.A., Sheetekela, S.P.: Optimal design of Power System Controller using Breeder Genetic Algorithm. In: Bio-Inspired Computational Algorithms and Their Applications, Intech, pp. 303–316 (2012)
7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley (1989)
8. Davis, L.: Handbook of Genetic Algorithms. International Thomson Computer Press (1996)
9. Yao, J., Kharma, N., Grogono, P.: Bi-objective Multipopulation Genetic Algorithm for Multimodal Function Optimization. IEEE Trans. On Evol. Comput. 14(1), 80–102 (2010)
10. Kennedy, J.F., Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufmann (2001)
11. Mulumba, T., Folly, K.A.: Design and Comparison of Multi-machine Power System Stabilizer base on Evolution Algorithms. In: In Proc. of the 46th International Universities' Power Engineering Conference (UPEC), Soest – Germany, September 5-8 (2011)
12. Abido, A.A.: Particle swarm Optimization for Multimachine Power System Stabilizer Design. IEEE Trans. on Power Syst. 3(3), 1346–1351 (2001)

13. Venayagamoorthy, G.K.: Improving the Performance of Particle Swarm Optimization using Adaptive Critics Designs. In: IEEE Proceedings on Swarm Intelligence Symposium, pp. 393–396 (2005)
14. Gosling, T., Jin, N., Tsang, E.: Population-Based Incremental Learning Versus Genetic Algorithms: Iterated Prisoners Dilemma. Technical Report CSM-40, University of Essex, England (2004)
15. Rastegar, R., Hariri, A., Mazoochi, M.: The Population-Based Incremental Learning Algorithm Converges to Local Optima. Neurocomputing 69(13-15), 1772–1775 (2006)
16. Folly, K.A., Venayagamoorthy, G.K.: Effect of learning rate on the performance of the Population-Based Incremental Learning algorithm. In: Proc. of the International Joint Conf. on Neural Network (IJCNN), Atlanta Georgia, USA (2009)
17. Folly, K.A.: An Improved Population-Based Incremental Learning Algorithm. International Journal of Swarm Intelligence Research (IJSIR) 4(1), 35–61 (2013)
18. Folly, K., Venayagamoorthy, G.: Power System Stabilizer Design using Multi-Population PBIL. In: Proc. of the 2013 IEEE Symposium Series on Computational Intelligence (2013)
19. Yang, S., Yao, X.: Experimental Study on Population-Based Incremental Learning Algorithms for Dynamic Optimization Problems. Soft Computing 9(11), 815–834 (2005)
20. Kundur, P.: Power System Stability and Control, McGraw-Hill, Inc. (1994)
21. Gibbard, M.J.: Application of Power System Stabilizer for Enhancement of Overall System Stability. IEEE Trans. on Power Systems 4(2), 614–626 (1989)

Vibration Adaptive Anomaly Detection of Hydropower Unit in Variable Condition Based on Moving Least Square Response Surface

Xueli An and Luoping Pan

China Institute of Water Resources and Hydropower Research, 100038 Beijing, China
an_xueli@163.com

Abstract. It is difficult to effectively analyze and identify the conditions of hydropower unit, due to its complex operation conditions, frequent start-stop conditions, continual working status switch, less fault samples, single static alarm threshold. Lots of test research shows that active power and working head are key factors which affect the operation conditions of hydropower unit. The health standard condition of unit is determined. An adaptive real-time anomaly detection model of hydropower unit vibration parameters is proposed based on moving least square response surface. In the proposed model, active power and working head are comprehensively considered. This model can adapt variable conditions of hydropower unit. The model is used to real time detect the anomaly of hydropower unit vibration parameters. The results show that this model can effectively evaluate the performance of unit vibration, can more accurately detect the abnormal of unit vibration.

Keywords: hydropower unit, vibration parameter, adaptive anomaly detection, moving least square response surface.

1 Introduction

Hydropower plant is the most favorable power source of power system. It is used to undertake the tasks of peaking, filling valley, frequency modulation, phase modulation, and emergency reserve. Hydropower plant can improve the efficiency of thermal power plants and nuclear power plants, increase the reliability of power grid. It has a significant role in ensuring the safety of the power grid operation and improving the economy of power system [1] and [2]. Due to the complexity of hydropower units' operating conditions, frequent start-stop and working conditions conversion, making the unit easy to malfunction. To ensure units' safe and stable operation, it is needed to mine their condition monitoring data. The data mining can better get the real operating condition of units and early warn the possible abnormalities.

The research of online monitoring and fault diagnosis of hydropower units mainly focuses on the development and integration of condition monitoring system and fault diagnosis methods. The current research achievements don't meet site requirements

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 146–154, 2014.
© Springer International Publishing Switzerland 2014

[1]. The studies for effectively analyzing the mass monitoring data aren't many. The studies for building the anomaly detection model based on online monitoring data are fewer [1, 3]. This makes the vast majority of hydropower plants can only use preventive maintenance strategy, namely scheduled overhaul strategy. This strategy will inevitably lead to the problem for inadequate maintenance or excess maintenance.

Anomaly detection aims to find the relationship between the abnormities of units' condition parameters and their potential failure, to reveal hidden information of abnormal parameters. The field personnel can timely take appropriate measures according to units' abnormal condition information, to restraint the further deterioration of abnormities [4], [5] and [6]. Using this method, faults can be nipped in the bud. The failure rate will be reduced. The security, stability of units operation and economy of maintenance will be increased.

In this paper, based on a long time condition monitoring data of hydropower units, the moving least square response surface is used to build adaptive anomaly detection model. This model considers the factors of active power and working head, which affect the operation condition of hydropower units. The proposed model provides a new way to online assessment of units' running condition.

2 Moving Least Square Response Surface

In a local domain Ω_x of the fitting area Ω, supposing the function values of a calculated function $f(x)$ of N sampling points x_I (I=1, 2, …, N) in the local domain Ω_x are known, $y_I=f(x_I)$. In the local domain, the calculated function $f(x)$ can be approximated as $g(x)\approx f(x)$, the fitting function $g(x)$ [7] can be expressed as:

$$g(x)=\sum_{i=1}^{m}\beta_i(x)\rho_i(x)=\rho^T(x)\beta(x)\cdot \tag{1}$$

where $\beta(x)=(\beta_1(x),\ \beta_2(x),\ \cdots,\beta_m(x))^T$ is calculated coefficient, $\rho(x)=(\rho_1(x),\ \rho_2(x),\ \cdots,\rho_m(x))^T$ is m-dimensional k-order complete polynomial, m is the number of basis function.

The fitting accuracy of $g(x)$ is varietal with the different order of basis function. As a secondary basis function in two-dimensional space $\rho(x)$ is

$$\rho(x)=\begin{pmatrix}1 & x & y & x^2 & xy & y^2\end{pmatrix}. \tag{2}$$

where m=6.

In moving least-squares fitting, the coefficient $\beta(x)$ is determined based on the weighted least squares. This makes the weighted sum of squares of each sampling point's errors minimized for the approximate function $g(x)$ in the neighborhood Ω_x of the point x.

$$\begin{aligned}\Psi &= \sum_{I=1}^{N} w_I(x)[g(x)-f(x_I)]^2 \\ &= \sum_{I=1}^{N} w_I(x)\left[\sum_{i=1}^{m}\beta_i(x)\rho_i(x_I)-f(x_I)\right]^2\end{aligned} \cdot \tag{3}$$

$$\frac{\partial\Psi}{\partial\beta_i(x)} = 2\sum_{I=1}^{N} w_I(x)\left[\sum_{i=1}^{m}\beta_i(x)\rho_i(x_I)-f(x_I)\right]\rho_i(x_I)=0 \cdot \tag{4}$$

where i = 1, 2, ..., m, N is number of sampling points within the neighborhood Ω_x of the point x. The $w_I(x)=w_I(x\text{-}x_I)$ is weighting function at the sampling point x_I. The $w_I(x)$ is greater than zero in a limited area Ω_I around the sample point x_I, beyond the Ω_I are zero. The Ω_I is support domain of weight function, also called impact domain of sampling points x_I. The accuracy of moving least square approximation depends largely on the weight function, which often using Gaussian weighting function in the application, detailed introduction refers to the [7].

3 Adaptive Anomaly Detection Model of Hydropower Unit Vibration

The real-time monitoring and timely warn of pumped storage unit is important for power grid’s stable operation. For the moment pumped storage power units have implemented monitoring system to online collect key parts’ monitoring signals. The measured values of monitoring signals are simply compared with a preset threshold to achieve the alarm, which guide the operation and maintenance of the unit. This method of a static alarm threshold is a single judge. It ignores the unit’s performance differences in different working conditions. When an alarm occurs, the equipment performance of unit may have largely deviated from the design conditions. A situation may occur. That is the unit’s equipment may have seriously deteriorated, but the alarm level of condition monitoring system has not yet been reached. It can be seen that static alarm threshold lacks the thorough study of monitoring signal’s hidden abnormities’ or faults’ information. And it lacks the warning capacity for early potential failures. It is far from insufficient to fully reflect the operational condition of the unit. Meanwhile, with the constantly expanding of hydropower plant capacity and the gradual improvement of monitoring auxiliary systems, the information quantity of unit’s control and monitoring data increase continuously. The operation personnel are often difficult to understand unit’s situation based on such a large amount of data [8]. They can’t timely find unit’s abnormity and fault. This can happen that there are huge amounts of data, but lack of information to guide decision.

In the same or similar operating conditions, when the equipments of pumped storage units are normal, their monitoring parameters should be random fluctuations in the mean nearby. After a long time's running, the unit will gradually deviate normal operation condition, enter the non-normal operation condition. As time went on, unit's deterioration will gradually accelerate, from quantitative change to qualitative change, may lead to serious consequences. Therefore, unit's abnormal operating condition should be paid great attention and closely observed. The reasons for an abnormality occurs should be analyzed. The effective solutions should be searched. Unit equipments' abnormality can be early found by effectively mining the useful implicit information of online monitoring data. This can guide the operator to adjust and control unit, ensure efficient and stable operation of the unit, minimize accidents probability.

A large number of field data analysis show that the main factors affecting hydropower unit's operating condition are active power and working head. In this paper, considering the effect of active power and working head, the health standard three-dimensional surface model $c=f(P, H)$ is built, where c is the unit's condition parameter, P is active power, H is working head. Based on the three-dimensional surface model, the adaptive anomaly detection of hydropower unit is made, concrete steps are as follows:

(1) Analyzing the conditions monitoring data of pumped storage power plant units in different operating condition, determining unit's standard health condition. Selecting the characteristic parameters which can reflect unit's operating condition.

(2) Inputting the unit's characteristic parameters in the health condition into LS-SVM to train, building a three-dimensional surface model $c=f(P, H)$, and validating the model.

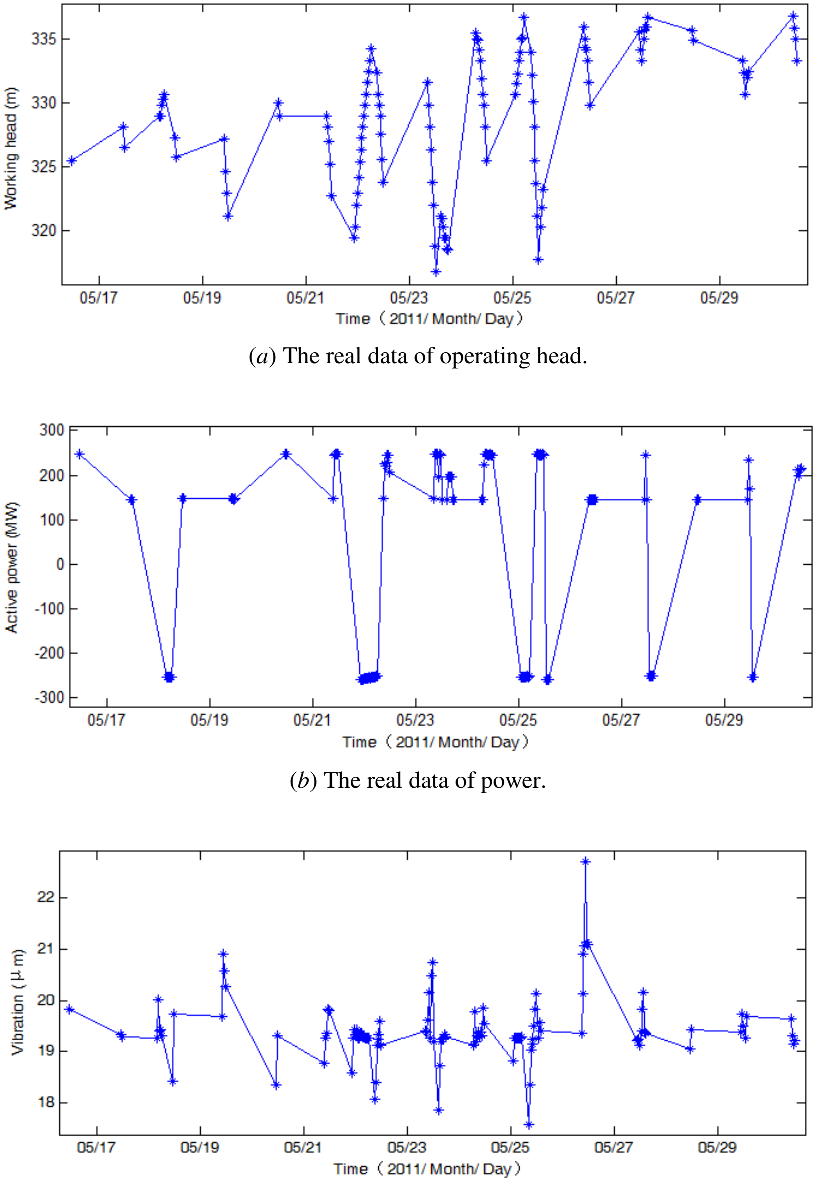
(3) Substituting the real-time condition monitoring data of active power and working head into the trained LS-SVM model to calculate the health standard value $c(t)$ of condition parameters in the current operating conditions. Comparing the current real value $r(t)$ and health standard value $c(t)$, calculate the vibration deviation of the unit in current condition:

$$d_v(t) = \frac{r(t) - c(t)}{c(t)} \times 100\% \cdot \tag{5}$$

where t is run time of pumped storage units. If $d_v(t)$ exceeds a preset threshold, an alert of abnormal vibration can be made. This can promptly find abnormal conditions of the units.

4 Case Study

The real condition monitoring data of a pumped storage power plant unit in September 22, 2008 ~ December 15, 2011 are studied to validate the effectiveness of adaptive anomaly detection model of hydropower unit's vibration parameters in varying conditions. The model is based on moving least squares response surface. Due to the complexity of pumped storage units' operating conditions, frequent starts and stops and working conditions switch, the validity of the proposed model in changing conditions can be better reflected by using pumped storage units' monitoring data. The x-direction horizontal vibration of upper bracket is selected as the study object.



(*a*) The real data of operating head.

(*b*) The real data of power.

(*c*) The real data of upper bracket horizontal vibration in direction-X.

Fig. 1. The condition monitoring data of pumped storage power station unit

Figure 1 shows unit's real data of working head, active power and upper bracket horizontal vibration in direction-X in from May 16, 2011 to May 30, 2011. It can be seen from Figure 1 that unit's active power focused on 250MW for pumping conditions; power is concentrated in 150MW, 200MW and 250MW for generating operation. The working head has strong volatility. The conversion of pumping and generating conditions is frequent. So the changes of upper bracket horizontal vibration in direction-X are complex. The effective information which reflects the true condition can't be obtained only from this Figure. Research shows that active power, working head have an important impact on the unit's vibration parameters. If setting a single static alarm threshold for units, the performance difference, hidden information of abnormality and fault will be greatly neglected. And the unit's real condition can't be truly reflected. Therefore, it is need to build a three-dimensional surface model to detect the abnormality of vibration parameters. This model should be adaptive the changes of working conditions for pumped storage units.

Firstly, determining unit's standard health condition, selecting the characteristic parameter which can reflect unit's operating condition.

The online monitoring data (unit has good condition and without fault) of unit initial operation are adopted to build vibration standard model of unit in healthy condition. The 800 sets online monitoring data from September 22, 2008 to September 18, 2009 are selected. The peak-peak value of x-direction horizontal vibration of upper bracket is selected as the characteristic parameter.

Then, inputting unit's healthy condition parameter into the moving least squares response surface to train, building a three-dimensional surface model $c=f(P, H)$, and validating the model.

To real-timely get a true operating condition of hydropower units, it is need to build a health condition model. Considering the important influence of power and working head on hydropower unit's vibration characteristics, and moving least square response surface has good fitting performance for scattered data, a vibration-power-working head three-dimensional surface model $v=f(P, H)$ of hydropower unit is built. This model is based on moving least squares response surface. Through this model, the mapping relationship in health condition among power (P), working head (H) and vibration parameter (v) can be obtained. For the 800 sets data from September 22, 2008 to September 18, 2009, 600 sets data are selected to establish a health standard model, the remaining 200 sets data as the test samples to validated the model. In order to make the moving least squares response surface model has good performance, the selected 800 sets health standard data should cover possible changes in working head and active power. The 200 test samples' active power and working head are inputted this model, the results can be seen that the health standard values of upper bracket horizontal vibration in direction-X based on moving least squares response surface model is consistent with the measured values. The average relative error is 3.36 %.

Finally, substituting the unit's online monitoring data of power and working head into the trained three-dimensional surface model (moving least squares response surface), calculating the health standard value $c(t)$ of the condition parameter in current condition. Using formula (5) to online assess unit's real-time operating condition, achieving early warn to unit's vibration anomalies.

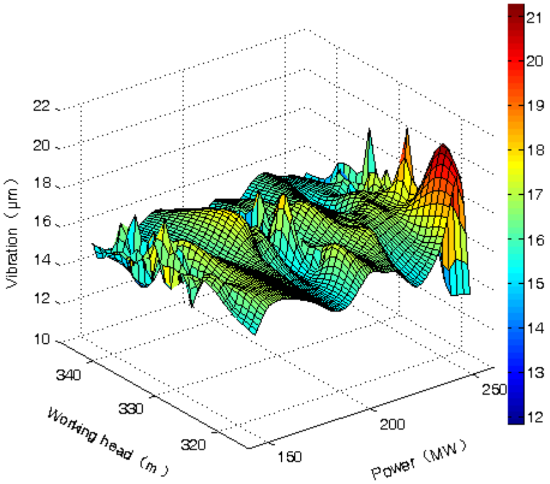


Fig. 2. The three-dimension surface of vibration-power-working head for hydropower unit

Substituting the power and working head of unit's condition monitoring data that after two years (May 12, 2011 ~ December 15, 2011) into unit's health model $v(t)=f(P(t), H(t))$, calculating the health standard values $v(t)$ of condition parameter in the current condition, and comparing the $v(t)$ and the real values $r(t)$. The comparison can be seen in Figure 3. The formula (5) is using to calculate the vibration deviation of the unit in current condition. The results are shown in Figure 4. It can be seen from Figure 4 that after two years of operation, the deterioration of pumped storage unit's component occurs, the unit is gradually deviating its healthy operation condition. If $d_v(t)$ exceeds a preset threshold, an alert of abnormal vibration is made. This can promptly detect the unit's abnormal condition.

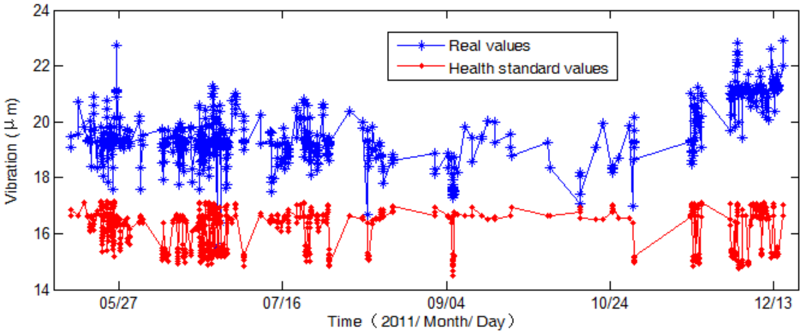


Fig. 3. The fault early warning results of pumped storage unit

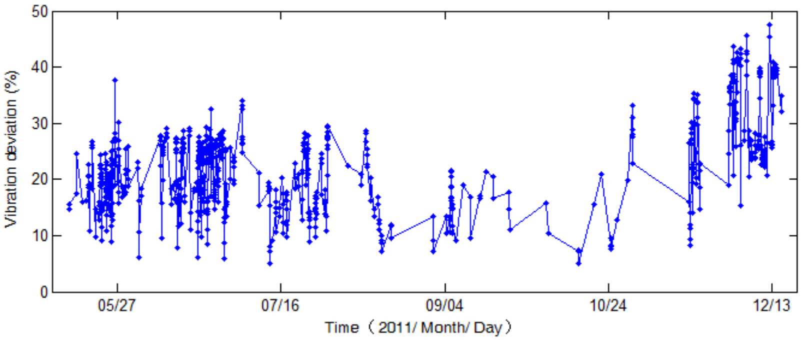


Fig. 4. The fault early warning results of pumped storage unit.

In summary, hydropower unit's vibration anomaly can be found early by using the presented three-dimensional surface anomaly detection model. When making unit's maintenance plan, the unit parts whose condition is abnormal, can be checked purposefully. This can effectively avoid the possibility of forced outages, truly realize the prevention of unit's fault. So field operator of hydropower plant can real-timely and comprehensivly obtain the health condition of unit's key components. The meaningless alarms will be reduced. The repair time will be shorten and the operation time will be increased.

5 Conclusions

Pumped storage unit has complex operating conditions. It has many condition monitoring points, less fault samples. It is difficult to effectively diagnose it's falut. Setting static alarm thresholds will ignore unit's differences of dynamic performance in varying operating conditions. So an adaptive real-time anomaly detection model of hydropower unit vibration parameters is proposed based on three-dimensional surfaces. Firstly, the unit's conditions monitoring data in different operating condition are analyzed. This reveals the key factors of active power and working head, which affect unit's performance. The unit's health standard condition is determined. Then, the characteristics parameters which can reflect of unit's operating condition are selected. The selected parameters of health condition are inputted into LS-SVM to train. Finally, unit's current condition monitoring data are inputted into the three-dimensional surface model to online assess unit's condition ment, achieve early warning of abnormal vibration. The example shows that the proposed model can effectively demonstrate the hidden information of condition monitoring data, real-timely track unit's operating condition, early warn unit's potential failures. The model has are good application prospects.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (grant number 51309258) and the Special Foundation for Excellent Young Scientists of China Institute of Water Re-sources and Hydropower Research (grant number 1421).

References

1. An, X.L., Pan, L.P., Zhang, F.: Condition degradation assessment and nonlinear prediction of hydropower unit. Power System Technology 37(5), 1378–1383 (2013)
2. An, X.L.: Vibration characteristics and fault diagnosis for hydraulic generator units. Thesis for the degree of doctor of engineering, Huazhong University of Science & Technology, Wuhan (2009)
3. Lu, W.G., Dai, Y.P., Gao, F.: A hydroelectric-generator unit faults early warning method based on distribution estimation. Proceedings of the CSEE 25(4), 94–98 (2005)
4. Renders, J.M., Goosens, A., Viron, F.D.: A prototype neural network to perform early warning in nuclear power plant. Fuzzy Sets and Systems 74, 139–151 (1995)
5. Karlsson, C., Larsson, B., Dahlquist, A.: Experiences from designing early warning system to detect abnormal behaviour in medium-sized gas turbines. In: 23rd International Congress on Condition Monitoring and Diagnostic Engineering Management, pp. 117–120. Sunrise Publishing, Hikone (2010)
6. Jardine, A., Lin, D., Banjevic, D.: A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical Systems and Signal Processing 20(7), 1483–1510 (2006)
7. Zhang, Y., Li, G.Y., Zhong, Z.H.: Design optimization on lightweight of full vehicle based on moving least square response surface method. Chinese Journal of Mechanical Engineering 44(11), 192–196 (2008)
8. Yan, J.F., Yu, Z.H., Tian, F.: Dynamic security assessment and early warning system of power system. Proceedings of the CSEE 28(34), 87–93 (2008)

Capacity and Power Optimization for Collaborative Beamforming with Two Relay Clusters

Bingbing Lu[1,2], Ju Liu[1,2], Chao Wang[1], Hongji Xu[1,2], and Qing Wang[1]

[1] School of Information Science and Engineering,
Shandong University, Jinan, 250100, China
juliu@sdu.edu.cn

[2] National Mobile Communications Research Laboratory,
Southeast University, Nanjing, 210096, China
hongjixu@sdu.edu.cn

Abstract. In this paper, we study two approaches to optimize the problems between capacity and power in a three-hop multi-relay network. In the first approach, two beamforming weight vectors are designed to maximize the capacity under the power constraints of relay clusters. While in the second approach, we minimize the power of total relay nodes as well as meet the minimal capacity demand. In both of design schemes, we turn into two beamformer vectors to only one though a series of mathematical manipulation. Then apply genetic algorithm (GA) to obtain the optimal weight value of the nonconvex problems. Simulation results show that our proposed approaches significantly outperform the previous methods conducted.

Keywords: cooperative communication, three-hop, multi-relay, genetic algorithm.

1 Introduction

Exploiting relay nodes to improve information capacity and link reliability has attracted increasing interest recently [1–4]. Many swarm intelligence algorithms [5–8] which can achieve optimal results also quickly development. Recently, the dual-hop relay systems have attracted attentions in the research academia. As the serious signal fading and path loss problems in some specific situation, we consider a three-hop relay system which consists of a transmitter, a receiver and two clusters of relay nodes. The relays at both terminals will form like a multi-input multi-output (MIMO) beamforming system [9, 10]. Some methods are proposed to optimize this problem like in [11], the cooperative relay weight coefficients are optimized by maximizing the destination SNR under the sum-power constraints at the relay clusters.

In this paper, we develop two distributed beamforming approaches in a three-hop AF cooperative communication system. In the first approach, we aim to

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 155–162, 2014.
© Springer International Publishing Switzerland 2014

maximize the information capacity subject to the separate total power constraint. In the second approach, we minimize the total transmit power which maintains the mutual information above a predefined threshold. As illustrated, we turning multi-variables into single one then solve by Genetic Algorithm (GA). Compared with the defect that be easy trapped into local optimal solution in [12], our proposed approaches can obtain the global optimal solution in statistically.

The remainder of the paper is organized as follows. In Section II, we present the system model and the optimization problem. The approaches of two relay weights optimization are presented in section III. Simulation results are provided in Section IV. Finally section V concludes the paper.

2 System Model and Optimization Problem

2.1 System Model

Consider a relay network in Fig. 1. We suppose that there is no direct link between the transmitter and receiver. Each node is equipped with a single antenna, and is subject to the half-duplex constraint. The Rayleigh flat fading channel coefficients between all the nodes are identical independent distributed.

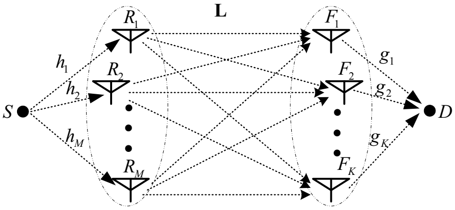


Fig. 1. System model [11]

The transmission information is divided into three parts. In the first part, the source node broadcasts the signal $\sqrt{P_0}s$ to the first relay cluster, where $E\{|s|^2\} = 1$. The signal $\mathbf{x}$ sent by $\{\mathrm{R}_m\}_{m=1}^{M}$ is given as $\mathbf{x} = \sqrt{P_0}\mathbf{Wh}s + \mathbf{Wn}_R$, where $\mathbf{W} = diag([w_1, w_2, \cdots w_M])$ and $\mathbf{n}_R$ with zero mean and variance of σ^2, similar as the noise $\mathbf{n}_F$ at $\{\mathrm{F}_k\}_{k=1}^{K}$ and n_D at the destination.

In the second part, $\{\mathrm{F}_k\}_{k=1}^{K}$ retransmits the received signal $\mathbf{y}$ multiplied by the beamforming matrix $\mathbf{P}$, which can be expressed as $\mathbf{y} = \mathbf{PLx} + \mathbf{Pn}_F$, where $\mathbf{P} = diag([p_1, p_2, \cdots p_K])$. Finally, the signal d received at the destination can be written as

$$d = \mathbf{g}^H\mathbf{y} + n_D = \sqrt{P_0}\mathbf{p}^H\mathbf{GLHw}\mathrm{s}{+}\mathbf{p}^H\mathbf{GLWn}_R{+}\mathbf{p}^H\mathbf{Gn}_F{+}n_D \qquad (1)$$

where $\mathbf{G}=\mathrm{diag}\,(\mathbf{g}), \mathbf{H}=\mathrm{diag}\,(\mathbf{h}), \mathbf{p} = [p_1, p_2, ..., p_K]^H$ and $\mathbf{w} = [w_1, w_2, ..., w_M]^H$.

2.2 Optimization Problem

We want to find the optimal solution from the capacity maximization and relay power minimization. By Shannon's second theorem, the mutual information between the source and destination is given by $I = \frac{1}{2}\log_2(1 + \mathrm{SNR})$. The capacity maximization is equivalent to maximized the receiving SNR at D since I is increasing along with SNR.

The two problems will be optimized with two relay beamformers $\mathbf{w}$ and $\mathbf{p}$, which are the capacity maximization under the relay power constraints and the relay power minimization as well as meet the capacity demand. In the both problems, the transmit power P_R of relay cluster $\{\mathrm{R}_m\}_{m=1}^{M}$ can be calculated as $P_R = \sum\limits_{m=1}^{M} E\left\{|x_m|^2\right\} = \mathbf{w}^H\mathbf{D}_R\mathbf{w}$, where $\mathbf{D}_{\mathrm{R}} = P_0 diag\left(E\{|h_1|^2\}, E\{|h_2|^2\} \cdots E\{|h_M|^2\}\right) + \sigma^2\mathbf{I}$. In the same way, the total transmit power P_F can be obtained as $P_F = \sum\limits_{k=1}^{K} E\left\{|y_k|^2\right\} = \mathbf{p}^H\mathbf{D}_F\mathbf{p}$, where $[\mathbf{D}_F]_{k,k} = \sum\limits_{m=1}^{M} |l_{k,m}|^2[\mathbf{D}_R]_{m,m}|w_m|^2 + \sigma^2$, $k = 1, 2 \cdots, K$.

3 Relay Weights Optimization Based on GA

GA is a metaheuristic optimization algorithm that use survival of the fittest evolutionary scheme to refine a set of solution candidates iteratively [13]. The computing process of GA includes mutation, crossover, selection and inheritance. The algorithm begins with creating an initial population of random individuals which represented as binary strings. Then we compute the objective values of the solution candidates to select the fittest individuals under some constraints. The selected individuals are reproduced by crossover and mutation so as to create the new individuals. The GA with the features of randomness and heuristic search nature has made the algorithm a suitable tool for finding the global optimal solution.

In the following two problems of capacity maximization and relay power minimization, the GA is a basic approach to solve the final optimization problems.

3.1 Capacity Maximization

In this subsection, our goal is to maximal the information capacity subject to the separated power constraint of the two relay clusters. Because of the relationship between I and SNR, the optimal I can be calculated by the optimal solution of SNR, then this optimization problem is equivalent to

$$\begin{aligned} \max_{\mathbf{w},\mathbf{p}} \quad & \frac{P_0}{\sigma^2}\frac{\mathbf{p}^H\mathbf{GLHw}(\mathbf{GLHw})^H\mathbf{p}}{\mathbf{p}^H\left\{\mathbf{GLw}(\mathbf{GLw})^H + \mathbf{gg}^H\right\}\mathbf{p} + 1} \\ \text{s.t.} \quad & \mathbf{w}^H\mathbf{D}_R\mathbf{w} \leqslant P_R^{\max}, \mathbf{p}^H\mathbf{D}_F\mathbf{p} \leqslant P_F^{\max} \end{aligned} \tag{2}$$

Our goal is to obtain the optimal vector **w** and **p** so that the SNR is maximized. We can see that, this problem is not convex, and the two beamforming vectors **w** and **p** depend on each other in problem (2), which make the problem more difficult to solve. We find that the initial problem can be regard as a question with variable **p** when **w** is a constant selected within the feasible region. And then for any available **w**, the maximum achievable SNR [10] can be expressed as

$$SNR_{\max}(\mathbf{w}) = P_F^{\max} P_0 \mathbf{M}^H \mathbf{D}_F^{-1/2} \mathbf{X} \mathbf{D}_F^{-1/2} \mathbf{M} \tag{3}$$

where $\mathbf{M} = \mathbf{GLHw}$, $\mathbf{X} = (\sigma^2 \mathbf{I} + P_F^{\max} \mathbf{D}_F^{-1/2} \mathbf{Q} \mathbf{D}_F^{-1/2})^{-1}$, $\mathbf{Q} = \sigma^2 \mathbf{GLw}(\mathbf{GLw})^H + \sigma^2 \mathbf{gg}^H$. Therefore the optimization problem (2) is simplified into a question with only one variable **w** with the constraint, which can be written as

$$\begin{aligned} &\max_{\mathbf{w}} \ P_F^{\max} P_0 \mathbf{w}^H (\mathbf{GLH})^H \mathbf{D}_F^{-1/2} \mathbf{X} \mathbf{D}_F^{-1/2} \mathbf{GLHw} \\ &s.t. \quad \mathbf{w}^H \mathbf{D}_R \mathbf{w} \leqslant P_R^{\max} \end{aligned} \tag{4}$$

Genetic Algorithm

```
Initial w
Calculate p by (3)
for g=1; g<=G; g=g+1                //G: generation limit
for i=1; i<= N; i=i+1                //N: Number of individuals
  if $\mathbf{w}_i$ don't meet constraints
    then utilize penalty function to generate new $\mathbf{w}_i$
  else
    $q(i) = \frac{\mathbf{w}_i{}^H \mathbf{A} \mathbf{w}_i}{\mathbf{w}_i{}^H \mathbf{B} \mathbf{w}_i + c}$            //$q(i)$:fitness faction of $\mathbf{w}_i$
    if $q(i) > \varepsilon$                   //$\varepsilon$: a predefined constant
      then reproduction $\mathbf{w}_i$
    else
      give up $\mathbf{w}_i$
    end if
  end if
$\mathbf{w}_i$ crossover with $\mathbf{w}_{i+j}$ of probability $p_c$
$\mathbf{w}_i$ mutation of probability $p_m$
end
end
```

It can be shown that, the problem (4) is not convex. GA is often applied as an approach to obtain a statistics global optimal solution for this nonconvex problems. The details process of the algorithm are expressed in **Genetic Algorithm**. The maximum I can be calculated by the obtained SNR. Because of this lengthy process, the computational complexity of GA is relatively high. Compared with using GA to solve the problem in (2) directly, the initial population of solving (4) is produced only by **w**. Thus the coding length of initial population individuals will reduce to half, which leads to a decrease of computation time through abundant crossover, mutation and duplication in every generational population of GA.

3.2 Power Minimization

In this subsection, our goal is to minimize the total transmit power while keeping the capacity at the destination above a certain preconcerted threshold. Similar as the first problem, the optimization of I can be converted to the problem about SNR, which meet the threshold γ. The problem can be expressed as

$$\begin{aligned} \min_{\mathbf{w},\mathbf{p}} \quad & \mathbf{w}^H\mathbf{D}_R\mathbf{w}+\mathbf{p}^H\mathbf{D}_F\mathbf{p} \\ \text{s.t.} \quad & \frac{P_0}{\sigma^2}\frac{\mathbf{p}^H\mathbf{GLHw}(\mathbf{GLHw})^H\mathbf{p}}{\mathbf{p}^{\mathbf{H}}\left\{\mathbf{GLw}(\mathbf{GLw})^{\mathbf{H}}+\mathbf{gg}^H\right\}\mathbf{p}+1}\geqslant\gamma \end{aligned} \tag{5}$$

Similar to the previous subsection, $\mathbf{w}^H\mathbf{D}_R\mathbf{w}$ in the objective function is fixed for a certain $\mathbf{w}$. Under the condition of every fixed $\mathbf{w}$, this question can be simplified as the problem about p. It can be known [10] that this optimization problem will get the global optimal solution when $\mathbf{p}$ is chosen as

$$\mathbf{p}=g(\mathbf{w})=\left(\frac{\gamma\sigma^2}{\mathbf{u}^H\mathbf{D}_F^{-\mathbf{1/2}}(\mathbf{R}-\gamma\mathbf{Q})\mathbf{D}_F^{-\mathbf{1/2}}\mathbf{u}}\right)^{1/2}\mathbf{D}_F^{-1/2}\mathbf{u} \tag{6}$$

where $\mathbf{u}=\mathcal{P}\{\mathbf{D}_F^{-\mathbf{1/2}}(\mathbf{R}-\gamma\mathbf{Q})\mathbf{D}_F^{-1/2}\}$ and the matrix $\mathbf{R}=P_0\mathbf{GLHw}(\mathbf{GLHw})^H$. We put (6) into (5), which turns (5) into a problem with only one variable $\mathbf{w}$. Mathematically, the simplified optimization problem can be expressed as

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathbf{w}^H\mathbf{D}_R\mathbf{w}+\mathrm{g}^H(\mathbf{w})\mathbf{D}_F\mathrm{g}(\mathbf{w}) \\ \text{s.t.} \quad & \frac{\mathbf{w}^H\tilde{\mathbf{A}}\mathbf{w}}{\mathbf{w}^H\tilde{\mathbf{B}}\mathbf{w}+\tilde{c}}\geq\gamma \end{aligned} \tag{7}$$

where $\tilde{\mathbf{A}}=P_0(g(\mathbf{w})^H\mathbf{GLH})^H(g(\mathbf{w})^H\mathbf{GLH})$, $\tilde{\mathbf{B}}=(g(\mathbf{w})^H\mathbf{GL})^H(g(\mathbf{w})^H\mathbf{GL})$ and the notation $\tilde{c}=\sigma^2g(\mathbf{w})^H\mathbf{gg}^Hg(\mathbf{w})+\sigma^2$.

We solve the problem in (7) to obtain the optimal solution using GA. For the implement of GA, the initial population is produced according to the vector $\mathbf{w}$, and the beamforming weight $\mathbf{p}$ can be calculated by $\mathbf{w}$. The fitness function and the constraints in GA are the same as the objective function and the restrict function expressed in (7), respectively. We can obtain the optimal value of vector $\mathbf{w}$, and calculate the corresponding optimal $\mathbf{p}$ and the global minimum of P_R+P_F through GA. By constrast, our proposed method has a lower complexity of computing than solving the problem in (5) with GA directly.

4 Simulation Result

In this section, simulations are designed to assess the performance of the proposed algorithms. Over all the simulation process, all the nodes are with the same noise power level and the transmit power P_0 of the source node is set to be

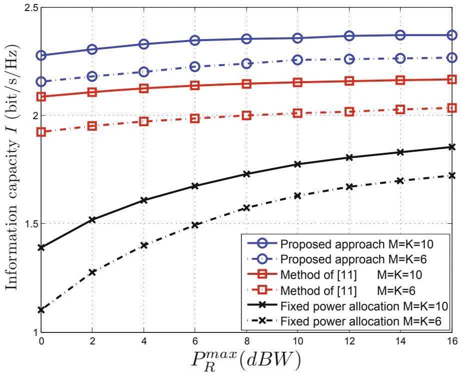


Fig. 2. Information capacity against $P_R^{\max}$ with 6 (dash line) and 10 (solid line) relay nodes, respectively

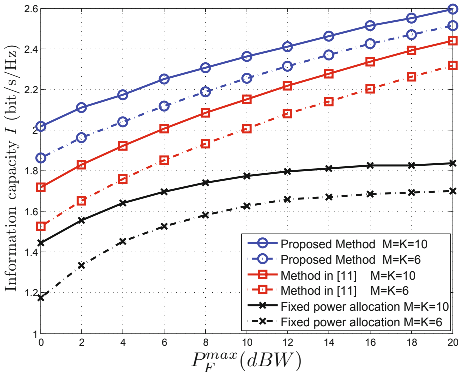


Fig. 3. Information capacity against $P_F^{\max}$ with 6 (dash line) and 10 (solid line) relay nodes, respectively

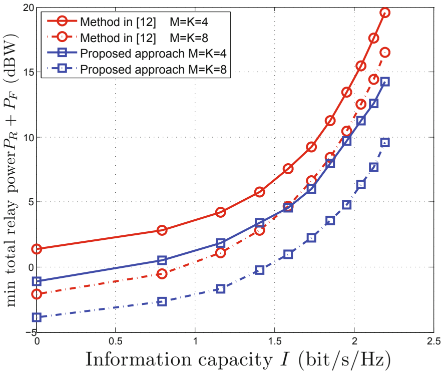


Fig. 4. Minimum total relay power $P_R + P_F$ versus capacity threshold for M=K=4, 6, 8, respectively

10dBW. In the simulation process, only beamforming vector $\mathbf{w}$ is the random variable, and $\mathbf{p}$ is calculated by $\mathbf{w}$. The objective function and constraints in GA are presented in (4) and (7).

Fig. 2 shows the maximum achievable information capacity versus the maximum allowable total transmit power ${P_R}^{\max}$. In Fig. 3, we have shown the maximum achievable mutual information versus the power ${P_F}^{\max}$. We can see that the proposed methods outperform the method presented in [11] and greatly exceed the fixed power allocation solution. As can be seen in these figures, the maximum information capacity tend to be saturated near some reaching threshold with the increasing of ${P_R}^{\max}$, while the maximum capacity almost keep linear increase with the rising of ${P_F}^{\max}$. Therefore, ${P_F}^{\max}$ has more contribution than ${P_R}^{\max}$ after they reaching a certain power.

In Fig. 4, we have plotted the minimum total power of relay nodes $P_R + P_F$ versus the capacity threshold. The total transmit power decrease with the rise of relay numbers due to the diversity gain. It can be observed that the proposed method suffers certain power reduction compared to the method in [12]. This is because the iterative computation between w and p in [12] is easy to fall into local optimal solution, while on the contrary, our proposed method based on GA can avoid this problem effectively.

5 Conclusion

In this paper, we consider the problem of distributed beamforming in a cooperative communication network which consists of a transmitter, a receiver and two relay clusters equipped at the transmitter and receiver side, respectively. The beamforming weight vectors are designed in two different approaches. In the first approach, we aim to obtain the maximum achieved information capacity subject to the power constraints of two relay clusters. In the second approach, we design the beamformer through minimizing the total transmit power of all the relay nodes subject to a constraint which guarantees the mutual information above a predefined threshold. In both of the two approaches, the two random variables can be reduced to only one. For this reason, the computational complexity will be reduced due to the halving of the initial population coding length. Simulation results show that the proposed methods can achieve great improvement compared to the existing solutions.

Acknowledgment. This work was supported by National Natural Science Foundation of China under Grant(61371188), Research Fund for the Doctoral Program of Higher Education under Grant(20130131110029), Natural Science Foundation of Shandong under Grant(ZR2011FM027), Open Research Fund of National Mobile Communications Research Laboratory under Grant (2012D10), China Postdoctoral Science Foundation funded project (2011M501092), Independent Innovation Foundation of Shandong University (2012ZD035), Special Fund for Postdoctoral Innovative Projects of Shandong Province (201103003), Scientific Research Foundation for the Excellent Young and Middle-aged Scientists of Shandong Province (BS2012DX024).

References

1. Laneman, J.N., Tse, D.N.C., Wornell, G.W.: Cooperative diversity in wireless networks: Efficient protocols and outage behavior. IEEE Transactions on Information Theory 50(12), 3062–3080 (2004)
2. Lim, G., Cimini, L.J.: Energy-efficient cooperative beamforming in clustered wireless networks. IEEE Transactions on Wireless Communications 12(3), 1376–1385 (2013)
3. Madan, R., Mehta, N.B., Molisch, A.F., Zhang, J.: Energy-efficient cooperative relaying over fading channels with simple relay selection. IEEE Transactions on Wireless Communications 7(8), 3013–3025 (2008)
4. Yang, Y., Li, Q., Ma, W.-K., Ge, J., Ching, P.C.: Cooperative secure beamforming for AF relay networks with multiple eavesdroppers. IEEE Signal Processing Letters 20(1), 35–38 (2013)
5. Ge, M., Wang, Q.-G., Chiu, M.-S., Lee, T.-H., Hang, C.-C., Teo, K.-H.: An effective technique for batch process optimization with application to crystallization. Chemical Engineering Research and Design 78(1), 99–106 (2000)
6. Precup, R.-E., David, R.-C., Petriu, E.M., Preitl, S., Radac, M.-B.: Novel adaptive gravitational search algorithm for fuzzy controlled servo systems. IEEE Transactions on Industrial Informatics 8(4), 791–800 (2012)
7. Saha, S.K., Ghoshal, S.P., Kar, R., Mandal, D.: Cat swarm optimization algorithm for optimal linear phase FIR filter design. ISA Transactions 52(6), 781–794 (2013)
8. Yazdani, D., Nasiri, B., Azizi, R., Sepas-Moghaddam, A., Meybodi, M.R.: Optimization in dynamic environments utilizing a novel method based on particle swarm optimization. International Journal of Artificial Intelligence 11(A13), 170–192 (2013)
9. Chalise, B.K., Vandendorpe, L.: MIMO relay design for multipoint-to-multipoint communications with imperfect channel state information. IEEE Transactions on Signal Processing 57(7), 2785–2796 (2009)
10. Nassab, V.H., Shahbazpanahi, S., Grami, A., Luo, Z.-Q.: Distributed beamforming for relay networks based on second-order statistics of the channel state information. IEEE Transactions on Signal Processing 56(9), 4306–4316 (2008)
11. Chen, L., Wong, K.-K., Chen, H., Liu, J., Zheng, G.: Optimizing transmitter-receiver collaborative-relay beamforming with perfect CSI. IEEE Communication Letters 15(3), 314–316 (2011)
12. Wang, C., Liu, J., Dong, Z., Xu, H., Ma, S.: Multi-hop Collaborative Relay Beamforming. In: 78th IEEE Vehicular Technology Conference, pp. 1–5. IEEE Press, Las Vegas (2013)
13. Tang, K.S., Man, K.F., Kwong, S., He, Q.: Genetic algorithms and their applications. IEEE Signal Processing Magazine 13(6), 22–37 (1996)

Solving Power Economic Dispatch Problem Subject to DG Uncertainty via Bare-Bones PSO

Yue Jiang, Qi Kang, Lei Wang, and Qidi Wu

Department of Control Science and Engineering,
Tongji University, Shanghai 201804, China
qkang@tongji.edu.cn

Abstract. Distributed generation (DG) is becoming increasingly important in power system. However the of DG will lead risks in power system due to its failure or uncontrollable power outputs which is usually relied on renewable energy. In this work, we solve the economic dispatch (ED) problem by considering controllable and uncontrollable DG in power system. This paper applies the bare-bones particle swarm optimization (BBPSO) method to solve the ED problem. The performance of BBPSO method is evaluated via IEEE 118-bus test system, and would be compared with other methods in terms of convergence performance and solution quality. The results may verify the effectiveness and promising application of the proposed method in solving the ED problem when we are considering both controllable and uncontrollable DG in power system.

Keywords: particle swarm optimization, bare-bones PSO, economic dispatch, uncontrollable.

1 Introduction

With the increasing use of distributed generation (DG), the power system will face many risks of system disruption. Because DG mainly relies on renewable energy such as solar and wind power which is unstable, thus the real power output is changing with weather. There are many types of DG, so the changing rules are different. For instance, the real power output of wind turbines is related to wind speed, which Weibull distribution is generally considered as the optimal probability density function. Therefore, this paper adopts Weibull distribution [1] as a probability distribution of wind speed over time.

Economic dispatch (ED) is an important problem in power system. ED is used in real-time energy management power system to control the production of thermal power stations. Its objective is to minimize the total cost of operating the generators, subject to load and operational constraints [10]. This paper proposes the notion which is when some DG units have a conspicuous rand feature, we can optimize the power output of available generators in order to ensure the safe, effectiveness and efficiency operation of the power system. Its objective is to minimize the total cost of operating the generators and optimize the network voltage profile.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 163–169, 2014.
© Springer International Publishing Switzerland 2014

Traditionally, ED problem can be addressed by a series of mathematical programming methods, such as lambda-iteration method [2], gradient method [3], and dynamical programming [4]. Nevertheless, such deterministic numerical methods cannot effectively solve nonsmooth and nonconvex cost function, moreover, it has larger scale dimension which is too difficult to calculate. In order to effectively solve such problems in power systems, many researchers used a variety of computational intelligence to handle it. For example the genetic algorithm [6], differential evolution algorithm [7], neural network [8], Tabu search algorithm [9] and so on. But the methods mentioned above have many shortcomings, including premature and poor convergence performance.

Usually speaking, people consider ED problem as a static optimization problem discussing in a normal power system. In this paper, however, we want to obtain a real-time power output of controllable generators in a dynamic environment. PSO [10]-[14] has been employed to handle the ED problem for many years. We propose the BBPSO [15]-[19] to solve ED problems.

2 Mathematical Model

The objective in this paper is to minimize the total short-term cost of operating the generators and network loss. It is given by:

$$\text{Minimize } F = \sum_{j=1}^{n} F_j(P_j) + P_{ploss} \tag{1}$$

where $F_j(P_j)$ is the cost function of the jth generator, P_j is the real output of the jth generator, n is the total # of the jth generators in the power system, P_{ploss} is the network loss in the power system.

$F_j(P_j)$ is related to the real power injected into the power system, which is modeled by the function.

$$F_j(P_j) = a_j + b_j P_j + c_j P_j^2 \tag{2}$$

where $a_j, b_j$and c_j are cost coefficient of the jth generator.

This paper adopts the Weibull distribution as probability distribution of wind speed over time .The probability density is given by.

$$f(v) = \frac{k}{c}(\frac{v}{c})^{k-1}\exp\,[-(\frac{v}{c})^{k}] \tag{3}$$

where, v is wind speed, k and c are shape parameter and scale parameter of Weibull distribution respectively. k and c can be calculated by the average wind speed μ and standard deviation .

$$k = (\frac{\sigma}{\mu})^{-1.086} \tag{4}$$

$$c = \frac{\mu}{\Gamma(1+\frac{1}{k})} \tag{5}$$

where Γ is Gamma function.

The relationship between the power output and wind is as followed[21].

$$\mathrm{P} = k_1 v + k_2 \qquad v_{c1} < v \le v_r \tag{6}$$

Where $\mathrm{v_{c1}}$ is cut-in wind speed, $\mathrm{v_r}$ is rated wind speed, $\mathrm{P_r}$ is rated power, $\mathrm{v_{c2}}$ is cut-out wind speed. $\mathrm{k_1}$, $\mathrm{k_2}$ can be calculated as follows:

$$k_1 = \frac{P_r}{v_r - v_{c1}} \tag{7}$$

$$k_2 = -k_1 v_{c1} \tag{8}$$

According to statistics, wind speeds stay between cut-in speed and the rated wind speed most of the time. In this paper DG is to simplify as PV node in the power flow caculation.

In this work, we consider the following constraints.

$$\begin{cases} \sum_{j=1}^{n} P_j = P_D + P_{ploss} & \\ P_j^{min} < P_j < P_j^{man} & (\mathrm{j} = 1, \dots, \mathrm{n}) \\ U_j^{min} \le U_j \le U_j^{man} & (\mathrm{j} = 1, \dots, \mathrm{n}) \\ P_j - P_j^0 \le UR_j & \\ P_j^0 - P_j \le DR_j & \end{cases} \tag{9}$$

Where $\mathrm{P_D}$ is the load demand of the power system; The operating range of all on-line is restricted by ramp rate limits. If power generation increases, then $\mathrm{P_j} - \mathrm{P_j^0} \le \mathrm{UR_j}$. If power generation decreases, then $\mathrm{P_j^0} - \mathrm{P_j} \le \mathrm{DR_j}$. $\mathrm{P_j^0}$ is the previous power output power, $\mathrm{UR_j}$ the up-ramp limit of the jth generator, and $\mathrm{DR_j}$ is down-ramp limit of the jth generator.

3 Bare-Bones PSO Algorithm

PSO commonly used decimal encoding. It was motivated by the social behavior of fish schooling and bird flocks. There are N particles in the PSO algorithm. The search space assumes dimension D. Each particle has three vectors. Its current position $\mathrm{x_i} = [\mathrm{x_i^1}, \mathrm{x_i^2}, \dots, \mathrm{x_i^D}]$, its velocity $\mathrm{v_i} = [\mathrm{v_i^1}, \mathrm{v_i^2}, \dots, \mathrm{v_i^D}]$ and its personal best position pBest.

However, F. van den Bergh proved the update formula which cannot guarantee convergence to the global optimal solution [22]. Therefore, Kennedy proposed the BBPSO[15].

The particles' position in BBPSO is updated by the following equations:

$$\boldsymbol{x}_i^d = \begin{cases} N\left(\frac{\mathrm{pBest}_i^d + g\mathrm{Best}^d}{2} \left|\mathrm{pBest}_i^d - gB\mathrm{est}^d\right|\right) & \mathrm{rand} > 0.5 \\ \mathrm{pBest}_i^d & \mathrm{else} \end{cases} \quad (10)$$

Where, if rand>0.5, current position $\mathrm{x}_\mathrm{i}^\mathrm{d} = \mathrm{pBest}_\mathrm{i}^\mathrm{d}$, otherwise the current position will be derived from a Gaussian distribution with the mean $\frac{\mathrm{pBest}_\mathrm{i}^\mathrm{d}+\mathrm{gBest}^\mathrm{d}}{2}$ and standard deviation $\left|\mathrm{pBest}_\mathrm{i}^\mathrm{d} - \mathrm{gBest}^\mathrm{d}\right|$.

By the formula (10) to update $\mathrm{x}_\mathrm{i}^\mathrm{d}$, it avoids adjusting parameter in the PSO algorithm and easy to implement. And it has the strong global convergence.

4 Solving ED Problems via BBPSO

Step 1. Input data of the power system, and set algorithm parameters. The power system data includes load demand, minimum $\mathrm{P}_\mathrm{j}^\mathrm{min}$ and maximum $\mathrm{P}_\mathrm{j}^\mathrm{man}$ of each generator, minimum $\mathrm{U}_\mathrm{j}^\mathrm{min}$ and maximum $\mathrm{U}_\mathrm{j}^\mathrm{man}$ of each node varies.

Step 2. Set k=1. Initialize the current position of each particle. Each particle's position is represented as matrix X. Set its personal best position equal to be its current position:

Step 3. By Power flow calculation, we can obtain the fitness value for each particle.

Step 4. Update pbest and gbest.

Step 5. Update the position according to (12). If the position cannot fulfill constraints in (9).

Step 6. Examine the termination condition. If it is not met, Set k=k+1 and return to Step 3. Otherwise, end and output results.

5 Simulations and Results

In order to verify the effectiveness of the BBPSO, the power system we used for simulation is the IEEE 118-bus system[25] in this work. This system consists of 118 buses and 54 generator nodes. 1,4,6,8,10,12,15,18,19,24 and 25 node are DGs. The rang of the real power output of GD is 0-50Mw. Node 69 is a slack one in power flow calculation. The entire load of the power system is 4242MW. It assumed that node 1 accessing a wind turbine that has a conspicuous rand feature. The wind speed will be derived from a weibull distribution with the $\mathrm{k} = 2.3466$ and $\mathrm{c} = 8.0928$. Rated power of wind turbine is 1Mw, cut-in wind speed is 3m / s, rated wind speed is 12m/s. The problem now is how to optimize the real power output of all controllable online generators and to satisfy the load demand.

According to (3)-(6), we can get the real power output of node 1 at interval of 10 minutes. Fig.1 shows the power output in each period.

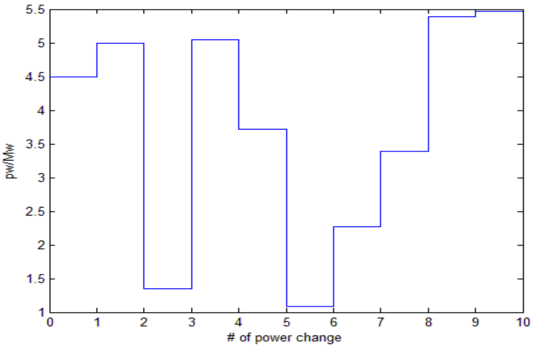


Fig. 1. Power change of node 1

The parameters of GA were set in[10], Pc=0.8 and Pm=0.1 are the crossover probability and mutation probability respectively. The parameters of RDPSO were set in[10], and are the thermal coefficient and the drift coefficient respectively, and the decrease from 0.9 to 0.3 on the course of the search. The parameters of PSO were set in[20], the range of inertia weight ω is decrease from 1.2 to 0.8 and the leaning factors were set as c1=c2=2.

Table 1 lists the total cost by each method mentioned when the power output of node 1 is 4.49Mw (in the first change). The mean cost and the standard deviation got by 100 runs of these methods. Fig.2 is convergence properties of the tested optimization methods. Both of them indicate that BBPSO has a better performance and robustness than other methods .

Table 1. The results for 100 runs

Methods	Min.Cost	Mean.Cost	Std.Dev	Mean.Time/min
GA	147856	161504	13648	10.0
PSO	150489	151815	1326	9.0
RDPSO	152352	153093	714	8.0
BBPSO	148023	148275	252	5.0

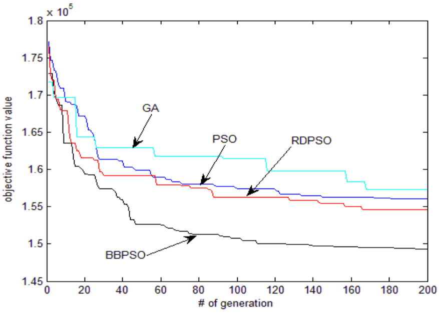


Fig. 2. Convergence properties of methods

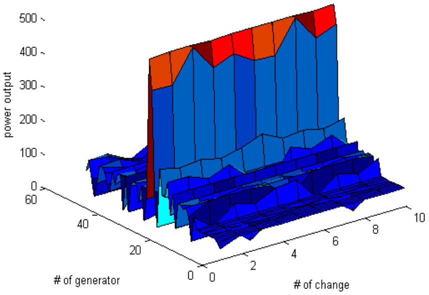


Fig. 3. The output power solution

Fig.3 is the solution of the power output of 53 generators when the power output of the node 1 keep changing in the interval of 10 minutes. It shows the power output solution of 52 generators in 10 times, when the wind speed of the node 1 is derived from a weibull distribution with the $k = 2.3466$ and $c = 8.0928$.

6 Conclusion

In this paper, we propose a BBPSO-based method to solve ED problems. The performance of the proposed method is evaluated in an IEEE 118-bus power system, and compared with other optimization methods in terms of convergence performance and solution quality. The results show that BBPSO method performs better than other compared methods in solving the ED problem. Our future work will focus on the dynamic economic dispatch problem.

Acknowledgement. This work was supported in part by the Natural Science Foundation of China (71371142, 61005090, and 61034004), the Fundamental Research Funds for the Central Universities, and the Research Fund of State Key Lab. of Management and Control for Complex systems.

References

1. Balouktsis, A., Chassapis, D., Karapantsios, T.D.: A nomogram method for estimating the energy produced by wind turbine generators. Solar Energy 72(3), 251–259 (2002)
2. Chowdhury, B.H., Rahman, S.: A review of recent advances in economic dispatch. Institute of Electrical and Electronics Engineers (1990)
3. Dhar, R., Mukherjee, P.: Reduced-gradient Method for Economic Dispatch. Proc. Inst. Elec., Eng. 120(5), 608–610 (1973)
4. Abarghooee, R., Aghaei, J.: Stochastic Dynamic Economic Emission Dispatch Considering Wind Power. In: IEEE Engineering and Automation Conference, vol. 1, pp. 158–161 (2011)
5. Loia, V., Vaccaro, A.: Decentralized Economic Dispatch in Smart Grids by Self-organizing Dynamic Agents. IEEE Trans. Systems, Man and Cybernetics: Systems 44(4), 397–408 (2014)
6. Chiang, C.: Improved Genetic Algorithm for Power Economic Dispatch of Units with Value-point Effects and Multiple Fuels. IEEE Trans. Power Syst. 20(4), 1690–1699 (2005)
7. Elsayed, S., Sarker, R., Essam, D.: An Improved Self-adaptive Differential Evolution Algorithm for Optimization Problems Industrial Informatics. IEEE Trans. on Industrial Informatics 9(1), 89–99 (2013)
8. Yang, C., Deconimck, G., Gui, W.: An Optimal Power-dispatching Control System for Electrochemical Process of Zinc Based on Back-propagation and Hopfield Neural Network. IEEE Trans. Electron 50(5), 953–961 (2003)
9. Lin, W., Cheng, F., Tsay, M.: An Improved Tabu Search for Economic Dispatch with Multiple Minima. IEEE Trans. Magn. 38, 1037–1040 (2002)

10. Sun, J., Palade, V., Wu, X.: Solving the Power Economic Dispatch Problem with Generation Constraints by Random Drift Particle Swarm Optimization. IEEE Trans. on Industrial Informatics 10(1), 222–232 (2014)
11. Faria, P., Soares, J.: Modified Particle Swarm Optimization Applied to Integrated Demand Response and DG Resources Scheduling. IEEE Transaction on Smart Grid 4(1), 606–616 (2013)
12. Chen, S., Manalu, G.: Fuzzy Forecasting Based on Two-factors Second-order Fuzzy-trend Logical Relationship Groups and Particle Swarm Optimization Techniques. IEEE Transaction on Cybernetics 43(3), 1102–1117 (2013)
13. Niknam, T., Doagou-Mojarrad, H.: Multiobjective Economic/Emission Dispatch by Multiobjective θ-particle Swarm Optimization, Generation, Transmission & Distribution, 6(5), 363–377 (2012)
14. Fu, Y., Ding, M.: Route Planning for Unmanned Aerial Vehicle (UAV) on the Sea Using Hybrid Differential Evolution and Quantum-behaved Particle Swarm Optimization. IEEE Trans. Systems, Man, and Cybernetics: Systems 43(6), 1451–1465 (2013)
15. Kennedy, J.: Bare Bone Particle Swarm. In: Proceedings of the IEEE Swarm Intelligence Symposium, pp. 80–87 (2003)
16. Blackwell, T.: A Study of Collapse in Bare Bones Particle Swarm Optimization. IEEE Transactions on Evolutionary Computation 16(3), 354–372 (2012)
17. Chen, C.: Cooperative Bare Bone Particle Swarm Optimization. Information Science and Control Engineering, 1–6 (2012)
18. Krohling, R., Mendel, E.: Bare Bones particle Swarm Optimization with Gaussian or Cauchy Jumps. In: IEEE Congress on Evolutionary Computation, pp. 3285–3291 (2009)
19. Wang, P., Shi, L.: A Hybrid Simplex Search and Modified Bare-bones Particle Swarm Optimization. Chinese Journal of Electronics 22(1), 104–108 (2013)
20. Kang, Q., Zhou, M.: Swarm Intelligence Approaches to Optimal Power Flow Problem with Distributed Generator Failures in Power Networks. IEEE Transactions on Automation Science and Engineering 10(2), 343–353 (2013)
21. Shapic, E., Balzer, G.: Power Fluctuation from a Large Wind Farm. In: International Conference on Future Power Systems, November 16-18 (2005)
22. van den Bergh, F.: An Analysis of Particle Swarm Optimizers, Pretoria, South Africa: Depatment of Computer Science, University of Pretoria (2002)
23. Pan, F., Hu, X., Eberhart, R.: An Analysis of Bare Bones Particle Swarm. In: IEEE Swarm Intelligence Symposium, pp. 1–5 (2008)
24. Kang, Q., Lan, T., Yan, Y.: Group Search Optimizer Based Optimal Location and Capacity of Distributed Generations. Neurocomputing 78(1), 55–63 (2012)
25. Power Systems Test Case Archive, `http://www.ee.washington.edu/research/pstca/pf118/pg_tca118bus.htm`

Extracting Mathematical Components Directly from PDF Documents for Mathematical Expression Recognition and Retrieval*

Botao Yu[1,2], Xuedong Tian[1,2,**], and Wenjie Luo[1,2]

[1] College of Mathematics and Computer, Hebei University, Baoding, Hebei, China
[2] Hebei key laboratory of Machine Learning and Computational Intelligence, Baoding, China
txdinfo@yahoo.com

Abstract. PDF document gains its popularity in information storage and exchange. With more and more documents, especially the scientific documents, available in PDF format, extracting mathematical expressions in PDF documents becomes an important issue in the field of mathematical expression recognition and retrieval. In this paper, we proposed a method of extracting mathematical components directly from PDF documents rather than cooperating indirectly with corresponding images converted from PDF files. Compared with traditional image-based method, the proposed method makes full use of the internal information of PDF documents such as font size, baseline, glyph bounding box and so on to extract the mathematical characters and their geometric information. The experimental result shows the method could meet the needs of the following processing of mathematical expressions such as formula structural analysis, reconstruction and retrieval, and has a higher efficiency than traditional image-based ways.

Keywords: PDF, Mathematical expression component, Mathematical expression recognition and retrieval, Font size, Baseline, Glyph bounding box.

1 Introduction

PDF (Portable Document Format) [1] documents present their contents and layouts in a manner independent of application software, hardware, and operating systems, which provides users with a consistent experience in sides of the displaying and printing pattern. With an increasing number of documents presented with PDF, more and more attentions are paid to this format of document for making good use of this resource.

Current researches on PDF documents involve extracting components from PDF documents or converting PDF documents into other formats such as XML and

* This work is supported by the National Natural Science Foundation of China (Grant No. 61375075) and the Natural Science Foundation of Hebei Province (Grant No. F2012201020).

** Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 170–179, 2014.

© Springer International Publishing Switzerland 2014

HTML. Chao and Fan [2] developed a method of extracting layout and content of PDF documents. The document was separated into text, image and vector graphics according to the object type. After that, words were formed to lines, then segments and images and vector graphics were saved. Marinai [3] developed a software tool to extract administrative metadata from PDF documents which could assist for building personal digital libraries. Déjean and Meunier [4] designed a system for converting PDF documents into structured XML format. In this system, streams that contain text, bitmap and vector images were extracted and converted respectively, and the extracted components were expressed in XML format. Rahman and Alam [5] proposed a method of converting PDF documents into HTML. By applying document image analysis techniques to retrieval logical and layout information of the document, the document was output in HTML format.

As an important component in PDF documents, mathematical expressions are also needed to be extracted for further recognizing and searching processes. Different from the images recognized and analyzed by mathematical formula recognition system, a PDF document that is not generated from scanned images has already contained the information of character code, baseline and font size. Therefore, the traditional operations applied to formula images for improving image quality and obtaining symbol codes such as image preprocessing (including binarization, denosing, skew detection and correction, etc.) and symbol recognition are not required. Nevertheless, the PDF documents do not provide the syntax and semantic information of symbols, it is necessary to locate the precise geometrical information of symbols and obtain their logic relationships for the following mathematical expression extraction, reconstruction and retrieval operations. Yang and Fateman [6] stated the significance of accessing mathematical expressions on line in digital documents like Postscript and PDF documents. And a method of extracting formulas from Postscript documents is proposed. First, a modified version of a program called Prescript was used to output information about strings and bounding boxes about typeset expressions. Then broken string fragments were assembled into words and items were determined as part of mathematical expressions by the characteristics of fonts and words (e.g. sin) commonly used in mathematical expressions and mathematical characters. Finally, the built-up mathematical expression in Lisp data structure was generated from stored data by applying clumping heuristics based on an existing Math/OCR program. In literature [7], Chan and Yeung summarized the existing work of mathematical expression recognition on symbol recognition and structural analysis. Lin et al. [8] combined rule-based and learning-based methods to detect both isolated and embedded mathematical expressions in PDF documents. For isolated formulas, they first used the character features to remove text lines which didn't seem to contain expressions. Then, the confidence level of classifying a line as a formula line was calculated by exploiting the geometric layout features. For embedded formulas, they focused much on the character features combined with additional layout features. Then the confidence level of a character being a math symbol was calculated. If the confidence level was higher than a threshold, the corresponding line was detected as an isolated formula or an embedded formula. In literature [9], they further discussed the identification of embedded mathematical formulas in PDF documents. First, text lines were segmented into words which are classified into formula type and ordinary text type with an SVM classifier. Then, formulas were extracted by merging formula type words as formulas. Baker et al. [10, 11] proposed a method of extracting mathematical expression by accessing the PDF document and a rasterized version of the PDF document. First, characters and their related information of font size, baseline and bounding box were extracted from the original PDF documents. In order to solve the problem of bounding box overlapping and obtain the exact character bounding boxes, they rendered PDF documents into images. After searching bounding boxes of the glyphs in the image, all the bounding boxes were registered with characters obtained from the original PDF documents. Then the expression parse tree was established with characters and related geometric information. This method paid attention to internal information in PDF documents such as font size, baseline, font name and font bounding box, which helped to locate the characters and got the minimal character bounding box from the corresponding images converted from original PDF document indirectly.

In this paper, we propose a method of extracting mathematical components directly from PDF documents for mathematical expression recognition and retrieval. Different from the method proposed in literature [10, 11], we obtain all information about glyph bounding box, baseline and font size, by directly accessing the original PDF documents, which makes full use of the internal information in the PDF documents and is also efficient. The glyph bounding boxes, together with baselines and character codes are used for the following processing.

A PDF document has complex structure. It belongs to a text and binary integrated format with compressed data, which leads to low readability of the original code of the documents. PDF documents can be generated by many tools and each tool has its own standard based on the PDF reference which has 7 editions so far. Font types used in PDF document could also frequently vary. When it comes to mathematical symbols, some symbols are generated by path construction operators (e.g. the long fraction symbol) or made of a character and a shape defined by the path construction operator (e.g. √ and a horizontal line make up the radical symbol). All these facts make it complex to extract components from a PDF document directly.

Our proposed method focuses on PDF documents with type1 font only and doesn't constrain tools that generate PDF documents.

2 Extracting Mathematical Components from PDF Documents Directly

The workflow of our method mainly contains 3 steps described as following and shown in Fig. 1.

1. Parsing of PDF documents. Parse the PDF documents to get the information of the fonts in Resource dictionary and the content of the Content stream of the Page objects.
2. Components extraction. Extract font and character information from the content parsed in step 2.
3. Expression output. Compose the mathematical expression with components in step 2 by using existing technique.

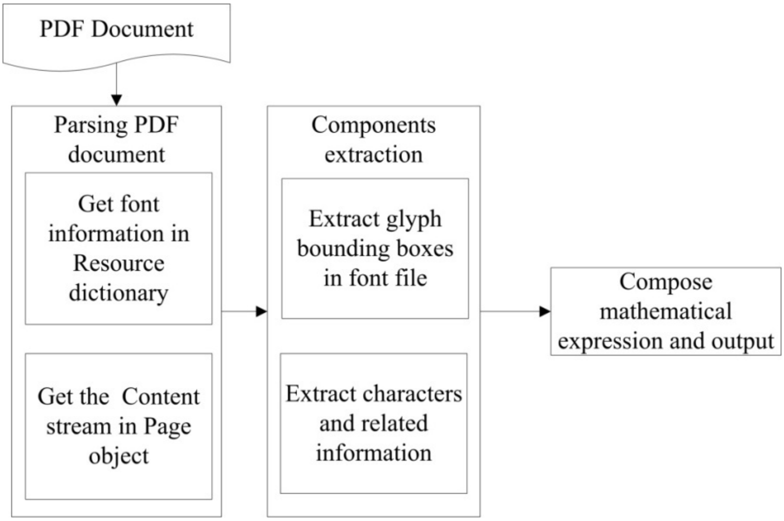

PDF Document
Parsing PDF document
Get font information in Resource dictionary
Get the Content stream in Page object
Components extraction
Extract glyph bounding boxes in font file
Extract characters and related information
Compose mathematical expression and output

Fig. 1. Workflow of the proposed method

2.1 Parsing of PDF Documents

For a better understanding of PDF and considering our specified research purpose, we decide to develop a special parsing algorithm of PDF documents, although there are many open source SDKs or open source software such as iTextSharp, iText, PDFBox and so on that can handle PDF documents. The parsing process mainly contains the following steps:

1. Read the global objects such as Xref Table and root object to get all the Page objects.
2. Get basic font information in Resource dictionary of the Page object.
3. Decode the content stream of the Page object.

Once the PDF document is parsed and the contents of content stream and font information in resource dictionary are obtained, we move to the extraction process.

2.2 Extraction of Glyph Bounding Box in Font File

By accessing the font dictionary and the font descriptor obtained in 2.1, we get the font type, font name, font bounding box, character encoding and width of the glyph in the font. Font bounding box and width are expressed in glyph coordinate system and are measured in units in which 1000 units corresponds to 1 unit in text space. When the font bounding box is transformed by the transformation method which will be described in 2.3, we find that the font bounding box is actually larger than the area the character pixels take and cannot be treated as the parameter to calculate the precise area the character occupy, as shown in Fig. 2.

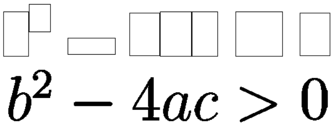


Fig. 2. An example of transformed font bounding boxes of the corresponding characters

In Fig.2, it is obvious that character a, b and c that share the same font have the font bounding boxes with the same height which can not reflect the real areas they occupy. The bounding box for the character - is also much bigger than expected. These boxes are grosser than the exact coordinates of each character's bounding box and cannot be used as geometric information for mathematical expression recognition. To solve the problem, we can take advantage of PDF documents in consistence that different PDF customer applications can display the PDF documents with same appearance, i.e. characters at the same position with the same appearance to get more precise geometrical data from the parameters in PDF documents that control the pattern of the layouts. Now we concentrate on the font file in PDF documents.

Font file is a stream containing Type 1 font program. It's in binary format and defines logic to render the character. Positional values in font file are also expressed in glyph coordinate system. Type1 font has two types, the Compact Font Format [12] and Adobe Type 1 Font Format [13]. Each type has its own format, but uses the same method to render the character. Type1 font uses commands to draw lines and Bezier curves to describe the appearance of the characters, as shown in Fig. 3.

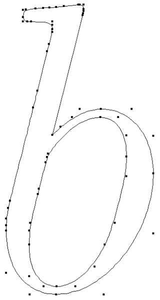


Fig. 3. An example of character b described in font file

In Fig. 3, there are many key position points on or off the outline of the character. These points control the display behaviors of the lines and Bezier curves that make up the outline, which could be used to calculate the exact bounding box, also called the glyph bounding box.

Fig. 4(a) shows one kind of binary data commands in font file that draws the character b and Fig. 4(b) shows the same commands but in decoded format which is more readable.

```
F8 34 81 9F F8 2F 9F F7 69 B3 01 B9 C2 F7 81 CC
03 F7 7B F9 31 15 8C 8F 8D 91 8B 90 08 95 81 8B
89 8A 67 88 89 79 1E 7A 8A 7C 89 79 8A 73 89 84
8A 8B 79 08 81 95 8B 95 BE 8B 82 81 84 83 6E 87
79 1E 73 2B 81 63 52 FB 77 87 79 19 86 72 8B 7A
8B 7E 08 25 C4 56 CC F7 08 F7 0C F7 29 F7 25 E7
57 CC 40 57 5C 60 77 78 1E 95 FC 04 15 6B 68 A3
D9 AC 8E 9E 9D D1 1F 8E 98 9B CB 8F 98 8D 93 C6
E4 CF 8B 08 B7 9F 5F 57 5B 6F FB 05 72 57 1F 72
55 5D 5B 5D 8B 08 0E
```

(a)

```
416 -10 20 411 20 213 40 hstem
46 55 237 65 vstem
231 699 rmoveto
1 4 2 6 0 5 rrcurveto
10 -10 0 -2 -1 -36 -3 -2 -18 vhcurveto
-17 -1 -15 -2 -18 -1 -24 -2 -7 -10 -18 rrcurveto
-10 10 0 10 51 0 -9 -10 -7 -8 -29 -4 -18 vhcurveto
-24 -96 -10 -40 -57 -227 -4 -18 rlinecurve
-5 -25 0 -17 0 -13 rrcurveto
-102 57 -53 65 116 120 149 145 92 -52 65 -75 -52 -47 -43 -20 -19 vhcurveto
10 -368 rmoveto
-32 -35 24 78 33 3 19 18 70 hvcurveto
3 13 16 64 4 13 2 8 59 89 68 0 rrcurveto
44 20 -44 -52 -48 -28 -113 -25 -52 hvcurveto
-25 -54 -46 -48 -46 0 rrcurveto
endchar
```

(b)

Fig. 4. Commands in font file rendering character b. (a) Binary format; (b) Decoded format

The process of our program that parses the decoded commands to get the glyph bounding box is composed of the following steps:

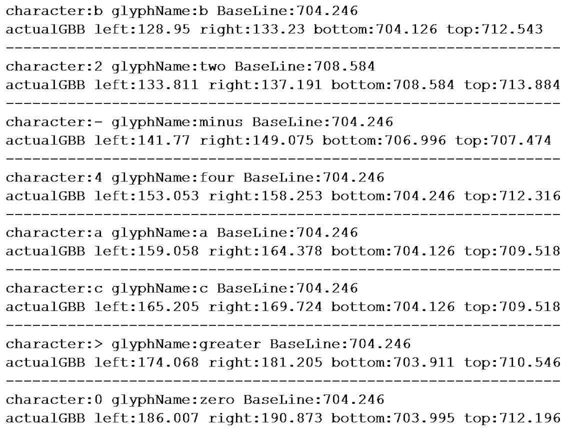
1. Read one single command and denote it as *CurrentCmd*.
2. If *CurrentCmd* is endchar, go to step 5; otherwise, go to step 3.
3. Obtain all the key points that control the boundary of the shape in *CurrentCmd* and calculate the minimal box that holds all the points. Denote the box as *CurrentBox*.
4. If *CurrentBox* is the first box got, denote it as *BOX*; otherwise calculate the minimal box that holds *CurrentBox* and *BOX*, and update *BOX* with the result. Go to step 1.
5. Output *BOX*. End.

By decoding the binary data and parsing the commands, the process of rendering characters is reproduced, from which we obtain the glyph bounding boxes. As with the font bounding box, the glyph bounding box is also expressed in glyph coordinate system. The glyph bounding box in font file is actually a unit bounding box, which means it equals to the exact area that the corresponding character whose font size is 1 takes. After the transformation in 2.3, the glyph bounding box could be used as the precise bounding box of the corresponding character with a certain font size. So far we have gotten the glyph bounding boxes in font file, the next is to extract geometric shapes and characters from content stream and transform the glyph bounding boxes.

2.3 Extraction of Mathematical Elements

Mathematical elements are classified into 4 types, single character, multi-characters, character with geometric shapes, and geometric shapes. In this process, all elements are extracted with their related attributes. We design a structure called *ElementInfo* to represent a mathematical element.

```
struct ElementInfo{
  string glyphName;/*element name, e.g. plus for + */
  double baseLine;
 TextSpaceBoundingBox actualGBB;/*transformed glyph
bounding box*/
};
```



```
character:b glyphName:b BaseLine:704.246
actualGBB left:128.95 right:133.23 bottom:704.126 top:712.543
-------------------------------------------------------------
character:2 glyphName:two BaseLine:708.584
actualGBB left:133.811 right:137.191 bottom:708.584 top:713.884
-------------------------------------------------------------
character:- glyphName:minus BaseLine:704.246
actualGBB left:141.77 right:149.075 bottom:706.996 top:707.474
-------------------------------------------------------------
character:4 glyphName:four BaseLine:704.246
actualGBB left:153.053 right:158.253 bottom:704.246 top:712.316
-------------------------------------------------------------
character:a glyphName:a BaseLine:704.246
actualGBB left:159.058 right:164.378 bottom:704.126 top:709.518
-------------------------------------------------------------
character:c glyphName:c BaseLine:704.246
actualGBB left:165.205 right:169.724 bottom:704.126 top:709.518
-------------------------------------------------------------
character:> glyphName:greater BaseLine:704.246
actualGBB left:174.068 right:181.205 bottom:703.911 top:710.546
-------------------------------------------------------------
character:0 glyphName:zero BaseLine:704.246
actualGBB left:186.007 right:190.873 bottom:703.995 top:712.196
```

Fig. 5. An example of the extracted data

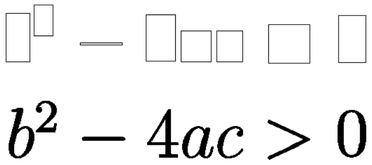


Fig. 6. An example of transformed glyph bounding boxes

In order to get the mathematical elements, we write a stack based program to extract characters, geometric shapes and the related attributes such as font size, start point and baseline from decoded content stream obtained in 2.1. Each character's position information can be calculated with the text showing commands in content stream. During the process, the glyph bounding boxes are transformed from glyph coordinate system to text coordinate system in 3 steps. First, the value of the glyph bounding box is divided by 1000. Then, multiply the value of the bounding box by the font size. Finally, move the origin of the glyph bounding box to the corresponding character's start point. For example, the glyph bounding box of character x in font file is [0 500 0 600] which takes the form [ll_x ur_x ll_y ur_y] specifying the lower-left x, upper-right x, lower-left y and upper-right y coordinates of the bounding box. In content stream, the start point of x is (100, 600) in text space and its font size is 6. After the first step, the value of the glyph bounding box is [0 0.5 0 0.6]. Then, it is changed to [0 3.0 0 3.6]. Finally, it becomes [100 103.0 600 603.6], which is the precise bounding box of x. All the positional information is expressed in the coordinate that sets the lower-left corner of the page as the origin. An example of the result obtained by our program for the expression is shown in Fig. 5. The transformed glyph bounding boxes of the expression are shown in Fig. 6.

From Fig. 5 and Fig. 6 we can see that all characters share the same baseline except character 2 and all bounding boxes are equal to the real sizes of the characters, which tell the exact position relationship between each character.

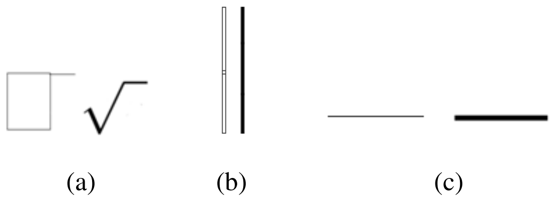


Fig. 7. Special mathematical elements. (a) Racial; (b) Long vertical bar; (c) Long fraction line.

Now the most of mathematical elements and their precise geometric information are ready for mathematical expression recognition. But some special situations still exist which would result in error results without analyzing and processing.

- Some mathematical elements are composed of characters and geometric shapes, e.g. the radical as shown in Fig. 7(a). The geometric shapes are added to the characters and the glyph bounding boxes are recalculated. For the radical in Fig. 7(a), the character √ intersects with the horizontal line, so the character and the line are treated as a whole and the glyph bounding box is recalculated to enclose the two items.
- For those who are composed of multi-characters, e.g. the vertical bar | as shown in Fig. 7(b), which may contain more than one shorter vertical bar to form a longer one, all the components are integrated as a new mathematical element with the same character name but a larger glyph bounding box and new baseline. In Fig. 7(b), the shorter vertical bars overlap with each other. The combining process is similar to the situation of the radical.
- The elements who are composed of geometric shapes, e.g. the longer fraction line as shown in Fig. 7(c) will be label as certain mathematical elements due to the context and their shapes.

2.4 Output of Expressions

After all the characters and their related attributes are extracted, the method of structural analysis of mathematical expressions proposed in [14] is used to analyze the structure to build the syntax tree for the expected expression. Once the syntax tree is build, the expression could be output in LATEX format. We will not talk about it much here.

3 Experimental Results

In this paper, MikTex and LATEX editors are used to generate PDF documents with Type1 fonts of varying font sizes and mathematical expressions in different forms such as matrix, integral, radical and so on. We use our proposed method to extract mathematical components directly from PDF documents that are similar to the layout as shown in Fig. 8(a).

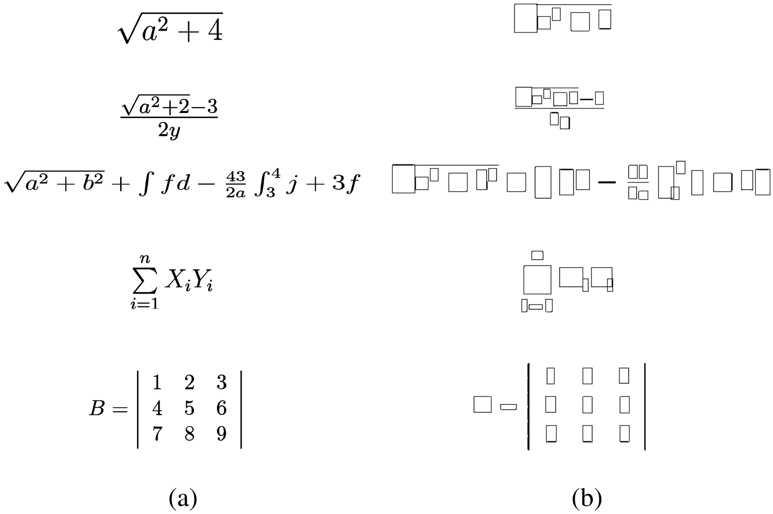

$\sqrt{a^2+4}$
$\frac{\sqrt{a^2+2}-3}{2y}$
$\sqrt{a^2+b^2}+\int fd-\frac{43}{2a}\int_3^4 j+3f$
$\sum_{i=1}^{n} X_iY_i$
$B=\begin{vmatrix}1&2&3\\4&5&6\\7&8&9\end{vmatrix}$
(a)
(b)

Fig. 8. Layout of a PDF file and its experimental results. (a) Layout; (b) Results.

Fig.8 (b) is the result showing exact geometric information of mathematical elements in Fig.8 (a) by applying our method. Experimental results show that all the information of the characters in the generated mathematical expressions, including character name, glyph bounding box, position, and baseline could be extracted correctly. The mathematical components could be used for further mathematical expression recognition and retrieval.

4 Conclusions

In this paper, a method of extracting mathematical components directly from PDF documents for mathematical expression recognition and retrieval is proposed, which makes full use of the internal information of the PDF documents and doesn't draw support from any other methods like format conversion. The extracted components could be used to mathematical expression extraction, reconstruction and retrieval.

Although the proposed method could extract the components correctly, it is only for Type1 fonts. Our further work is to transferred this method to other type fonts and improve its robustness.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (Grant No. 61375075) and the Natural Science Foundation of Hebei Province (Grant No. F2012201020).

References

1. Adobe Systems Incorporated, PDF Reference, 6th edn. (November 2006)
2. Chao, H., Fan, J.: Layout and Content Extraction for PDF Documents. In: Marinai, S., Dengel, A.R. (eds.) DAS 2004. LNCS, vol. 3163, pp. 213–224. Springer, Heidelberg (2004)
3. Marinai, S.: Metadata Extraction from PDF Papers for Digital Library Ingest. In: 10th International Conference on Document Analysis and Recognition, pp. 251–255. IEEE Press, New York (2009)
4. Déjean, H., Meunier, J.-L.: A System for Converting PDF Documents into Structured XML Format. In: Bunke, H., Spitz, A.L. (eds.) DAS 2006. LNCS, vol. 3872, pp. 129–140. Springer, Heidelberg (2006)
5. Rahman, F., Alam, H.: Conversion of PDF Documents into HTML: a Case Study of Document Image Analysis. In: Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 87–91. IEEE Press, New York (2004)
6. Yang, M., Fateman, R.: Extracting Mathematical Expressions from Postscript Documents. In: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation, pp. 305–311. ACM (2004)
7. Chan, K.-F., Yeung, D.-Y.: Mathematical Expression Recognition: a Survey. J. International Journal on Document Analysis and Recognition. 3(1), 3–15 (2000)
8. Lin, X.Y., Gao, L.C., Tang, Z., Lin, X.F., Hu, X.: Mathematical Formula Identification in PDF Documents. In: 2011 International Conference on Document Analysis and Recognition, pp. 1419–1423. IEEE Press, New York (2011)
9. Lin, X.Y., Gao, L.C., Tang, Z., Hu, X., Lin, X.F.: Identification of Embedded Mathematical Formulas in PDF Documents Using SVM. In: IS&T/SPIE Electronic Imaging, pp. 82970D–82970D. International Society for Optics and Photonics (2012)
10. Baker, J.B., Sexton, A.P., Sorge, V.: Extracting Precise Data on the Mathematics Content of PDF Documents. Towards Digital Mathematics Library, Birmingham , pp. 75–79 (2008)
11. Baker, J.B., Sexton, A.P., Sorge, V.: A Linear Grammar Approach to Mathematical Formula Recognition from PDF. In: Carette, J., Dixon, L., Coen, C.S., Watt, S.M. (eds.) MKM 2009, Held as Part of CICM 2009. LNCS, vol. 5625, pp. 201–216. Springer, Heidelberg (2009)
12. Adobe Systems Incorporated, The Compact Font Format Specification, Version 1.0, 4 (December 2003)
13. Adobe Systems Incorporated, Adobe Type 1 Font Format, Version 1.1 (February 1993)
14. Tian, X.D., Li, N., Xu, L.J.: Research on Structural Analysis of Mathematical Expressions in Printed Documents. J. Computer Engineering 32(23), 202–204 (2006)

An Efficient OLAP Query Algorithm Based on Dimension Hierarchical Encoding Storage and Shark

Shengqiang Yao and Jieyue He*

School of Computer Science and Engineering,
MOE Key Laboratory of Computer Network and Information Integration,
Southeast University, Nanjing 210096, China
shengq_yao@163.com, jieyuehe@seu.edu.cn

Abstract. The on-line analytical processing (OLAP) queries always include multi-table joins and aggregation operations in their SQL clauses. As a result, how to reduce multi-table joins and effectively aggregate the query data with "big data" is the key issue for query processing. Therefore, the novel OLAP query algorithm is proposed in this paper based on the dimension hierarchical encoding (DHE) storage strategy with the In-Memory computing in Shark. With DHE and Shark, a star join with hierarchy level is mapped to a multidimensional range query on the fact table and the large-scale data by transformations and actions are computed on resilient distributed datasets (RDDs). The experimental results show that, compared with the data analysis operations in Hive, complex multi-table joins and I/O overhead are reduced by DHE and Shark. The query performance is greatly improved than that of the ordinary star schema.

Keywords: Big data, Data warehouse, Dimension hierarchical encoding, Shark.

1 Introduction

Data warehousing and on-line analytical processing (OLAP) [1] are essential elements of decision support system. However, the rapidly growing size of the data sets for business intelligence makes traditional warehousing solutions unsuitable. The main idea underneath this evolution is that those data sets need to be stored in the cloud and be accessed by a set of services. Following this consideration, there have been several proposals to store and process extremely large data sets.

Hadoop [2] is a popular open-source map-reduce implementation inspired by Google's MapReduce [3]. It is being widely used in web search, log analysis and other large-scale data processing filed. Hive [4] is a data warehousing solution built on top of Hadoop. It supports SQL-like declarative language (HiveQL) and is widely used. HiveQL is compiled into map-reduce jobs executed on Hadoop. However, expensive data materialization for fault tolerance and costlier execution strategies [5, 6] makes Hive slow. Shark [7, 8] is a new data warehouse system capable of deep data

* Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 180–187, 2014.
© Springer International Publishing Switzerland 2014

analysis using the Resilient Distributed Datasets (RDDs) memory abstraction. Though Shark shows quite better performance than hive, complex OLAP queries still take lots of system resources and affect the query efficiency, like pre-shuffle and intermediate data outputs, especially when the data sets become larger.

OLAP over big data repositories has recently received a great deal of attention from the research communities [10]. For example, there is a research focus on the performance of aggregation operations by using dimension-oriented storage in HBase [11]. A decomposed snowflake schema was proposed in [12] to get better performance in their parallel database. The work in [13] shows that HBase can't original support the complex OLAP queries. The performance of "data loading time" and "grep select time" in HBase is poorer than Hive. Other studies, like [14], they use a distributed cube model in the cloud platform. However, cube needs to be pre-computed and extra space to store. It is also quite difficult to decide which queries to pre-compute. In traditional warehouse system, a multidimensional hierarchical method [15, 16] is used to reduce table joins and improve the efficiency of queries. Therefore, an efficient OLAP query method is proposed in this paper by using dimension hierarchical encoding (DHE) storage and shark. The code for representing the hierarchies of each dimension replaces the foreign keys in the fact table. As a result, the complex table joins reduced and the efficiency of OLAP queries improved. Our experimental results on star schema benchmark (SSB) [9] show that with DHE storage and shark, the OLAP query method has better performance.

Briefly then, the outline of this paper is as follows. In Section 2, the method of OLAP query based on DHE and shark are described in detail. In Section 3, our algorithm is applied to SSB and the results are analyzed. In Section 4, the conclusions are given.

2 OLAP Query Based on DHE and Shark

2.1 Dimension Hierarchical Encoding

In data warehouse systems, a central fact table contains the measures, and the dimension tables connect to it via foreign keys. In most situations, the size of dimension table is far less than the fact table and the number of hierarchy level members is small. So we need only a little effort to tackle with the DHE when an OLAP query comes without losing any original semantics.

Assuming that L_{ij} is the j-th level attribute of dimension table D_i and $B^{L_{ij}}$ is the binary code of L_{ij}. $B^{L_{ij}}$ can be denoted as formula (1):

$$B^{L_{ij}} = \{b_{k-1} \dots b_i \dots b_0\}. \tag{1}$$

The value of b_i is 0 or 1 and k is the binary code length of L_{ij}. Its value can be calculated by formula (2):

$$\mathrm{k} = \mathrm{Bit}(L_{ij}) = \lceil \log_2 m \rceil \ \ (\mathrm{m} = |L_{ij}|). \tag{2}$$

m is the number of different members in L_{ij}.

Assuming that B^{D_i} is the dimension hierarchy encoding of dimension table D_i. To obtain B^{D_i}, DHE combines each hierarchy level of dimension D_i in a level-descending order (from the top level to low level). B^{D_i} can be calculated by formula (3):

$$B^{D_i} = (\ldots((B^{L_{i1}} \ll Bit(L_{i2})|B^{L_{i2}}) \ll Bit(L_{i3})|B^{L_{i3}})\ldots) \ll Bit(L_{im})|B^{L_{im}}. \tag{3}$$

"≪" is the left shift operator and "|" is the bitwise-Or operator.

For example, as shown in table 1, the "Customer" dimension has 3 different hierarchy levels {Region, Nation, City}. "Region" is the top level attribute and has total 5 different members, which is originally stored as string type.

According to formula (1) and (2), 3 bits are used to replace the 5 different members in Region with {001,010,011,100,101}. "Nation" and "City" can be respectively represented by 4bit and 8bit. By formula (3), we can obtain the DHE of dimension table "Customer". Table 1 displays the part of the DHE of "Customer".

Table 1. Part of the DHE of Customer

Customer(Dimension)		Region(Level)		Nation(Level)		City(Level)	
CustKey	$B^{Customer}$	Region	B^{Region}	Nation	B^{Nation}	City	B^{City}
1	0010001…00001	AFRICA	001	ALGERIA	0001	ALGE0	00000001
…	010…010100001	…	…	ARGENTINA	0010	……	……
300	…	ASIA	011	…	…	BRAZ0	00010101
…	…	…	…	BRAZIL	1111	…	…

In order to speed up the OLAP query with "big data" and reduce storage space, we use DHE to replace the dimension foreign keys in fact table and this phase can be done by ETL tools. Fig.1 shows the example of the storage strategy with DHE in star schema.

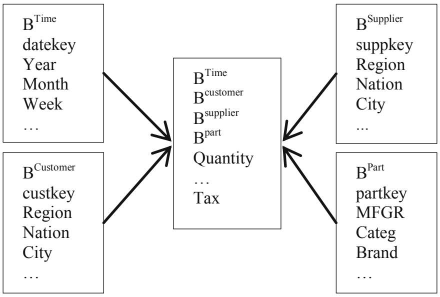


Fig. 1. Star schema with DHE

2.2 Overview of Spark and Shark

Spark [7] is a fast and general engine for large-scale data processing which outperforms Hadoop. Spark introduces an abstraction called RDDs (resilient distributed datasets). An RDD is a read-only collection of objects partitioned across a set of machines. RDDs achieve the abstraction to manipulate distributed data sets in local operations. RDDs can be calculated in two ways: Transformations (e.g. map, filter) and Actions (e.g. count, collect). Transformations are lazy operations that define a new RDD, while actions launch a computation to return a value.

Spark is composed of one Master node and several Worker nodes. The program developed by users called driver program. The workers are long-lived processes that can store RDD partitions in RAM across operations. The driver defines one or more RDDs and invokes actions on them.

Shark [8] is a data warehousing implementation on Spark. Shark is compatible with Hive and uses Hive query compiler. However, it transforms the operators into the operation on RDDs rather than MapReduce jobs. The Hadoop Distributed File System (HDFS) data obtained by shark will be computed by spark.

2.3 The OLAP Query Algorithm Based on DHE and Shark

Usually OLAP users are not interested in single measures but in some form of summarized data. An important concept of OLAP data models is the notion of dimension hierarchies. Hierarchies provide an appropriate method of describing the level of aggregation for a dimension. Thus, typical OLAP queries in star schema contain restrictions on multiple dimension tables that are then used as restrictions on the very large fact table:

```
Select F.attributes,D.attributes,Agg(Measures)
From Fact,Dimension
Where <Join Constraint> and <Dimension Restriction>
and <Fact Restriction>
Group By Gb-attributes
Order By Ob-attributes
```

Attributes in “<Dimension Restriction>”, “Gb-attributes” and “Ob-attributes” mostly contain hierarchy level attributes and a few non-hierarchy-level attributes.

By DHE, a star join with hierarchy level is mapped to a multidimensional range query on the fact table. In Shark, the large-scale data is computed by transformations and actions on RDDs. Map Sets <$L_{ij}, B^{L_{ij}}$> are preprocessed according to formula (1), (2) and B^{D_i} is got according to Formula (3). As the example shown in Fig.2, the hierarchy level restriction in dimension table “Customer” is Region=“AFRICA” and Nation=“ALGERIA”. According to formula (3), the first 3 bits in $B^{Customer}$ represent “Region” and the middle 4 bits represent “Nation”. So we create the mask code $B^{mask} = "111111100000000"$. $B^{AFRICA} = "001"$ and $B^{ALGERIA} = "0001"$ are obtained from formula (1) and (2). Thus, we create the filter code $B^{filter} = "001000100000000"$ for this restriction. Bitwise-And operator is then used between $B^{Customer}$ and B^{mask} to filter the tuples to new RDDs.

Thus, the query output can be obtained by the main steps as follows:

- Step1: Analyze the level attributes in “<Dimension Restriction>”. Obtain the hierarchy level code $B^{L_{ij}}$ from Map Set $<L_{ij}, B^{L_{ij}}>$ and their corresponding offset in B^{D_i}.
- Step2: create the mask B^{mask} and the filter key B^{filter} from Step 1.
- Step3: Sequential Scan the Fact table with B^{mask} and B^{filter}. Create new RDDs for those filtered tuples in Fact table as the example shown in Fig.2.
- Step4: If there exists non-hierarchy-level attribute in <Dimension Restriction>, Group By and Order By clause, join the corresponding Dimension table with the RDDs created in Step 3 and then generate new RDDs too.
- Step5: GroupByPreShuffle (part-aggregate the data at map-side) is executed on the RDD got from Step4 according to $B^{L_{ij}}$ in B^{D_i} and the other attributes. As a result, the amount of data processed at reduce-side will decrease.
- Step6: According to Gb-attributes, value will be distributed to different reducer. Then merge the values and compute the sum at the reduce side, sort and extract data.
- Step7: Submit the final results from worker nodes to master node.

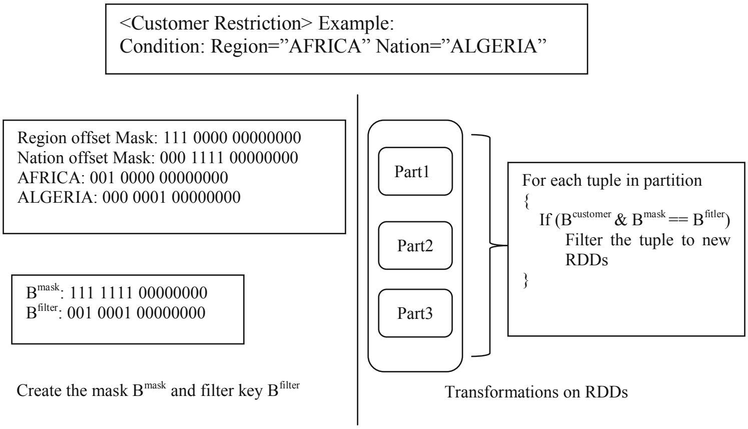

<Customer Restriction> Example:
Condition: Region=”AFRICA” Nation=”ALGERIA”
Region offset Mask: 111 0000 00000000
Nation offset Mask: 000 1111 00000000
AFRICA: 001 0000 00000000
ALGERIA: 000 0001 00000000
B^{mask}: 111 1111 00000000
B^{filter}: 001 0001 00000000
Create the mask B^{mask} and filter key B^{filter}
Part1
Part2
Part3
For each tuple in partition
{
If ($B^{customer}$ & B^{mask} == B^{fitler})
Filter the tuple to new RDDs
}
Transformations on RDDs

Fig. 2. Create the filter key for <Customer Restriction> and transform actions on RDDs

For example, the following complex multi-table joins query in SSB:

```
Select sum(revenue),year,brand
From datetbl join lineorder
on (datetbl.datekey=lineorder.orderdate) join part
on (part.partkey=lineorder.partkey) join supplier
on (supplier.suppkey=lineorder.suppkey)
Where category='MFGR#12' and region='AMERICA'
Group by year,brand
Order by year,brand
```

It filters data set according to region attribute in dimension table “supplier” and category attribute in “part”. The summarized “revenue” is group by and order by year in dimension table "datetbl" and brand in "part". As shown in Fig.3, new RDDs reflect an operator's transformation on the RDD that resulted from the previous operator's transformation. Based on DHE and Shark, We do not need to write intermediate results to HDFS in a temporary file, and only simply write results to local disk. The complex multi-table joins and I/O overhead are also reduced by DHE.

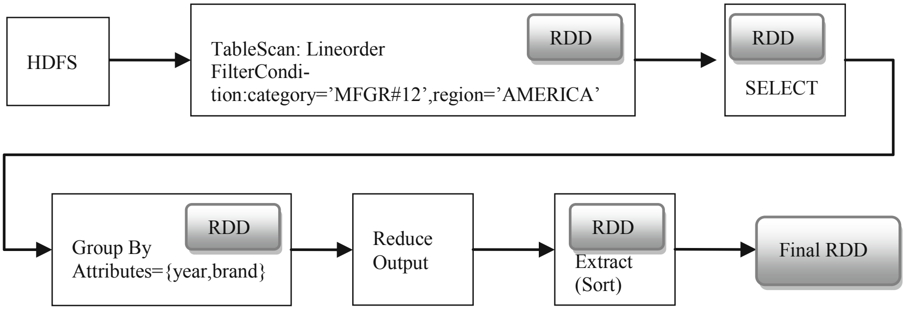


Fig. 3. Query plan based on DHE and Shark

3 Experiments and Results

SSB [9] is used in the experiment which is based on the TPC-H benchmark to measure the performance of data warehousing applications. SSB provides both functional coverage (different common types of Star Schema queries) and selectivity coverage (varying fractions of the fact table that must be accessed to answer the queries). There are four major query types in SSB. The number of dimension tables increases from query type 1 (Q1) to query type 4 (Q4). For example, Q1 is shown in the below code:

```
Select sum(lo_extendedprice*lo_discount) as revenue
From lineorder,date
Where lo_orderdate=d_datekey
and d_year=[YEAR]
and lo_discount between [DISCOUNT]-1
and [DISCOUNT]+1 and lo_quantity<[QUANTITY]
```

The above query is meant to quantify the amount of revenue increase that would have resulted from eliminating certain company-wide discounts in a given percentage range for products shipped in a given year. This is a "what if" query to find possible revenue increases.

The experiments utilized IBM high performance computing platform. The platform has total 279 computing nodes and 3500 CPU cores. We used 1 master node and 3 worker nodes. The available RAM of each worker node is set to 2GB.

We use three data sizes of 5G, 10G and 20G with DHE and pre-load them into HDFS. The records number in fact table is respectively 30 million, 60 million and 120 million. For each query type, we complete the query with different values several times (e.g. discount=0.5, discount=0.6) and use the average time as the result.

Q1 to Q4 on Hive-SSB and Shark-SSB are based on the original SSB. Q1 to Q4 on Shark-DHE are based on our OLAP query algorithm and the DHE star schema which is generated from the original SSB.

Fig.4, Fig.5 and Fig.6 illustrate that the data analysis performance built on Shark is much better than Hive. Moreover, Our OLAP algorithm based on DHE and Shark further improves the query time than the original star schema.

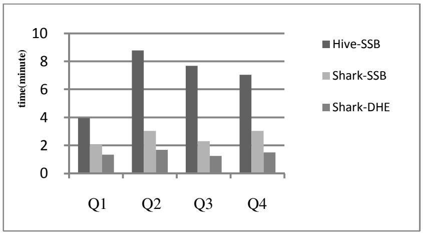


Fig. 4. Query performance on 5G data

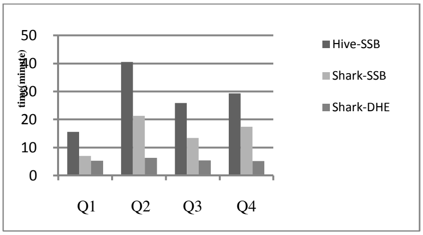


Fig. 5. Query performance on 20G data

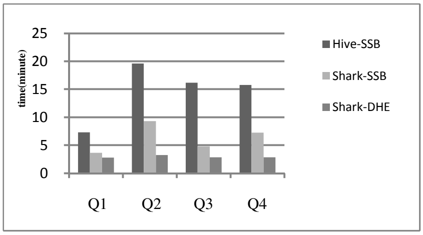


Fig. 6. Query performance on 10G data

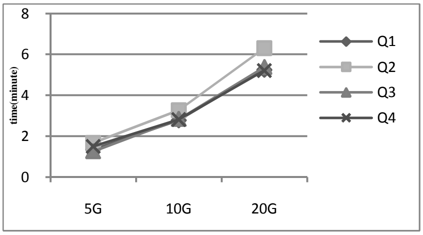


Fig. 7. Performance trends on Shark-DHE

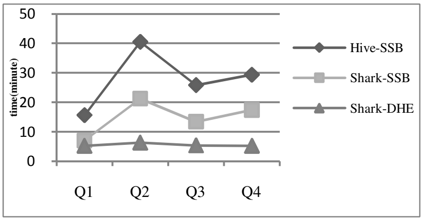


Fig. 8. Q1 to Q4 trends on 20G data

Fig.7 shows that the OLAP query time increases along with the size of the data. The trend of query time with 20G data is demonstrated in Fig.8. The trend in our method is quite smooth because most join operations can be removed by using DHE and map the query to a range query on the fact table. Q1 involves only one dimension and does not contain group by operations, so the improvement of the query time is not obvious between Shark-SSB and Shark-DHE.

4 Conclusion

With the era of "big data" and expanding amount of data size, the processing technology which computing distributed data set to satisfy the complex OLAP queries has become a research hot spot. In this paper, the novel OLAP query algorithm is proposed based on the dimension hierarchical encoding storage strategy with the In-Memory computing in Shark. This method reduces the complex multi-table joins in star schema. The results of experiment reveal that our OLAP algorithm based on DHE and Shark greatly improves the query time.

Acknowledgements. This work was supported by the Natural Science Foundation of Jiangsu Province under Grant No.BK2012742.

References

1. Chaudhuri, S., Dayal, U.: Data warehousing and OLAP for decision support. ACM Sigmod Record 26(2), 507–508 (1997)
2. Apache Hadoop, `http://wiki.apache.org/hadoop`
3. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Communications of the ACM 51(1), 107–113 (2008)
4. Thusoo, A., Sarma, J., Jain, N.: Hive: a warehousing solution over a map-reduce framework. Proceedings of the VLDB Endowment 2(2), 1626–1629 (2009)
5. Pavlo, A., Paulson, E., Rasin, A.: A comparison of approaches to large-scale data analysis. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, pp. 165–178. ACM (2009)
6. Stonebraker, M., Abadi, D., DeWitt, D.: MapReduce and parallel DBMSs: friends or foes. Communications of the AC 53(1), 64–71 (2010)
7. Zaharia, M., Chowdhury, M., Das, T.: Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, p. 2. USENIX Association (2012)
8. Pavlo, A., Paulson, E., Rasin, A.: A comparison of approaches to large-scale data analysis. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, pp. 165–178. ACM (2009)
9. O'Neil, P., O'Neil, E., Chen, X.: The star schema benchmark (SSB). Pat. (2007)
10. Jing-hua, Z., Ai-mei, S., Ai-bo, S.: OLAP Aggregation Based on Dimension-oriented Storage. In: IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), pp. 1932–1936. IEEE (2012)
11. Apache HBase, `http://www.hbase.apache.org`
12. Wang, H., Qin, X., Zhang, Y., Wang, S., Wang, Z.: LinearDB: A relational approach to make data warehouse scale like mapReduce. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) DASFAA 2011, Part II. LNCS, vol. 6588, pp. 306–320. Springer, Heidelberg (2011)
13. Shi, Y., Meng, X., Zhao, J.: Benchmarking cloud-based data management systems. In: Proceedings of the Second International Workshop on Cloud Data Management, pp. 47–54. ACM (2010)
14. Brezany, P., Zhang, Y., Janciak, I.: An Elastic OLAP Cloud Platform. In: IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), pp. 356–363. IEEE (2011)
15. Fa, H., Sheng, Y., Zhen, X.: A Novel Aggregation Algorithm for Online Analytical Processing Queries Evaluation Based on Dimension Hierarchical Encoding. J. Computer Research and Development. 4, 608–614 (2004)
16. Markl, V., Ramsak, F., Bayer, R.: Improving OLAP performance by multidimensional hierarchical clustering. In: International Database Engineering and Applications Symposium, pp. 165–177. IEEE (1999)

The Enhancement and Application of Collaborative Filtering in e-Learning System

Bo Song and Jie Gao

College of Software, Shenyang Normal University, Shenyang City,
Liaoning Province, China, 110034
songbo63@aliyun.com, gaojiexy@126.com

Abstract. Collaborative Filtering recommendation algorithm is one of the most popular approaches for determining recommendations at present and it can be used to solve Information Overload issue in e-Learning system. However the Cold Start problem is always one of the most critical issues that affect the performance of Collaborative Filtering recommender system. In this paper an enhanced composite recommendation algorithm based on content recommendation tags extracting and CF is proposed to make the CF recommender system work more effectively. The final experiment results show that the new enhanced recommendation algorithm has some advantages on accuracy compared with several existing solutions to the issue of Cold Start and make sure that it is a feasible and effective recommendation algorithm.

Keywords: Recommendation Algorithm, Cold Start, Collaborative Filtering, e-Learning.

1 Introduction

With the development of Internet technology, huge information resource is presented to us constantly, thus the issue of information overload formed. It becomes more difficult for users to find out what they need or they are interested in. Facing information overload issue, recommender system is one of the solutions. Collaborative Filtering is one of the most popular approaches for determining recommendations at present and the CF recommendation algorithm is proposed originally in 1992 by Goldberg, Nichols and Oki [1]. It is an intelligent and personalized information service system and describes the user's lone-term information need by user modeling, based on which it can customize the personalized information with the specific recommendation strategy [2]. The implementation of CF recommender system must rely on users' explicit rating, such as Amazon, Drugstore; these electronic commerce platforms possess a large number of users who have rated their purchased goods. The CF recommendation algorithm can be divided into two parts. One is user-based CF and the other one is item-based CF.

The e-Learning system stays on the situation that they may confront with Cold Start issue while using Collaborative Filtering. When new users register the system or new items are added into the system, the CF recommender system may recommend

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 188–195, 2014.
© Springer International Publishing Switzerland 2014

no items for the new users or the new items may not be recommended for existing users because of there is no relevant rating information in the user-item rating matrix known as user modeling. This is the problem of Cold Start of CF recommendation algorithm. Aiming at above problems, in this paper an enhanced composite recommendation algorithm based on content information tags extracting and Collaborative Filtering will be proposed to solve the new users' Cold Start problem to make the recommender system works more effectively.

2 Collaborative Filtering Recommendation Algorithm

Personalized recommendation is an important part in user behavior analysis. In simple term, it is a process of searching the resource which the user might interest in. At present, Collaborative Filtering is a relatively mature and popular recommendation algorithm. In CF, an item is considered as a black box-we don't look at its content and user interactions (such as rating, saving, purchasing) with the item are used to recommend an item of interest to the user. The main idea of Collaborative Filtering is to exploit information about the past behavior or opinions of an existing user community for predicting which items the current user will most probably like or be interested in [3]. The recommender process is simply as follows: first given a user-item rating dataset and a current user as input and calculate the similarity between users; second use the similarity matrix made up of the above similarities to find similar users that had similar preferences to those of the current user, that is called nearest neighbors; third for every item that the current user has not yet seen, the prediction ratings will be calculated based on the ratings for the items made by the similar users; finally sort the prediction ratings and the top-N items will be recommended for the current user. Such a top-N list should not contain items that the current user has already rated.

The matrix of given user-item ratings is known as user modeling which is obtained by collecting and arranging users interactions. If the number of items is "n" and the number of users is "m" in the system, the user modeling is a rating matrix of "m*n". We use $\mathrm{U} = \{User_1, User_2, ..., User_m\}$ to denote the set of users, $\mathrm{I} = \{Item_1, Item_2, ..., Item_m\}$ for the set of items, and R as a $m*n$ matrix of ratings $\mathrm{r}_{(i,j)}$, with $i \in \{1 ... m\}$ and $j \in \{1 ... n\}$. The possible rating values are defined on a numerical scale from 1 to 5 in this paper and the higher the rating value, the more strongly the user likes the item.

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| * |\vec{b}|} \tag{1}$$

To find nearest neighbors, the formulae applied to calculate the similarity between users or items must be used. There are several similarity calculation methods, such as the cosine similarity measure, adjusted cosine similarity measure, Pearson correlation coefficient measure. Cosine similarity is established as the standard metric in item-based Collaborative Filtering; as it has been shown that it produces the most accurate results [3]. The formula (1) shows the cosine similarity measure which calculates the rating similarity between Item a and Item b in item-based Collaborative Filtering. The range of the ratio in formula (1) represents similarity which is from 0 to 1. The larger the ratio is, the higher the similarity between two items is. However, the basic cosine measure does not take the differences in the average rating behavior of the users into account [3]. The adjusted cosine measure solves the problem and subtracts the user average. And the values for the adjusted cosine measure correspondingly range from −1 (strong negative correlation) to +1(strong positive correlation) as is shown in formula (2). Here $r_{u,a}$ is the rating given by user u to the item a, $r_{u,b}$ is the rating given by user u to the item b and $\bar{r}_u$ represents the average of user u ratings.

$$sim(a,b) = \frac{\sum_{u \in U}(r_{u,a} - \bar{r}_u)(r_{u,b} - \bar{r}_u)}{\sqrt{\sum_{u \in U}(r_{u,a} - \bar{r}_u)^2}\sqrt{\sum_{u \in U}(r_{u,b} - \bar{r}_u)^2}} \tag{2}$$

As in the adjusted cosine measure, the Pearson correlation coefficient measure shows in the formula (3). The symbol $\bar{r}_a$ corresponds to the average rating of user a and $\bar{r}_b$ corresponds to the average rating of user b and takes the values from −1 to +1. Pearson correlation coefficient is used to calculate the similarity between users in user-based Collaborative Filtering.

$$sim(a,b) = \frac{\sum_{i \in I}(r_{a,i} - \bar{r}_a)(r_{b,i} - \bar{r}_b)}{\sqrt{\sum_{i \in I}(r_{a,i} - \bar{r}_a)^2}\sqrt{\sum_{i \in I}(r_{b,i} - \bar{r}_b)^2}} \tag{3}$$

After the nearest neighbors were found, the recommender engine will work according to the prediction rating of non-rated items based on the current user. The prediction rating formula is shown in formula (4).

$$predictedRating = \frac{\sum(sim(a,b) * itemrating)}{\sum sim(a,b)} \tag{4}$$

Where, the prediction rating is the ratio of similar users' rating weighted sum and similarity sum. Then add up all similar users' prediction rating and sort them according to the values of sum and the top N items will be the recommendation items for the current user. The larger the prediction rating sum of an item rated by all similar users is, the more effective recommendation results the current user will get. Correspondingly the scale of the sum will not be more than "5".

3 The Enhancement of Collaborative Filtering

The success of Collaborative Filtering relies on the availability of a sufficiently large set of quality preference ratings provided by users, but it method may suffer from the new user problem, in which there is no rating record on new users in user modeling

[4]. This is the issue of new users' Cold Start. Usually there are several solutions to Cold Start problem such as random recommendation and Mean Value recommendation. In recent years, combining content information with CF recommendation proved to be an effective solution to Cold Start problem and has become a hot research [5].

3.1 Content-Based Tags Extracting Solution to Cold Start

The recommendation process of the new enhance composite recommendation algorithm can be divided into four steps: firstly content information of new users and all existing items will be analyzed and the tags that represent their feature will also be extracted; secondly find the similar matrix between the tags matrix of new users and the tag matrix of all items with the method of linear algebra, then recommend the items for the new users, this is the first time recommendation in the new algorithm; Thirdly the new users can choose the items and rate them, then the interaction can fill in the user modeling and the new users will acquire the final more accurate recommendation results.

When new users register the e-Learning system, it is necessary for them to fill in some individual information, especially education background, preference and forte. Usually the items in the e-Learning system are added in some text depiction about them, for instance lesson name, major classification, lesson introduction. We can use the above information to build tags matrix. Tags may contain a single term or multiple terms and it is very important to convert terms into numerical value.

The typical steps involves in building tags matrix are tokenization, normalization, eliminating stop words and quantitative filling. At this stage, we will introduce the use of term frequency and document frequency to compute the weight associated with each term. Tokenization is to parse the text to generate terms and sophisticated analyzers can also extract phrases from the text. The *TokenStream* class and *Analyzer* class of Apache Lucene jar file can help to extract tags from the text [6]. Apache Lucene is a Java-based open source search engine developed by Doug Cutting [7]. And we can create a Map collection class *buildTermFrequencyMap* to build a HashMap to store the terms and their frequencies. It is with all the text information in the system as input and uses Loops and conditional statements to put the terms and frequencies into the *termFrequencyMap* Map. Then *getTopNTermFrequencies* method will find the top N terms appeared most frequently in each text information file by sorting the frequencies.

The second step is normalization to convert the terms to lowercase. To make the matrix more accurately, we will also use the *StandardAnalyzer* class of Apache Lucene to eliminate stop words such as "a, an, the" that appear in the text too often [7]. The stop words list can use a String type array to preserve. Eliminating stop words can not only enhance the importance of meaningful words, but also abate noise apparently at the same time. The Java static code block shows the above methods.

```
static {
   List<String> allStopWords = new ArrayList<String>();
   allStopWords.addAll(Arrays.asList(StandardAnalyzer.STOP_WORDS));
   allStopWords.addAll(Arrays.asList(ADDITIONAL_STOP_WORDS));
```

```
    MERGED_STOP_WORDS = allStopWords.toArray(new String[0]);    }
  public CustomAnalyzer() {
    super(CustomAnalyzer.MERGED_STOP_WORDS);    }
```

ADDITIONAL_STOP_WORDS is the String type array that contains stops words. It is added into the *allStopWords* ArrayList.

When the first top N terms were picked out and then they can consist of coordinate space of tags. So every new user $User_i$ can be described by term vectors: $User_i = \{U_1, U_2, ..., U_n\}$. Among them, U_j express a tag which is a column vector. For each new user' registration text information, take 80% term frequency as the cut-off point, If the term frequency is higher than 80%, we can set the term vector element value as 1, or else is 0. For each item, we can also use the above method to build the corresponding tags matrix. So, all the text information of new users and existing items can be indicated by a 0-or-1 matrix of tags by quantitative filling.

After defining the tags matrix, the tags matrixes of new users and items have the same dimensions. Then we can use the method of linear algebra to find similarity matrixes. Methods are as follows: Assume A, B as the matrixes of order n, if there exist invertible matrices P of order n, such that $P^{-1} * A * P = B$ was established, we call matrix A is similar to B, denoted by A~B. P^{-1} is the inverse matrix of P. Here also involves matrix multiplication algorithm. The steps for finding similarity matrix are as follows. Firstly calculate the Eigen values of the matrix: $|A - \lambda E| = 0$, here E is a unit matrix, from upper left to lower right corner of the main diagonal, the elements are all 1, outside are 0. The matrix A can be seen as tags matrix of random new user and matrix B can be tags matrix of existing item. Then for every Eigen value, calculate the solution space of matrix equation $(A - \lambda E)X = 0$. We can exploit the feature vector provided by basic solutions as column vector to comprise the matrix P. The situation shows that the invertible matrix P exists and matrix A is similar to B. If the above mentioned matrix equation has no solution space, there will not be invertible matrix P. On this situation, matrix A is not similar to B; there is no similarity between the new user and the item. Using this algorithm, the tags matrix of new user will be compared with every tags matrix of existing item, if there is tags matrix of item is similar to the tags matrix of the new user, the corresponding item will be recommended to the new user. The number of matrix B that can enable the equality $P^{-1} * A * P = B$ succeeds is certainly more than one in the system. So the first recommendation list for the new user can form. Similarity matrix is an equivalence relation and has reflexive, symmetry and transitivity three properties. The items that tags matrix are similar to the tags matrix of the new user can recommend to him or her. We call it the first-time recommendation because the recommendation this time is not enough accurate and we can further make efforts to let the recommender system work more effectively. Therefore, the items of first time recommendation will be chosen by the new user. The new user then can rate the items by ones preference and education background. After this step, the new user' interactions are added into the User Modeling. After the new user' interactions are added the User Modeling, we can use Collaborative Filtering Recommendation algorithm to obtain more accurate recommendation results. A series of process, calculating similarity between the new user and existing users, predicting ratings for the new user, sorting the predicting ratings, will work as usual. The recommendation results of first-time recommendation process will be as seed candidates to create the final more accurate recommendation results. Therefore we can name the new solution to Cold Start problem as two-times recommendation. Using the new content-based tags extracting composite recommendation algorithm, Collaborative Filtering recommendation algorithm can provide better service for new users and improve the performance of recommender system.

3.2 Experimental Results and Analysis

In order to verify the effectiveness of the algorithm, this paper will use the public data set MoviLens provided by Grouplens working group to test. MoviLens is a research recommender system developed by the researchers of Grouplens working group in USA University of Minnesota based Web. It accepts user evaluation of the films and provides the corresponding movie recommendation list. At present, the system has more than 43000 users and the user ratings of the items are more than 1600 [8].

In this paper, the ML data set provided by the Movielens will be used, which is composed of 943 user evaluations of the 10000 items with 1-to-5 ratings. The data set has a total of 1682 items and each user makes evaluation on the 20 items at least [8]. In this paper we will select 10% of the data set as the experimental data randomly. Then we will choose 10% users data of the experimental data as new users' information. Their rating items are more than 100 so that enough information can be used. In this experiment, the new users' ratings will be deleted and stored in another place. Using the three different recommender algorithms of Random Recommendation, Mean Value Recommendation and the new recommendation algorithm proposed in this paper, compare the predict ratings with real ratings of the new users. This experiment process will be carried out 9 times totally. The accuracy of recommendation algorithm usually uses MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error) to measure. They express that what degree the new users will like or dislike the items recommended by recommendation algorithm.

$$MAE = \frac{\sum_{i\in N} p_{i,i} - r_{i,j}}{N} \tag{5}$$

Where $p_{i,j}$ is on behalf of the predicted values. In fact, MAE is the average value of the sum of differences between predicted values and real ratings values. The lower the MAE value, the better the recommendation accuracy. Fig.1 shows the MAE curve of three recommendation algorithms that solve Cold Start problem. The recommendation results of Random Recommend are random and the MAE curve is a random fluctuating curve that the value is more than 0.6. As the ratings from new user are more, the accuracy of Random Recommend is much bad. The MAE of Mean Value Recommend is almost a fixed value. The MAE of the new recommendation algorithm is a low value firstly. When ratings from new user are added, the MAE becomes much lower and the accuracy of new recommendation is best of all.

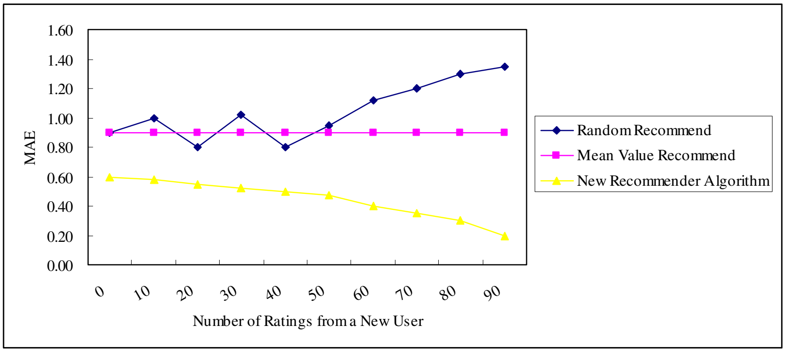


Fig. 1. The MAE Curve of Three Solutions to Cold Start

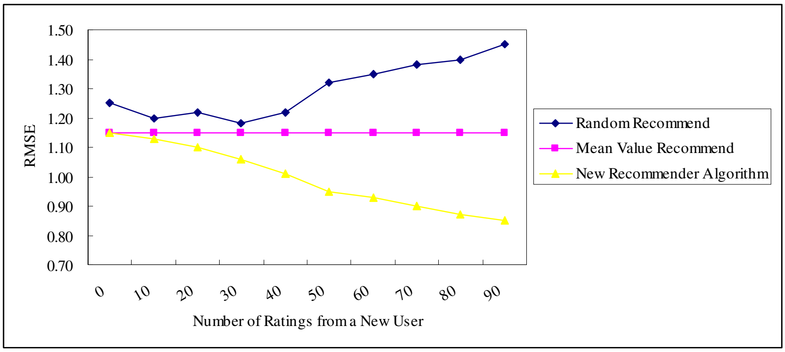


Fig. 2. The RMSE Curve of Three Solutions to Cold Start

RMES in formula (6) is another metric to measure accuracy of a recommender system. MAE is widely used because of its simple calculation, easy to understand. But MAE also has some limitations, because the low ratings that are difficult to predict accurately usually make great contribution to the MAE metric. RMSE firstly square each absolute error, so it will have heavier punishment for the larger absolute error relatively [9]. In formula (6), N is the total number of experimental items and E^p is the entire experimental data set. The lower the RMSE value, the better the recommendation accuracy, like MAE. Fig.2 shows the curve of three recommendation algorithms that solve Cold Start problem. The curve changes of RMSE are almost unanimously with MAE, just the RMSE value is more precise. The two metrics of accuracy prove that the new recommendation algorithm in this paper is a feasible and effective recommendation algorithm to solve the problem of Cold Start.

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in E^p} \left(r_{u,i} - p_{u,i}\right)^2}{N}} \tag{6}$$

4 Conclusions

In this paper an enhanced composite algorithm based on content information and tags extracting is proposed to solve the issue of Cold Start to make Collaborative Filtering recommender system works more effectively. The final experiment results show that the new enhanced recommendation algorithm has some advantages on accuracy compared with several existing solutions to the problem of Cold Start and make sure that it is a feasible and effective recommendation algorithm. Therefore, we can adopt the new recommendation algorithm in e-Learning system to recommend appropriate lessons for learners. When new learners register e-Learning system, they can also acquire recommendation lessons which meet their demand. The new enhanced recommendation algorithm can be applied to the e-Learning system used in small and medium enterprises and if the scale of e-Learning system is too large, tags extracting workload will be correspondingly become too volume.

Acknowledgment. This work was supported by the Science and Technology Project of Education Department of Liaoning Province, China (Research of e-Learning System of small and medium-size Enterprises Based on SaaS, No.L2013417).

References

1. Goldberg, D., Nichols, D., Oki, B.M.: Using collaborative filtering to weave an information tapestry. Communications of the ACM 35(12), 145–147 (1992)
2. Lei, R.: The Key Technology Research of Recommender System. East China Normal University (2012)
3. Dietmar, J., Markus, Z., Alexander, F., Gerhard, F.: Recommender System an Introduction, pp.13–14, 19. Cambridge University Press (2011)
4. Liu, Q., Gao, Y., Peng, Z.: A novel collaborative filtering algorithm based on social network. In: Tan, Y., Shi, Y., Ji, Z. (eds.) ICSI 2012, Part II. LNCS, vol. 7332, pp. 164–174. Springer, Heidelberg (2012)
5. Dongting, S., Tao, H., Fuhai, Z.: Summary for Research on the Cold Start Problem in Recommender Systems. Computer and Modernization 5, 59–62 (2012)
6. Haralambos, M., Dmitry, B.: Algorithms of the Intelligent Web, pp.100–101. Publishing House of Electronics Industry (2011)
7. Satnam, A.: Collective Intelligence in Action, pp. 349–350. Manning Publications (2009)
8. Yanhong, G.: Hybrid Recommendation Algorithm of Collaborative Filtering Cold Start Problem of New Items. Computer Engineering 34(23), 11–13 (2008)
9. Yuxiao, Z.: Summary for Evaluating Indicator of Recommender System. Journal of University of Electronic Science and Technology of China 41(2), 163–172 (2012)

A Method to Construct a Chinese-Swedish Dictionary via English Based on Comparable Corpora

Fang Li, Guangda Shi, and Yawei Lv

Intellingence Engineering Lab, Beijing University of Chemical Technology,
Beijing 100029, China
lifang@mail.buct.edu.cn, shiguangdabuct@163.com,
lvyawei7@gmail.com

Abstract. Taking advantage of existing bilingual dictionaries and a third language can construct a new bilingual dictionary. However, the access of some existing bilingual dictionaries is difficult, which makes it impossible to construct a new bilingual dictionary. For solving the problem, the paper proposes a method that we can construct the bilingual dictionaries of the source language and the target language with a third language by using the corresponding comparable corpora, respectively. The proposed method is applicable to the languages which are lack of comparable corpora and the available dictionaries, such as Chinese and Swedish. This paper constructs a Chinese-Swedish bilingual dictionary by using English as a third language to prove the validity of the proposed method. The result of experiment shows that the proposed method has a good performance in the construction of the Chinese-Swedish bilingual dictionary.

Keywords: bilingual dictionary, intermediary third language, comparable corpora.

1 Introduction

With the rapid development of natural language processing such as machine translation and cross-language information retrieval, a bilingual dictionary plays a more and more important role due to the growing demand for the bilingual dictionary. In the process of constructing bilingual dictionary, especially for the languages which are not widely available, the existing bilingual dictionaries via a third language (usually English) are utilized as the usual method. But there is a problem in this process, which is difficult or even impossible to get the existing bilingual dictionaries for less common languages such as Swedish. For the problem, this paper proposes a new method of constructing a dictionary with the help of the comparable corpora.

The goal of the experiment is to construct Chinese-Swedish bilingual dictionary by using Chinese-English and English-Swedish comparable corpora to construct Chinese-English and English-Swedish dictionary, eventually we get a Chinese-Swedish bilingual dictionary.

The remainder of this paper is structured as follows: Section 2 describes the related work of the bilingual dictionary construction. Section 3 proposes a new method in detail to solve the problem of the lack of dictionary when constructing

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 196–203, 2014.
© Springer International Publishing Switzerland 2014

Chinese-Swedish dictionary by using English as a third language. Section 4 provides our experiment and analysis. The last section is a summary of this paper and prospects.

2 Related Work of Bilingual Dictionary Construction

Tanaka and Umemura[1] built a Japanese-French bilingual dictionary by using English as third language intermediary. They made use of the existing Japanese-English bilingual dictionary and the English-French bilingual dictionary to construct Japanese-French bilingual dictionary. At the beginning, they searched the English translation candidates of Japanese words by looking up the Japanese-English bilingual dictionary, and found the French translation candidates of the words in English-French bilingual dictionary. Similarly, the English translation candidates of French translation candidates were available by looking up the French-English bilingual dictionary. After that, they got the French translation candidates of Japanese words by comparing the English translation candidates of Japanese words with the English translation candidates of French words. Bond and other scholars[2] took advantage of the Japanese-English bilingual dictionary and the English-Malay bilingual dictionary to construct Japanese-Malay bilingual dictionary. They used a method called semantic classification to sort final candidate translations.

The construction of another bilingual dictionary using comparable corpora is based on the following hypothesis: if two words have similar meaning, the contexts of the two words should also be similar. That's to say, the words in source language and its translations in target language have the similar contexts. Fung (1998)[3] extracted the contexts of the words in the comparable corpora, and then the similarity between two words is calculated by using co-occurrence vector of the words, and constructed a bilingual dictionary. Zhang Yongchen and other scholars captured the datas on the web and built a Chinese-English bilingual dictionary by using Vector Space Model. Haghighi[4] used the Matching Canonical Correlation Analysis to construct a Chinese-English bilingual dictionary. Rapp[5] considered the sequence and relationship of words based on Vector Space Model when constructing a English-German bilingual dictionary. Daille and Morin[6] used the variant methods of the Vector Space Model to construct a French-English bilingual dictionary. About the construction of bilingual dictionary, a lot of scholars did a plenty of researches, involved a lot of languages. However, the research about the dictionary construction of Chinese-Swedish is seldom seen.

3 Chinese-Swedish Dictionary Construction via English

This section detailedly describes how to extract the words of the source language and the target language to construct a bilingual dictionary when we do not have the dictionaries of the source language and the target language with a third language.

As you know, Chinese-Swedish comparable corpora are different to obtain. Our method is to construct a Chinese-English dictionary and an English-Swedish dictionary using the corresponding corpus respectively.

3.1 The Construction of Chinese-English Dictionary and English-Swedish Dictionary

In the process of constructing the Chinese-English dictionary, we make use of a method called the standard method, the technological process can be seen in Figure 1. This detailed process of the method is as follows:

1. Preprocessing. Word segmentation and filtering stop words should be done in this process. As to the preprocessing of English words, stemming and filtering stop words should be used[7].
2. Extracting the contexts. We select 3 as our window size and extract the contexts of the words in the range of window size.
3. The construction of the context space vector. We make use of the bag-of-words to create the context vectors and use the TF-IDF (term frequency-inverse document frequency) of each word to measure the importance of each word. They are calculated as follows:

$$TF_{w_i} = \frac{n_{w_i}}{N} \tag{1}$$

$$IDF_{W_i} = \log\frac{K}{k+1} \tag{2}$$

$$TF - IDF = TF_{w_i} * IDF_{W_i} \tag{3}$$

In the formulas above, w_i represents a word. n_{wi} represents the frequency of the word w_i in the documents. N represents the number of the same words in the documents. K represents the number of the documents. k represents the number of documents which contain the word w_i.

4. Calculating similarity. We select the Cosine Similarity[8] method to calculate the similarity between Chinese-English word vectors. The highest similarity words in the target language are selected as the translation candidates for words in the source language.

$$Cos(W_s, W_t) = \frac{\sum_{i=1}^{t}(w_{is} \times w_{it})}{\sqrt{\sum_{i=1}^{t} w_{is}^2 \times \sum_{i=1}^{t} w_{it}^2}} \tag{4}$$

W_S and W_t represent a source word and a target word respectively.

Analogously, the process of constructing English-Swedish bilingual dictionary is similar to the process of constructing Chinese-English bilingual dictionary.

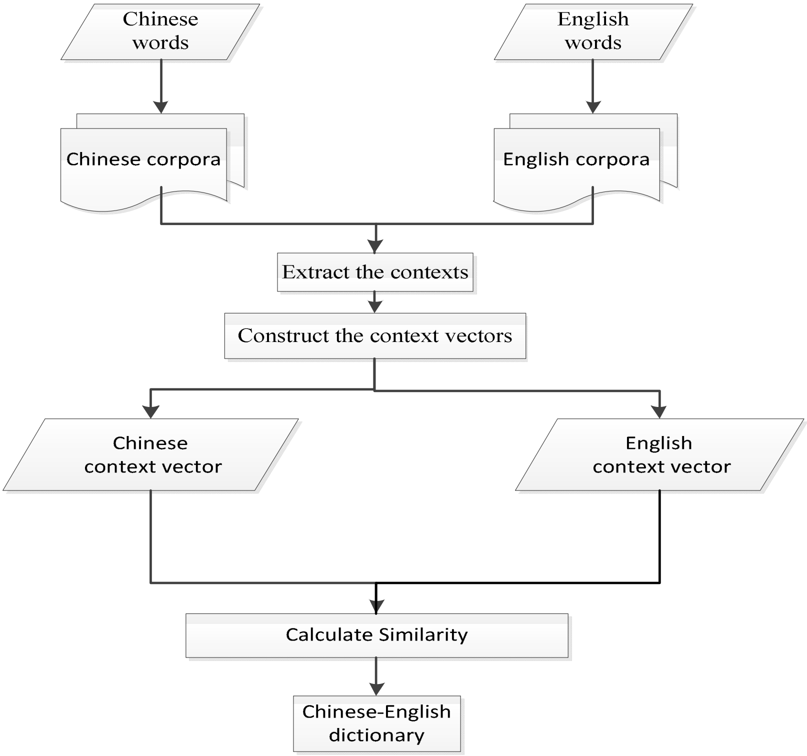


Fig. 1. The method of the Chinese-English bilingual dictionary

3.2 The Construction of Chinese-Swedish Dictionary

The process of constructing Chinese-Swedish can be seen in Figure 2.

We construct Chinese-Swedish bilingual dictionary by extracting words in the Chinese-English bilingual dictionary and the English-Swedish bilingual dictionary that we get in section 3.1.

We look up English translations for Chinese words, and then look for Swedish translations of these English translations. Then, for each Swedish word, we look up all of its English translations. After that, we count the number of shared English translations. If they have the common English translations, they are treated as the translation pairs.

We use a similarity score *S* for a Chinese word *c* and a Swedish word *s* is given in Equation (5), where *E(w)* is the set of English translations of *w*.

$$S(c,s) = \frac{2 * |E(c) \cap E(s)|}{|E(c)| + |E(s)|} \tag{5}$$

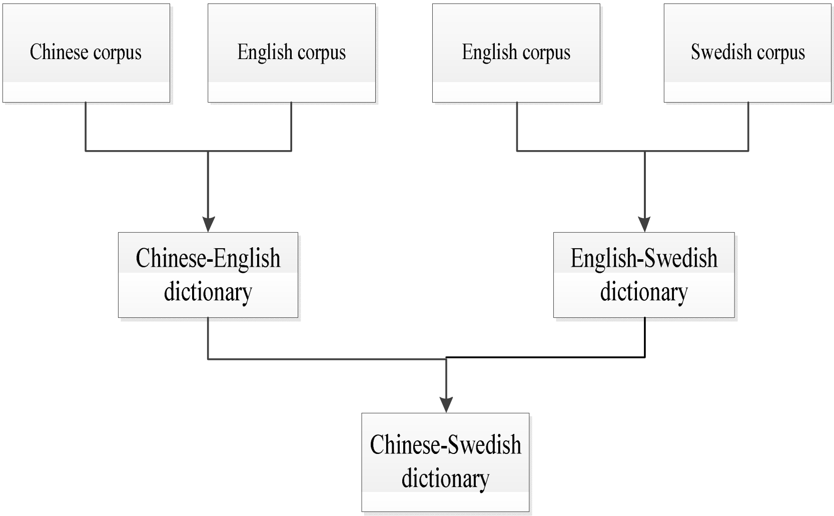

Chinese corpus
English corpus
English corpus
Swedish corpus
Chinese-English dictionary
English-Swedish dictionary
Chinese-Swedish dictionary

Fig. 2. The method of the Chinese-Swedish bilingual dictionary

4 Experiments

This section mainly describes the evaluation criteria of the related work and experimental results and analysis.

The paper uses the Chinese-English comparable corpora that are the XinHua news about finance in the Gigawords and the English-Swedish comparable corpora that are derived from Wikisource.

4.1 Evaluation Criterion

We use the precision and the Mean Reciprocal Rank (MRR) as the evaluation criteria. In the process of the bilingual word of construction, the precision of the evaluation criteria is often used and it is the average accuracy of the top n translation candidates. The Mean Reciprocal Rank (MRR) is that the mean reciprocal of the rank of the correct translation. It is a measure of the ranking of the correct translation in the translation candidates.

The Mean Reciprocal Rank (MRR) is a more lenient evaluation criteria than the precision. The precision and the Mean Reciprocal Rank(MRR) are calculated as follows:

$$Precision = \frac{count_{top5}}{N} \tag{6}$$

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{7}$$

$Count_{top5}$ represents the total numbers of the top 5 translation candidates, $rank_i$ represents the ranking of the correct translation, N represents the numbers of the word pairs. Different from the precision, the Mean Reciprocal Rank (MRR) is not considering the number n of the word translation candidates; therefore, it can better measure the performance of the bilingual dictionary construction.

4.2 Results and Analysis

Table 1 shows how the link is realized and the similarity scores in section 3. The similarity score shows how many English words are shared by the two dictionaries. The higher the score, the higher the probability of successful linking. As Table 1 shows, we can see that, if the number of shared English translated words is more, then we get a higher possibility of accurate matching of Chinese and Swedish. However, the accuracy reduces when the number of the shared English translated words decreases. Sometimes we have to inappropriately sort out the matched pairs such as 首都 (kapital). The reason is that a lot of intermediate words are polysemous such as capital.

Table 1. Example of linking through English translations

Score	Chinese→English	Swedish→English	Chinese-Swedish	
2	支票(check, cheque)	Check(check, cheque)	支票(check)	Yes
2	现金(cash, money)	Kassa(cash, money)	现金(kassa)	Yes
1	资本(capital)	Kapital(capital)	资本(kapital)	Yes
1	费用(cost, expense)	Kostnad (cost)	费用(kostnad)	Yes
1	首都(capital)	Kapital (capital)	首都(kapital)	No

Table 2 shows the experiment results of the method to solve the problem of the lack of dictionary when constructing Chinese-Swedish dictionary by using English as a third-party language. We evaluate the precision of the top 5 translation candidates.

As Table 2 shows, we totally extract 17368 translation pairs from the Chinese-English comparable corpora, and the precision is 54.38%, the Mean Reciprocal Rank (MRR) is 58.13%, moreover, there are 94 translation pairs that are not quite accurate, but they are accepted. We get 14273 translation pairs in total, and the precision is 51.26%, the Mean Reciprocal Rank (MRR) is 56.34%, moreover, there are 137 translation pairs that can be accepted. Finally, we have a Chinese-Swedish dictionary that contain 7369 translation pairs, its precision is 46.17%, the Mean Reciprocal Rank (MRR) is 58.26% and there are 63 translation pairs accepted.

Table 2. The Experiment Results

	Translated	Precision	MRR	Accepted
Chinese-English	17368	54.38%	58.13%	94
English-Swedish	14273	51.26%	56.34%	137
Chinese-Swedish	7369	46.17%	51.26%	63

5 Conclusion and Future Work

As a conclusion, we proposes a new method in detail to solve the problem of the lack of dictionary when constructing Chinese-Swedish dictionary by using English as a third language, and this method is applicable to other language resources. We make use of the public available resources Gigawords and Wikisource for the construction of a new language pair. The process is divided into three parts, the first part is constructing Chinese-English bilingual dictionary by using the Chinese-English comparable corpora, and the second part is constructing English-Swedish dictionary by using the English-Swedish comparable corpora, finally we can get the Chinese - Swedish dictionary based on the two above dictionary. The accuracy obtained is 46.17%， which proves the effectiveness of the proposed method when lack of the dictionary between the source language and the third language or between the target language and third language.

The current study about the Chinese-Swedish bilingual dictionary construction is relatively few, during the construction of Chinese-Swedish bilingual dictionary process, the proposed method proves its feasibility and effectiveness, and this method is also applicable to other language resources, which has an important contribution to the study of related work.

However, there is still a lot of work to do. As future work, firstly, we plan to compare different definitions of context in the process of constructing the context vector, such sentence-based context and syntax-based context. Secondly, we plan to conduct experiments on other comparable corpora such as Wikipedia and different language pairs. Finally, we plan to extend our method to deal with some compound and rare words.

Acknowledgments. This research is supported by the Chinese-English comparable corpora of Gigawords and the English-Swedish comparable corpora of Wikisource. We would also like to thank everyone who offers help.

References

1. Tanaka, K., Umemura, K.: Construction of a bilingual dictionary intermediated by a third language. In: 15th COLING International Conference on Computational Linguistics, pp. 297–303. ACL, Stroudsburg (1994)

2. Bond, F., Sulong, R., Yamazaki, T., Ogura, K.: Design and construction of a machine-tractable Japanese-Malay dictionary. In: 8th MT Summit, pp. 53–58. Santiago de Compostela (2001)
3. Fung, P.: Compiling bilingual lexicon entries from a non-parallel English-Chinese corpus. In: 3rd Annual Workshop on Very Large Corpora, Boston, pp. 173–183 (1995)
4. Haghighi, A., Liang, P., Berg-Kirkpatrick, T.: Learning Bilingual Lexicons from Monolingual Corpora. In: the Association for Computational Linguistics on Computational Linguistics, pp. 771–779. ACL, Stroudsburg (2008)
5. Rapp, R.: Automatic identification of word translations from unrelated English and German corpora. In: 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics. Association for Computational Linguistics, pp. 519–526. ACL, Stroudsburg (1999)
6. Daille, B., Morin, E.: French-English Terminology Extraction from Comparable Corpora. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 707–718. Springer, Heidelberg (2005)
7. Fung, P.: A Statistical View on Bilingual Lexicon Extraction: From Parallel Corpora to Non-parallel Corpora. In: Farwell, D., Gerber, L., Hovy, E. (eds.) AMTA 1998. LNCS (LNAI), vol. 1529, pp. 1–17. Springer, Heidelberg (1998)
8. Kaji, H., Erdenebat, D.: Automatic Construction of a Japanese-Chinese Dictionary via English. In: the International Conference on Language Resources and Evaluation, Marrakech, pp. 699–706 (2008)

The Design and Implementation of the Random HTML Tags and Attributes-Based XSS Defence System

Heng Lin, Yiwen Yan, Hongfei Cai, and Wei Zhang

School of Software, Beijing Institute of Technology, Beijing 100081 China
anonymous.joker.lin@gmail.com,
278567461@qq.com

Abstract. At present, cross site scripting (XSS) is still one of the biggest threat for Internet security. But the defensive approach is still feature matching mostly; that is, to check for a matching and filter in all information submitted. However, filtering technology has many disadvantages as heavy-workload, complex-operation, high-risk and so on. For this reason, our system use the randomization techniques of HTML tags and attributes innovatively, based on the prefix of HTML tags and attributes, to determine the tags and attributes are Web designers expect to generate or other users insert in, and then we follow the results to carry out different policies, only tags and attributes that Web designers expected to generate can be rendered and implemented. By this way, we can defend against XSS attacks completely. The test results show that the system is able to solve a variety of problems in filtering technology. It uses simple and convenient operation and safe and secure effect to free developers from heavy filtering work. System has a good compatibility and portability across platforms, it also can connect with all web-based applications seamlessly. In all, system defend against XSS better and meet the need of today's XSS attacks defence.

Keywords: random, tag prefix, cross-site scripting, defence system.

1 Project Background

Cross Site Script Execution (usually abbreviated as XSS) is a kind of attack that attacker using the lack of filtering on the user's input, manufacturing malicious input which can affect other users, so as to achieve the purpose of attack, stealing user data, using user identity to make some things or using virus to attack the visitors.

In recent years, XSS vulnerability has been ranked in the top three Web security .According to the data published by OWASP(Open Web Application Security Project) in 2010 and 2013,in the top ten Web security vulnerabilities, XSS is both ranked the top three[1].

In November 2012, Anheng Institute for information security found XSS vulnerability exist in the Web applications of Baidu, Tencent, Sina and other Internet companies[2]; In April 2013,a fact was exposed by WooYun that Taobao had Cross Site Vulnerabilities [3];In June 2013, WooYun reported again that the main Web site

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 204–211, 2014.
© Springer International Publishing Switzerland 2014

of Tencent and Baidu had XSS vulnerability [4-5]. XSS vulnerability has been a serious threat to the information security of majority of Internet users.

To prevent from XSS attack, someone present model-based testing evolved as one of the methodologies which offer several theoretical and practical approaches in testing the system under test (SUT) that combine several input generation strategies like mutation testing, using of concrete and symbolic execution etc. by putting the emphasis on specification of the model of an application [6]; someone detecting Cross-Site Scripting Vulnerability using Concolic Testing[7]. Existing cross-site scripting attack protection mainly contains four aspects: the input validation, escape, filtering, and character set specifies, there are Http-Only, input check, output check, three kinds of defense in total. But the reality is that each method has its own defensive shortcomings, it's also very hard to handle the text properly.

So we develop XSS-Defender, a system that allows the server to identify untrusted content and reliably convey this information to the client, and that allows the client to enforce a security policy on the untrusted content. Analogously, XSS-Defender randomizes HTML tags and attributes to identify and defeat injected malicious web content. These randomized tags and attributes serve two purposes. First, client proxy distinguish a tag or attribute is legal or not through whether the tag or attribute has a random prefix client proxy generated so that identify untrusted content. Second, they prevent the untrusted content from distorting the document tree.

We make the following contributions:

- We innovatively develop a system randomizing HTML tags and attributes to defend against XSS attacks.
- We lead a simple way for current web application to defend against popular XSS attack efficiently.

2 Systematic Realization

2.1 Processing of Client Proxy for the User Data

When a user opens a client proxy, the client proxy automatically connect to the server proxy, to get the public key that server proxy stored in the cache, after that, the public key will use as a basis of the RSA encryption algorithm to transfer randomized prefix. When a user of Web site requests a Web page or submit data to the server, first of all, the user send data to the client proxy that is deployed on the user's computer. Via the client proxy, system forwards the request to the server proxy that is deployed on the server. Meanwhile, the client proxy use Python random number generator to generate a random string as the identification of random prefix, after a non-equivalent encryption, it will be sent to server proxy with user's requests together. The random string is effective only at this time of page request and getting response to server, it will be destroyed at once after using it one time, in order to ensure the security of the system.

2.2 Processing of Server Proxy for the User Data

As shown in Fig. 1, when the server proxy starts, it generates a pair of public and private keys used for RSA encryption. When the server proxy receives requests or data, it detects the requested type first, if the request is a request of RSA public key, proxy will send the generated public key to client proxy. If the request is other networks' interactive request, proxy will judge whether it has a random prefix the client proxy generates or not. If not, the request or data will be discarded; if so, it will forward requests or submitted data to the Web server, then the one-time random prefix the client proxy generates will be stored in the cache.

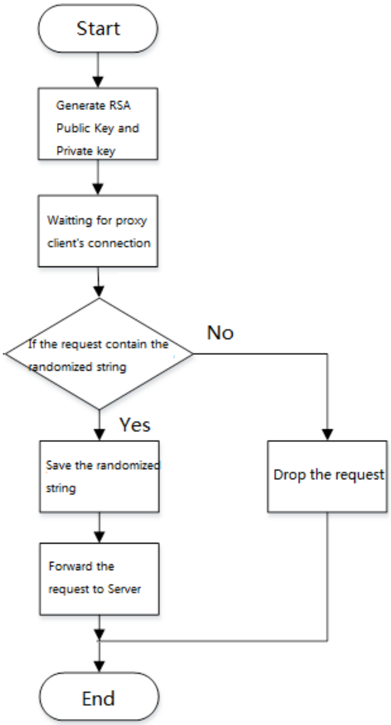

Start
Generate RSA Public Key and Private key
Waitting for proxy client's connection
If the request contain the randomized string
No
Yes
Save the randomized string
Drop the request
Forward the request to Server
End

Fig. 1. Processing flow chart of server proxy for the user data

2.3 Processing of Web Server for the User Data

In the system, the Web server pre-stores a confidential 16-bit string which can be modified at any time as a tag prefix to prove that the tags or attributes come from the server. In a static page stored on the server, each HTML tag and attribute will have the confidential prefix, and in the code of Web page dynamically generated by the server, each HTML tags and attributes generated within the plan of Web site developer (legal) will have the confidential prefix, tags and attributes not in the developers' plan (illegal) does not have this prefix. Confidential prefix only transfer across the local network connection between the server proxy and server which are deployed on the same machine, in addition to the site maintainers, others all could not get the confidential prefix.

When receiving a request, the Web server dynamically generates or read the user requests' website from the cache. In this page, all legal tags and attributes have the confidential prefixes, meanwhile illegal tags and attribute does not have the confidential prefixes. In this way, the server generates a page that is able to distinguish whether each tab comes from the server or not. Then, the server sends this page to the server proxy through a local connection inside the computer.

2.4 Processing of Server Proxy for the Page Sent to Users

After receiving the Web server's response to the request, server proxy will replace all confidential prefixes in the code of Web page with confidential prefix the client proxy generates. Then, the server proxy sends modified code of Web page to the client proxy. As the prefix for the Web site's users cannot be revealed to other users, so that the replaced page still can distinguish this tag is legal or not by the tag or attribute whether the client has proxy generated random prefix. This way, the client proxy can classify and deal with them according to the different tag prefix.

2.5 Processing of Client Proxy for the Page Sent to Users

When the client proxy receives the treated page sent by server proxy, first of all, it extracts all tag and attribute information in the code of Web pages. Then, it can classify according to the different tag and attribute prefix. (Process as Fig. 2).

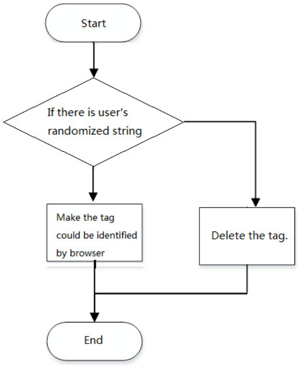

Start
If there is user's randomized string
Make the tag could be identified by browser
Delete the tag.
End

Fig. 2. Processing flow chart of client proxy for each tag in the code of Web pages

Client proxy distinguish a tag or attribute is legal or not through whether the tag or attribute has a random prefix client proxy generated. According to the result of judgments, client proxy deletes the illegal tags; but deletes the random prefix of legal tags, transforms tags into a format that can be rendered and implemented by browsers. After the sorting, the client proxy will send the last generated and transformed page to the user's browser. In this way, legal tags and attributes can be rendered and implemented on the user's browser, while illegal tags and attributes cannot be rendered and implemented on the user's browser. To avoid the browser executing malicious code submitted by the attackers efficiently, and thus system can prevent XSS occurred.

3 The Performance of Testing and Experiment

Through a large number of comprehensive testing of the system, the testing is able to detect the operations whether efficient or stable. To achieve protection against XSS attacks, we get the performance of the each module and the overall operational status of the system.

Test content is divided into system functional testing and system performance testing in two parts.

3.1 System Functional Testing

System Resources

Examine the CPU and memory usage. Turn on the system, using the Windows Task Manager to see the CPU and memory usage. See the memory usage of server and the clients which run XSS-Defender and detect the whether exist the presence of a memory leak.

Response Speed Test

Statistics the user's access speed, page load time when using XSS-Defender system and using XSS-Defender system. Then we can conclude the comparisons of running the system which affect the ability of the server's response and effect.

Server Performance Impact

Specific test operation to record the XSS-Defender server for each user request response time is calculated for each user request processing response speed; recording XSS-Defender client browser page request to the server's response time is calculated browser page request response speed.

3.2 The Results of System Testing

Functional Test for XSS Protection

We designed four different methods to attack by XSS.

Table 1. Four pieces of script code

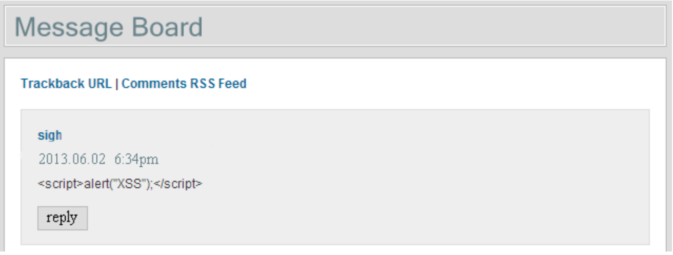
No.	script code
1	`><script>alert(document.cookie)</script>`
2	`<img scr=″javascript: alert(/XSS/)></img>`
3	`<div style={left:\0065\0078ression (alert(’xss’))}></div>`
4	`<style/style″ STYLE=″background:expre><ssion(alert(>″\\<XSS=″));″>`

We established a blog site. Then we inject four script codes into the blog message board.

Based on the experimental results, the system makes the harmful input invalid, so it defends against the several of XSS attacks efficiently.

Function Test that Submission does not Affect the Contents of the User's Normal Display

Inject Code:
<script>alert(“XSS”);</script>
The browser of client shows:



Message Board

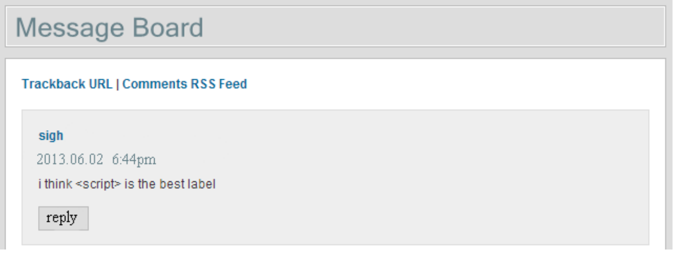
Trackback URL | Comments RSS Feed

sigh
2013.06.02 6:34pm
<script>alert("XSS");</script>

Fig. 3. The result of injecting XSS code

The injected code is displayed as text and not executed properly in the site content:
i think <script> is the best label

The browser of client shows:



Message Board

Trackback URL | Comments RSS Feed

sigh
2013.06.02 6:44pm
i think <script> is the best label

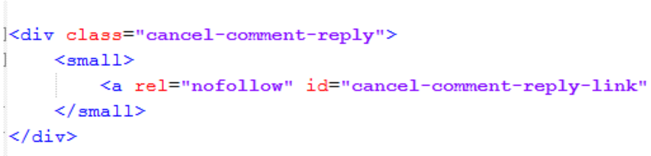
Fig. 4. Submit a normal script

Users fill out a script tag containing the ‘script’ contents and the browser is normally displayed.

Based on the above experimental results, the system can tell the difference between legal and illegal entry.

The Function Test of Adding the Prefix on the Service Side of the Page File

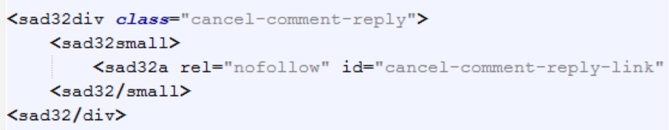
Source page on the server part:



```
<div class="cancel-comment-reply">
    <small>
        <a rel="nofollow" id="cancel-comment-reply-link"
    </small>
</div>
```

Fig. 5. Source code

The client sends a request to obtain the corresponding part of the html file:

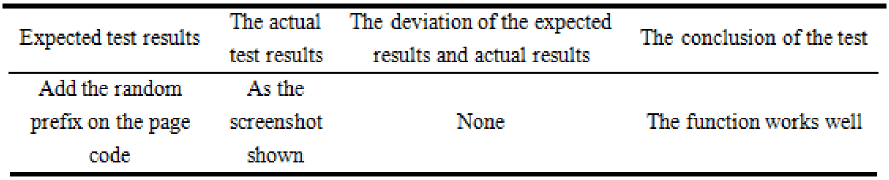


```
<sad32div class="cancel-comment-reply">
    <sad32small>
        <sad32a rel="nofollow" id="cancel-comment-reply-link"
    <sad32/small>
<sad32/div>
```

Fig. 6. The code received

All tags add prefix ‘sad 32’.

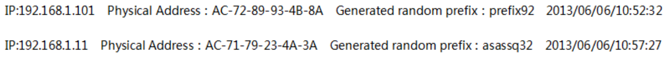
Table 2. The function test of adding the prefix on the service side of the page file



Expected test results	The actual test results	The deviation of the expected results and actual results	The conclusion of the test
Add the random prefix on the page code	As the screenshot shown	None	The function works well

Test of Log Function

Try to request the server, the log on the server side shows as follows:



```
IP:192.168.1.101   Physical Address : AC-72-89-93-4B-8A   Generated random prefix : prefix92   2013/06/06/10:52:32
IP:192.168.1.11   Physical Address : AC-71-79-23-4A-3A   Generated random prefix : asassq32   2013/06/06/10:57:27
```

Fig. 7. Test of log function

Logs completely records every request and its generated random prefix.

Function Test of Cross-Platform

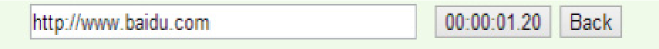
The use of Python development in Linux, Mac OS X, Windows under all operating normally, the client implementation for the client agent, it can be supported by all browsers.

Table 3. Function test of adding prefix on the server side

Expected test results	The actual test results	The deviation of the expected results and actual results	The conclusion of the test
Log records	As the screenshot shown	None	No problem found

Performance Test on the Client Side

When the client program is not running, the s peed of opening a web page:

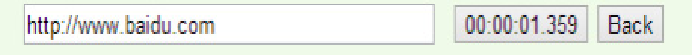

http://www.baidu.com

00:00:01.20

Fig. 8. The speed of opening a web page

When client program is not running.

After running the client program :


http://www.baidu.com

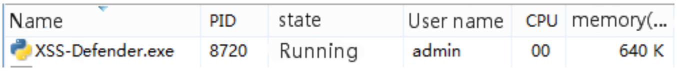
00:00:01.359

Fig. 9. The speed of opening a web page

When client program is running

The speed is almost unchanged.

When running the client, open the Task Manager, we can see that:



Name	PID	state	User name	CPU	memory(...
XSS-Defender.exe	8720	Running	admin	00	640 K

Fig. 10. The resources occupancy after running the client system

Based on the above experimental results, the system occupy very small CPU and memory.

References

1. OWASP: OWASP Top- 2013 10 rcl The Ten Most Critical Web Application Security Risks (2013)
2. eNet, `http://www.enet.com.cn/article/2012/1112/A20121112190987.shtml`
3. WooYun, `http://www.wooyun.org/bugs/wooyun-2010-022080`
4. WooYun, `http://www.wooyun.org/bugs/wooyun-2010-025030`
5. WooYun, `http://www.wooyun.org/bugs/wooyun-2010-025002`
6. Top 25 most dangerous software errors, `http://cwe.mitre.org/top25/.CWE/SANS`
7. Bozic, J., Wotawa, F.: XSS Pattern for Attack Modeling in Testing. In: 8th International Workshop on Automation of Software Test (AST), pp. 71–74. IEEE (2013)

DWT and GA-PSO Based Novel Watermarking for Videos Using Audio Watermark

Puja Agrawal[1] and Aleefia Khurshid[2]

[1] Department of Electronics and Communication Engineering,
Ramdeobaba College of Engineering and Mangement, Nagpur, India
[2] Department of Electronics Engineering,
Ramdeobaba College of Engineering and Mangement, Nagpur, India
{agrawalps,khurshidaa}@rknec.edu

Abstract. This paper presents a digital video watermarking scheme that can embed invisible and robust watermark information into the video streams of MPEG-1, MPEG-2, H.264/AVC, MPEG-4 standards. Watermark embedding process is in Discrete Wavelet Domain. Trade off between transparency and robustness is considered as optimization problem and is solved by Genetic Algorithm - Particle Swarm Optimization (GA-PSO) based hybrid optimization technique. An audio signal is converted into 9 bit planes by using bit plane slicing and then embedded into the frames of video signals as watermark. The performance evaluation results based on the Peak Signal to Noise Ratio (PSNR) and Normalized Correlation (NC) confirm that the proposed video processing method shows reliable improvements for various sequences compared to existing ones for geometrical attacks like rotation and cropping.

Keywords: Watermarking, Robustness, Transparency, GA-PSO, Bit Plane Slicing.

1 Introduction

Protection of multimedia data has become one of the major challenges due to the rapid growth of unauthorized access and copy of digital media objects like images, audio and video. Digital Watermarking is a process where some valuable information is embedded into the host media like images, video and audio etc. The secret message embedded as watermark can be almost anything, for example: a serial number, plain text, image, random signal, an organization's trademark, or a copyright message for copy control and authentication. Potential applications of digital watermarking includes, copy control, transaction tracking, authentication, and legacy system.

In general, digital watermarking involves two major operations: (i) Watermark embedding, and (ii) Watermark extraction. The two most important properties viz. robustness and transparency are required for preserving the security of videos from unauthorized access. The ability to detect the watermark content after application of common signal processing distortions like filtering, lossy compression, color

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 212–220, 2014.
© Springer International Publishing Switzerland 2014

correction, noise, contrast distortions, geometric distortions is known as robustness. Imperceptibility/Transparency means that the presence of watermark is not noticed by the human eyes. Watermarking techniques can be classified according to the nature of host data (text, image, audio or video), or according to the working (spatial or frequency) domain.

1.1 Literature Review

Most of the proposed video watermarking scheme based on the techniques of the image watermarking and applied to raw video or the compressed video. As some issues in video watermarking are not present in image watermarking, such as video object and redundancy of the large amount video data, researchers have made use of those characteristics to develop different schemes [1].

Hui-Yu Huang et.al. have presented an approach consists of a pseudo-3-D DCT. The watermark message represents an index for selection of a particular quantizer from a set of possible quantizers. The selected quantizer is applied to the host data to encode the watermark message [1]. This is invisible, robust for Raw Videos, but complex process for implementation.

Jing Zhang et.al have described that a grayscale watermark pattern is first modified to accommodate the H.264/AVC computational constraints, and then embedded into video data in the compressed domain. With the proposed method, the video watermarking scheme can achieve high robustness and good visual quality without increasing the overall bit-rate [2].

Gwenael Doerr et.al have given the in depth overview of video watermarking and have pointed out that video watermarking is not just a simple extension of still image watermarking[3].

Ersin ELBASI [4]¸ has proposed a novel video watermarking system based on the Hidden Markov Model (HMM). This novel watermarking scheme splits the video sequences into a Group of Pictures (GOP) with HMM. Portions of the binary watermark are embedded into each GOP with a wavelet domain watermarking algorithm.

Sanjoy Deb Roy et.al. have presented a hardware implementation of a digital watermarking system that can insert invisible, semi fragile watermark information into compressed video streams in real time [5].

Sourav Bhattacharya et.al. have presented a survey on different video watermarking techniques and comparative analysis with reference to H.264/AVC [6].

Salva A. K. Mostafa et. al. have presented a video watermarking scheme based on principal component analysis and wavelet transform [7].

Chuen-Ching Wang et. al. have presented a simple but effective digital watermarking scheme utilizing a context adaptive variable length coding method for wireless communication systems [8].

In this paper, we have presented an efficient video watermarking technique using discrete wavelet transform and GA-PSO based hybrid optimization to protect the copyright of digital videos. In the watermark, first of all the input video is segmented into shots. The frames of all the shots are decomposed into four sub-bands as HH, HL, LH and LL. Watermark is embedded into the high middle frequency HL and LH sub-bands, where acceptable performance of imperceptibility and robustness could be achieved. An audio signal is chosen as watermark which is unusual. Before embedding into the sub-bands, the watermark audio signature is processed into a 9 bit plane slices. Then it is embedded into HL sub-band, and LH sub-band. Here, every audio bit is embedded into the chosen sub-bands with the aid of our proposed embedding process. Subsequently, the watermark audio bits are extracted with the help of our proposed extraction process. Powerful GA-PSO optimization guarantees the performance. Since, the watermarking is performed in the wavelet domain; the attained watermark image is of good quality. The efficiency of our proposed watermarking technique is proved by good PSNR and NC values obtained for the watermarked videos in the experimental results.

2 Watermark Embedding and Extraction

This process is the most important in this scheme where all the different parts of watermarks are embedded into different scenes of the video. Video Preprocessing and Embedding is done by changing position of some DWT coefficients with the following condition:

2.1 Selection of Best Embeddable Locations

Human visual system has a very strong error correction mechanism. An image contains lot of redundancies. Small changes made to an image remain undetected by the human eyes. On the other hand, it has been observed that if an effort is made to increase the invisibility of a watermark, then robustness of the scheme suffers and vice a versa. A compromise therefore has to be made in order to get an optimum system.

Wavelet based transforms gained popularity recently since the property of multi-resolution analysis that it provides [12]. The higher level sub bands are more significant than the lower level sub bands. They contain most of the energy coefficients, so embedding in higher level sub bands is providing more robustness. On the other hand lower level sub bands have minor energy coefficients so watermark in these sub bands are defenseless to attacks. The sub band LL is not suitable for embedding a watermark since it is a low frequency band that contains important information about an image and easily causes image distortions. Embedding a watermark in the diagonal sub band HH is also not suitable since the sub band can easily be eliminated, for example by lossy compression as it has minor energy coefficient. So the middle frequency sub bands LH and HL are the best choice for embedding [12].

There has been a considerable amount of research proposals on the applications of DWT in digital image and video watermarking systems by virtue of its excellent and exceptional properties mentioned above, but the scope of optimization in this area is tremendously less. An optimized DWT for digital image watermarking is capable of producing perceptual transparency and robustness among the watermarked and the extracted images [13].

Extending the above concept to videos with suitable modifications, we have proposed the DWT and GA-PSO hybrid optimization based scheme for video watermarking.

2.2 Embedding

Input: Original video sequence: V_o [a,b] , watermark audio A_w [a,b]
Output: Watermarked video V_w [a,b]

Initially, the shot segmentation technique is applied to original input video sequence V_o [a,b] is segmented into number of non-overlapping shots D[a,b] . For embedding purpose, we identify number of frames E[a,b] in all the segmented shots D[a,b]. Then watermark audio signal A_w[a,b] is converted into 9-bit plane W[a,b], by the use of bit plane slicing. The video frames have R, G and B components. The blue channel is selected for embedding because this channel is more resistant to changes compared to red and green channels and the human eye is less sensitive to the blue channel, a perceptually invisible watermark embedded in the blue channel can contain more energy than a perceptually invisible watermark embedded in the luminance channel of a color image [12]. The blue components E_B [a,b] of all the separated frames are extracted. Then each bit of 9-bit plane sliced audio watermark W [a,b] is applied into the blue components of each frame; Discrete Wavelet Transform is applied to blue component E_B[a,b]. Discrete Wavelet Transform converts each frame into four sub-bands such as HH, HL, LH and LL to attain the transformed T[a,b] frames. Then we select the middle frequency sub-bands (HL, LH) from the transformed frames to embed the watermark audio A_w[a,b] into the appropriate sub-bands. In order to choose the embedding locations in the sub-bands, we find the similarity matrix for the video signal. The similarity matrix for HL sub-band is denoted by U_p(x,y) and the LH sub-band similarity matrix is denoted by lower part L_p(x,y).Then we calculate the mean value T_{LH}(m) and the maximum value T_{LH}(M) of the chosen embedding part T_{LH}.

Watermark bits are embedded in the video frames according to following conditions:

```
Condition 1:For embedding the Watermark Pixel 1
If $T_{LH}$(a) > 1 then $L_p$(x,y) << [$T_{LH}$(a)]
else $L_p$(x,y)<< $T_{LH}$ (a) + $T_{LH}$(M)
end if

Condition 2: For embedding the watermark Pixel 0
If $T_{LH}$(a)>0 then
$L_p$(x,y) << Abs[$T_{LH}$(a)]
else
$L_p$(x,y) << $T_{LH}$(a) -$T_{LH}$(M)
end if
```

Likewise the watermark bits can also be embedded into the HL band. Then the modified sub-bands are mapped into its original position and inverse wavelet transform is applied to attain the watermarked video sequence V_w [*a*,*b*].

2.3 Extraction Process

Input: Watermarked video sequence $V_w[a, b]$
& the size of the audio watermark A_w[a, b].
Output: Recovered watermark audio A_{rw}[*a*', b']

The watermarked video sequence $V_w[a, b]$ is segmented into number of non-overlapping shots $D'[a, b]$ by the use of shot segmentation technique. Then we identify number of frames $E'[a, b]$ in each segmented shots. Blue components of the partitioned frames are extracted. Afterwards Discrete Wavelet Transform is applied to the each partitioned frame $E_B'[a, b]$. The DWT is splits each frame into four sub-bands such as HH, HL, LH and LL and then to attain the transformed $T'[a, b]$ frames. The middle frequency LH and HL sub-bands are selected from the transformed frames. The watermark audio bits are extracted from LH and HL sub-bands. If the embedded bit value is greater than the mean pixel value, then the extracted pixel value is one. If it is lesser, then the extracted pixel is zero.

Matrix with the size of the audio watermark is prepared and the extracted bits are placed to attain the watermark audio. The extracted audio watermark A_{rw} [a',b'] is obtained by the use of reverse process of vector finding operation.

3 GA-PSO Based Hybrid Optimization

In order to achieve both imperceptibility and robustness of the watermarked media, we use the Genetic algorithm (GA) and Particle swarm Optimization (PSO) based hybrid optimization. GA is applied for generating the chromosome and PSO for selecting the optimal location for embedding the watermark media into host media. GA-PSO based hybrid optimization techniques are applied in embedding as well as extraction process.

The function of the randomly generated set of genes is the generation of chromosomes. Population size plays an important role in presenting the solution to the problem at hand. The beginning population set up is done by producing a population set P that comprises of set of chromosome vectors having half size of the HL or LH sub-band. Subsequently, we initialized it with “1” according to the size of the watermark in that vector in a random manner, and the remaining places are filled down with “0” values. Then, the beginning set of chromosomes is brought forth at random with minimum number.

The watermark embedding process is iterated till the optimal locations are obtained for each chromosome in the population set. Embedding and extraction process is carried out using these procedures which were defined in the section 2. Fitness computation formula is depicted below,

$$\text{Fitness} = \text{PSNR} + \text{NC} . \tag{1}$$

PSNR: The Peak-Signal-To-Noise Ratio (PSNR) is used to measure deviation of the watermarked and attacked frames from the original video frames.
NC: The normalized coefficient (NC) gives a measure of the robustness of watermarking and its peak value is 1. For calculating the PSNR and NC we have used standard formula as mentioned in [1].

Selection of optimized chromosomes is done based on the values for fitness. Select the optimized chromosome = N_P / 2.Where, N_P is Number of Parent chromosomes. Remaining chromosomes enter the next iteration in the search of finding the optimal solutions according to fitness function.

Based on the selected optimal chromosomes, we have the value of optimal solution. This solution is fed to the crossover operation. These set of fitness value corresponding to chromosomes provide the new offspring by the use of crossover operation. Every two individuals are chosen from the better set of chromosome to produce two new offspring by single crossover point.

In mutation operation, the output of crossover operation is used as input. The process of this function is to modify one gene value that is randomly selected and then that chromosome is fed to the fitness computation operation. Here mutation operation is replaced with a velocity computation.

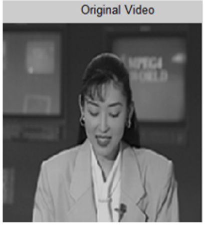
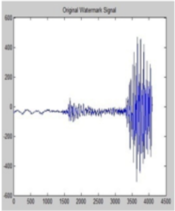
The velocity computation operation is the part of the Particle swarm optimization. During each iteration, each particle accelerates in the direction of its own personal best solution found so far as well as in the direction of the global best position discovered so far by any of the particle in the swarm. This means that if a particle discovers a promising new solution, all the other particles will move closer to it, exploring the region more thoroughly in the process. The velocity computation is done with the standard formula as described in [15].

Newly obtained set of chromosomes velocity computation operation can be evaluated for best fitness using fitness function. If the optimal solution for embedding the watermark media into original media is obtained then and then this process will be terminated, otherwise that solution will move to fitness computation operation again and the selection and velocity computation operators are performed iteratively. The PSO process will be iteratively performed until the desired termination is satisfied.

4 Experimental Results

We used many different videos of varying standards, framerate, framesize, payload for experimentation. Which includes Akiyo, Coastguard, Claire, Carphone, Shutttle, container, football and silent. Apart from these we used bradman.mpg, barryrichards.mpg, and chrisold.mpg and many others.

The original 30th frame and its corresponding audio watermark are shown in Fig. 1. Watermarked frame appears visually identical to the original. The performance of algorithm can be measured in terms of its imperceptibility and robustness against the possible attacks. Watermarked frame is subjected to a variety of attacks such as Salt and Pepper Noise, Median Filtering, Gaussian Noise, and geometric attacks etc. In case of geometric attacks, the scheme is tested against 90^0, 180^0, 270^0 frame rotation, and 25 % frame cropping. To evaluate the performance of any watermarking system, Peak Signal to Noise Ratio (PSNR) is used as a general measure of the visual quality. And the NC values are used as a measure of robustness. The NC values for extracted watermark for test videos by our proposed system are greater than .9. The most important aspect is the consistency of the results for different attacks. For robustness of compressions, our proposed system can effectively resist the MPEG-1, MPEG-2, MPEG-4 and H.264 compressions. Proposed system causes very slight distortion and simultaneously provides high visual quality. The embedded watermark is an audio signal which is unusual. An audio signal Example.wave, 529kb size is used and converted into suitable dimensions using wavread and signalslices functions in Matlab and transformed into a suitable watermark. Human Auditory system is more sensitive than the human visual system. Any modification in the extracted audio watermark will be noticed more easily as compared image or other watermarks. Evidently audio watermark bits are spread over LH and HL sub bands and it would be difficult to extract and reconstruct proper watermark and achieve high NC values. Our proposed system is strongly resistant to geometrical attacks like rotation and cropping in comparison to [1], [2] and [7].

Video	Original media	Watermark media	Attacks		Output	
					PSNR	NC
Akiyo			Without		35.5613	0.99209
			Salt & Pepper Noise	0.02	35.5609	0.98659
				.001	35.5612	0.9870
			Cropping		35.4617	0.98985
			Rotate	90	35.4617	0.98985
				180	35.5222	0.98814
				270	35.5271	0.98898

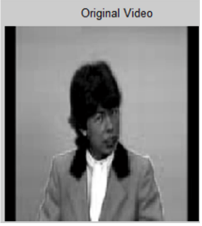
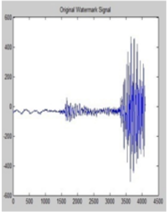
Video	Original media	Watermark media	Attacks		Output	
					PSNR	NC
Claire			Without		32.9895	0.9675
			Salt & Pepper Noise	0.02	32.9889	0.97265
				.001	32.9889	0.9675
			Cropping		32.9895	0.9675
			Rotate	90	32.9486	0.9509
				180	32.9628	0.9288
				270	32.9474	0.9253

Fig. 1. Experimental Results for Akiyo, Carphone of frame size 720X 480, 80 frames

5 Conclusion

A DWT and GA-PSO based novel video watermarking technique has been proposed using an audio signal as watermark. The performance of our purposed watermarking scheme is evaluated with common image processing attacks such as salt and pepper noises, rotation, cropping, Experimental results demonstrate this watermarking technique is robust against various attacks including the geometrical attacks. This proposed method is an extension to (HWT- Haar Wavelet Transform) HWT-GA-PSO based Image watermarking method [10].

References

1. Hui-Yu, H., Cheng-Han, Y., Wen-Hsing, H.: A Video Watermarking Technique Based on Pseudo 3-D DCT and quantization Index Modulation. IEEE Transactions on Information Forensics and Security 5(4), pp. 625–637 (2010)
2. Jing, Z., Anthony, T., Ho, S., Gang, Q., Pina, M.: Robust Video Watermarking of H.264/AVC. IEEE Transactions on Circuits and Systems-II: Express Briefs 54(2), 205–209 (2007)
3. Gwenael, D., Jean-Luc, D.: Guide Tour of Video Watermarking Signal Processing: Image Communication Elsevier. Signal Processing Image Communication 18, 263–282 (2003)
4. Elbasi, E.: Robust Multimedia Watermarking: Hidden Markov Model Approach for Video Sequences. Turk J. Elec. Eng. & Comp. Sci. 18(2), 159–170 (2010), doi:10.3906/elk-0906-85
5. Sonjoy, D.R., Xin, L., Yonatan, S., Alexander, F., Orly, Y.-P.: Hardware Implementation of a Digital Watermarking System for Video Authentication. IEEE Transactions on Circuits And Systems For Video Technology 23(2), 289–301 (2013)
6. Saurav, B., Chattopadhyay, T., Arpan, P.: A Survey on Different Video Watermarking Techniques and Comparative Analysis with Reference to H. 264/AVC. IEEE (2006)
7. Mostafa, S.A.K., Tolba, A.S., Abdelkader, F.M., Elhindy, H.M.: Video Watermarking Based on Principal Component Analysis and Wavelet Transform. International Journal of Computer Science and Network Security 9(8), 45–52 (2009)
8. Chuen-Ching, W., Yao-Tang, C., Yu-Chang, H.: Post-Compression Consideration in Video Watermarking for Wireless Communication. World Academy of Science, Engineering and Technology, 199–204 (2011)
9. Noorkami, M., Marsereau, R.M.: Digital Video Watermarking in P-Frames With Controlled Video Bit Rate Increase. IEEE Transactions on Information Forensics and Security 3(4), 441–455 (2008)
10. Puja, A., Khurshid, A.: Novel Invisible Watermarking for Various Images using HWT-GA-PSO based Hybrid Optimization. International Journal of Advanced Research in Computer Science and Software Engineering 3(8), 1093–1101 (2013) ISSN: 2277 128X
11. Martin, Z.: Master Thesis on Video Watermarking, Department of Computer Science Education, Charles University Prague (2007)
12. Shekhawat., R.S., Rao, S., Shrivastava, V.K.: A Robust Watermarking technique based on Biorthogonal Wavelet Transform. IEEE (2012) 978-1-4673-0455-9/12

13. Surekha, P., Sumathi, S.: Application of GA and PSO to the Analysis of Digital Image Watermarking Process. International Journal of Computer Science & Emerging Technologies 1(44), 350–362 (2010) (E-ISSN: 2044-6004)
14. Ramesh, S.M., Shnamugam, A.: An Efficient Robust Watermarking Algorithm in Filter Techniques for Embedding Digital Signature into Medical Images Using Discrete Wavelet Transform. European Journal of Scientific Research 60(1), 33–44 (2011) ISSN 1450-216X
15. Wang, Z., Sun, X., Zhang, D.: A Novel Watermarking Scheme Based on PSO Algorithm. In: Li, K., Fei, M., Irwin, G.W., Ma, S. (eds.) LSMS 2007. LNCS, vol. 4688, pp. 307–314. Springer, Heidelberg (2007)

Application and Comparison of Three Intelligent Algorithms in 2D Otsu Segmentation Algorithm

Lianlian Cao[1,2], Sheng Ding[1,2], Xiaowei Fu[1,2], and Li Chen[1,2]

[1] College of Computer Science and Technology,
Wuhan University of Science and Technology, China
[2] Hubei Province Key Laboratory of Intelligent Information Processing
and Real-time Industrial System, China

Abstract. 2D Otsu thresholding algorithm has been proposed based on Otsu algorithm, it is more effective in image segmentation. However, the computational burden of finding optimal threshold vector is very large for 2D Otsu method. In this paper, three kinds of intelligent algorithm are applied to improve and compare the efficiency of search. Experimental results show that these methods can not only obtain the ideal segmentation results but also greatly reduce the launch time. Moreover, it is proved that the quantum particle swarm optimization (QPSO) algorithm has the highest efficiency.

Keywords: 2D Otsu, image segmentation, intelligent algorithm, QPSO.

1 Introduction

In the image processing field image segmentation is a very important part, it is the basis of image analysis and understanding as well. The shareholding method is an effective method of image segmentation, and the most representative method is Otsu [1]. 2D Otsu [2, 3] method was presented on the basis of Otsu method, and it is based on image pixel and the two-dimensional histogram of pixel domain average. This method can get better segmentation results. However, it increases the computational complexity and limits the application of the algorithm. In order to overcome these disadvantages, the intelligent algorithms [4] are proposed to improve search efficiency. These three algorithms are respectively particle swarm optimization (PSO) algorithm [5], quantum particle swarm optimization (QPSO) algorithm [6, 7] and genetic algorithm (GA) [8].

The experiment results show that the threshold search efficiency of the three intelligent algorithms is greatly increased, and the QPSO algorithm searching efficiency is highest in these intelligent algorithms.

2 Three Intelligent Algorithms and the Main Parameter Setting

2.1 Particle Swarm Optimization Algorithm

PSO is a stochastic search method that was developed in 1995 based on the sociological behavior of bird flocking. This algorithm is easy to implement and has

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 221–227, 2014.
© Springer International Publishing Switzerland 2014

been successfully applied to solve a wide range of optimization problems. Now, the PSO technique is used to solve the problem of threshold based segmentation.

Main Parameter Setting. Set the learning factor c_1 and c_2, $c_1 = c_2 = 2$; the inertia weight w, $w = w_{\max} - \frac{w_{\max} - w_{\min}}{Maxiter} \times iter$, where the $w_{\max} = 0.9, w_{\min} = 0.4$, $Maxiter$ is the maximum number of iterations.

2.2 Quantum Particle Swarm Optimization Algorithm

QPSO is combining the classical PSO algorithm and the quantum theory, and it is based on the concept of quantum theory. The main iterative formula for particles

$$mbest(\mathrm{t}) = \frac{1}{m}\sum_{i=1}^{m} p_i(\mathrm{t}) = [\frac{1}{m}\sum_{i=1}^{m} p_{i1}(\mathrm{t}), \frac{1}{m}\sum_{i=1}^{m} p_{i2}(\mathrm{t}), \ldots, \frac{1}{m}\sum_{i=1}^{m} p_{iD}(\mathrm{t})]. \tag{1}$$

$$\mathrm{p}_{id}(t) = (\mathrm{r}_1\mathrm{p}_{id} + \mathrm{r}_2\mathrm{p}_{bd})/(\mathrm{r}_1 + \mathrm{r}_2). \tag{2}$$

$$X_{id}(\mathrm{t+1}) = \mathrm{p}_{id}(\mathrm{t}) \pm \beta \,|\, mbest(\mathrm{t}) - \mathrm{X}_{id}(\mathrm{t}) \,|\, \ln(\frac{1}{u}). \tag{3}$$

where $mbest$ is the average best position in group; p_{id} as a random point between p_{id} and p_{bd}; r_1 and r_2 for the interval [0, 1] random number; t for the current iteration number; D is the dimension of particles, u is a random number in [0, 1]; β as the contraction coefficient of expansion of the algorithm.

Main parameter Setting. $\beta = m - (m-n)\frac{t}{Maxiter}$, where $m = 1, n = 0.5$.

2.3 Genetic Algorithm

GA is a kind of adaptive global optimization probability search algorithm, which can speed up the overall algorithm to complement real-time processing. Because the essence of 2D Otsu threshold is seeking an optimal solution process, so the available genetic algorithm has the speediness of its optimization, in order to achieve the purpose of improving the efficiency.

Main Parameter Setting. Crossover probability, $pc = 0.8$; Mutation probability, $pm = 0.02$.

3 Image Segmentation Based on Intelligent Algorithm

3.1 2D Otsu Segmentation Methods

Given an image f represented by L gray levels, size of $M \times N$, each pixel in the image of the value corresponds to a greyscale.

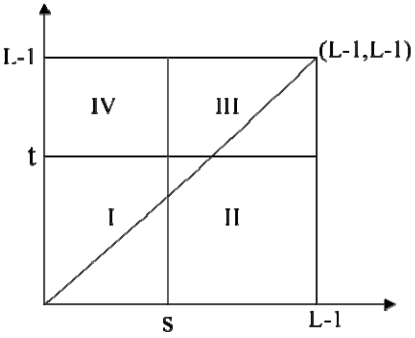


Fig. 1. Planar projection of two-dimensional gray histogram

The two-dimensional vector (s,t) in Fig.1 is the threshold, and it divides two-dimensional histogram into four parts (I, II, III and IV). The regions I and III contain the distributions of object and background classes, respectively.

Let S_0 and S_1 represent the object and the background respectively. The between-class discrete matrix is defined as

$$\sigma_B(s,t) = \sum_{k=0}^{1} P(S_k)[(\mu_k - \mu_T)(\mu_k - \mu_T)^T]. \tag{4}$$

where w_0 and w_1 are the probabilities of class occurrence; μ_0 and μ_1 are the mean value vectors of S_0 and S_1; μ_T is the total mean level vector of the 2D histogram.

$$w_0(s,t) = P(S_0) = \sum_{i=0}^{s}\sum_{j=0}^{t} p_{ij}, w_1(s,t) = P(S_1) = \sum_{i=s+1}^{L-1}\sum_{j=t+1}^{L-1} p_{ij}. \tag{5}$$

$$\mu_0(s,t) = (\sum_{i=0}^{s}\sum_{j=0}^{t}\frac{ip_{ij}}{\omega_0}, \sum_{i=0}^{s}\sum_{j=0}^{t}\frac{jp_{ij}}{\omega_0})^T, \mu_1(s,t) = (\sum_{i=s+1}^{L-1}\sum_{j=t+1}^{L-1}\frac{ip_{ij}}{\omega_1}, \sum_{i=s+1}^{L-1}\sum_{j=t+1}^{L-1}\frac{jp_{ij}}{\omega_1})^T. \tag{6}$$

$$\mu_T = (\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} ip_{ij}, \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} jp_{ij})^T. \tag{7}$$

The trace of discrete matrix could be expressed as

$$tr(\sigma_B) = \sum_{k=0}^{1} (\omega_k [(\mu_{ki} - \mu_{Ti})^2 + (\mu_{kj} - \mu_{Tj})^2]). \tag{8}$$

3.2 QPSO Concrete Realization of Image Segmentation

In this paper, the trace of discrete matrix function $t_r(\sigma_B)$ is the fitness function; find the maximum value, namely

$$f(s,t) = \max t_r(\sigma_B). \tag{9}$$

QPSO Algorithm Process:

1) Initialize position x_i of each particle.

2) Calculate the fitness of each particle (according to the formula (9)).

3) Update individual extremum: let the current fitness value of the i-th particle is compared with the particle individual extreme p_{id}. If the former is more optimal, then update the p_{id}, otherwise the p_{id} unchanged.

4) Update the global extremum: choose the optimal value from all p_{id} (choose the maximum value of the algorithm) as the global extremum p_{bd}.

5) Optimization process, according to the formula (1), (2) and (3) update all the particles in the QPSO algorithm.

6) Check whether meet the termination conditions, if satisfied, exit; Otherwise, $k = k + 1$ (k is the number of iterations) return to step 2, until meet the termination conditions.

4 Experimental and Result Analysis

All the algorithms are implemented on a personal computer with CPU of 2.6GHz using MATLAB R2012a programming language. In the experiments, 4 images (Lena, Cameraman, Peppers and Rice) used in this study and they are shown in Fig. 1. In order to demonstrate the effectiveness of the algorithms, below, will do some specific analysis of these algorithms and their results.

4.1 Segmentation Results

In this paper, the total number of particles is set to 10, and the largest number of iterations is set to 100. The experimental results show that the algorithm achieves the same segmentation results with the 2D Otsu algorithm. Results are as shown below:

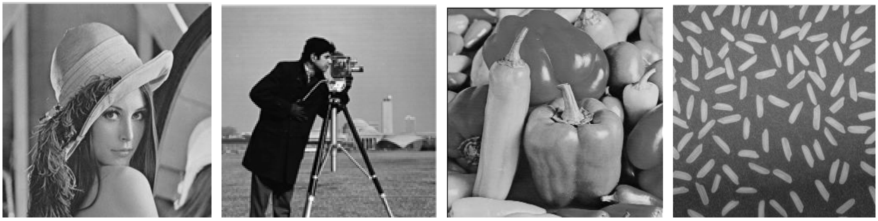


Fig. 2. Test Images. (a) Lena; (b) Cameraman; (c) Peppers; (d) Rice

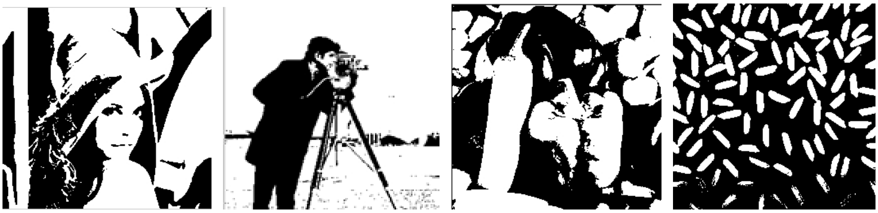


Fig. 3. Segmentation by 2D Otsu

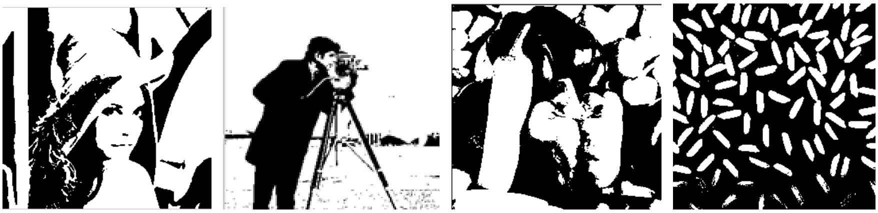


Fig. 4. Segmentation by PSO+ 2D Otsu

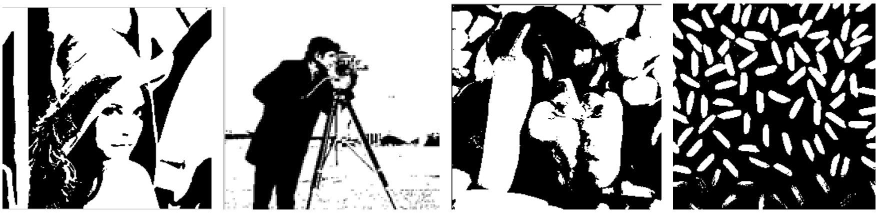


Fig. 5. Segmentation by QPSO + 2D Otsu

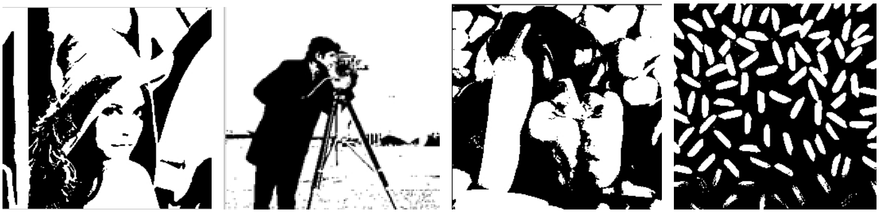


Fig. 6. Segmentation by GA + 2D Otsu

All the figures show the experimental results of these algorithms. Because PSO, QPSO and GA algorithms can achieve the same effect as the 2D Otsu algorithm, so we can obtained that the PSO, QPSO and GA algorithms are as good as the 2D Otsu.

4.2 Calculation Results

Firstly, 2D Otsu method is used to calculate the maximum variance threshold, get the optimal threshold value of image a is (125,116), the corresponding maximum variance is 2975.5; the image b is (89,103) and corresponding maximum variance is 6209.5; the image c is (125,134) and corresponding maximum variance is 3912.3; the image d is (133,138) and corresponding maximum variance is 2433.7. The contrast of the thresholds and searching time of the various methods is shown in Table 1.

Table 1. Performance comparison

Image	2D Otsu	GA+2D Otsu	PSO+2D Otsu	QPSO+2D Otsu
Lena(125,116)	0.8452s	0.4359s	0.4164s	0.3220s
Cameraman(89,103)	0.8408s	0.4446s	0.4137s	0.3351s
Peppers(125,134)	0.8314s	0.4490s	0.4226s	0.3325s
Rice(133,138)	0.8388s	0.4638s	0.4154s	0.3644s

Table 1 lists the average elapsed time of 40 experiments. For each figure, each algorithm tested 40 times. During the experiment, the time efficiency of these algorithms is repeated comparative. The experimental results is as shown in Table 1, it can come to a conclusion: the introduction of the three intelligent algorithms has increased 2D Otsu operation efficiency and reduced the search time.

The following is the efficiency comparison among the three kinds of intelligent algorithm. It is already to know, the efficiency of these algorithms is higher than the 2D Otsu algorithm. However, in the three algorithms whose efficiency is the highest. Similarly, it can find the answer from Table 1. In respect of the consumption time of three algorithms, QPSO algorithm has the shortest search time. So it is proved that the efficiency of QPSO algorithm is optimal.

5 Conclusions

2D Otsu method is an effective method of image segmentation, and it considers the image gray level information and the space between the pixel neighborhood information. Usually, the method can get better segmentation result than one-dimensional Otsu method, but the consumption of time is greatly increased. In order to solve this problem, the GA, PSO and QPSO algorithms are used to search the optimal two-dimensional threshold vector. The experiment results show that the use of the three intelligent algorithms can reduce the search time, thereby it increase search efficiency. What is more, among these algorithms, the search time of QPSO is shortest, so we can draw a conclusion that the QPSO algorithm is optimal.

Acknowledgments. This work was supported by Open Project Program of Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System (znss2013A008) and National Natural Science Foundation of China (No.61201423, 61375017)

References

1. Sthitpattanapongsa, P., Srinark, T.: A two-stage Otsu's thresholding based method on a 2D histogram. In: 2011 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 345–348. IEEE (2011)
2. Lu, C., Zhu, P.: The Segmentation Algorithm of Improvement a Two-dimensional Otsu and application research. In: 2nd International Conference on software Technology and Engineering (ICSTE) V1-76–V1-79 (2010)
3. Wang, X., Chen, S.: An improved image segmentation algorithm based on two-dimensional Otsu method. Inf. Sci. Lett 1, 77–83 (2012)
4. Kennedy, J., Eberhart, R.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
5. Tang, H., Wu, C., Han, L., Wang, X.: Image Segmentation Based on Improved PSO. In: The Proceedings of the International Conference on Computer and Communication Technologies in Agriculture Engineering (CCTAE 2010), pp. 191–194 (2010)
6. Yang, S., Wang, M., Jiao, L.: A quantum particle swarm optimization. In: Congress on Evolutionary Computation, CEC 2004, vol. 1, pp. 320–324. IEEE (2004)
7. Chao, Z., Jun, S.: Hybrid-Search Quantum-Behaved Particle Swarm Optimization Algorithm. In: 2011 Tenth International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), pp. 319–323. IEEE (2011)
8. Sheta, A., Braik, M.S., Aljahdali, S.: Genetic Algorithms: A tool for image segmentation. In: 2012 International Conference on Multimedia Computing and Systems (ICMCS), pp. 84–90. IEEE (2012)

A Shape Target Detection and Tracking Algorithm Based on the Target Measurement Intensity Filter

Weifeng Liu, Chenglin Wen, and Shuyu Ding

Hangzhou Dianzi University, Hangzhou, Zhejiang 310018, China
{liuwf,wencl}@hdu.edu.cn

Abstract. The probability hypothesis density (PHD) is the expectation intensity in a point in state space. The intensity integral in any region of the state space is the expected number of targets contained in that region. In this paper, we propose a target measurement intensity (TMI) filter. Compared with the existing methods, the proposed approach is simpler. Since the conventional PHD filter can not directly deal with the shape target detection and tracking, we give the detection and tracking algorithm based on the TMI filter by modeling the parameter dynamics and measurement function of the shape target.

1 Introduction

The probability hypothesis density (PHD) was first proposed by Stein and Winter [1]. Literally, it is a hypothesis density and does not exist in practice. Its physical meaning is the expected number of targets in a point in state space. Therefore, its integral in certain region in state space proposes the number of targets in that region. Mahler showed that the first order moment of multitarget random finite set (RFS)[2], which is an extension of the first order moment of random point process [3], is equal to the PHD almost everywhere. He also proposed the PHD recursive filter as an alternative of RFS Bayesian equation. One can estimate target state from the PHD filter. The PHD filter is a joint decision and estimation algorithm. It can be seen as an implicit association-estimation algorithm for the association step is substituted by an estimation step. It can deal with the uncertain number of targets such as the surviving targets, the spontaneous birth of new targets, and the spawned targets. Similar to the traditional approaches, the PHD filter is used in the point target tracking.

The PHD is equal to the expected number of measurement originating from a point $\boldsymbol{x}$ in state space. This is built on the following viewpoint: under the assumption of target being a point, a target produces at most one measurement. Therefore, the number of targets statistically equals the number of measurements. The PHD thus can be seen as the target measurement intensity (TMI). In RFS framework, Mahler got the PHD filter through probability generating functionals (PGF). Erdinc et al alternatively derived the PHD filter by using the physical-space approach - a bin model [4], where the PHD is interpreted

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 228–235, 2014.
© Springer International Publishing Switzerland 2014

as the bin-occupancy probability in the traditional probability framework. In another ref.[5], Streit derived a multitarget intensity filter from a Bayesian first principles approach using a Poisson point process approximation at one step. Streit's intensity filter is very similar to the PHD filter except the estimation of the target birth and measurement clutter processes. In this paper, an alternative derivation of the PHD filter is first proposed by using the TMI.

A point target produces one measurement at most in the traditional researches. In contrast, a shape target might give multiple measurements in each scan. The TMI can also be used in shape target tracking, not only a point target. we here refer it to be as the TMI filter and extend the point target tracking to the shape target tracking. Nevertheless, the detection and survival probability also have different means. For example, how to define a shape target is detected when the shape is partly covered. How to define the probability of detection in this case. Besides, the key to shape target tracking is to model dynamics and measurement function. All these implies that the original PHD filter cannot be directly used to the shape target tracking. Note that in this paper we confine the shape target to be with a parameter model, i.e., the shape target can be described by using a parameter model.

2 Background and Problem Description

Two methods were proposed to derive the PHD filter. The first is the Mahler's PGF method. The second is the physical space method given by Erdinc et al. Roy Streit proposed the intensity filter using the Poisson point process (PPP) method. In single sensor case, the intensity filter and the PHD filter have the same form. In this section, we review the PHD filter and the three methods.

Mahler's PHD filter consists of the following two recursive steps of predicted and update step [2],[6].

3 The Derivation of the Target Measurement Intensity Filter

3.1 Non-parameter State Mixture Models

The finite mixture models (FMM) are used to describe observations coming from various random sources and the models have a finite number of distributions

$$f(y|\theta) = \pi_1 f_1(y|\theta_1) + \cdots + \pi_m f_m(y|\theta_m) \tag{1}$$

Where $y \triangleq \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\}$ are observations, $\pi_1, \cdots, \pi_m$ are the mixing weights, $\theta_1, \cdots, \theta_m$ are the parameters for distributions $f_1(\cdot|\cdot), \cdots, f_m(\cdot|\cdot)$, called components here. we defined the following state mixture models:

$$f_k(\boldsymbol{z}_k|\pi_k, X_k) = \pi_{k,0} f_{k,0}(z_{k,i}|x_{k,0}) + \pi_{k,1} f_{k,1}(z_{k,i}|x_{k,1}) + \cdots + \pi_{k,m_k} f_{k,m_k}(z_{k,i}|x_{k,m_k}) \tag{2}$$

For consistence, the former observations y are replaced by $z_k \triangleq \{\boldsymbol{z}_{k,1}, \cdots, \boldsymbol{z}_{k,n_k}\}$, k is the sampling time, , $\pi_k \triangleq \{\pi_{k,0}, \pi_{k,1}, \cdots, \pi_{k,m_k}\}$, $X_k \triangleq \{\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_{k,m_k}\}$, where $\boldsymbol{x}_{k,0}$ is the clutter state, $\boldsymbol{x}_{k,1}, \cdots, \boldsymbol{x}_{k,m_k}$ are the states of targets, $f_{k,0}$ is the clutter density, $\pi_{k,0}$ is the clutter mixing weight, m_k can be also interpreted as the number of targets. $\{\pi_{k,j}\}_{j=1}^{m_k}$ are the mixing weights of the targets and $\{f_{k,j}(\boldsymbol{z}_{k,i}|\boldsymbol{x}_{k,1})\}_{j=1}^{m_k}$ are the corresponding measurement densities of targets. Streit et al introduced the state mixture models in the probabilistic multiple hypothesis tracking (PMHT) [12]-[13]. The state mixture models describes how the measurements are generated, but the derivation of the measurement likelihood function is still a difficult one. Under the independent assumption, the measurement likelihood can be given by product $L_z(x) = \prod_{i=1}^{n_k} f_k(\boldsymbol{z}_{k,i}|\pi_k, X_k)$. But this expression is an implicit form for the state X_k need to be first estimated. Then, a problem is: can we obtain the states of targets while avoid the calculation of the likelihood function? This needs one to consider certain characteristic function of the target state $\boldsymbol{x}$ in the target state space. Thus the states of targets can be estimated from the characteristic function. For RFS, the characteristic function is corresponding to its first order, i.e., the PHD $D_k(\boldsymbol{x})$. In this paper we consider the target measurement intensity (TMI) $e_k(x)$ in the state space. The TMI describes the distribution of the number of the target measurements in the state space. The further research shows that the TMI are statistically equal to the PHD under certain assumptions.

Assume that the target states are all in the same state space. Consider the state models of a point $\boldsymbol{x}$ in state space

$$\begin{aligned} f_k(z_{k,i}|\boldsymbol{x}) = P(D_0|\boldsymbol{x}) f_k(\phi|\boldsymbol{x}, D_0) + P(D_1|\boldsymbol{x}) f_k[\boldsymbol{z}_{k,i}|\boldsymbol{x}, D_1] \\ = (1 - P_D(\boldsymbol{x}))\pi_{k,\phi} f_k[\phi|x, e_{k,0}(\boldsymbol{x})] + \\ P_D(\boldsymbol{x})\{\pi_{k,c}(\boldsymbol{x}) c_k[z_{k,i}|\boldsymbol{x}, e_{k,i}(\boldsymbol{x}) = 0] + \pi_{k,t}(\boldsymbol{x}) g_k[\boldsymbol{z}_{k,i}|\boldsymbol{x}, e_{k,i}(\boldsymbol{x}) = 1]\} \end{aligned} \tag{3}$$

Where D_1, D_0 are the events that the target is detected and is not detected, respectively. ϕ is the event of no measurements, $c_k(\cdot|\cdot)$ is the clutter distribution, $g(\cdot|\cdot)$ is the target measurement distribution, $e_{k,i}(\boldsymbol{x})$ is the indicating variables defined in $\{0, 1\}$. $e_{k,i}(\boldsymbol{x}) = 1$ implies that the ith measurement produced by state $\boldsymbol{x}$. We extend the above state mixture models to the total state space S as follows.

$$\begin{aligned} & f_k(z_{k,i}|S) = f_k(\boldsymbol{z}_{k,i}|S, D_0)P(D_0) + f_k(\boldsymbol{z}_{k,i}|S, D_1)P(D_1) \\ & = \int_{x \in s} [f_k(\boldsymbol{z}_{k,i}|\boldsymbol{x}, D_0)P(D_0|\boldsymbol{x}) + f_k(\boldsymbol{z}_{k,i}|\boldsymbol{x}, D_1)P(D_1|\boldsymbol{x})] d\boldsymbol{x} \\ & = \int_{x \in s} \{[1 - P_D(\boldsymbol{x})] f_k[\phi|\boldsymbol{x}, e_{k,0}(\boldsymbol{x})]\} dx + \\ & \int_{x \in s} \{\pi_{k,c}(\boldsymbol{x}) c_k[\boldsymbol{z}_{k,i}|\boldsymbol{x}, e_{k,i}(\boldsymbol{x}) = 0] + P_D(\boldsymbol{x})\pi_{k,t}(\boldsymbol{x})) g_k[\boldsymbol{z}_{k,i}\boldsymbol{x}, e_{k,i}(\boldsymbol{x}) = 1]\} dx \end{aligned} \tag{4}$$

Where $\pi_{k,c}(\boldsymbol{x})$ and $\pi_{k,t}(\boldsymbol{x})$ are the weights of clutter measurements and target measurements. We define the mixing weight to be the probability of target existing in the point $\boldsymbol{x}$

$$\pi_{k,c}(\boldsymbol{x}) = p_k[e_{k,i}(\boldsymbol{x}) = 0|\boldsymbol{x}]$$
$$\pi_{k,t}(\boldsymbol{x}) = p_k[e_{k,i}(\boldsymbol{x}) = 1|\boldsymbol{x}]$$

Then the target measurements intensity in the state $\boldsymbol{x}$ is defined by:

$$e_k(\boldsymbol{x}) = e_{k,0}(\boldsymbol{x}) + e_{k,1}(\boldsymbol{x}) + \cdots + e_{k,n_k}(\boldsymbol{x}) \tag{5}$$

The TMI consists of two types of terms: the intensity $e_{k,0}(\boldsymbol{x})$ of no measurements and individual measurement intensities $\{e_{k,i}(\boldsymbol{x})\}_{i=1}^{n_k}$. In the following subsection, we focus on deriving the recursive equations of the TMI and the mixing weights.

3.2 Derivation of the Recursive Equation for the TMI Filter

The recursive equations of the TMI filter involve two step, i.e., the predicted step and the update step.

The Predicted TMI. The assumption in the predicted step is proposed as follows.
A.1: Each target moves and generates individually and independently of all the other targets.

Proposition 1. *Under the assumption A1. Assume that at time k the intial TMI is $e_k(\boldsymbol{x})$. Then, the predicated TMI is given by:*

$$e_{k+1|k}(\boldsymbol{x}) = e^{\gamma}_{k+1}(\boldsymbol{x}) + e^{s}_{k+1|k}(\boldsymbol{x}) + e^{\beta}_{k+1|k}(\boldsymbol{x}) \tag{6}$$

Where

$$e^{s}_{k+1|k}(\boldsymbol{x}) = \int_{\boldsymbol{\omega}\in S} P_S(\boldsymbol{x}) f_{k+1|k}(\boldsymbol{x}|\omega) e_k(\boldsymbol{\omega}) d\boldsymbol{\omega} \tag{7}$$

$$e^{\beta}_{k+1|k}(\boldsymbol{x}) = \int_{\boldsymbol{\omega}\in S} P_S(\boldsymbol{x}) \beta_{k+1|k}(\boldsymbol{x}|\boldsymbol{\omega}) e_k(\boldsymbol{\omega}) d\boldsymbol{\omega} \tag{8}$$

Where $e^{\gamma}_{k+1}(\boldsymbol{x}), e^{s}_{k+1|k}(\boldsymbol{x})$, $e^{\beta}_{k+1|k}(\boldsymbol{x})$ are respectively the TMI for the spontaneous births, the surviving target and the spawned target. $P_S(\boldsymbol{x})$ is the survival probability. $f_{k+1|k}(\boldsymbol{x}|\omega)$ and $\beta_{k+1|k}(\boldsymbol{x}|\boldsymbol{\omega})$ are respectively the Markov models of the surviving target and the spawned target, which are the same as eq.(1). It can be derived based on the weighted sums of the TMI. This is similar as the PHD filter.

3.3 The Update TMI

In this step, we propose the following assumptions
A.2 The number of target measurements and the number of clutter measurement follow Poisson with intensities M_{k+1} and λ_{k+1}, respectively.
A.3 Clutter measurements are independent of target states.

Proposition 2. *Assume that predicted TMI is $e_{k+1|k}(\boldsymbol{x})$. Then, under the assumptions A.2 and A.3, the update TMI is given by:*

$$e_{k+1}(\boldsymbol{x}) = e_{k+1,0}(\boldsymbol{x}) + e_{k+1,1}(\boldsymbol{x}) + \cdots + e_{k+1,n_{k+1}}(\boldsymbol{x}) \tag{9}$$

Where $e_{k+1,0}(\boldsymbol{x})$ is the TMI of no measurements, $e_{k+1,i\geq 1}(\boldsymbol{x})$ is the TMI of the ith measurement. And they can be calculated by the following equation:

$$e_{k+1,0}(\boldsymbol{x}) = (1 - P_D(\boldsymbol{x}))e_{k+1|k}(\boldsymbol{x}) \tag{10}$$

$$e_{k+1,i\geq 1}(\boldsymbol{x}) = \frac{P_D(\boldsymbol{x})g_{k+1}(z_{k+1,i}|\boldsymbol{x})e_{k+1|k}(\boldsymbol{x})}{\lambda c_{k+1}(\boldsymbol{z}_{k+1,i}) + \int P_D(\boldsymbol{x})g_{k+1}(z_{k+1,i}|\boldsymbol{x})e_{k+1|k}(\boldsymbol{x})d\boldsymbol{x}} \tag{11}$$

4 The Shape Detection and Tracking

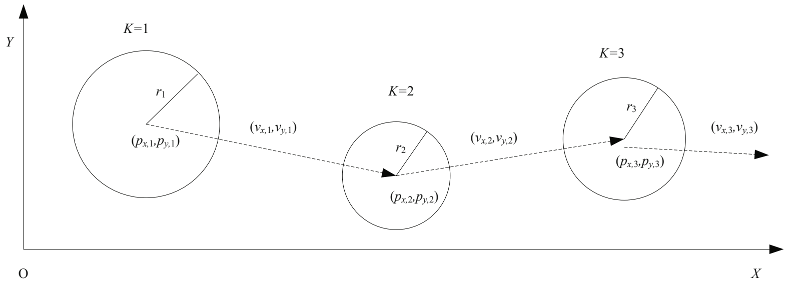


Fig. 1. A movement of a circular shape in x-y plane which is described by a five-dimensional parameter vector

4.1 The Target Dynamics and Measurement Function

A shape can be modeled in the parameter space and be described by a parameter vector. We therefore propose the dynamics and measurement function in the parameter space. The movement of the shape target can be modeled by the dynamics of the parameter vector. This is the same as the usual state function. For a nonlinear function and a linear system function, we can describe it by using the following functions:

$$\boldsymbol{X}_k = f(X_{k-1}, \boldsymbol{\omega}_k) \qquad \text{nonlinear parameter function} \tag{12}$$

$$\boldsymbol{X}_k = A_{k-1}X_{k-1} + B_{k-1}\boldsymbol{\omega}_k \qquad \text{linear parameter function} \tag{13}$$

We example two targets with a circular shape and a linear shape, respectively. Fig.1 shows a circular target moving in the planar. We select the center point and the radius of the circle as the parameter vector $\boldsymbol{X} = (p_{x,k}, \dot{p}_{x,k}, p_{y,k}, \dot{p}_{y,k}, r_k)$, where $(p_{x,k}, p_{y,k})$ is the center coordinate of the circular shape, $(\dot{p}_{x,k}, \dot{p}_{y,k})$ is the velocity of the center point, r_k is the circle radius.

In Figure 2, a movement of a linear target is given. It involves two types of movements which include rotation around the center of the linear shape and CV movement of the center point. A seven-dimensional parameter vector is proposed as $X_k = (p_{x,k}, \dot{p}_{x,k}, p_{y,k}, \dot{p}_{y,k}, \theta_k, \dot{\theta}_k, l_k)$, where $(p_{x,k}, p_{y,k})$ is the the center coordinate of the linear shape, $(\dot{p}_{x,k}, \dot{p}_{y,k})$ is the velocity of the center point, θ_k and $\dot{\theta}_k$ are respectively the angle and the angular velocity of the linear shape, l_k is the length of the target.

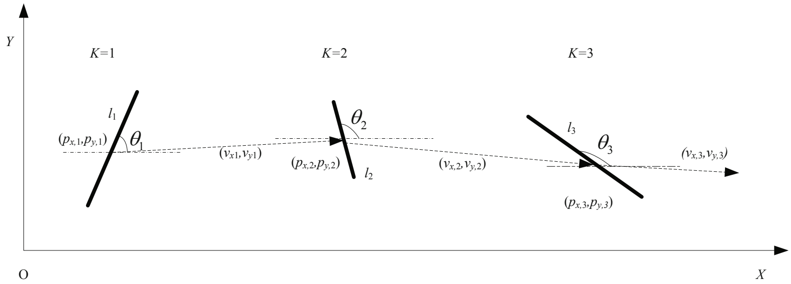
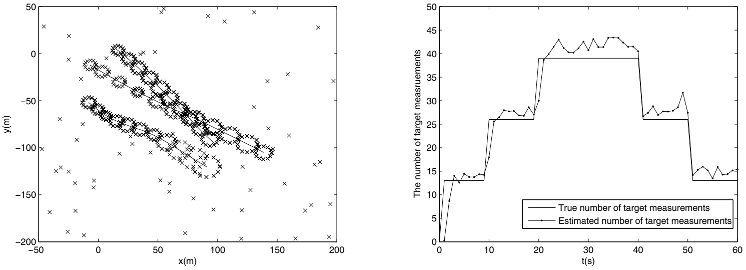


Fig. 2. A movement of a linear shape in x-y plane which is described by a seven-dimensional parameter vector

5 Simulations

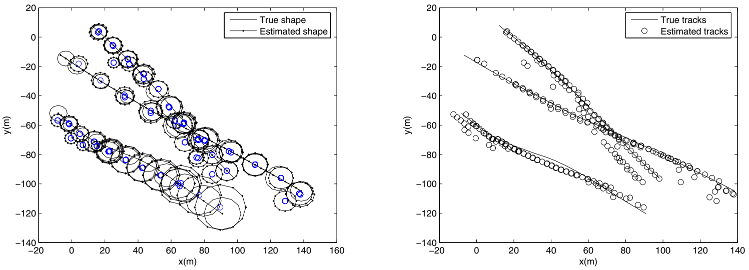
In this section, a simulation of detection and tracking of three circular targets is proposed to verify the proposed TMI filter. Three circular targets with CV movement are proposed in this simulation. The parameter vector is given in



(a) The clutter measurements and target measurements

(b) The measurement intensities

Fig. 3. The measurements and intensities



(a) The true shapes and the estimated shapes

(b) The true tracks and the estimated tracks

Fig. 4. The Estimated shapes and tracks

eq.(26). The initial states for the three targets are respectively: $X_0^1 = (-10\text{m}, 3.2\text{m/s}, -10\text{m}, -2.2\text{m/s}, 5\text{m})$, $X_0^2 = (10\text{m}, 2.2\text{m/s}, 10\text{m}, -2.2\text{m/s}, 5\text{m})$ and $X_0^3 = (-10\text{m}, 1.8\text{m/s}, -50\text{m}, -1.9\text{m/s}, 5\text{m})$. Covariance matrixes: $P_0^i = \text{diag}([10, 2, 2.5, 10, 2.5, 1])$, $Q_k^i = \text{diag}([0.01, 0.01, 0.2])$ and $R_k^i = \text{diag}([0.04, 0.04]), i = 1, 2, 3$. Clutter density $\rho(\boldsymbol{x}) = 1.0 \times 10^{-3}\text{m}^{-2}$. The surveillance region is $[-50, 200] \times [-200, 50]\text{m}^2$. The measurements are produced by eqs.(27)-(29), where parameters $\{\psi_k(i)\}$ samples in $[0, 2\pi]$ per $\pi/6$ and thus $\lambda_k = 13$.

The Gaussian mixture based TMI filter, which is analogous to the Gaussian mixture PHD filter, is proposed here. Sub figure (a) of Fig.3 shows the clutter measurements and shape measurements every 4 seconds. Obviously, these two types of measurements are mixed. Thus, our first step is to detect the target. It can be seen from (b) of Fig.3 that the measurement intensities and number of targets indicate the existing of targets and their values are approximated to the true value. In estimation of shape parameter, it can be seen from sub figure (a) of Fig.4 that the circle radiuses are close to the true values. Sub figure (b) of Fig.4 shows that the estimations of position are near the true tracks.

6 Conclusion

This paper proposed a target measurement intensity (TMI) filter based on the mixture distributions. We provided the predicted TMI and the update TMI. Under some assumptions, the TMI filter is equal to the original PHD filter. The PHD filter focus on point target tracking. A potential advantage of the TMI filter is that it can be extended the target tracking with parameter shape. Thus we can use it in the extended target tracking. Correspondingly, our next work is to extend the TMI to the target with parameter shape. The key is to model the parameter dynamics and the parameter measurement function. Based on these functions, we extend the TMI filter to the shape target tracking. Finally, we propose two experiments involving three circular targets and three linear targets to verify the proposed TMI filter.

Nevertheless, under the multiple measurements condition, estimation of the number of targets is still an intractable problem. We propose a simple formulation under the Poisson assumption of the target measurements. Besides, in this paper we confine our object to the targets with parameter shapes and they are in the same parameter space. Future works are still needed for the general shape target.

Acknowledgements. This work was supported in part by the NSFC (61175030, 61273170, 61333011, and 61271144)

References

1. Winter, C.L., Stein, M.C.: IES/BTI system overview. In: Proceedings of 8th National Symposium on Sensor Fusion, Dallas TX, vol. I, pp. 15–17, 27–46 (1995)
2. Mahler, R.: Multitarget Bayes Filtering via First-Order Multitarget Moments. IEEE Transactions on Aerospace and Electronic System 39(4), 1152–1178 (2003)
3. Daley, D.J., Vere, J.D.: An Introduction to the Theory of Point Processes, Elementary Theory and Methods, 2nd edn., vol. I, pp. 123–131. Springer, New York (2003)
4. Erdinc, O., Willet, P., Bar-Shalom, Y.: A Physical-Space Approach for the Probability Hypothesis Density and Cardinalized Probability Density Filters, Signal and Data Processing of Small Targets. In: Proc. of SPIE, vol. 6236, pp. 1–12 (2006)
5. Streit, R.L., Stone, L.D.: Bayes Derivation of Multitarget Intensity Filters. In: The 11th International Conference on Information Fusion, Colgon, Germany, pp. 1686–1693 (July 2008)
6. Mahler, R.: An Introduction to Multisource-Multitarget Statistics and Its Applications. Technical Monograph, Lockheed Martin: 1–20 (Mar 15, 2000)
7. Vo, B.N., Ma, W.K.: The Gaussian Mixture Probability Hypothesis Density Filter. IEEE Transactions on Signal Processing 54(11), 4091–4104 (2006)
8. Vo, B.T., Vo, B.N., Cantoni, A.: Bayesian Filtering with Random Finite Set Observations. IEEE Transactions on Signal Processing 56(4), 1313–1326 (2008)
9. Challa, S., Vo, B.N., Wang, X.Z.: Bayesian Approaches to Track Existence - IPDA and Random Sets. In: The 11th International Conference on Information Fusion, Annapolis, Maryland, USA, pp. 1228–1235 (2002)
10. Ulmke, M., Erdinc, O., Willett, P.: Gaussian Mixture Cardinalized PHD Filter for Ground Moving Target Tracking. In: The 10th International Conference on Information Fusion, Quebec, Canada, pp. 1–8 (July 2007)
11. Liu, W.F., Han, C.Z., Lian, F.: An alternative derivation of a Bayes tracking filter based on finite mixture models. In: Proceedings of the 12th International Conference on Information Fusion, Seattle, USA, pp. 842–849 (2009)
12. Streit, R.L., Luginbuhl, T.E.: Maximum likelihood method for probabilistic multihypothesis tracking. In: Proceedings of SPIE International Symposium, Signal and Data Proceeding of Small Targets, Bellingham, WA, USA, pp. 394–405 (1994)
13. Streit, R.L., Luginbuhl, T.E.: A Probabilistic multi-hypothesis tracking algorithm without enumeration and pruning. In: Proceedings of the Sixth Joint Service Data Fusion Symposium, Laurel, MD, pp. 1015–1024 (1993)

Multi-cell Contour Estimate Based on Ant Pheromone Intensity Field

Qinglan Chen[1], Benlian Xu[2], Yayun Ren[2], Mingli Lu[2], and Peiyi Zhu[2]

[1] School of Mechanical Engineering,
Changshu Institute of Technology, 215500 Changshu, China
[2] School of Electrical & Automatic Engineering, Changshu Institute of Technology,
215500 Changshu, China
chenql@cslg.cn, xu_benlian@cslg.cn, zpy2000@126.com

Abstract. In this paper, we propose an ant pheromone based approach to accurately extract the contours of multiple small cells in low contrast bio-medical images. With the local information of intensity variation of each pixel, the initial distribution of ant colony is generated as ants' starting positions. Following the heuristic information, such as the pixel grayscale variance, the ant inertial heading and the image intensity, ant's searching behavior is modeled appropriately to make each of ants move along the edge of interested object as possible. Due to modeling an accurate depositing mechanism of pheromone, the corresponding ring pheromone field is formed and used to extract interested cells' contours after simple morphological operations. Experiment results show that our algorithm could give an accurate contour estimate of each cell for several different image sequences.

Keywords: Image Processing, Ant Colony, Contour Estimate.

1 Introduction

As an important branch of cell motion analysis, the estimate of cell contour could directly or indirectly encompass rich contents about each individual cell, and the related research is challenging and emerging due to poor image quality, small size and discontinuities or sharp changes in intensity, etc.. In most computer applications, image contour extraction constitutes a crucial initial step before performing the task of object recognition and representation. The conventional approaches are computationally expensive because each set of operations is conducted for each pixel. Thus, many researchers resort to other promising techniques. An ACO-based approach, a nature-inspired optimization algorithm, has the potential of solving these intractable problems because of its parallelized and intelligent searching mechanisms, which makes the algorithm easily adaptable for processing multiple objects simultaneously. In terms of the combination between the ACO-based approaches and image segmentation, the related work can be divided into two main strands of research. The first strand focuses on the fusion of ACO and other edge detection and contour extraction algorithms [1, 2], mainly because of the strong and effective

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 236–243, 2014.
© Springer International Publishing Switzerland 2014

optimization capabilities of ACO. Oliver *et al.*[3] formulate the shape correspondence as a Quadratic Assignment Problem (QAP), incorporating proximity information into the point matching objective function, and propose the first ACO algorithm directly aimed at solving the point and contour correspondence problems. Lai *et al.* [4] present a novel system for active contour tracking of moving objects in video sequences, incorporating the use of edge flows in the ACO algorithm to improve the efficiency. Li [5] presents a novel image contour extraction by ant colony algorithm and B-snake model. The method can enhance the flexibility of B-snake to describe complex shape. The second strand is those approaches [6-9] which construct the edge map based on the pheromone matrix. In these methods, each entry of the pheromone represents the edge information at each pixel of the image. Anna *et al.* [10] propose a ACO-based edge detection method which takes advantage of the improvements introduced in ant colony system, a extension of AS. Aminu *et al.* [11] use the discrete wavelet transform (DWT) as a preprocessing step with ACO to enhance image edge detection. Carla *et al.* [12] employ the ACO-based algorithm preceded by anisotropic diffusion to segment the optic disc in color images, and good performance is achieved as the optic disc was detected in most of all the images, even in the images with great variability.

2 Ants for Multi-cell Contour Estimate

2.1 Ants' Initial Distribution

In the original ant system, the initial ant colony is usually uniformly distributed in a searching space, however, an alternative layout is proposed to assign a given number of ants to pixels of an image where objects (cells) probably occur. For this purpose, the grayscale distribution of current image is utilized to measure the relative intensity variation of each pixel within its local region. As illustrated in Fig.1, an 8-neighbouring pixel configuration, a neighboring region of a given pixel (i, j) with intensity $I_{(i,j)}$, is defined, and its local region grayscale variance is computed as

$$\Delta\sigma_{(i,j)} = \frac{1}{|N_{(i,j)}|} \sum_{(i',j')\in N_{(i,j)}} (I_{(i',j')} - \bar{I}(N_{(i,j)}))^2 \quad , \tag{1}$$

where $N_{(i,j)}$ denotes the neighboring pixels of pixel (i, j), $\bar{I}(N_{(i,j)})$ denotes the average gray intensity of $N_{(i,j)}$, and $|N_{(i,j)}|$ is the number of neighboring pixels of pixel (i, j). As implicated in Eq. (1), the grayscale variance has a smaller value in the area of background and interior of cell, whereas a larger value is probably taken between two sides of edge, as well as in the vicinity of each edge pixel.

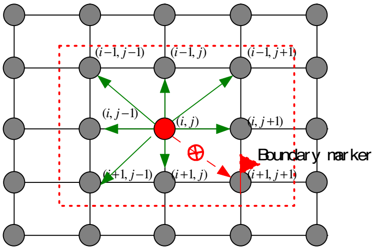


Fig. 1. 8-neighbouring pixel configuration

2.2 Ants' Working Mechanism

To emulate clumping behavior in ant colony (i.e., aggregation), ant working environment and ant decision constraint should be defined appropriately. The ant operating environment corresponds to the current cell image and is denoted by a tuple $\langle P, N\rangle$, where P is a finite set of image pixels, and $N \subseteq P \times P$ is a *closest* propagation neighboring relation among pixels. For any pixel index $g \in P$, an 8-neighbouring configuration is represented as $N_{(g)} = \{s \in P : gNs\}$ with ($\left|N_{(g)}\right| \le 8$). With this definition, each ant can move directly towards one of its neighbors at a time, and any pixel can be visited simultaneously by several ants.

Cell contour extraction is challenging due to incomplete or discrete grayscale variance curves and irregular or partial ant colony distribution in the neighboring region of each edge pixel. Therefore, a novel ant decision is designed and modeled as

$$(i', j') = \begin{cases} \underset{\substack{(i',j')\in N_{(i,j)} \\ (i',j')\notin \Omega_v}}{\arg\max} \left((\tau^C_{(i',j')}(\hat{t}))^{\lambda} (\Delta\tilde{\sigma}_{(i',j')})^{\varsigma} (W(\Delta\theta_{(i',j')}))^{\gamma} \right) & \text{if } \ q' < \tilde{q}_0 \\ (\tilde{i}, \tilde{j}) & \text{otherwise} \end{cases}, \qquad (2)$$

where λ, ς, and γ are the adjustment parameters of contour pheromone $\tau^C_{(i',j')}(\hat{t})$, heuristic grayscale variance $\Delta\tilde{\sigma}_{(i',j')}$, and weight factor of ant heading change $W(\Delta\theta_{(i',j')})$, respectively; Ω_v is the set of pixels visited by the current ant; $\tilde{q}_0$ is a threshold which takes the value between 0 and 1.

The pixel $(\tilde{i}, \tilde{j})$ will be visited according to the probability distribution given by:

$$P^C_{(i,j)\to(\tilde{i},\tilde{j})}(\hat{t}) = \begin{cases} \dfrac{(\tau^C_{(\tilde{i},\tilde{j})}(\hat{t}))^{\lambda} (\Delta\tilde{\sigma}_{(\tilde{i},\tilde{j})})^{\varsigma} (W(\Delta\theta_{(\tilde{i},\tilde{j})}))^{\gamma}}{\sum\limits_{\substack{(m,n)\in N_{(i,j)} \\ (m,n)\notin \Omega_v}} (\tau^C_{(m,n)}(\hat{t}))^{\lambda} (\Delta\tilde{\sigma}_{(m,n)})^{\varsigma} (W(\Delta\theta_{(m,n)}))^{\gamma}} & \text{if } \ N_{(i,j)} \not\subset \Omega_v \\ 0 & \text{otherwise} \end{cases}. \qquad (3)$$

It is implicated that the above model not only propels an ant towards edge pixels of cell, but also forces an ant keeping on moving along the edge of cell instead of staying in the vicinity of its starting pixel. According to the configuration in Fig.1, the destination candidates for each ant decision are up to eight, and the angle between two neighboring candidates is 45^0. Therefore, we assume that if the current heading of an ant is known, the weight factor of ant heading change in the following decision is defined as: $W(\pm 0^0) = 1/3$, $W(\pm 45^0) = 1/3$, $W(\pm 90^0) = 1/10$, $W(\pm 135^0) = 1/16$, and $W(\pm 180^0) = 1/20$.

Once an ant has made $\bar{m}$ decisions, it will deposit an amount of pheromone on corresponding visited pixels (up to $\bar{m}$ pixels at a time) with two levels

$$r^C(\hat{t}) = \begin{cases} c_2 \cdot (1 - e^{-\hat{t}/T2}) / \sum_{m'=1}^{\bar{m}} \left(1/\max\left\{\Delta\tilde{\sigma}_{(m')}, \bar{\mu}_{\min}\right\}\right) & \textit{if } std\left\{1/\Delta\tilde{\sigma}_{(m')}\right\}_{m'=1}^{\bar{m}} < \delta_0, d_{\bar{m}}^{\max} > \dfrac{\bar{m}}{6} \\ c_0 \cdot (1 - e^{-\hat{t}/T2}) & \textit{otherwise} \end{cases}, \quad (4)$$

where c_2, δ_0, $T2$, and $\bar{\mu}_{\min}$ are constants, $std\{\cdot\}$ denotes the standard deviation of a given set, and $\Delta\tilde{\sigma}_{(m')}$ denotes the grayscale variance value of pixel corresponding to the m'-th decision of ant.

We observe that not all ants deposit the same amount of pheromone according to the above two constraints, but they do offer a hint of where continuous cell edges are located when an ant makes a decision. Specifically, the first constraint $std\left\{1/\Delta\tilde{\sigma}_{(m')}\right\}_{m'=1}^{\bar{m}} < \delta_0$ tries to strengthen tour pixels within the same level of grayscale invariance, while the second constraint $d_{\bar{m}}^{\max} > \bar{m}/6$ encourages ant to move as far as possible along the edges.

2.3 The Formation of Pheromone Field

In this work, we model three pheromone working mechanisms to jointly produce pheromone field. First, the pheromone deposited by different ant individuals are aggregated and merged on the corresponding pixel, which results in pheromone peaks in the interested areas. Second, pheromone performs evaporation over time, and this simulates the memory ability of ant individuals as we observe in nature. Finally, pheromone propagation is considered to build a connection between neighboring pixels, and it builds a bridge for access by neighboring agents.

Pheromone aggregation is defined as a combination of evaporation, external input, and propagation. For any pixel (i, j), the evolution of pheromone amount follows

$$\tau_{(i,j)}^C(\hat{t}+1) = E \cdot \tau_{(i,j)}^C(\hat{t}) + r_{(i,j)}(\hat{t}) + q_{(i,j)}(\hat{t}) \quad , \quad (5)$$

where $r_{(i,j)}(\hat{t})$ denotes the pheromone external input to pixel (i, j) at the $\hat{t}$-th iteration, and $q_{(i,j)}(\hat{t})$ models the propagation input to pixel (i, j). Note that the above model applies to both location and contour pheromone fields.

It is observed that, unlike the traditional ant system (AS), the pheromone propagation $q_{(i,j)}(\hat{t})$ is introduced to coincide with the pixel intensity continuity in an image, and its evolution form is defined as

$$q_{(i,j)}(\hat{t}) = \sum_{(i',j') \in N_{(i,j)}} \frac{D}{|N_{(i',j')}|}\left(r_{(i',j')}(\hat{t}-1) + q_{(i',j')}(\hat{t}-1)\right) \ , \tag{6}$$

where $|N_{(i',j')}|$ defines the cardinality of $N_{(i',j')}$, D denotes the propagation coefficient with $0 < D < 1$, and $\frac{D}{|N_{(i',j')}|}$ characterizes the averaged propagation proportion of total received pheromone intensity on pixel (i',j') at the $\hat{t}-1$ -th iteration to its neighboring pixels.

As defined in Eq. (6), the propagation pheromone field at next iteration is the propagating results of the field of itself and the external input pheromone field both at current iteration. Furthermore, we assume that the pheromone amount on each pixel is an un-weighted sum of pheromones of its neighbors, thus the Eq. (6) could be divided into two parts and rewritten as

$$\begin{aligned} q_{(i,j)}(\hat{t}) &= \sum_{(i',j') \in N_{(i,j)}} \frac{D}{|N_{(i',j')}|} r_{(i',j')}(\hat{t}-1) + \sum_{(i',j') \in N_{(i,j)}} \frac{D}{|N_{(i',j')}|} q_{(i',j')}(\hat{t}-1) \\ &= D \cdot r_{(i,j)}(\hat{t}-1) + D \cdot q_{(i,j)}(\hat{t}-1) \end{aligned} \ . \tag{7}$$

Upon the contour pheromone field $\tau^C(\hat{t})$ is formed, we first treat it as an input image, and then three steps of morphological operations, including bridging unconnected pixels, filling image regions and holes, and removing interior pixels, are done to generate the contour of each cell.

3 Experiments

The performance of our proposed cell contour estimate algorithm is evaluated using two challenging low-contrast multiple cell image sequences, which considers various cases including cell dynamic difference, cell shape variation, and varying number of cells. In terms of the initial ant colony distribution, a predefined threshold is used to allocate ants to corresponding pixels. A larger value of threshold could generate fewer ants and save computational burden at the expense of the loss of more useful information, whereas a smaller value improves tracking accuracy with more ants generated and more computational cost required. To obtain the desirable tracking results, we set the threshold to be 0.1 in both image sequences.

Figs. 2 and 3 give the multi-cell contour estimates for these two sequences, and it can be observed that our method could give an accurate contour estimate of each existing cell of an image.

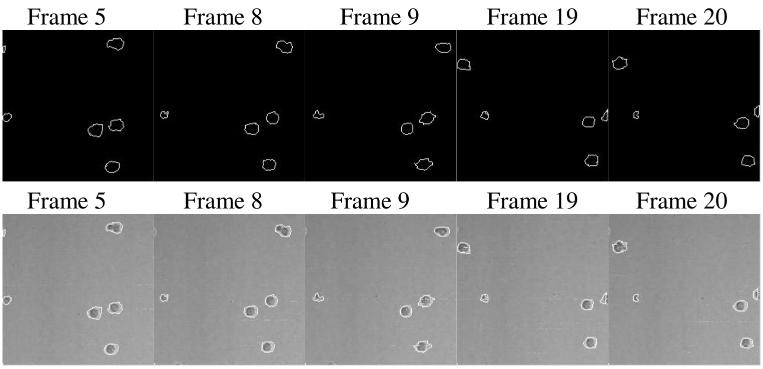


Fig. 2. Cell contour estimates of image sequence 1

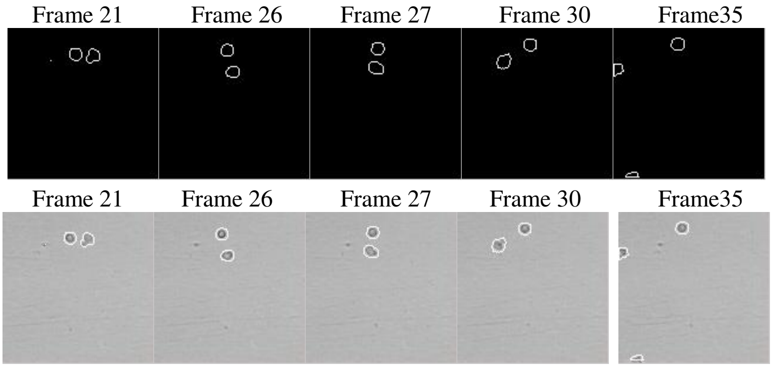
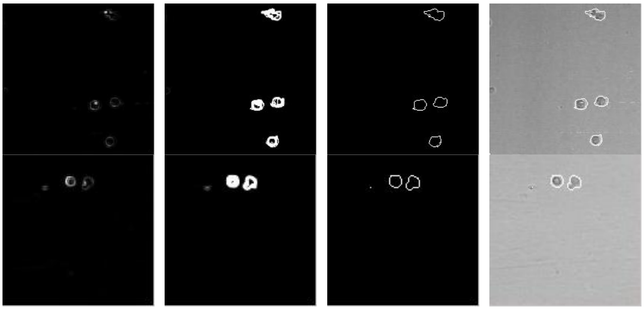


Fig. 3. Cell contour estimates of image sequence 2

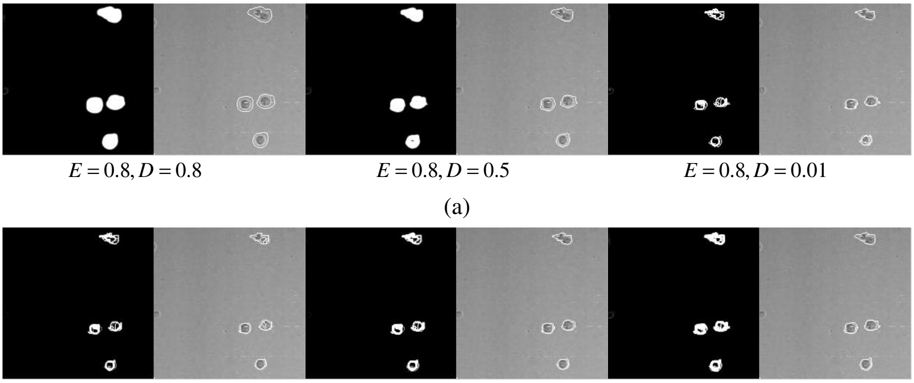
Also, we note that, despite the weakness in intensity and the discontinuity of potential edges in the normalized grayscale variance field $\Delta\tilde{\sigma}_{(i,j)}$ (Fig.4(a)), our algorithm could obtain the cell contour pheromone field in the form of a series of continuous and close loops, as shown in Fig.4(b). Through three steps of morphological operations, all cell-related contours are extracted in the end, as illustrated in Fig.4 (c).

Since cell contour estimate is dependent directly on the contour pheromone field, and the working mechanism of pheromone is appropriately adjusted to form multiple close, smooth and continuous belt loops. As shown in Fig.5 (a), if the propagation coefficient D increases, it means that the effect of propagation increases as well, and the continuity of each contour is guaranteed but the size of contour is larger than the true one. For a smaller value of D, it will result in contour discontinuity and debris due to lack of link bridge between neighboring pixels in terms of pheromone. Similarly, for a smaller value of E, which means that more current pheromones evaporate and only few are used for the following iteration, an undesirable estimate of each contour is achieved as a net structure. However, with less pheromone evaporation, i.e., a larger value of E, more pheromone are utilized as a guide for ant to search for possible segment of contour, as illustrated in Fig.5 (b).



(a) (b) (c) (d)

Fig. 4. Cell contour estimate



$E = 0.8, D = 0.8$ $E = 0.8, D = 0.5$ $E = 0.8, D = 0.01$

(a)

$E = 0.1, D = 0.1$ $E = 0.5, D = 0.1$ $E = 0.9, D = 0.1$

(b)

Fig.5. Sensitivity analysis of various combination of E and D on contour estimate.

4 Conclusions

Cell motion analysis has become a major research direction for understanding the full potential of time-lapse microscopy in biological research or drug discovery. In this paper, we propose an ant pheromone based approach to accurately extract the contours of multiple small cells. Experiment results show that 1) our algorithm could give an accurate estimate of contour of each cell in various scenarios; 2) the tracking accuracy depends on how the pheromone field models, which is affected mainly by parameters E and D. As part of future work, we would like to expand the scope of cell contour estimate to multi-parameter joint estimate, which could give us a broad quantitative view of cell cycle progression.

Acknowledgments. This work is supported by national natural science foundation of China (61273312) and natural science foundation of higher education colleges in Jiangsu (14KJB510001).

References

1. Li, L., Ren, Y.M., Gong, X.P.: Medical Image Segmentation Based on Modified Ant Colony Algorithm with GVF Snake Model. In: 2008th IEEE International Seminar on Future BioMedical Information Engineering, pp. 11–14. IEEE Press, Washington (2008)
2. Ruberto, C.D., Morgera, A.: ACO contour matching a dominant point approach. In: 2011 4th IEEE International Congress on Image and Signal Processing, pp. 1391–1395. IEEE Press, Shanghai (2011)
3. Kaick, O.V., Hamarneh, G., Zhang, H., Wighton, P.: Contour Correspondence via Ant Colony Optimization. In: 2007 15th Pacific Conference on Computer Graphics and Applications, pp. 271–280. IEEE Press, Washington (2007)
4. Chang, Y.H., Lai, D.X., Zhong, Z.H.: Active Contour Tracking of Moving Objects Using Edge Flows and Ant Colony Optimization in Video Sequences. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 1104–1116. Springer, Heidelberg (2009)
5. Li, J.: Image Contour Extraction Based on Ant Colony Algorithm and B-snake. In: Huang, D.-S., Zhao, Z., Bevilacqua, V., Figueroa, J.C. (eds.) ICIC 2010. LNCS, vol. 6215, pp. 197–204. Springer, Heidelberg (2010)
6. Tao, W., Jin, H., Liu, L.: Object segmentation using ant colony optimization algorithm and fuzzy entropy. J. Pattern Recognition Letters 28(7), 788–796 (2007)
7. Agrawal, P., Kaur, S., Kaur, H., Dhiman, A.: Analysis and Synthesis of an Ant Colony Optimization Technique for Image Edge Detection. In: ICCS 2012th International Conference on Computing Sciences, pp. 127–131. IEEE Press, Washington (2012)
8. Haase, R., Böhme, H.J., Perrin, R., Zöphel, K., Abolmaali, N.: Self-reproduction versus transition rules in ant colonies for medical volume segmentation. In: Dorigo, M., Birattari, M., Blum, C., Christensen, A.L., Engelbrecht, A.P., Groß, R., Stützle, T. (eds.) ANTS 2012. LNCS, vol. 7461, pp. 316–323. Springer, Heidelberg (2012)
9. Huang, P., Cao, H.Z., Luo, S.Q.: An artificial ant colonies approach to medical image segmentation. J. Comput Methods Programs Biomed 92, 267–273 (2008)
10. Baterina, A.V., Oppus, C.: Image Edge Detection Using Ant Colony Optimization. J. WSEAS Transactions on Signal Processing 6, 58–67 (2010)
11. Muhammad, A., Bala, I., Salman, M.S., Eleyan, A.: Discrete Wavelet Transform-based Ant Colony Optimization for Edge Detection. In: 2013th International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), pp. 280–283. IEEE Press, Konya (2013)
12. Pereira, C., Goncalves, L., Ferreira, M.: Optic disc detection in color fundus images using ant colony optimization. J. Med Biol Eng Comput 51, 295–303 (2013)

A Novel Ant System with Multiple Tasks for Spatially Adjacent Cell State Estimate

Mingli Lu, Benlian Xu, Peiyi Zhu, and Jian Shi

School of Electrical & Automatic Engineering, Changshu Institute of Technology,
215500 Changshu, China
luxiaowenwp@sohu.com, xu_benlian@cslg.cn,
zpy2000@126.com, yievans2010@hotmail.com

Abstract. Multi-cell tracking is an important problem in studies of dynamic cell cycle behaviors. This paper models a novel multi-tasking ant system that jointly estimates the number of cells and their individual states in cell image sequences. Our ant system adopts an interactive mode with cooperation and competition. In simulations of real cell image sequences, the multi-tasking ant system integrated with interactive mode yielded better tracking results . Furthermore, the results suggest that our algorithm can automatically and accurately track numerous cells in various scenarios, and is competitive with state-of-the-art multi-cell tracking methods.

Keywords: Ant System, Cell Tracking, Object Motion Analysis.

1 Introduction

During cell image sequencing, two or more cells will very likely contact or present occlusions. In such cases, the image is not easily associated with spatially adjacent images because the joint observation cannot be easily segmented. Because the components of corresponding cell states are now coupled, the tracking of spatially adjacent cells or cell occlusions becomes a challenging task, rendered more complicated by low SNR in the image data. For efficiency and accuracy, the development of automated tracking methods for spatially adjacent cell is of great importance.

Many efforts have been made over the past decades. Dufour *et al.* [1] presented a fully automated technique for segmenting and tracking cells in 3-D+time microscopy data. This method uses coupled active surfaces with or without edges, together with a volume conservation constraint and several optimizations to handle touching and dividing cells, and cells entering the field of view during the sequence. In the method of Nguyen *et al.* [2], multiple cell collisions cells are automatically tracked by modeling the appearance and motion of each collision state, and testing collision hypotheses of possible state transitions. Although some of the above algorithms have resolved special challenges in spatially adjacent objects, they are problem-dependent and not applicable to generic cell tracking problems.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 244–252, 2014.
© Springer International Publishing Switzerland 2014

In this paper, we employ a novel ant system with multiple tasks that jointly estimates the number of cells and their individual states in sequences of cell images. Depending on the initial distribution of the ant colony, the colony is roughly divided into several groups, each assigned the task of finding a potential cell through the defined ant working mode, namely interactive mode with cooperation and competition.

2 Methods

In this section, we introduce a novel ant system with multiple tasks (AS-MT) for estimating multi-cell parameters. It is noted that our algorithm builds solutions in a parallel way on $N+1$ pheromone fields (where N is the initial number of divided ant groups),whereas the conventional ACO algorithm computes solutions in an incremental way on only one pheromone field.

2.1 Initial Distribution of Ant Colony

Since the background in most cell image sequences exhibits slowly varying background signals, such a problem can be solved by simplistic, static-background models. In our work, the approximate median method, a kind of recursive technique, is employed for the purpose of fast background subtraction [3], in which each pixel in the background model is compared to the corresponding pixel in the current frame, and finally to be incremented by one if the new pixel is larger than the background pixel or decremented by one if smaller. With the iteration evolves, a pixel in the background model converges to a value where half of the incoming pixels are greater than and half are less than its value, and this value is known as the median.

The approximate median foreground detection compares the current frame to the background model and further identifies the foreground pixels $I_1(i)$ and binary image pixels $I_2(i)$ as

$$I_1(i)=\begin{cases} I(i), & if\ |I(i)-B(i)|>Th \\ 0, & otherwise \end{cases} \tag{1}$$

$$I_2(i)=\begin{cases} 1, & if\ |I(i)-B(i)|>Th \\ 0, & otherwise \end{cases} \tag{2}$$

where $I(i)$ denote the current frame pixels, $B(i)$ are background pixels estimated by recursive technique , $B(i)=\begin{cases} B(i)+1 & if\ I(i)>B(i) \\ B(i)-1 & if\ I(i)<B(i) \end{cases}$, and Th is a predefined threshold. If the absolute difference between current frame pixel and background pixel is larger than the predefined threshold Th , the foreground pixel $I_1(i)=I(i)$,

otherwise $I_1(i)=0$. Binary image pixel $I_2(i)$ follows the same rule as Eq. (2). We further assume that an ant is distributed at the location of pixel i if $I_2(i)=1$ in the binary image, and thus a given number of birth ants are generated in the current frame. Using the K-means clustering method[4], these ants are further divided into N subgroups.

2.2 Ant System with Multiple Tasks

In this section, working mode in ant system with multiple tasks (AS-MT) is proposed and investigated in details. Each ant can move directly towards one of it neighbors at each time, and any pixel can be visited simultaneously by several ants with the guidance of our proposed pheromone update mechanism.

Interactive Mode with Cooperation and Competition (IMCC). In our defined interactive mode with cooperation and competition (IMCC), ants with different tasks are modeled to work together with appropriate cooperation and repulsion. The repulsion term is characterized by $\frac{\tau_j^s(t)}{\tau_j(t)}$ indicated the ratio of pheromone level of task s to total pheromone level at the t-th iteration, and the larger the pheromone amount of task s, the more important this pheromone field will play in ant decision. while the cooperation is represented by the total pheromone $\tau_j(t)$. Therefore, the model of ant decision is a function of the pheromone amount $\tau_j(t)$, heuristic information function η_j and the pheromone amount of current corresponding task $\tau_j^s(t)$, which is formulated as

$$P_{i,j}^{s,k}(t)=\begin{cases}\dfrac{\left[\tau_j(t)\right]^{\alpha}\eta_j^{\beta}\left[\dfrac{\tau_j^s(t)}{\tau_j(t)}\right]^{\gamma}}{\sum\limits_{j\in H(i)}\left[\tau_j(t)\right]^{\alpha}\eta_j^{\beta}\left[\dfrac{\tau_j^s(t)}{\tau_j(t)}\right]^{\gamma}}, & \textit{if } j\in H(i)\\ 0, & \textit{otherwise}\end{cases}\tag{3}$$

where $H(i)$ denotes the set of neighbors of pixel i , $\tau_j(t)$ is the total sum of pheromone amount left by all ants with different tasks on pixel j, and parameters α, β and γ regulate the relative importance of corresponding terms. It is noted that, during the process of searching solutions, each ant of a given task is assumed to sense some information of its neighboring pixels such as the total pheromone amount $\tau_j(t)$, the ratio of pheromone level of a given task to total pheromone level

$\frac{\tau_j^s(t)}{\tau_j(t)}$, and the heuristic function η_j. In the definition of IMCC, if both the relative proportion of pheromone s and the total pheromone amount keep in high level at pixel j, the ant of task s will select the corresponding pixel j as its next position with a lower probability than the relative proportion term is considered only, since both the ant cooperation and competition between different tasks are in effect simultaneously in a trade-off mode.

Heuristic Information. If an ant moves from pixel i to pixel j, the corresponding heuristic value can be defined as

$$\eta_j = e^{-u(1-\frac{1}{|T|}\sum_{i=1}^{|T|}\sum_{j=1}^{M}\min(w_i(j),\tilde{w}_i(j)))^{\upsilon}} \tag{4}$$

Where μ and υ are the adjustment coefficients designed for achieving better likelihood difference comparison between the candidate blob and cell sample blobs, η_j lies in the range of 0 and 1, $\tilde{w}_i(j)$ denotes the value of the j-th element of $\tilde{w}_i$ in cell sample pool, $w_i(j)$ denotes the histogram at pixel j, M is the total number of elements in histogram w, and $|T|$ is the number of cell samples in template pool.

2.3 Merge and Prune Processes

Considering the different ant pheromone fields, if more than one ant groups tend to search for the same cell, the corresponding pheromone fields are probably partially overlapped and the absolute distance between pheromone peaks is relatively small. However, for those spatially distant ant groups, they naturally search for different objects, and the peak distances between ant pheromone fields are easily discriminated and well separated. Therefore, the overlapping ratio $O_{overlap}$ based on pheromone peak is calculated between pheromone peak blobs and treated as a criterion. In our experiments, the merging procedure is performed between two pheromone peak blobs if the overlapping ratio $O_{overlap} > \sigma$, where σ is threshold and set to $\sigma = 0.3$ in our studied cell image data.

In order to remove the false alarms caused by noise and clutters, prune procedure is employed. Suppose that we have the prior information on cell size, if the number of ants of group is less than threshold, the prune processes is carried out and the irrelevant object is removed. Finally, data association based on the easily-implementing nearest neighboring method between frames is done to establish individual trajectories of interested cells.

To visualize our proposed algorithm in a full view, we summarize the procedure in Table 1.

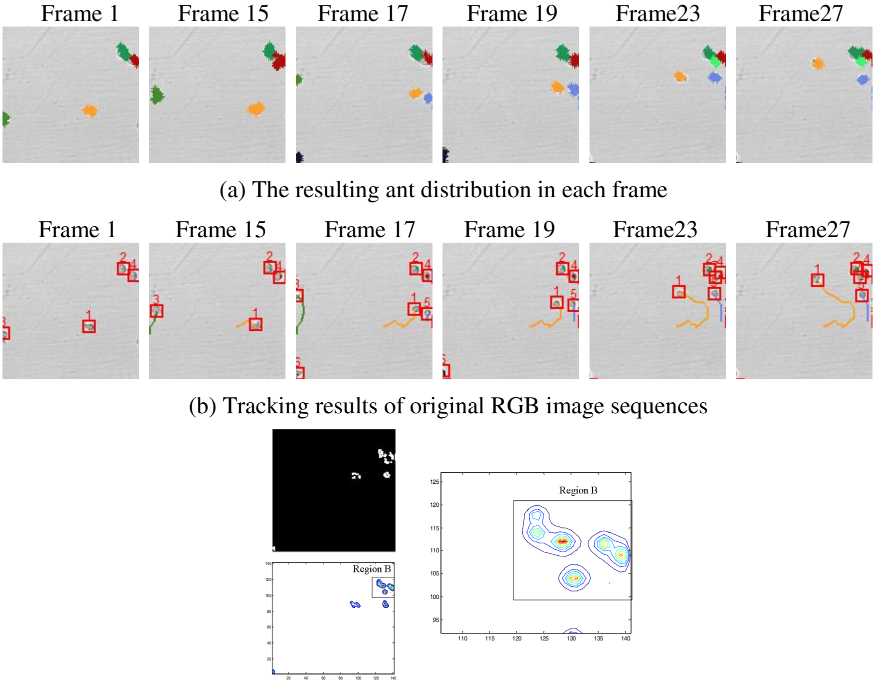
Table 1. Pseudo-code of our proposed algorithm (not considering data association)

```
Input: Image frame by frame
Generate initial distribution of ant groups by the approximate median method;
The initial ant distribution is roughly divided into $N$ groups using K-means method;
$t = 1, q_j^s(0) = 0, \tau_j^s(0) = c$;
While $t < t_{\max}$
  For task $s = 1:N$
    For ant $k = 1:K$
      Ant $k$ moves from pixel $i$ to pixel $j$ with a probability:
      $P_{i,j}^{s,k}(t) = \begin{cases} \dfrac{\left[\tau_j(t)\right]^{\alpha} \eta_j^{\beta} \left[\dfrac{\tau_j^s(t)}{\tau_j(t)}\right]^{\gamma}}{\sum_{j \in H(i)} \left[\tau_j(t)\right]^{\alpha} \eta_j^{\beta} \left[\dfrac{\tau_j^s(t)}{\tau_j(t)}\right]^{\gamma}}, & if\ j \in H(i) \\ 0, & otherwise \end{cases}$
      Deposit corresponding pheromone amount according to $\Delta r_j^{k,s}(t) = \Delta\tau_0$ ;
    end
    Propagated input to pixel $j$
      $q_{1,j}^s(t) = \sum_{j' \in H(j)} \frac{P}{|H(j')|} r_{j'}^s(t) = \frac{P}{|H(j')|} \sum_{j' \in H(j)} \sum_k \Delta r_{j'}^{s,k}$ ;
    Pheromone update on each pixel at task $s$ $\tau_j^s(t+1) = \rho\tau_j^s(t) + r_j^s(t) + q_j^s(t)$ ;
    Propagated input evolution $q_j^s(t) = q_{1,j}^s(t) + \sum_{j' \in H(j)} \frac{P}{|H(j')|} q_{j'}^s(t)$ ;
  end
    Total pheromone $\tau_j(t+1) = \sum_s \tau_j^s(t+1)$ ;
end
If the blob overlap ratio between two pheromone field peaks is greater than a given threshold then
  The merge process is performed.
end
If the number of ant group is less than the given threshold then
  the prune processes is carried out.
end
Output: Cell state
```

3 Experiments

In this section, we will test the tracking performance of our proposed algorithm in terms of cells dividing, different dynamics and varying number in cell image sequences. To evaluate the tracking accuracy between frames, we adopt three measure criterions, namely, label switching rate (LSR), lost tracks ratio (LTR) and false tracks ratio (FTR). The label switching rate is the number of label switching events normalized over total number of ground truth tracks crossing event which happen when two objects get very close each other(and they are sometimes merged into one object) and after they are separated, one object would be treated as a new object and its label is changed. The lost tracks ratio is the number of tracks lost over total number of ground truth tracks. The false tracks ratio is the number of false object that are tracked over total number of ground truth tracks. All experiments were performed in MATLAB (R2012a) on a 1.7 GHz processor computer with 4G random access memory.

Fig.1 presents an example of tracking results of selected images with our proposed algorithm.



(a) The resulting ant distribution in each frame

(b) Tracking results of original RGB image sequences

(c) Initial ant colony distribution and the resulting of ant pheromone field in frame 23

Fig. 1. Tracking results of multi close moving cells($\rho = 0.8, P = 0.6, \alpha = 2.5, \beta = 1, \gamma = 1.1$)

According to the tracking results presented in Fig.1(b), our proposed algorithm could tackle the following challenging cases: cell 3 partly enter the field of view in frame 1, then moves left, partially leaves the field of view in frames 15, and fully leaves the field of view in frame 17. New cells 6 and cell 5 enter the field of view in frame 15, cells 6 leaves the field of view in frame 19. Cell 4 divides into two cells (cell 4 and cell 7) in frame 23. All cells are kept on being tracked with our algorithm in the following frames. It can be observed that initial ant distribution of three spatially adjacent cells is adhered in frame 23, with the cooperation and compete of our proposed algorithm all spatially adjacent cells are successfully separated and tracked. After 50 times of iteration, the adhesion of pheromone field is well separated. All these are illustrated in Fig. 1(c). In addition, Figs. 2 and 3 plot the position and instant velocity estimates of each cell. It can be seen that cell 1 undergoes fast dynamics, and cell 3 also moves quickly both in x and y direction.

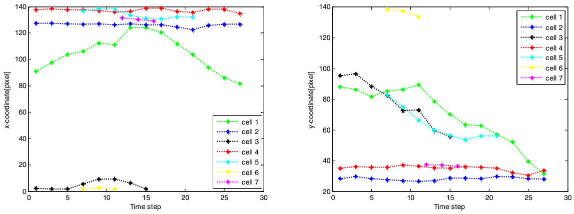


Fig. 2. The position estimate of each cell in x and y directions

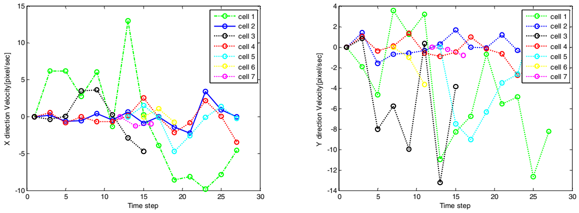


Fig. 3. The instant velocity estimate of each cell in x and y directions

To get insight into tracking performance of our proposed algorithm, we have thoroughly compared our proposed algorithm based on mode IMCC with other three recently developed multi-cell tracking algorithms, i.e., the particle filter (PF) [5] , the multi-Bernoulli filter [6] and Gaussians Mixture Probabilistic Hypothesis Density (GM-PHD) filter [7]. To perform an objective and fair comparison, both the PF and GM-PHD filter use the same detection data obtained by a hybrid cell detection algorithm [8] due to the fact that these two belong to "detect-before-track" methods; meanwhile, for the multi-Bernoulli filter, the likelihood function takes the same form as the heuristic information used in our ant system, denoted by Eq. (4), since both fall in the category of "track-before-detect" techniques. We record all label switching reports, tracks lost reports and false track reports in each frame over 50 Monte-Carlo simulations, and the averaged LSR, LTR and FTR are computed as illustrated in Table 2. According to the statistic results in Table 2, the averaged LSR, LTR and FTR are only ***1.57%***, ***3.17%*** and ***1.57%***, respectively, using our algorithm. The comparison results demonstrate that our algorithm performs better than the other methods in the case cells closely.

Table 2. Comparison results for tracking performance of various methods

Method	LSR(%)	LTR(%)	FTR(%)
PF	15.87	9.52	19.05
Multi-Bernoulli filter	12.39	11.11	17.46
GM-PHD	9.52	6.34	3.17
Our method	1.57	3.17	1.57

Without loss of generality, we present the averaged position errors using the manual tracking result as the ground truth. The comparison of cell 1 position error estimates per frame by various methods is shown in Fig.4 and the same conclusions are drawn as the above.

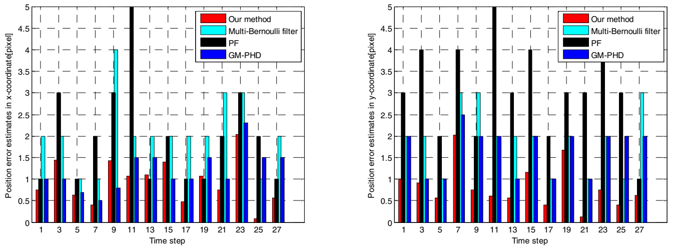


Fig. 4. The comparison of cell 1 position error estimates by various methods

4 Conclusions

The problem of properly tracking spatially adjacent objects is one of the most difficult issues in automated cell tracking. In this paper, a novel ant system with multiple tasks is modeled for jointly estimating the number of cells and individual states in cell image sequences. According to statistic results, our ant system with multiple tasks algorithm demonstrates a robust tracking performance in terms of the measures of LSR, LTR and FTR when comparing with other three recently developed multi-cell tracking algorithms.

Acknowledgments. This work is supported by national natural science foundation of China (No.61273312) and the natural science fundamental research program of higher education colleges in Jiangsu province(No. 14KJB510001).

References

1. Dufour, A., Shinin, V., Tajbakhsh, S., Guillen-Aghion, N., Olivo-Marin, J.C., Zimmer, C.: Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. IEEE Transactions on Image Processing 14, 1396–1410 (2005)

2. Nguyen, N.H., Keller, S., Norris, E., Huynh, T.T., Clemens, M.G., Shin, M.C.: Tracking Colliding Cells In Vivo Microscopy. IEEE Transactions on Biomedical Engineering 58, 2391–2400 (2011)
3. Bandi, S.R., Varadharajan, A., Masthan, M.: Performance evaluation of various foreground extraction algorithms for object detection in visual surveillance. Comput. Eng. Res. 2, 1339–1443 (2012)
4. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics) 28, 100–108 (1979)
5. Smal, I., Draegestein, K., Galjart, N., Niessen, W., Meijering, E.: Particle Filtering for Multiple Object Tracking in Dynamic Fluorescence Microscopy Images: Application to Microtubule Growth Analysis. IEEE Transactions on Medical Imaging 27, 789–804 (2008)
6. Hoseinnezhad, R., Vo, B.-N., Vo, B.-T., Suter, D.: Visual tracking of numerous targets via multi-Bernoulli filtering of image data. Pattern Recognition 45, 3625–3635 (2012)
7. Juang, R.R., Levchenko, A., Burlina, P.: Tracking cell motion using GM-PHD. In: IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2009, pp. 1154–1157 (2009)
8. Lu, M., Xu, B., Sheng, A.: Cell automatic tracking technique with particle filter. In: Tan, Y., Shi, Y., Ji, Z. (eds.) ICSI 2012, Part II. LNCS, vol. 7332, pp. 589–595. Springer, Heidelberg (2012)

A Cluster Based Method for Cell Segmentation

Fei Wang, Benlian Xu, and Mingli Lu

School of Electrical & Automatic Engineering,
Changshu Institute of Technology,215500 Changshu, China
wangleea@aliyun.com

Abstract. In the field of cell biology, cell segmentation is an essential task in biomedical application. For this purpose, a cluster based method for cell segmentation is proposed. Firstly, an ant colony clustering algorithm is used to make pre-segmentation from which cell candidates are identified, then some noise spots are filtered with area feature, after that, a novel cluster algorithm is proposed to divide adhering cells into individuals. Finally, good results of segmentation can be achieved. Experimental result show that the method remains both the advantage of image segment of ant colony cluster and the ability of further process of pre-segmentation, which improves the performance of cell segmentation.

Keywords: Ant colony algorithm, clustering algorithm, cell segmentation.

1 Introduction

Many scientific biological applications as well as experiments for drug development require the observation of cell responses to a variety of stimuli. Some of the responses that need to be quantified are cell migration, cell proliferation, and cell differentiation. The corresponding conclusions require the observation of cells over extended periods of time. An effective way to achieve this is with microscopy images. However, the resulting data sets of images are large and their manual analysis is tedious, subjective, and restrictive. Thus, an automated technology for analysis is needed urgently, of all the image based research of cells, cell segmentation is the very first step.

Many image segmentation method have been used in cell segment, like thresholding [1], gradient based methods [2], the watershed algorithm [3], level sets [4], dynamic programming [5] and various other pattern analysis and machine learning algorithms [6]. Ant colony clustering algorithm is a new swarm intelligence method that has been used in many fields, like data mining, document retrieval, image segmentation and so on. In this paper, an ant colony clustering algorithm is applied to achieve cell pre-segmentation process, then, a noise filter process is implemented, and after that a novel clustering algorithm is proposed to divide the adhering cells into individuals.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 253–258, 2014.
© Springer International Publishing Switzerland 2014

2 Framework of Cell Segmentation

In the process of cell segmentation, an ant cluster based method is applied to execute rough segmentation, then another cluster algorithm and features filter step are used to improve the performance of cell segmentation, its flow chart is shown as follows.

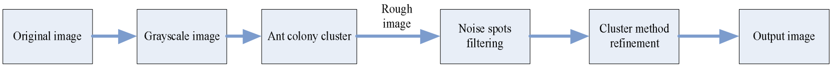


Fig. 1. Flow chart of cell segmentation

2.1 Ant Colony Clustering

Ant colony clustering algorithm is an improved ant colony method which was proposed by Indian researcher P.S. Shelokar. As a globally optimized heuristic method, it is used to address clustering problems and applied to integrate data according to the information around the clustering centers [7]. During application of the algorithm, every data sample is treated as an ant who has different attributes, and the process for searching food by ants is regarded as clustering procedure.

Give the initial image X, and look every pixel X_i (i = 1, 2, …, n) as an ant, every ant stands for the feature vector of pixel. Image segmentation is the process that these ants with different feature vector searching food source. The distance of any pixel X_i to X_j is d_{ij}, using Euclidean distance to calculate:

$$d_{ij} = \sqrt{\sum_{k=1}^{m} p_k (x_{ik} - x_{jk})^2} \quad . \tag{1}$$

where m is the number of feature vector, p is the weighted factor, which is set by its attribute and subjected to equation: $\sum p_k = 1, p_k \geq 0$. $\tau_{ij}(t)$ is the pheromone information value of the arc linking the data X_i to data X_j at time t, and is given as:

$$\tau_{ij}(t) = \begin{cases} 1 & d_{ij} \leq r \\ 0 & d_{ij} > r \end{cases} \quad . \tag{2}$$

r is the radius of clustering. The probability of path X_i to X_j is given by the following equation:

$$P_{ij}(t) = \frac{\tau_{ij}{}^{\alpha}(t)\eta_{ij}{}^{\beta}(t)}{\sum_{s \in S} \tau_{ij}{}^{\alpha}(t)\eta_{ij}{}^{\beta}(t)} \quad . \tag{3}$$

$$\eta_{ij} = r/d_{ij} \quad . \tag{4}$$

where η_{ij} is the Heuristic Guide function, which reflects the similarity between the pixels and the clustering center, α and β are regulating factors, which respectively reflect the accumulated information during the ants moving process, and the relative importance of the heuristic information in the ants path selecting process. If $p_{ij}(t) \geq p_0$, X_i would be clustered into the cluster where its X_j belongs to. Let $C_j = \left\{ X_k \mid d_{kj} \leq r, k = 1, 2, ..., J \right\}$, C_j denotes the collection of data which converge into the neighborhood of X_j . Then, the clustering center is given as:

$$C_j = \frac{1}{J} \sum_{k=1}^{J} X_k \quad . \tag{5}$$

With the ants moving, the pheromone information value on every path is changing. Through one circulation, the pheromone information value on each path is adjusted according the following formula:

$$\tau_{ij}(t) = \rho \tau_{ij}(t) + \Delta \tau_{ij} \quad . \tag{6}$$

where ρ represents the evaporating degree of pheromone information value with the elapse of time, $\Delta \tau_{ij}$ is the augmentation of path pheromone information value in this circulation.

$$\Delta \tau_{ij} = \sum_{k=1}^{n} \Delta \tau_{ij}^{k} \quad . \tag{7}$$

$\Delta \tau_{ij}^{k}$ is the pheromone information value remained by the *k*th ant.

The following sections discuss a new way to solve the pre-segmentation image problems by combining a novel clustering algorithm and a noise spots filtering process.

2.2 Refinement Process

After ant colony clustering process, a rough cell segmentation image is obtained, from which we can get adhering cells and some noise spots. So a refinement process is needed in which the noise spots filtering process and adhering cells segment process are included.

Aim at filtering noise spots, we use area feature of cells which expressed as:

$$S = \sum s_x \quad s_x = \begin{cases} 1 & x \in cell \\ 0 & x \notin cell \end{cases} \quad . \tag{8}$$

s_x is unit pixel and subjects to constraints: $\min(S) < S < \max(S)$, where $\max(S)$ and $\min(S)$ are areas that corresponding to the maximum area and minimum area of cells. Any objects that do not satisfied cell features are considered as noise. Comparing with noise spots, cells have relatively stable feature, which facilitates removing mismatch objects and reduces or eliminates noise spots.

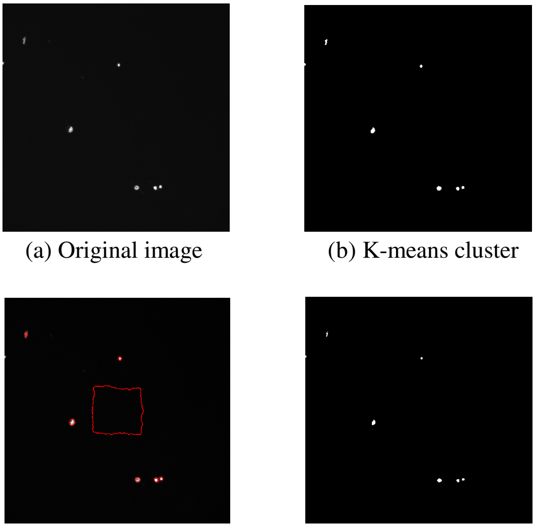
After noise filtering process, a clustering method is used to find individuals of cells, the clustering algorithm is shown below:

1. Initialize the set of clusters to the empty set, $S = \phi$
2. Find a cluster C in S, such that for all C_i in S $dist(C, x_i) \leq dist(C_i, x_i)$
3. If $dist(C, x_i) \leq w$, then associate x_i with the cluster C. otherwise a new cluster is created $S \leftarrow S \bigcup \{C_n\}$, where C_n is a cluster with x_i
4. Repeat step 2 and 3, until now instances are left.

where x_i is data samples, C_i is the cluster in S. After this step all cells should be identified.

3 Results and Discussion

The experiment was made under the environment of Inter I5 2410M, win7 64bit, Matlab 2011b. As for parameters $\alpha = 2, \beta = 3, r = 10, p_0 = 0.9$ were selected. Fig 2 is a simple cell image which has few cells. We use level set method, k-means method and our method to process this image, and the results are shown as follows:



(a) Original image

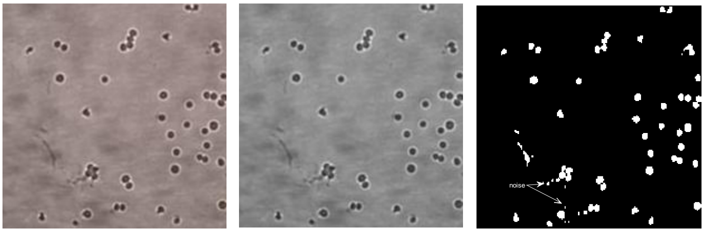
(b) K-means cluster

(c) Level set

(d) Our method

Fig. 2. Simple cell image segmentation

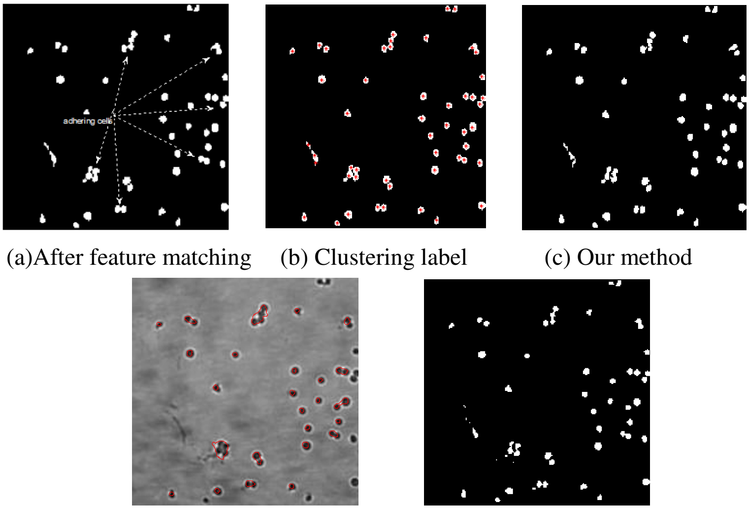
From the results shown above we can tell that at this degree, three methods share the same detection results, but level set had the longest time consuming for its iteration. We then choose an image from a frame of the sequence images which is shown as Fig 3(a). From the image, we can see that there are many cells, some of which are adhering cells or the noises of background. Fig 3(b) is the gray scale image of Fig 3(a). The gray scale image is operated by ant colony clustering method and the pre-segmentation image is obtained as Fig 3(c).



(a) Original image (b)Gray-scale image (c)Pre-segmentation

Fig. 3. Pre-segmentation of multi-cell image

In Fig 3(c), most of cells are classified as white spots. Some spots are cells, some are noise spots, which must be filtered. Then filtering process is carried out as mentioned in section 2, the result is shown as Fig 4(a). From Fig 4(a), we can see that many noise spots have been erased but still one big noise spot remained. After this step, the cluster method that proposed in this paper is used to make refinement. We first use this method find all possible cells in the image, as we can see from Fig 4(b), each red dot denote one cell. Then, find the connected objects, we get the conclusion that it is the adhering cell if more than one clusters are included in one connected object. The clusters were found with the method of cut off the line between two



(a)After feature matching (b) Clustering label (c) Our method

(d) Level set (e) K-means cluster

Fig. 4. Multi-cell segmentation

clusters to divide adhering cells, which is shown as Fig 4(c). Comparing with Fig 4(b) and Fig 4(d), we can see that some adhering cells are separated successfully.

Fig 4(d) is cell segmentation with level set, from this image we can see that some of cells are identified but some connected cells are clustered as one cell and some individuals are not detected. The last figure is k-means based segmentation method, most of cells are classified with this method, but some of them are adhering cells that can hardly be used at subsequent applications. In addition k- means based method still remains more noise spots compared with our method.

4 Conclusions

The cell segmentation is an important problem in biological study. A cluster based method is proposed in this paper, which combines ant colony cluster algorithm with image refinement process. This method remains both the advantage of image segment of ant colony cluster and the ability of further process of pre-segmentation, which improves the performance of cell segmentation. The experiment results confirmed the above point that the method proposed in this paper is effective.

Acknowledgments. This work is supported by National Natural Science Foundation of china (No.61104186) and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No.14KJB510001).

References

1. Xiaobo, Z., Fuhai, L., Jun, Y.: A Novel Cell Segmentation Method and Cell Phase Identification Using Markov Model. J. IEEE Transaction on Information Technology in Biomedicine 13(2), 152–157 (2009)
2. Li, G., Liu, T., Nie, J., Guo, L., Wong, S.T.C.: Segmentation of touching cells using gradient flow tracking. In: 4th IEEE International Symposium on Biomedical Imaging, pp. 77–80. IEEE Press, New York (2007)
3. Muhimmah, I., Kurniawan, R., Indrayanti, I.: Automated cervical cell nuclei segmentation using morphological operation and watershed transformation. In: 2012 IEEE International Conference on Computational Intelligence and Cybernetics, pp. 163–167. IEEE Press, New York (2012)
4. Bergeest, J.-P., Rohr, K.: Efficient globally optimal segmentation of cells in fluorescence microscopy images using level sets and convex energy functionals. J. Medical Image Analysis 16(7), 1436–1444 (2012)
5. McCullough, D.P., Gudla, P.R., Meaburn, K., Kumar, A., Kuehn, M., Lockett, S.J.: 3D Segmentation of whole cells and cell nuclei in tissue using dynamic programming. In: 4th IEEE International Symposium on Biomedical Imaging, pp. 276–279 (2007)
6. Chankong, T., Theera-Umpon, N., Auephanwiriyakul, S.: Automatic cervical cell segmentation and classification in Pap smears. J. Computer Methods and Programs in Biomedicine 13(2), 539–556 (2014)
7. Shelokar, P.S., Jayaraman, V.K., Kulkarni: An ant colony approach for clustering. J. Analytica Chimica Acta 509(2), 187–195 (2004)

Research on Lane Departure Decision Warning Methods Based on Machine Vision

Chuncheng Ma, Puheng Xue, and Wanping Wang

CCCC First Highway Consultants Co., Ltd., Xi'an, Shaanxi, China
{496056424,1464925547,329833165}@qq.com

Abstract. To improve the driving safety of drivers, an effective lane detection algorithm was proposed upon the research on lane departure decision warning system based on machine vision. Firstly, the lane images were preprocessed to adapt to various lighting conditions and improve the efficiency of the lane detection. Then, by means of hough transform, actual lane line features were extracted according to the different image lane line features. Finally, after the study of different lane departure models based on lane line detection, this article put forward a lane departure decision algorithm. Experimental results demonstrate that the developed system exhibits good detection performances in recognition reliability and warning decision. It has proved that this system has high accuracy, large detection range and high practicability.

Keywords: machine vision, lane detection, hough transform, lane departure decision, warning methods.

1 Introduction

Safety Driving Assist (SDA) is the current concern in the research of intelligent transportation system, which mainly solves the traffic security problems. As an essential component in the research field of security auxiliary driving, lane departure decision warning system gets more and more attention in recent 20 years. If the vehicle deviates from the lane or there is any trend of vehicle deviation, the system will warn the tired or absent-minded drivers to alter driving directions, thus reduce lane accidents. As a result, it is of great significance for driving safety.

2 Lane Departure Decision Warning System Based on Machine Vision

Lane departure decision warning system based on machine vision mainly consists of CCD camera, video capture card, PC processor, alarm unit, display equipment and other components [1]. During the high speed running of vehicles, the system uses camera and video capture card to get lane images, and computer processes the digital images to detect real-time image of left and right lane, on which the lane departure decision is made. If the vehicle deviates from the normal lane or has the trend to

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 259–266, 2014.
© Springer International Publishing Switzerland 2014

deviate, the system will send warning information through display and alarm circuit, to remind or warn the driver to alter driving direction.

3 Research on Lane Detection Algorithm

Lane detection in road image is the basis of lane departure decision warning system. The model of lane detection is shown in Figure 1. The road detection image includes some interferential information, while lane line is mainly in the below of image. For comprehensive feature analysis of road image, a threshold was set, and the region below is our research part, and only that part is processed, so that we can enhance the real-time of lane detection.

Researches show that the vehicle lane departure decision can be made by lane field image of myopia [2]. In order to simplify the processing, we can select an appropriate cut point to process the lane line locating in the below of as straight line Therefore, the lane detection becomes the detection of straight line in the region of interest, whose algorithm includes image equalization grayscale processing, edge detection, Hough transform, lane feature extraction and so on.

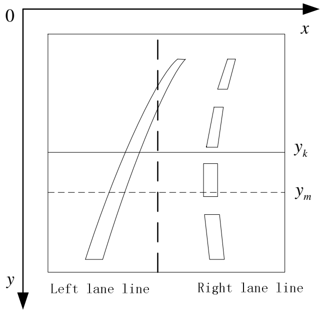


Fig. 1. The model diagram of lane detection

3.1 Image Equalization Grayscale Processing

Lamps and lighting directly affect the quality of machine vision system. As for the road images under different lights, the gray values vary greatly. In strong or low light images, the lane line presents low contrast to the background, and the range of the gray changes is narrow. To improve the adaptability and immunity of the system, the research applies the method of histogram equalization grayscale processing for lane image process.

Histogram equalization is expressed as the cumulative distribution function:

$$S_k = \sum_{i=0}^{k} \frac{n_i}{N} = \sum_{i=0}^{k} p_r(r_i) \tag{1}$$

In this function, $i = 0,1,\ldots k$, $k = 0,1,\ldots \mathrm{L}$, L is the gray level range, $p_r(r_i)$ is the appearance probability of i level gray in the image, N is the total pixels of the road image.

3.2 Edge Detection and Binary Image Processing

Lane detection mainly focuses on the detection of the edge of lane line, and edge detection is realized through the use of an algorithm to extract the lines between objects and background in the image [3]. Changes in the gray image can be reflected by the gradient of image gray distribution, so that differential technique based on local image can be used to obtain edge detection operator [4].

To achieve the purpose of rapid detection, Sobel operator is applied to provide relatively accurate information about edge direction [5].

After edge detection, the road image is processed by binary to reduce the image storage capacity, thus raise lane detection speed. Figure 2 shows the image after edge detection and binary image processing.

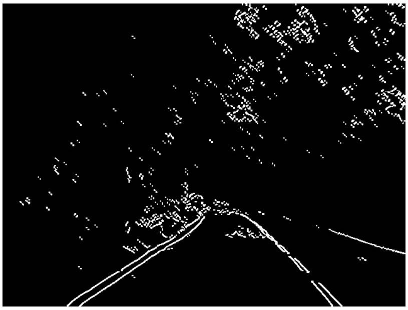


Fig. 2. Edge detection and binary image processing

3.3 Image Hough Transform

The key of lane detection is to detect the line representing real lane. The common method for straight line extraction is Hough transform, the polar coordinates equation of straight line based on Hough transform is as follows [6]:

$$\rho = x\cos\theta + y\sin\theta \tag{2}$$

ρ is the distance from straight line to the origin in image space; θ is the angle between the line and the x-axis. There are two planes: xy is the image plane, $\rho\theta$ is the parameter plane. Linear equation is used to transform the points of the same line on the image planes to a coordinate point (ρ, θ) on the parameter plane. The image after Hough transform is shown in Figure 3.

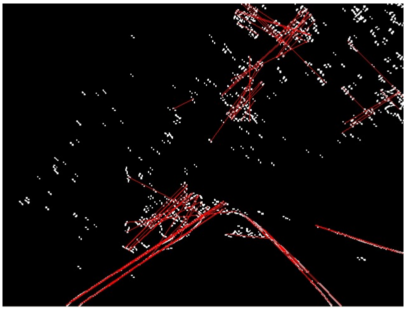


Fig. 3. The image after Hough transform

3.4 Lane Feature Extraction

From Figure 3, it can be seen that there are still some interferential lines after Hough transform, so that it is necessary to filter out such lines. After Hough transform, parameters of the linear can be extracted, gaining all coordinate equations as well as the starting and ending coordinates of line segments [7].

Generally speaking, left lane line locates in the left half of the image and right lane line in the right half; accordingly, the interest image region is divided into two parts, left and right, to identify the coordinating lane line. In the interest image region, the middle part is the road which is relatively flat, with minimal disruption to the Hough transform. Moreover, the road lines locate in the middle of the road, so line searching algorithm from bottom to up and middle to left or right is employed to extract lane features.

Steps for feature extraction of lane lines are as follows:

1) The angles formed by the left and right lane lines with x-axes are set to α and β; the range of α is $(100°,170°)$, and range of β is $(10°,80°)$. At the same time the straight lines which do not meet the requirements are removed. Detection results are shown in Figure 4.

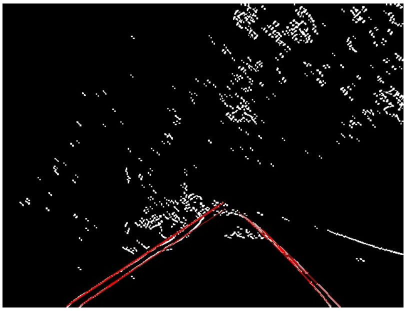


Fig. 4. Lane detection

2) Extract three factors: the length of straight line itself L_Line, the length from the middle of the image to the line L_Center, and the length from the bottom to the

line L_Bottom .Calculate the value of the three factors with each range as R_l , R_c , R_b in the left and right half of the interest region.

3) Figure out the straight lines with the biggest value of comprehensive determine coefficient in left and right half interest region respectively. Comprehensive determine coefficient of lane lines comes to:

$$A = \sum_{i=1}^{3} w_i V_i \tag{3}$$

w_i is the weight of factor values, which is set to $W = (0.4, 0.3, 0.3)$, V_i is the value of factor values.

$$V = \begin{cases} L\_Line / R_l & i = 1 \\ 1 - L\_Center / R_c & i = 2 \\ 1 - L\_Bottom / R_b & i = 3 \end{cases} \tag{4}$$

The straight lines with the biggest value of comprehensive determine coefficient in the left and right half interest region are the lane lines in left and right. Extracted lane lines are shown in Figure 5.

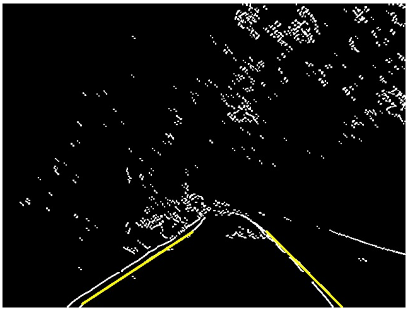


Fig. 5. Extraction Lane Line

4 Lane Departure Decision Warning

When vehicles are on their ways, because of environment changes, lane turning and changing, some errors may appear, resulting in lane line changes between the current frame and former frame in the video, or detection failure of lane line. For such situation, the current image frames are determined to be ineffective and current lane departure warning decision is also eliminated.

Vehicles' driving states mainly contain normal driving, slight left skew, slight right skew, vehicle departure from left lane, and vehicle departure from the right lane. The model of lane departure is given in Figure 6. Two auxiliary horizontal straight lines are set as $y = y_b$ and $y = y_t$ for the lane departure decision. In the Hough transform, the upper and lower endpoints of right and left lane segments are set as (x_{Lt}, y_{Lt}), (x_{Lb}, y_{Lb}) and (x_{Rt}, y_{Rt}) , (x_{Rb}, y_{Rb}) .Then the horizontal coordinates for intersections of left and right lane line with y_b are produced, X_{Lb} and X_{Rb} :

$$X_{Lb} = \begin{cases} x_{Lb} & x_{Lt} = x_{Lb} \\ x_{Lb} - \dfrac{y_b - y_{Lb}}{y_{Lb} - y_{Lt}}(x_{Lt} - x_{Lb}) & x_{Lt} \neq x_{Lb} \end{cases} \tag{5}$$

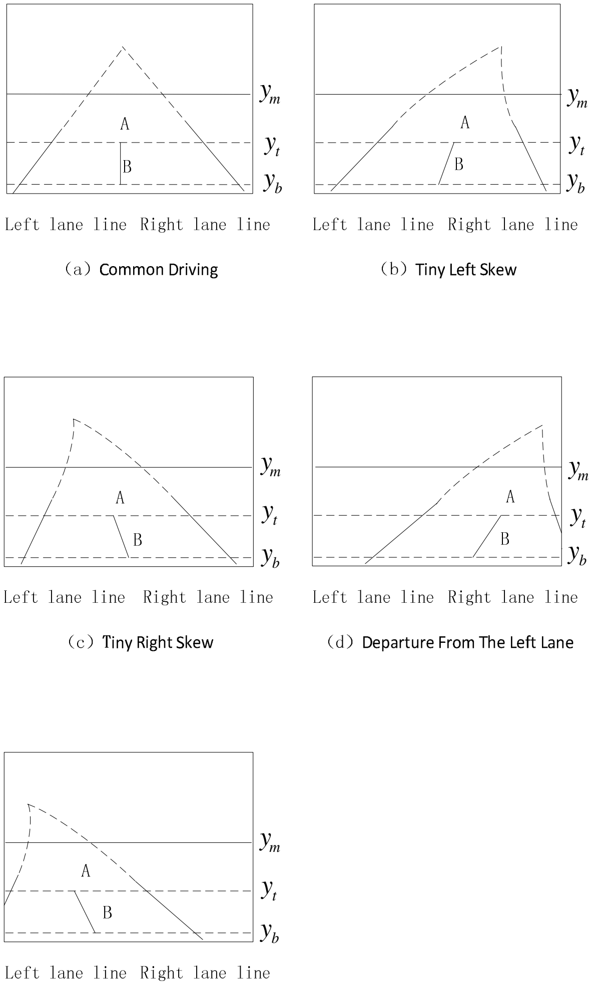
$$X_{Rb} = \begin{cases} x_{Rb} & x_{Rt} = x_{Rb} \\ x_{Rb} + \dfrac{y_b - y_{Rb}}{y_{Rb} - y_{Rt}}(x_{Rb} - x_{Rt}) & x_{Rt} \neq x_{Rb} \end{cases} \tag{6}$$

$$X_{Lt} = \begin{cases} x_{Lb} & x_{Lt} = x_{Lb} \\ x_{Lb} - \dfrac{y_t - y_{Lb}}{y_{Lb} - y_{Lt}}(x_{Lt} - x_{Lb}) & x_{Lt} \neq x_{Lb} \end{cases} \tag{7}$$

$$X_{Rt} = \begin{cases} x_{Rb} & x_{Rt} = x_{Rb} \\ x_{Rb} + \dfrac{y_t - y_{Rb}}{y_{Rb} - y_{Rt}}(x_{Rb} - x_{Rt}) & x_{Rt} \neq x_{Rb} \end{cases} \tag{8}$$

The middle point A of the intersection between Left and right lane line with $y = y_t$ is $(\dfrac{X_{Lt} + X_{Rt}}{2}, y_t)$, and the middle point B of the intersection between Left and right lane line with $y = y_b$ is $(\dfrac{X_{Lt} + X_{Rt}}{2}, y_t)$. The slope of the straight line AB is K_{AB} .

$$K_{AB} = \begin{cases} \infty & \dfrac{X_{Lt} + X_{Rt}}{2} = \dfrac{X_{Lb} + X_{Rb}}{2} \\ \dfrac{y_b - y_t}{\dfrac{X_{Lt} + X_{Rt}}{2} - \dfrac{X_{Lb} + X_{Rb}}{2}} & \dfrac{X_{Lt} + X_{Rt}}{2} \neq \dfrac{X_{Lb} + X_{Rb}}{2} \end{cases} \tag{9}$$



(a) Common Driving

(b) Tiny Left Skew

(c) Tiny Right Skew

(d) Departure From The Left Lane

(e) Departure From The Rihgt Lane

Fig. 6. The model of lane departure

After analyzing a great quantity of vehicle driving states and lane images, setting a left and a right threshold X_{Lb}' and X_{Rb}' as well as threshold K_L and K_R for the slope of straight line AB are assigned to decide whether there is lane departure.

5 Experimental Results and Analysis

The lane departure decision system based on machine vision is developed by the combination of Visual C++ and Open CV Open computer vision library. The processed video image size is 640*480. After a large number of simulation experiments on 1000 continuous frames image sequences from Video, through the lanes detection and the analysis of lane departure decision, we can come to the conclusion as follows: the rate of detection driveway is 98.23%; the success rate of lane departure decision is 97.64%; the time that deciding a road image nearly need 30ms. The final results show that the algorithm of lane detection and lane departure decision can effectively detect and track lane line and accurately make the right decision of lane departure. It is more real-time and anti-jamming. Therefore the system can improve the safety of drivers driving, playing the role of Vehicle Auxiliary Navigation.

6 Conclusion

The paper studies and implements a kind of lane departure decision warning system based on machine vision. It has shown that it can effectively and instantly detect the left, right lane lines and make right decisions on the condition of lane deviations. With favorable ability of lane detection and lane departure decision, this approach can satisfy the requirements of the lane departure decision warning system. Still this algorithm needs a further improvement so as to make sure that the system has the ability to make more accurate decision and lane detection under different atrocious weather circumstances and promotes the lane departure decision warning system to be perfect.

References

1. Jiuqiang, H.: Machine vision technology and applications. Higher Education Press, Beijing (2009)
2. Jung, C.R., Kelber, C.R.: A lane departure warning system using lateral offset with uncalibrated camera. In: Proceedings of the 2006 IEEE Intelligent Transportation Systems, pp. 102–107. IEEE (2005)
3. Huang, S.S., Chen, C.J., Hsiao, P.Y., Fu, L.C.: On-board vision system for lane recognition and front-vehicle detection to enhance driver's awareness. In: 2004 IEEE International Conference on Robotics and Automation, Proceedings. ICRA 2004, vol. 3, pp. 2456–2461. IEEE (2004)
4. Jung, C.R., Kelber, C.R.: A lane departure warning system based on a linear-parabolic lane model. In: 2004 IEEE Intelligent Vehicles Symposium, pp. 891–895. IEEE (2004)
5. Li, X., Zhang, W.: Research on lane departure warning system based on machine vision. Chinese Journal of Scientific Instrument 29, 1554–1558 (2008)
6. Yang, Y., Farrell, J.A.: Magnetometer and differential carrier phase GPS-aided INS for advanced vehicle control. IEEE Transactions on Robotics and Automation 19(2), 269–282 (2003)
7. Zhang, R., Wang, Y., Yang, R.: Researches on Road Recognition in Landsat TM Images. Journal of Remote Sensing 2, 016 (2005)

Searching Images in a Textile Image Database

Yin-Fu Huang and Sheng-Min Lin

Department of Computer Science and Information Engineering
National Yunlin University of Science and Technology
huangyf@yuntech.edu.tw, bhujmn2007@gmail.com

Abstract. In this paper, a textile image search system is proposed to query similar textile images in an image database. Five feature descriptors about the color, texture, and shape defined in the MPEG-7 specification, which are relevant to textile image characteristics, are extracted from a dataset. First, we tune the feature weights using a genetic algorithm, based on a predefined training dataset. Then, for each extracted feature descriptor, we use K-means to partition it into four clusters and combine them together to obtain an MPEG-7 signature. Finally, when users input a query image, the system finds out similar images by combining the results based on MPEG-7 signatures and the ones in three nearest classes. The experimental results show that the similar images returned from an image database to a query textile image are acceptable for humans and with good quality.

Keywords: CBIR, genetic algorithm, K-means, MPEG-7 specification, weight tuning.

1 Introduction

At present, multimedia data have played an important role in our daily life. However, querying a multimedia database by keywords is gradually insufficient to meet users' needs. Thus, facing a huge amount of images in an image database, content-based image retrieval has become a popular and required demand.

In the past years, many general-purpose image retrieval systems have been developed [5, 6, 10], and these systems rely mainly on visual features. King and Lau used MPEG-7 descriptors to retrieve fashion clothes [5]. In order to improve query results, Lai and Chen proposed a user-oriented image retrieval system by iteratively interacting with users about query results [6]. Smeulders et al. presented a review of 200 references in content-based image retrieval [10].

In this paper, we propose a textile image search system for querying similar textile images in an image database. This system consists of an offline phase and an online phase. In the offline phase, we tune the feature weights using a genetic algorithm [11], based on a predefined training dataset. Then, for each extracted feature descriptor, we use K-means to partition it into four clusters and combine them together to obtain an MPEG-7 signature [3]. In the online phase, when users input a query image, the system extracts its MPEG-7 visual features first, and then finds out similar images by combining the results based on MPEG-7 signatures and the ones in three nearest classes.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 267–274, 2014.
© Springer International Publishing Switzerland 2014

The remainder of this paper is organized as follows. In Section 2, we present the system architecture and briefly describe the procedure of searching similar textile images. In Section 3, we introduce five feature descriptors relevant to textile image characteristics. In Section 4, the methods of tuning the feature weights and generating MPEG-7 signatures are proposed to facilitate searching similar images. In Section 5, we present the experimental results to evaluate the effectiveness of three search modes provided in our system. Finally, we make conclusions in Section 6.

2 System Architecture

In this paper, we propose a textile image search system consisting of an offline phase and an online phase, as shown in Fig. 1. In the offline phase, five feature descriptors about the color, texture, and shape defined in the MPEG7 specification are extracted from training images; i.e., ColorLayout descriptor, ColorStructure descriptor, EdgeHistogram descriptor, HomogeneousTexture descriptor, and RegionShape descriptors including a total of 221 dimensions. Because each feature plays a different role in distinguishing a textile image from others, feature weights should be determined in order that the discrimination among textile images could be boosted. Here, we use a genetic algorithm to determine feature weights. Then, we build MPEG-7 signatures using k-means clustering on all textile images where the weighted Euclidean distance calculated in k-means clustering takes the feature weights determined in the genetic algorithm.

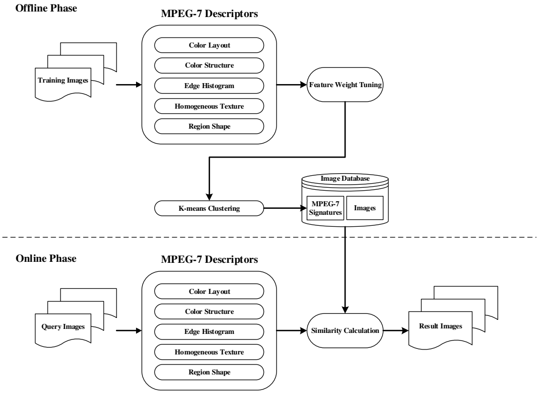


Fig. 1. System architecture

In the online phase, we also extract the same features, as mentioned, from a query image. Then, we find out 1) the images with the same MPEG-7 signature as the query image as the first candidates and 2) the images in three nearest classes to the query image as the second candidates. Finally, we can find out result images most similar to the query image, which appear in both groups of candidates.

3 Feature Extraction

In this paper, we adopt the bag-of-feature MPEG-7 [1, 2, 4, 7-9] defined by the MPEG organization, which consists of color description (i.e., two color descriptors), texture description (i.e., two texture descriptors), and shape description (i.e., one shape descriptor), as shown in Table 1. Among them, the descriptors relevant to textile image characteristics are as follows.

1) ColorLayout Descriptor describes the layout and variation of colors and this reflects the color combinations in a textile image.

2) ColorStructure Descriptor counts the contents and structure of colors in a textile image by using a sliding window.

3) EdgeHistogram Descriptor counts the number of edge occurrences in five different directions in a textile image.

4) HomogeneousTexture Descriptor calculates the energies of a textile image in the frequency space, which are the level of gray-scale uniform distribution and texture thickness, and this reflects the texture characteristics in a textile image.

5) RegionShape Descriptor relates to the spatial arrangement of points (pixels) belonging to an object or a region in a textile image.

Table 1. MPEG-7 visual descriptor features

Type	Feature description	Dim.	Overall statistics	Total number
1	ColorLayout	12	1	12
2	ColorStructure	32	1	32
3	EdgeHistogram	5	16	80
4	HomogeneousTexture	2	31	62
5	RegionShape	35	1	35

4 Method

K-means clustering is an effective approach to find similar images for a query image, but it is usually dependent on how well stored images are clustered. In reality, the Euclidean distance between two images, used in K-means clustering, plays a major role in determining good clustering. In calculating the Euclidean distance between two images, each kind of involved features (or descriptors) mentioned in Section 3 has its own semantic in measuring their similarity; thus, these features should be assigned with weights in measuring the similarity between two images.

In our system, a finest weight set is tuned to represent the weight of each feature involved in an image by using a genetic algorithm. Next, we generate MPEG-7 signatures based on K-means clustering with weighted features. Finally, in the online phase, we find out result images most similar to a query image after the similarity calculation.

4.1 Feature Weight Tuning

In this step, we use a genetic algorithm to tune the weights of features. A genetic algorithm is a search heuristic used to generate useful solutions to optimization and search problems. In general, a typical genetic algorithm requires 1) a genetic representation of a solution domain and 2) a fitness function to evaluate the solution domain. Here, we also use the weighted Euclidean distance as the fitness function to measure the similarity between two images as follows.

$$Dist(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2 \cdot w_i} \tag{1}$$

where A, B are two images with feature sets $\mathrm{A} = \{a_1, a_2, a_3, \dots, a_n\}$ and $\mathrm{B} = \{b_1, b_2, b_3, \dots, b_n\}$.

Before starting the genetic algorithm, the values of all the 221 features extracted from an image have been normalized in the range [0, 1]. For an initial population of 100 individuals in the genetic algorithm, each one is a set of weight values randomly generated. Only the best 20 individuals with higher fitness values can be alive in the next generation. Then, the 20 individuals are randomly selected to generate 40 children by crossover; in crossover, each feature weight of a child is selected from the corresponding feature weight of parent A or parent B, respectively with 50% probability. Furthermore, in order to avoid being trapped into a local optimal, we also generate another 40 children by crossover plus mutation. In mutation, 10% probability of the feature weights of a child are replaced with new random values.

To measure the fitness of each individual, 24 centroids are used to represent 24 pre-defined classes of 679 training textile images, which are calculated from the images in each class. Then, the weighted Euclidean distance can be treated as a classifier; if a training image has the shortest distance with the centroid of a class using the weights of individual *x* and the matching class is indeed the class of the training image, 1 point is added to the score of individual *x*, and this score is the fitness of individual *x*.

By iteratively doing so, the best individual (or best feature weights) with the highest score would be found. The genetic algorithm will terminate when the finest weight set becomes stable; i.e., the finest weight set is always the best for 1000 iterations. During the iterations, if a new individual has higher fitness than the old best one, the iteration counter is reset to 0 and the new individual will be examined for the next 1000 iterations.

4.2 MPEG-7 Signatures Based on K-means Clustering

For each extracted visual descriptor, we use K-means to partition it into four clusters respectively, and number them from 0 to 3. Then, we combine the cluster numbers from the five visual descriptors together and obtain a 5-digit MPEG-7 signature. Thus, an MPEG-7 signature can represent the characteristics of an image. An MPEG-7 signature has 5 digits and each digit can be 4 different values, so that $(4)^5 = 1024$ bins could be used to distinguish the characteristics of images. Since K-Means compresses images into clusters, we would be able to build an index structure more easily using these signatures. Besides, the centroids of K-means on the five visual descriptors are also stored for the similarity measures in the online phase.

4.3 Similarity Calculation

First, we extract the MPEG-7 visual features of a query image. Then, the similarity measures between the visual features extracted from the query image and the recorded centroids of K-means on the five visual descriptors are performed to determine cluster numbers, respectively. The cluster numbers of the most similar centroids are combined together to become the query signature. Then, we can find out the images with the same MPEG-7 signature as the query image as the first candidates. This approach can be treated as the similarity measures based on local views. Next, we find out the images in three nearest classes to the query image as the second candidates where 24 centroids of pre-defined classes have been mentioned in Section 4.1. This approach can be treated as the similarity measures based on global views. Finally, we can find out the most similar images to the query image, which appear in both groups of candidates.

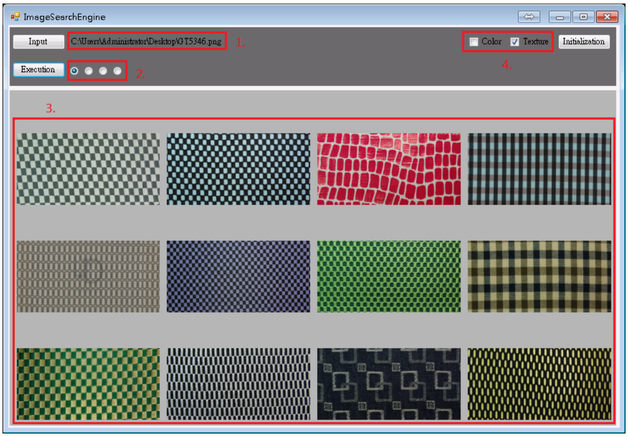
5 Implementation

We have implemented an "**I**mage **S**earch **E**ngine" system to search similar images from an image database to a query textile image. Totally the 4069 images in the textile image database are from Globle-Tex Co., Ltd. [13], where 679 images are training images with pre-defined classes and the others are input to their nearest classes subsequently according to the weighted Euclidean distance as mentioned in Section 4.1.

5.1 User Interface

The user interface of the ISE system is shown in Fig. 2. The **"Initialization"** button is used to initialize the system (or to do data preprocessing in the offline phase). The "**Input**" button can be clicked to input a query image, and after the query image is input, Area 1 will record the path name of the query image. Furthermore, the **"Execution"** button is used to show result images. The radius buttons shown in Area 2 are page switches used to display result images of different pages in Area 3. The number of radius buttons is dynamic and dependent on the number of all result images. The check boxes shown in Area 4 are used to select a search mode; i.e., full (or default) mode, texture-concern mode, and color-concern mode. For the texture-concern mode, we use only the EdgeHistogram Descriptor, the HomogeneousTexture Descriptor, and the RegionShape Descriptor to search similar images. On the contrary, for the color-concern mode, we use only the ColorLayout Descriptor and the ColorStructure Descriptor to search similar images.

The result images displayed in Area 3 are ranked according to the similarity degree to a query image. The most similar image is put at the upper-left and the others are sequentially shown in the left-to-right and top-to-bottom way. The similarity degree to a query image is also according to the weighted Euclidean distance.



C:\Users\Administrator\Desktop\GT5346.png

● ○ ○ ○

□ Color ☑ Texture

Fig. 2. User interface

5.2 Similarity Evaluation

Here, we invite ten evaluators to rate the similar images returned by the ISE system, to their query images. In order to observe the effectiveness of the ISE system, we use the acceptable percentage measure defined by Zhang et al. [12] for rating each result image as a 1-to-5 scale (1: not similar, 2: poorly similar, 3: fairly similar, 4: well similar, and 5: strongly similar). Besides, we also use the quality value measure to evaluate the quality of result images as follows.

The acceptable percentage measure:

$$m_1 = \frac{n_3 + n_4 + n_5}{\sum_{i=1}^{5} n_i} \tag{2}$$

The quality value measure:

$$m_2 = \frac{\sum_{i=1}^{5} n_i \times i}{\sum_{i=1}^{5} n_i} \tag{3}$$

where n_1, n_2, n_3, n_4, and n_5 are the number of result images with a score of 1, 2, 3, 4, and 5, respectively.

5.3 Experiments and Discussions

In Experiment 1, each evaluator tests the effectiveness of three search modes using images in the database. Since the interpretations of the evaluators on textures in the same image are diverse (i.e., someone focuses on patterns, but someone focuses on tiny formations), their measures on the full and texture-concern modes are with major differences. However, on average, the acceptable percentages on the full and texture-concern modes are 81% and 83%, and the quality values on the full and texture-concern modes are 3.6 and 3.7, respectively. Furthermore, since the interpretations of the evaluators on colors are more consistent, the average acceptable percentage and quality value on the color-concern mode are 92% and 4.1, respectively. Thus, the system works well in all three modes, when using images in the database.

In Experiment 2, each evaluator tests the effectiveness of three search modes using images out of the database. We found that their measures on these three modes are a little decreased, when compared with using images in the database. The average acceptable percentages on the three modes are 73%, 80%, and 73%, and the average quality values are 3.1, 3.6, and 3.6, respectively. The reason could be that the query images out of the database do not pertain to the pre-defined classes in the system, and the system cannot but return the similar images in three nearest classes to the query images.

6 Conclusions

In this paper, we propose and implement a textile image search system to search similar images from an image database to a query textile image. In the system, a finest weight set is tuned for the extracted features involved in an image by using a genetic algorithm. Then, we generate MPEG-7 signatures based on K-means clustering with weighted features. In the online processing, users can find out result images most similar to a query image after the similarity calculation. The experimental results show that the similar images returned from an image database to a query textile image are acceptable for humans and with good quality in all three modes.

Although our content-based ISE system can work well for searching textile images, there are still two issues to be overcome in the future. First, the descriptors we used here are still not good enough to describe all the classes of images in our system so that some of them cannot be well classified in the system. Second, for a query image without a pre-defined class, this will lead the system to return unpredictable results. For example, when users input a car image in the worst case, the system has no way to exclude this situation.

Acknowledgments. This work was supported by National Science Council of R.O.C. under grant MOST 103-2221-E-224-049.

References

1. Bober, M.: MPEG-7 visual shape descriptors. IEEE Transactions on Circuits and Systems for Video Technology 11(6), 716–719 (2001)
2. Chang, S.F., Sikora, T., Puri, A.: Overview of the MPEG-7 standard. IEEE Transactions on Circuits and Systems for Video Technology 11(6), 688–695 (2001)
3. Huang, Y.F., Chen, H.W.: A multi-type indexing CBVR system constructed with MPEG-7 visual features. In: Zhong, N., Callaghan, V., Ghorbani, A.A., Hu, B. (eds.) AMT 2011. LNCS, vol. 6890, pp. 71–82. Springer, Heidelberg (2011)
4. ISO/IEC 15938-3, Information Technology – Multimedia Content Description Interface-Part3: Visual (2002)

5. King, I., Lau, T.K.: A feature-based image retrieval database for the fashion, textile, and clothing industry in Hong Kong. In: Proc. International Symposium on Multi-technology Information Processing, pp. 233–240 (1996)
6. Lai, C.C., Chen, Y.C.: A user-oriented image retrieval system based on interactive genetic algorithm. IEEE Transactions on Instrumentation and Measurement 60(10), 3318–3325 (2011)
7. Manjunath, B.S., Ohm, J.R., Vasudevan, V.V., Yamada, A.: Color and texture descriptors. IEEE Transactions on Circuits and Systems for Video Technology 11(6), 703–715 (2001)
8. Martinez, J.M., Koenen, R., Pereira, F.: MPEG-7: the generic multimedia content description standard, part 1. IEEE Multimedia 9(2), 78–87 (2002)
9. Martinez, J.M.: MPEG-7 overview (version 10). ISO/IEC JTC1/SC29/WG11 N6828 (2004)
10. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12), 1349–1380 (2000)
11. Whitley, D.: A genetic algorithm tutorial. Statistics and Computing 4(2), 65–85 (1994)
12. Zhang, Y., Milios, E., Zincir-Heywood, N.: Narrative text classification for automatic key phrase extraction in web document corpora. In: Proc. the 7th Annual ACM International Workshop on Web Information and Data Management, pp. 51–58 (2005)
13. Globle-Tex Co., Ltd., `http://www.globle-tex.com/`

IIS: Implicit Image Steganography

K. Jithesh[1,*] and P. Babu Anto[2]

[1.] Department of Computer Science, M.G College, Iritty, Kannur Uniersity, Kerala,India
[2.] Department of Information Technology, Kannur University, Kerala, India
jithukotheri@gmail.com

Abstract. In steganography secrets are imposed inside the cover medium either by replacing bits in the spatial domain or changing the frequency domain. Instead the proposed Implicit Image Steganography (IIS) scheme does not alter or replace bits in the original cover image for hiding information. As the name implies there is no explicit embedding of data inside the image. Before beginning the communication, entities should agree upon a cover image with maximum ranges of intensity values. At least, it should contain intensity values that can represent ASCII of all characters. Coordinate positions of each pixel with intensity which can be the ASCII of a letter in the secret will be the stego-key. In this scheme, transferring of cover image is not done as in the case of usual procedures. The communicating entities have to transfer only the key. The big advantage of this technique is that the cover is not required to transmit each other. Hence nobody can even know about the cover. So it is not only difficult but impossible to attack this communication. Also it is not required to worry whether distortion happens while embedding.

Keywords: Cover, Steganography, Stego-key, Stego-image, Steganalysis.

1 Introduction

The standard concept and practice of ‘‘What You See Is What You Get (WYSIWYG)’’ which we encounter sometimes while printing images or other materials, is no longer precise and would not deceive a steganographer as it does not always hold true. Images can be more than what we see with our Human Visual System (HVS); hence, an image can convey more than merely 1000 words. For decades people have been trying to develop innovative methods for secret communication. Networking and digitization have become part of the technological features in the rapid economic development of the society. The convenient and timely acquisition of on-line services through accessing the internet has assumed the proportion of a tidal current for individuals and organizations in their pursuit of work. However, the relay of sensitive information via an open Internet channel increases the risk of attacks. Thus many techniques have been proposed to deal with this problem. Data hiding, known as information hiding, plays an important role in the information security. For

[*] Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 275–283, 2014.
© Springer International Publishing Switzerland 2014

content authentication and perceptual transparency, the main idea of data hiding is to conceal the secret data into the cover medium, and thereby to avoid attracting the attention of attackers in the Internet channel. The growing numbers of internet-based applications nowadays have made digital communication an essential part of infrastructure. Confidentiality in some digital communication is absolutely necessary when sensitive information is being shared between two communicating parties over a public channel.

Steganography [1, 2, 3] and Cryptography [4] are two sides of the same coin for providing confidentiality and protecting sensitive information. The former is the art and science of hiding sensitive information within innocuous documents in an undetectable way. A thorough history of steganography can be found in the literatures. The latter is the art and science of writing sensitive information in such a way that no one but the intended recipient can decode it. The innocuous documents (also known as hosts/covers/carriers) can also be of any kind as long as they do not seem suspicious. However, with the advent of the digital technology, digital hosts such as image, audio and video files etc. have become nowadays the most commonly used host files. On account of their insensitivity for the human visual system, digital images [5, 6] can be regarded as an excellent choice for hiding sensitive information. One of the most commonly used data hiding approaches is the substitution technique [7, 8], [12, 13]. The embedding algorithm may require a secret key, referred to as stego-key.

Each and every method introduced in the field of information security has the sole purpose of achieving the triple pillars of information security. They are confidentiality, availability and authentication. The available algorithms of steganography have been embedding secret data inside the cover medium. A few exceptions are there like Quantum steganography protocol [9] and Multi-party covert communication with steganography and quantum secret sharing [10]. Abbas Cheddad et al. reported the current techniques of image stegnography with its pros and cones [11]. The problem associated with these methods is that they cause image distortion. Also the current techniques lose its security when the algorithm behind the communication is once revealed or an intruder has hacked the very content. Once a new method is introduced it is possible for many types of attacks to happen.

A different methodology has been introduced here for the purpose of accomplishing the goals of information security. The proposed study is not intended to go along with the usual way. Usually information is hidden inside the cover by replacing the bits or changing the frequency level. But in this scheme we do not change or make any distortion in the spatial or in the frequency domain of the cover medium. Here the cover will not be transmitted. Instead, before beginning, they should agree upon the stego-image and should keep a copy of the same. They need to send only the bit positions inside the cover which comprises the secret. They will just inform each other about the bit positions inside the image. Since they hold the same copy of the cover, they can easily extract the secret

From this perspective the authors have tried to develop a system which can give maximum security by avoiding steganalysis and distortion to the cover medium. Our primary objective here is to present a new and simple steganographic scheme that gives high security. On account of the complexities in steganography and progressive power of steganalysis methods, it has turned out to be a challenge to systematically develop techniques with much better performance. Key exchanging is also a big problem in the area of communication. This study aims to deal with these problems by proposing a scheme to increase the non-detecting power of the secret. Since the original cover image is not transmitted it can thwart all conventional attacks. The results of our experiments illustrate that since there is no stego-image, the steg-analysis is not possible. Hence the security of the system would be compromised only if communicating entities cheated each other.

The remainder of this paper is organized as follows: Section 2 contains a brief discussion of related works. In Section 3, the proposed IIS scheme is discussed. Section 4 presents the experimental results and performance of the proposed method, in terms of comparison with conventional schemes. Section 5 provides a comparative visual analysis of cover images and finally section 6 gives the conclusion of this study.

2 Related Work

Some literature proposed varieties of data hiding techniques. Most of them are irreversible. It means that, after the secret data are extracted from the stego image, the original image suffers some distortion and cannot be completely reconstructed. Nevertheless, in some fields (i.e., medical, military applications), the restoration of original image is essential after extracting the embedded secret data. Therefore, reversible data hiding schemes, also called as distortion-free data hiding or lossless data hiding, have drawn much attention of researchers. In principle, reversible data hiding schemes can be classified into three types, i.e., spatial domain, frequency domain, and compressed domain. In the spatial domain schemes, all pixel values are modified directly to embed secret data. In the frequency domain schemes, the coefficient values of image are computed by using some transformation methods (i.e., integer discrete cosine transform, integer wavelet transform). In the compressed domain schemes, the original image are first compressed based on some popular compression algorithms, such as vector quantization, block truncation coding etc. Then, according to the peculiarity of compressed codes, the compressed image is encoded to conceal secret data.

There are numerous techniques available in the literatures of steganography. The Least Significant Bit [LSB] replacement steganographic methods [7, 8], [12, 13] are the simplest one and are widely used in the fields of information security due to its high hiding capacity and quality. It can embed a secret bit stream into the LSB plane of an image. LSB replacement, LSB matching (LSBM), LSB matching revised (LSBMR) [12], and LSBMR-based edge-adaptive (LSBMR-EA) [13] image steganography are well-known LSB steganographic methods. The LSB-replacement embedding method replaces the LSB plane with embedded message bits, but the others do not. In LSB matching, if the embedded bit does not match the LSB of the cover image, then the pixel value of the corresponding pixel is randomly changed by ±1. Unlike LSB replacement and LSBM, which embed message bits pixel by pixel, LSBMR deals with two pixels at a time and allows fewer changes to the cover image. The steganalysis resistance and image distortion of LSBMR are better than those of the previous two methods. In general, the choice of embedding positions within a cover image depends on a pseudorandom sequence without consideration of the relationship.

The proposed method is a new one of its kind. From the literature survey and to the best of our knowledge there is no such steganography scheme similar to the proposed one. In the available technologies of steganography the embedding is done inside the cover image. A work which is an exception and which can be related to the proposed work in terms of not embedding secret inside an image is introduced by Guo et al. [9] in 2003 with the title Quantum Secret Sharing without Entanglement. Also, Xin Liao et al. [10] in 2010 presented a novel multi-party covert communication scheme by elegantly integrating steganography into Guo et al.'s QSS. This scheme is good in terms of security but the payload of this method is comparatively very less. That is, it communicates only one bit per transaction. Nevertheless, their idea of not embedding secret inside a digital image motivated us to introduce a method which is better in payload capacity and security. The proposed IIS scheme does nothing over the cover image. It only keeps a location map. Another important feature of the proposed stego system is that it does not require a secret key. Thus, the constructions presented demonstrate that in order to achieve perfect steganographic security no secret has to be shared between the communicating parties. The main idea behind the stegosystems we propose is to conceal the cover from outside.

3 Proposed IIS Scheme

As stated earlier this is a simple but innovative technique. Here nothing is done with the cover, but copy of the cover is shared by the communicating entities. The algorithm is as follows:

1. Select appropriate cover image carefully.
2. It can be either grey-scale or color.
3. If possible it should contain intensity values that can easily represent ASCII of each character.
4. Share the copy of the digital cover medium with the other end.
5. Exchanging of the cover should be made very confidential.
6. Hand to hand exchange is more reliable and secure, otherwise use a trusted third party.
7. After the successful exchanging of the cover the communication can be started.
8. Take the secret.
9. Digitize it.
10. Find out pixels which can fully represent characters of the secret.
11. Mark up the coordinates of the respective pixels.
12. This is treated as the stego-key.
13. If pixel [intensity] values are not enough to represent all characters then try to find consecutive bits of a pixel inside the image to represent such letters.

14. If consecutive bits of a pixel are used, the starting and ending position is required to be kept.
15. In such case two or more pixels can be used to hold a character.
16. In such cases key should be the combination of coordinates and starting as well as ending positions of the bits.
17. If color image is used coordinate position and RGB position should be the key.
18. If RGB is used a single pixel can represent 3 characters a time.
19. Stego-key can be sent to the other side with or without encryption.
20. With the key the receiving end can easily extract the secret from the copy of the image.

4 Experimental Results

Always there will be new threat in the field of communication. So no technique is good for a long time. A method that combines both logic and craft can only survive in the contest of constant changes in the field. In order to improve the security of the secret communication, here we propose a technique which gives better performance. In this section, we present the results from the experiments we conducted in order to evaluate the reliability of this method. We have used both color and grey-scale images. Care has been taken while selecting images. That is, when grey-scale image is the cover; it is assured that it contains maximum shades. Images with 0-255 ranges of intensity are the optimal case. Any character of English language can be represented with 0-255 ASCII values. So a grey scale image is enough. In a grey-scale cover, coordinate positions will be the stego-key. When the cover is a color image it is possible to represent characters of all human languages. If range of color variations of the image is more, it is very easy to represent secret.

As mentioned earlier, this method does not require anything to be done over the cover image. It requires selecting a cover image of grey-scale or color with maximum ranges. Then, hand over the copy of the same image to the other entity that belongs to the same channel of communication. The exchanging should be very confidential. It is better to exchange in a handshake mode. If it is not possible, trust a third party. The secrecy and security of the communication rely on the cover transfer. If it fails, the entire security will be compromised. As the title of this paper indicates, there is no direct or explicit hiding of information inside the cover. In fact, it is a steganogrpahy without steganography. Here follows a case in point.

Let the secret to be transmitted is “Pay”.

The binary form of the information is

P = 80 → 1010000, a = 97 → 1100001, y = 121→ 1111001

Fig.1 is the stego-image. Make a copy and hand it over to the other end. As mentioned the security of this approach lies in this step. Information about the stego-image should not be revealed at any cost. If it is revealed, the entire essence of the security will be easily broken.

In this experiment the first letter is P. Its binary is 101000. Select the appropriate pixel position from the cover. We have a number of techniques to select the pixel coordinate from an image with a single mouse click. For example matlab provides impixelinfo function for the same. Keep the coordinate position as the stego-key. The (x, y) coordinates of letters inside the image are (101, 63), (119, 60) and (149, 29) respectively. Here the stego-key is 101631196014929. Send this stego-key to the other end. We can use either public key or private key cryptography to transfer the key. Public key cryptosystem is the best method to exchange the key. Even if the key is lost or broken, it does not affect the security of the system. As far as the cover image is not known to anybody else other than communicating entities, the confidentiality of the information is preserved at the maximum. It is possible to send the stego-key without doing any encryption or hiding.

The proposed scheme is compared with the currently popular steganographic schemes namely, Secure Bit-Plane Based Steganography for Secret Communication [14], Adaptive Image Steganography Based on Depth-Varying Embedding [15] and A Novel Technique for Image Steganography Based on a High Payload Method and Edge Detection [16]. Table 1 shows the results of comparison. The default optimal parameter settings suggested in the respective works are adapted for experimentation.

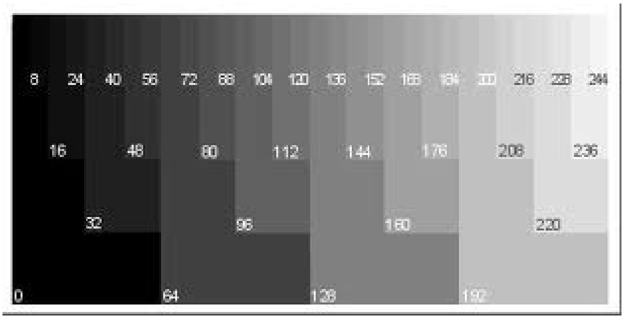


Fig. 1. Cover or Stego Image

Table 1. Comparison between conventional methods and IIS

Factors	Proposed IIS Scheme	Conventional schemes
Distortion	NIL	Caused due to hiding
Payload	Large	Comparatively less
Security	high	Comparatively less
Key exchange	It Can be revealed	Big problem
Authentication	Very easy	Difficult.

5 Visual Analysis

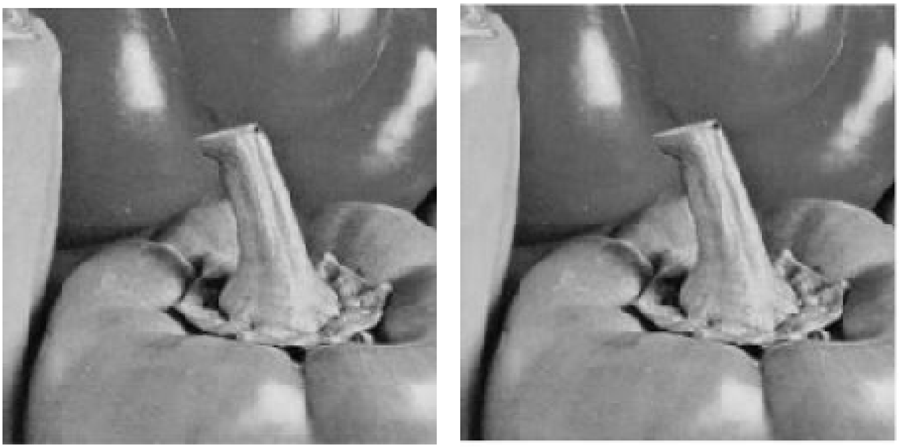
As indicated earlier, the available steganography leads to visual distortions to the original view of the cover image. Fig.2 shows the distortion caused to the image Pepper while using popular Chang et al.'s [17] steganography scheme. It is obvious that using any scheme except the one proposed by Xiu-Bo Chen et al. will lead to image distortion. But Xiu-Bo Chen et al.'s scheme has other limitation in that it can only send one bit of data at a single transaction.

To evaluate the visual quality of stego images by using the human eye, we have enlarged the partial area of original cover image and stego-image of Chang et al.'s scheme, as shown in Fig.2. Fig.2 (a) shows the cropped area in the original Pepper image and Fig.2 (b) shows the cropped area in the stego-image. The distortion between original cover image and stego-image is visually almost imperceptible. The table.2 shows the MSE and PSNR of popular Yang et al.'s Scheme [18], Lin-Tsai Scheme [19] and Chang et al.'s Scheme. The results reveal that even though they are good techniques, they are not free from image distortion.

Steganalysis is an interesting topic that focuses on the detection of the presence of embedded secret messages. RS attack [20], proposed by Fridrich et al., and x2 detection [21], proposed by Westfeld and Pfitzmann, are the two most effective LSB steganalytic techniques. The RS-attack technique can detect both sequentially embedded messages and randomly embedded messages. No current steganogrpahy method can claim that they are free of image distortion while embedding secret inside because PSNR never becomes infinity. Hence these techniques cannot escape from steganalysis. They are not absolutely secure against attacks. Now that the proposed method never embeds a secret directly or explicitly to an image, it can claim that it is free of any such kinds of steganalysis. Enhancing the visual quality and authentication capability of stego-image are new challenges for researchers working on the development of novel secret image sharing scheme. The results and discussion clearly indicate that the proposed scheme provide high confidentiality and authentication.

Table 2. The PSNR and MSE of stego-images with the same payload capacity (The unit of PSNR is db)

Secret image (256×256)	Stego-image (512× 512)	Lin-Tsai Scheme		Yang et al's Scheme		Chang et al.'s Scheme	
		MSE	PSNR	MSE	PSNR	MSE	PSNR
General test pattern	Cameraman	7.74	39.25	4.43	41.66	5.49	40.73
	Baboon	7.85	39.18	4.55	41.55	6.59	39.94
	Lena	7.80	39.20	4.50	41.60	5.98	40.37
	Pepper	7.86	39.17	4.54	41.56	7.63	39.30
	Sailboat	7.89	39.16	4.59	41.51	8.45	38.86
	Average	7.83	39.19	4.52	41.58	6.83	39.84



(a) (b)

Fig. 2. Cover image and stego image of Change et al.'s scheme

6 Conclusion

Here a novel spatial steganography scheme realized with IIS paradigm is introduced. It can be also referred as steganography without steganography. Nothing is done with the cover image. Hence, as usual distortion to the medium is not caused. In fact, it is a different strategy to achieve high security to communication. Its biggest advantage is that it is impossible to break the confidentiality even if we lost the key. Since stego-key alone can do nothing, it is not necessary to encrypt or hide the key. The greatest risk of this method lies in transmitting the cover. Exchanging by hand is absolutely secure. The additional concern of this approach is to get images with ranges varying from 0 to 255. For any stego-system the next property to be considered after its security is its capacity. The capacity of a stego-system can be defined as the number of hidden bits transmitted per letter of the cover image. We show that our stego-system has the maximum possible capacity: Though it can adopt large payload, it is optimal for small secrets. In public key cryptography like RSA and others the length of the key is very big. It is common to have a key length of 1024 bits and more. In decimal notation it can easily exceed 300 digits. Here a key length with 300 digits can easily represent a secret of about 150 letters. The length of the key is directly proportional to the secret to be encoded. The common problem of recovering the original cover image from the stego-image is also resolved here. Further studies can be undertaken to increase the payload. This study has also taken advantages of human psychology in its formulation so that it become more effective and practical. Since the cover image is not transmitted all kinds of conventional threats become insignificant and ineffective thereby making it more viable for its purpose.

References

1. Johnson, N.F., Jajodia, S.: Exploring steganography: seeing the unseen. IEEE Computer 31(2), 26–34 (1998)
2. Shih, Y.F.: Digital Watermarking and Steganography Fundamentals and Techniques. CRC Press, Taylor & Francis Group, Boca Raton (2008)
3. Petitcolas, F.A.P., Stefan, K.: Information Hiding Techniques for Steganography and Digital Watermarking, Canton Street, p. 685. Artech House, Inc., Norwood (2000)
4. Stalling, W. (ed.) Cryptography and Network Security-Principles and Practices, 4th edn., Pearson Prentice Hall, India P. Ltd., India (2000)
5. Salivahanan, S.: Digital Signal Processing. Tata McGraw-Hill, India (2000)
6. Gonzalez, Woods: Digital Image Processing, 3rd edn. Prentice Hall, USA (2008)
7. Hu, C.H., Lou, D.C.: LSB Steganographic method based on reversible histogram transformation function for resisting statistical steganalysis. Information Sciences 188, 346–358 (2012)
8. Chan, C.K., Chen, L.M.: Hiding data in images by simple LSB substitution. Pattern Recognition 37, 469–474 (2004)
9. Bo, X.C., Guo, Z.Q., Jie, X.Z., Xin, X.N., Xian, Y.Y.: Novel quantum steganography with large payload. Optics Communications 283, 4782–4786 (2010)
10. Liao, X., Wena, Q.V., Suna, Y., Zhangb, J.: Multi-party covert communication with steganography and quantum secret sharing. The Journal of Systems and Software 83, 1801–1804 (2010)
11. Cheddad, A., Condell, J., Curran, K., McKevitt, P.: Digital image steganography: Survey and analysis of current methods. Signal Processing 90, 727–752 (2010)
12. Mielikainen, J.: LSB Matching Revisited. IEEE Signal Process. Lett. 13(5), 285–287 (2006)
13. Huang, J., Luo, W.: Edge adaptive image steganography based on LSB matching revisited. IEEE Trans. Inf. Forens. Security 5, 201–214 (2010)
14. Cong-Nguyen, B., Lee, H.Y., Joo, J.J.C., Lee, H.K.: Secure bit-plane based steganography for secret communication. IEICE Transactions on Information and Systems E93, 79 (2010)
15. He, J., Tang, S., Wu, T.: An adaptive image steganography based on depth-varying embedding. In: Image and Signal Processsing, CISP 2008, vol. 5, pp. 660–663 (2008)
16. Halkidis, S.T., Ioannidou, A., Stephanides, G.: A novel technique for image steganography based on high payload method and edge detection. Expert Systems with Application 39, 11517–11524 (2012)
17. Chang, C.C., Hsieh, Y.P., Lin, C.H.: Sharing secrets in stego images with authentication. Pattern Recognition 41, 3130–3137 (2008)
18. Yang, C.N., Chen, T.S., Yu, K.H., Wang, C.C.: Improvements of image sharing with steganography and authentication. Journal of Systems and Software 80, 1070–1076 (2007)
19. Lin, C.C., Tsai, W.H.: Secret image sharing with steganography and authentication. Journal of Systems and Software 73, 405–414 (2004)
20. Fridrich, J., Goljan, M., Du, R.: Reliable detection of LSB steganography in color and grayscale images. In: Proceedings ACM Workshop Multimedia and Security, pp. 27–30 (2001)
21. Westfeld, A., Pfitzmann, A.: Attacks on steganographic systems. In: Proceedings of the 3rd International Workshop on Information Hiding, Dresden, Germany, pp. 61–76 (1999)

Humanized Game Design Based on Augmented Reality

Yanhui Su[1,2], Shuai Li[1], and Yongsong Zhan[1]

[1] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology,
Guilin, 541004, China
[2] Institute of Computer, Zhejiang University, Hang Zhou, 310058, China
suyanhui@gmail.com

Abstract. Currently, the primary research of AR (Augmented Reality) is focus on how to improve the accuracy of identification and reduce the dependency to markers. Successful cases are scarce on how to play the special advantage of augmented reality, combine it with mobile devices and generate practical value. We start form the actuality of AR technology characteristics, achieved the first blind game system which based on mobile platform Android. This system is not only regard game`s emotion design as starting point and pay closely attention to the vulnerable groups, especially the blind community, but also provide them opportunities that compete with average person. Besides, it offers a diversification choice of game against for blind users by set different difficulty level and game player modes. The user experience indicates that the system combines AR and the innovation of game design together, and help vulnerable group achieve flexible user experience and operation.

Keywords: Augmented reality, Game design, Humanized design, Emotional design.

1 Introduction

Augmented Reality (AR) is an emerging field and a hot spot of research in recent years as an important branch of virtual reality technology. Based on information in real scenarios, AR technology superposed virtual objects or other information that computer generated to the real scenario and fused them. By this way, a bridge would be set up between virtual and reality world, thus, we could realize the “Augment” to reality world and presented a new environment with real sensory effect to users.

AR technology has a widely application and a more obvious superiority than VR technology. Zhongwang Jiang`s article[1] introduced the development history and application field of AR technology in detail. Sui Yi`s paper[2] introduced the AR technology based on a handheld device. From 1990, Tom Caudell and David Mizell, engineer of Boeing Co, proposed the concept of “Augmented reality” firstly when they designed the auxiliary wiring system[3]. HIT laboratory at university of Washington released a develop tool of AR system which is named ARToolKit in 1999. Now, just several decades, AR technology has formed a relatively complete workflow and implementation system, its basic principle is shown as Fig.1.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 284–291, 2014.

© Springer International Publishing Switzerland 2014

Games developed rapidly as an emerging industry and various types of games emerged, but games' control mode and scene effect still changed little. In recent years, with the improvement of smartphone`s processing capacity and PDA, AR technology applied to game gradually, so that the game scene is more realistic and users` interactive experience is more immersive. In 2000, Bruce H. Thomas developed the first outdoor mobile AR game which is named AR Quake. In 2004, Mohring[4] et al developed the first application which could completely do all the AR tasks by smart phone. In recent years, research of AR mobile game attracted more and more scholars` attention, and kinds of handheld games based on AR technology constantly emerge. Mobile Maze game[5] require user through the maze by push a car ball. In 2005, Video Processing at VIT developed a multi-user table tennis game which is named Symball[6]. HIT lab NZ developed an AR Tennis game[7]. In 2006, Siemens developed a free throw game, AR Soccer[8]. Besides, Kurt D.Squire team developed an AR scientific education game, Mad City Mystery[9]. From last year, a series of mobile AR game appeared constantly, such as Drop Defender, Zombie Room AR and so on. The application of AR technology made our visual information more intuitive and richer, the real-time interactive experience more real, it will be a development tendency for mobile application in the future.

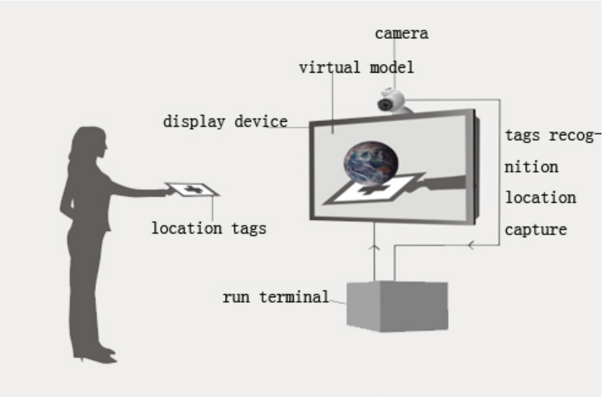


Fig. 1. Augmented reality principle

2 Humanization of Game Design

Except the play ability, game design also needs to pay attention to players` emotional needs. The book "Creating Emotion in Games"[10] told us that emotional design could help game players to realize the additional value of game as an important part of game design, especially for vulnerable groups. Such as: blind person, how to let them to fully experience game`s joy as normal people, the key point is to provide them the opportunities that competitive with normal players. So Shaking World, a puzzle action game which we will introduce in this paper, used the third-person design and AR technology, built a fair competitive platform for blind players. Through the humanized touch board design and with the help of the blind touch habits, blind community could fully experience fun of this game.

3 Game System and Mechanism

3.1 Game System

Shaking World is a fantastic third-person puzzle action game developed by Unity 3d. It has two modes: online mode or console mode, where online mode set up two characters in the same scene, player A and player B. Player A is responsible for shaking the game world and player B should try best to control the character which named little fool against the shaking. They can earn coins and buy props they like by play games. Besides, Shaking World also designed custom mode what players can increase the fun by building themselves level with game props. In a word, this game easy to play, numerous levels plus thrilling scenes make it the best choice for recreation.

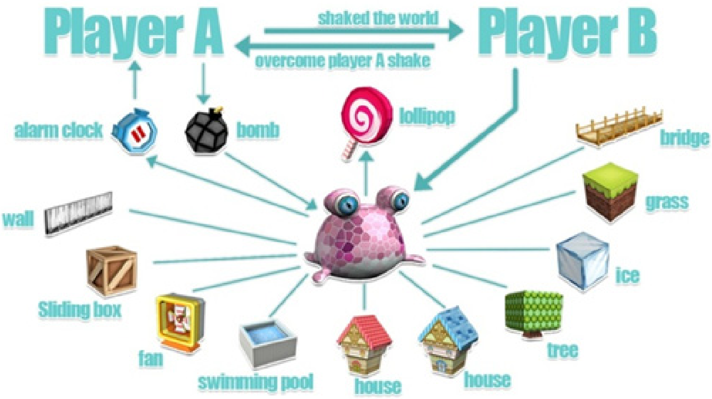


Fig. 2. Game system

3.2 Game Props Design and Mechanism Design

Game set in a lot of props in order to increase its playability, such as: bomb, alarm clock, lollipop and so on. The relationship of every element in the game is as shown in Fig. 2. Where, mostly props used to decorate the scene or hinder player B win the game. For example, if character falling into the swimming pool, player B needs a quick click on the character to climb back on the ground, but if player B's clicking too slowly, character will drown and the player A will win the game; The fan has a fixed direction to blow, when the character close to the source of wind, he will be blown away; If player A bought bombs and let player B touched it, player B will be killed, and player A will win the game, but if player B bought the alarm clock and touched it, player A will lose control of the game world 3 seconds, and he will win if he lets the character get the lollipop. In a word, what Player A need to do is that he should try his best to shake player B out of the game world and do the winner, as shown in the left of Fig. 3. On the contrary, player B should make a great effort to overcome player A`s shaking and get the lollipop successfully, as shown in the right of Fig. 3.

Fig. 3. Game mechanism design

3.3 Technology Implementation

Vuforia SDK is based on AR application and display mobile device as a “magic camera” or show the scene as a world that coexistence of virtual and reality. We used this SDK in Shaking World, adopted the AR technology which based on mobile terminal and transplanted the style of AR to cellphone platform. The virtual scene could be stacked to the paper when used the camera of cellphone to shoot it. Besides, in order to assure the consistency between operation and braille, we used the style of virtual button, set a touch area on the paper for blind person. Game runs on Android platform, and its technical principle hierarchy Chart is showed as Fig. 4. Players take photos by camera, find out the target image and determine its coordinate; besides, identify the target images, overlay virtual image on real image and use virtual button interactive with real world. Its bottom technology is as follows: bottom is on Android system, around the operating system to build two big modules, including: camera and rendering module; besides, on the basis of the camera module, we add target tracking module, including: virtual buttons, multi-target tracking, target image recognition, space target recognition and so on, and in the form of SDK provides the chance that developers call low-level interface, can achieve all kinds of special effects effectively.

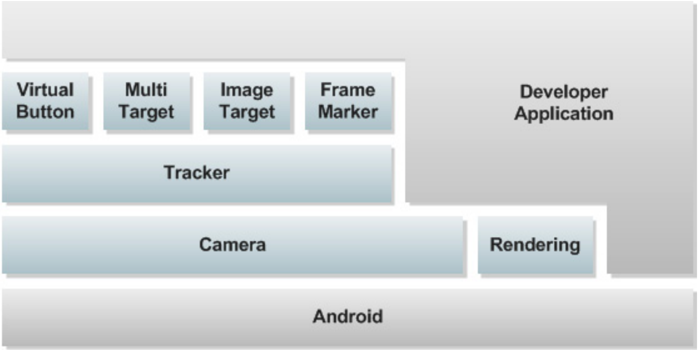


Fig. 4. Bottom framework of AR engine

Virtual Button System. Virtual buttons are developer-defined rectangular regions on image targets that when touched or occluded in the camera view, can trigger an event.

Virtual buttons can be used to implement events such as a button press or to detect if specific areas of the image target are covered by an object. Virtual buttons are evaluated only if the button area is in the camera view and the camera is steady. Evaluation of virtual buttons is disabled during fast camera movements. Define virtual buttons in the Database Configuration XML as children of an image target. To add virtual buttons, insert a section similar to the following:

```
<ImageTarget size="247 173" name="wood">
<VirtualButton name="red" rectangle="-108.68 -53.52 -
75.75 -65.87" enabled="true" />
<VirtualButton name="blue" rectangle="-45.28 -53.52 -
12.35 -65.87" enabled="true" />
</ImageTarget>
```

The Virtual Button state can be requested from active targets in the scene by iterating through the button child objects:

```
// Iterate through this targets virtual buttons:
for (int i = 0; i< target->getNumVirtualButtons(); ++i)
{ constVirtual Button* button = target-
  >getVirtualButton(i);
  if (button->isPressed())
    { textureIndex = i+1;
       break; }
}
```

User-defined the identified image. In this section we show how to use the user-defined target feature to instantiate objects of classes from TrackableSource which can be used to create new Trackables at runtime.

Two new classes, ImageTargetBuilder and ImageTargetBuilderState are introduced: where, class ImageTargetBuilder exposes an API for controlling the building progress, retrieving a TrackableSource for instantiating a new trackable upon successful completion. The flow chart is as shown in Fig. 5.

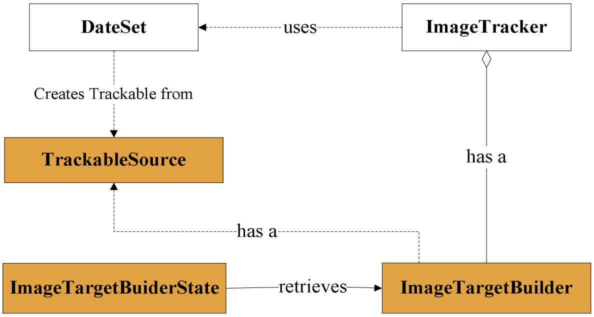


Fig. 5. The process of instantiate object

Expression form of identified images. The expression forms of identified images are as follows:

```
?xml version="1.0" encoding="UTF-8"?>
<QCARConfigxmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="qcar_config.xsd">
<Tracking>
<ImageTarget size="247 173" name="stones" />
<ImageTarget size="247 173" name="chips" />
</Tracking>
</QCARConfig>
```

4 Game Results

Relative to other games, Shaking World has few bright spots: (1)Pay attention to user`s emotional experiences, provide humanized attention and care design for vulnerable groups. (2) Using augmented reality mode to increase fun of the game and expand the original dimension of game experience which based on screen to a three-dimensional game space. (3) Using the virtual button mode of AR technology, provided game operation based on braille contact for blind user.(4) Combined the play ways of virtual and reality, and provide an interaction with virtual game by manipulating physical paper. (5) With the aid of AR technology and use the levels mode of blind person book, blind man could touch the dots on graph to operate this game. Besides, different maps represent different level, so with different maps, blind person can take part in every level of game as normal people.

The game Shaking World which based on AR technology could achieve different effects. The left of Fig. 3 stands that player B was shaking out the game world, the right of Fig. 3 stands that the character eats the lollipop. Fig.6 is the whole scene of this game. Fig.7 is the game mode that designed for blind person. They could play the game by touch dots on the graph paper when run the game. Every elements of this game drawn exquisite, character designed personality, and visual effect is preferably.

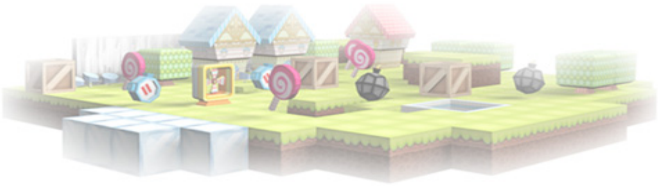


Fig. 6. Humanized design of game scene

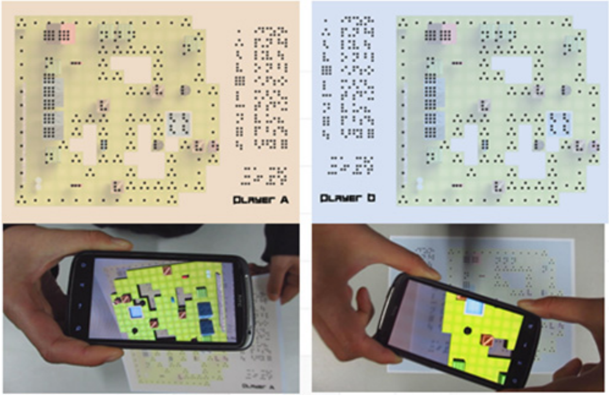


Fig. 7. Blind person mode of the AR game

5 Conclusion

Shaking world game used the AR technology, fully considered humanized design, the research result indicated that it has a certain appeal and also is a better choice for entertainment. Beyond that, it also has much commercial value: (1) Convenient development peripheral products: the “Shaking World” building blocks, gift boxes, store content boxes, alarm clocks and so on. (2) Can implant the SDK advertising on the top of houses in the game scenes and don't destroy the beautiful game. (3) Can implant some products in the art style of the game for joint operations of products. (4) Can promote to many operators, such as: App Store, CUCC, CM, CHA. (5)To cooperate with shopping website: Play the game to earn the points or gold of the shopping website. The more important is that the game also has much public value: (1) Pay attention to vulnerable groups, to provide fair and competitive opportunities for the blind, let them feel the game happiness. (2)With the help of a third party social media channels, for example the game feeds, cause people concern for the vulnerable groups and give them more help.

In the future, AR technology is an emerging and active field of research. It could bring people new visual experience, and has a broad market prospect and application scope. For the game industry, apply the AR technology to game could greatly rich the content of game frames and increase game`s entertainment.

Acknowledgments. This research work is supported by the grant of Guangxi science and technology development project (No: 1355011-5), the grant of Guangxi Key Laboratory of Trusted Software of Guilin University of Electronic Technology (No: kx201309), the grant of Guangxi Education Department (No: SK13LX139), the grant of Guangxi Undergraduate Training Programs for Innovation and Entrepreneurship (No: 20121059519).

References

1. Jiang, Z.: The development of Educational Augmented Reality Game. East China Normal University, Shang Hai (April 2012)
2. Sui, Y.: Research and application of augmented reality technology based on handheld device. Qing Dao University, Qing Dao (March 6, 2009)
3. Caudell, T.P., Mizell, D.W.: Augmented reality: An application of heads-up display technology to manual manufacturing processes. In: 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, vol. 2, pp. 659–669. IEEE (1992)
4. Mohring, M., Lessig, C., Bimber, O.: Video see-through ar on consumer cell-phones. In: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 252–253. IEEE Computer Society (2004)
5. Bucolo, S., Billinghurst, M., Sickinger, D.: Mobile maze: a comparison of camera based mobile game human interfaces. In: Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services, pp. 329–330 (2005)
6. Hakkarainen, M., Woodward, C.: SymBall: camera driven table tennis for mobile phones. In: ACM International Conference Proceeding Series, vol. 265, pp. 391–392 (2005)
7. Henrysson, A., Billinghurst, M., Ollila, M.: Face to face collaborative AR on mobile phones. In: 2005. Proceedings. Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 80–89. IEEE (2005)
8. Geiger, C., Paelke, V., Reimann, C.: Mobile entertainment computing. In: Göbel, S., Spierling, U., Hoffmann, A., Iurgel, I., Schneider, O., Dechau, J., Feix, A. (eds.) TIDSE 2004. LNCS, vol. 3105, pp. 142–147. Springer, Heidelberg (2004)
9. Squire, K.D., Jan, M.: Mad City Mystery: Developing scientific argumentation skills with a place-based augmented reality game on handheld computers. Journal of Science Education and Technology 16(1), 5–29 (2007)
10. Freeman, D.: Creating Emotion in Games. Beijing Hope Electronic Press, Beijing (2005)

A Hybrid PSO-DE Algorithm for Smart Home Energy Management

Yantai Huang[1,*], Lei Wang[1,2], and Qidi Wu[1]

[1] Department of Electronics and Information Engineering, Tongji University, Shanghai, China
huangyantai@sina.com
[2] Shanghai Key Laboratory of Financial Information Technology, Shanghai, China
wanglei@tongji.edu.cn

Abstract. Home energy management system is an important part in smart home, smart home is assumed to be equipped with smart meter which smart control of generators, storages, and demand response programs. In this paper, we study a versatile convex optimization framework for the automatic energy management of various household loads in a smart home. The scheduling algorithm determines how energy resources available to the end-users considering a number of constraints. Hence, a hybrid PSO-DE algorithm approach is proposed. We devise a model accounting for a typical household user, and present computational results showing that it can be efficiently solved in real-life instances.

Keywords: smart home, home energy management, hybrid PSO-DE algorithm.

1 Introduction

With the increase of the electric demand, energy crisis worldwide is one of the most serious challenges in the 21st century. The residential sector is experiencing the strongest increase on its electric demand. Therefore, the study of the electric demand in the residential sector is an important task for the electric grid controllers as already shown in [1, 2].

Home energy management system provides an opportunity for improving the efficiency of energy consumption in residential sector [3, 4]. A typical Smart home for residential sector integrates the operation of electrical and thermal energy supply and demand.

Developing efficient demand response models of electrical appliances is a key problem in an energy management system in a smart home, which have received considerable attention recently [5-12]. In general, the main objective of Home energy management system is to minimize the electricity bill or maximize their users' satisfaction by allocating available resources and managing the load of appliances.

The home energy management in a smart home can be formulated as a complex mathematical optimization problem. Dynamic programming may be used if the

[*] Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 292–300, 2014.
© Springer International Publishing Switzerland 2014

optimal schedule is required, however, the immense computational effort involved might require a powerful computer and long computation times. On the other hand, the complex optimization problem can be solved by the commercial solver such as CPLEX and MOSEK. However, these commercial solvers are specialized optimization package whose computational requirements are not suitable for a smart meter. Thus, heuristic techniques have been applied to the original problem for obtaining a global optimal solution to the problem. Thus, this paper proposes a particle swarm optimization algorithm to solve the original problem for obtaining a global optimal solution to the problem. In order to improve the performance of the algorithm, we integrate particle swarm optimization with differential evolution (PSO-DE) to solve constrained numerical and engineering optimization problems.

The rest of this paper is organized as follows. In Section 3, the mathematical problem formulation is presented. In Section 4, the scheduling problem is resolved by a hybrid PSO-DE algorithm. Section 5 includes necessary figures, results, and discussions of the typical study case. The conclusion is drawn in Section 6. References are appended last.

2 Mathematical Formulation

It is assumed that most electric appliances are networked together and are controlled by a home energy management system. The smart home consists of micro-CHP unit, storage devices, local loads, thermal loads and must run electrical loads. The output power energy of micro-CHP unit is supplied to the kinds of appliance loads, the surplus power energy is used to charge storage batteries or sail to the main grid. In contrast, the deficit power energy is provided by batteries and main grid. In this paper, the whole problem of the smart home energy management system is defined as an optimization problem. In so doing, we have used simple models of the micro grid's components which will be described in the following subsections.

2.1 Objective Function

The objective function is to minimize the operation cost: the cost of the power purchased from the main grid during a day, and the cost of gas consumed by the micro-CHP unit (G_{CHP}). In this study, the micro-CHP's startup time and consequent cost is neglected.

$$\cos t = \sum_{h=1}^{24} [TOU(h) p_{grid}(h) + G_p G_{CHP}(h)] \tag{1}$$

where TOU(h) is the main grid's time-of-use price tariff; Pgrid(h) is the power transferred between main grid and smart home; Gp is the natural gas price; And Gchp(h) is the micro-CHP's consumed gas.

2.2 Constraints

Electrical Demand Supply Balance. The loads contain thermal appliances (hot water and air conditioner) and electrical appliances. We assume electrical appliances are must run loads that can not be scheduled. While thermal appliances consumption can be scheduled to avoid peak hours. For each household, let $P_{ms}(h)$ denote the total energy consumption of electrical(must run) appliances. while p_{heat}(h) and p_{water}(h) represent the air conditioner and hot water energy consumption one-hour respectively. P_{batt}(h) is the battery's output power and P_{chp}(h) denote the micro-CHP's electrical output power. So, electrical demand-supply balance constraint would be as follow:

$$P_{grid}(h)-p_{batt}(h)+p_{CHP}(h)=p_{ms}(h)+p_{heat}(h)+p_{water}(h) \tag{2}$$

Constraint of Battery. The battery can charge or discharge at different condition, so their can avoid peak time energy and can more efficient use of the overall energy.

$$0\leq\frac{p_{batt}^{ch}(h)}{\eta_{ch}}\leq p_{ch}^{\max} \tag{3}$$

$$0\leq p_{batt}^{dch}(h)\bullet\eta_{dch}\leq p_{dch}^{\max} \tag{4}$$

$$p_{batt}(h)=\frac{p_{batt}^{ch}(h)}{\eta_{ch}}-p_{batt}^{dch}(h)\bullet\eta_{dch} \tag{5}$$

$$SOC(h+1)=SOC(h)+\frac{p_{batt}^{ch}(h)-p_{batt}^{dch}(h)}{E_{batt}} \tag{6}$$

$$SOC^{\min}\leq SOC(h)\leq SOC^{\max} \tag{7}$$

where SOC(h) is the state of charge in the battery; SOC$^{\min}$ and SOC$^{\max}$ are the minimum and the maximum SOC, respectively; P_{batt}^{ch}(h) is the battery's charging power; P_{batt}^{dch}(h) is the batter's discharging power; η_{ch} and η_{dch} are the battery's charging and discharging efficiency, respectively.

Desired Hot Water and Building Temperature. The building is modeled within the context of desired hot water temperature and building temperature.

In modeling the hot water storage, the energy equivalent of the hot water storage at each time step is taken in consideration and the dynamic of the water flow is not considered. So, the water storage temperature at each hour is calculated as the following equation [13], [8]:

$$T_{st}(h+1)=\frac{V_{cold}(h)*(T_{st}^{cold}-T_{st}(h))+V_{total}*T_{st}(h)}{V_{total}}+\frac{H_{air}(h)}{V_{total}*C_{water}} \tag{8}$$

$$T_{st}^{\min}\leq T_{st}(h)\leq T_{st}^{\max} \tag{9}$$

where $T_{st}^{\min}$ and $T_{st}^{\max}$ are the minimum and the maximum desired hot water temperature respectively.

According to the thermal modeling for a building presented in [13], the building temperature at each hour is obtained by:

$$T_{in}(h+1)=T_{in}(h)*e^{\frac{-\Delta}{\tau}}+(R*H_{air}(h)+T_{out}(h))*(1-e^{\frac{-\Delta}{\tau}}) \tag{10}$$

$$T_{in}^{\min}\leq T_{in}(h)\leq T_{in}^{\max} \tag{11}$$

where Δ=1h and $\tau=RC$. The values used are R=18°C/kW, C=0.525kWh/°C, and the initial room temperature=20°C.

Constraint of Micro-CHP Operation. Electrical and thermal output power limits for micro-CHP:

$$V_{CHP}(h)*P_{CHP}^{\min}\leq P_{CHP}(h)\leq V_{CHP}(h)*P_{CHP}^{\max} \tag{12}$$

$$V_{CHP}(h)*H_{CHP}^{\min}\leq H_{CHP}(h)\leq V_{CHP}(h)*H_{CHP}^{\max} \tag{13}$$

Micro-CHP electrical and thermal efficiency:

$$P_{CHP}(h) = H_{CHP}(h) * \frac{\eta_e}{\eta_{th}} \tag{14}$$

Micro-CHP output power ramp rate:

$$-P_{rr} \le P_{CHP}(h) - P_{CHP}(h-1) \le P_{rr} \tag{15}$$

$$-\frac{\eta_{th}}{\eta_e} * P_{rr} \le H_{CHP}(h) - H_{CHP}(h-1) \le \frac{\eta_{th}}{\eta_e} * P_{rr} \tag{16}$$

where $H_{chp}(h)$ is the micro-CHP's thermal output power. $V_{chp}(h)$ is the micro-CHP's status. η_e and η_{th} are the micro-CHP's electrical and thermal efficiency, respectively.

3 Scheduling Using Hybrid PSO-DE Algorithm

The objective of the scheduler is to maximize the net benefits. Based on (1), the corresponding optimization problem is to find the real-value variable for battery, micro-CHP and thermal appliances. Therefore, it can be considered as a constraint real-value optimization problem. This paper mixes particle swarm optimization and differential evolution (PSO-DE) to solve the constraint optimization problem.

3.1 Overview of PSO

Particle swarm optimization is a stochastic global optimization method inspired by the choreography of a bird flock. PSO relies on the exchange of information between individuals, called particles. In PSO, each particle adjusts its trajectory stochastically towards the positions of its own previous best performance and the best previous performance of its neighbors or the whole swarm. At the each iteration, the velocity and position updating rules are given by:

$$v_{i,j}^{t+1} = wv_{i,j}^t + c_1 r_1 (pbest_{i,j}^t - x_{i,j}^t) + c_2 r_2 (gbest_j^t - x_{i,j}^t) \tag{17}$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^t \tag{18}$$

where w is an inertia weight factor, c_1 is a cognition weight factor, c_2 is a social weight factor, r_1 and r_2 are two random numbers uniformly distributed in the range of [0,1]. In this version, the variable $V_{i,j}$ is limited to the range $\pm V_{\max}$ [17-19] analyzed and introduced the velocity adjustment as

$$v_{i,j}^{t+1} = R_1 (pbest_{i,j}^t - x_{i,j}^t) + R_2 (gbest_j^t - x_{i,j}^t) \tag{19}$$

where R1 and R2 are generated using abs(N(0,1)) According to the statistical knowledge, the mean of abs(N(0,1))is 0.798 and the variance is 0.36.

3.2 Overview of DE

Different evolution (DE) [15] has become a popular algorithm in global optimization. DE starts the search with an initial population containing NP individuals, which are randomly sampled from the search space. Then, one individual called the target vector in the population is used to generate a mutant vector by the mutation operation. So far, several mutation strategies have been proposed [14].

Subsequently, DE applies a crossover operator to generate the offspring individual, the crossover is employed and executed as follows:

$$u_{i,j} = \begin{cases} y_{i,j}^{t}\ if \quad rand \le CR \quad OR \quad j = j_{rand} \\ x_{i,j}^{t} \qquad otherwise \end{cases} \tag{20}$$

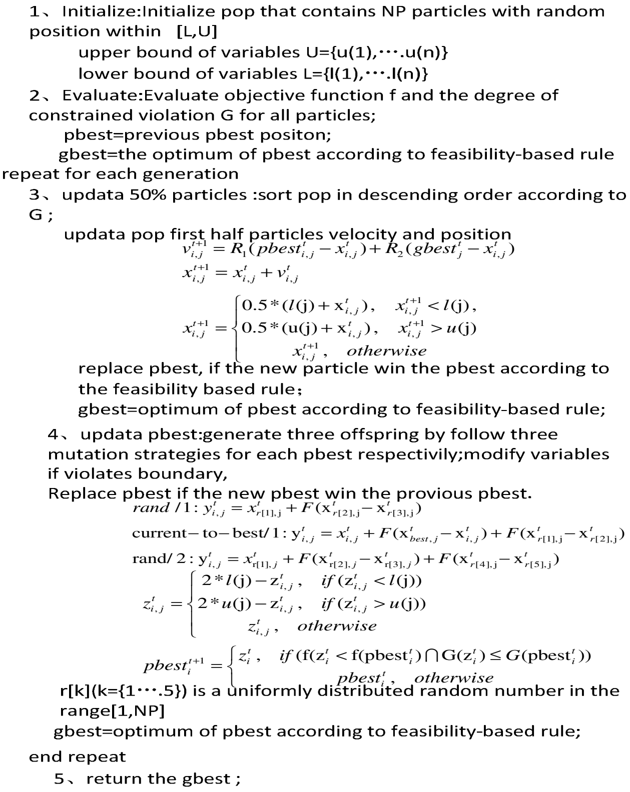
where $i \in \{1,2,...,NP\}$, $\mathrm{j} \in \{1,2,...,\mathrm{n}\}$,rand is a uniformly distributed rand number between [0,1], $\mathrm{j}_{\mathrm{rand}}$ is a randomly selected integer from [1,n],CR is the crossover control parameter, $u_{i,j}^{t}$ is the jth component of the trial vector .

Finally, the target vector x_i is compared with the trial vector u_i in terms of the objective function value and the better one survives into the nest generation:

$$x_i^{t+1} = \begin{cases} u_i^t\ ,if \quad f(\mathrm{u}_i^t \le \mathrm{f}(\mathrm{x}_i^t)) \\ x_i^t \quad ,otherwise \end{cases} \tag{21}$$

3.3 PSO-DE for Real-Value Constraint Optimization

The PSO-DE's main procedure can be summarized in Fig.1 [19].



```
  1、Initialize:Initialize pop that contains NP particles with random
  position within  [L,U]
        upper bound of variables U={u(1),···.u(n)}
        lower bound of variables L={l(1),···.l(n)}
  2、Evaluate:Evaluate objective function f and the degree of
  constrained violation G for all particles;
      pbest=previous pbest positon;
      gbest=the optimum of pbest according to feasibility-based rule
repeat for each generation
  3、updata 50% particles :sort pop in descending order according to
  G ;
      updata pop first half particles velocity and position
          $v_{i,j}^{t+1} = R_1(pbest_{i,j}^t - x_{i,j}^t) + R_2(gbest_j^t - x_{i,j}^t)$
          $x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^t$
          $x_{i,j}^{t+1} = \begin{cases} 0.5*(l(\mathrm{j}) + \mathrm{x}_{i,j}^t), & x_{i,j}^{t+1} < l(\mathrm{j}), \\ 0.5*(\mathrm{u(j)} + \mathrm{x}_{i,j}^t), & x_{i,j}^{t+1} > u(\mathrm{j}) \\ x_{i,j}^{t+1}, & otherwise \end{cases}$
        replace pbest, if the new particle win the pbest according to
        the feasibility based rule;
        gbest=optimum of pbest according to feasibility-based rule;
    4、updata pbest:generate three offspring by follow three
    mutation strategies for each pbest respectivily;modify variables
    if violates boundary,
    Replace pbest if the new pbest win the provious pbest.
          $rand/1: y_{i,j}^t = x_{r[1],\mathrm{j}}^t + F(\mathrm{x}_{r[2],\mathrm{j}}^t - \mathrm{x}_{r[3],\mathrm{j}}^t)$
          $\mathrm{current-to-best/1}: \mathrm{y}_{i,j}^t = x_{i,j}^t + F(\mathrm{x}_{best,j}^t - \mathrm{x}_{i,j}^t) + F(\mathrm{x}_{r[1],\mathrm{j}}^t - \mathrm{x}_{r[2],\mathrm{j}}^t)$
          $\mathrm{rand/2}: \mathrm{y}_{i,j}^t = x_{\mathrm{r}[1],j}^t + F(\mathrm{x}_{\mathrm{r}[2],j}^t - \mathrm{x}_{\mathrm{r}[3],j}^t) + F(\mathrm{x}_{r[4],\mathrm{j}}^t - \mathrm{x}_{r[5],\mathrm{j}}^t)$
          $z_{i,j}^t = \begin{cases} 2*l(\mathrm{j}) - \mathrm{z}_{i,j}^t, & if(\mathrm{z}_{i,j}^t < l(\mathrm{j})) \\ 2*u(\mathrm{j}) - \mathrm{z}_{i,j}^t, & if(\mathrm{z}_{i,j}^t > u(\mathrm{j})) \\ z_{i,j}^t, & otherwise \end{cases}$
          $pbest_i^{t+1} = \begin{cases} z_i^t, & if(\mathrm{f}(\mathrm{z}_i^t < \mathrm{f}(\mathrm{pbest}_i^t) \cap \mathrm{G}(\mathrm{z}_i^t) \le G(\mathrm{pbest}_i^t)) \\ pbest_i^t, & otherwise \end{cases}$
      r[k](k={1···.5}) is a uniformly distributed random number in the
      range[1,NP]
     gbest=optimum of pbest according to feasibility-based rule;
  end repeat
    5、return the gbest ;
```

Fig. 1. PSO-DE main procedure

As described above, the paper is to solve a constrained optimization problem, the strategy for handling constraints is usually the use of penalty function methods. However, the main problem is that they require a careful fine tuning of the penalty factors .In order to overcome the drawback of choice penalty factors, this paper applies PSO-DE method to handle constraints, which does not require setting any additional parameters in comparison to the original PSO. The feasibility-based rules are applied to handle constraints of the problem.

In order to handle the constraints, we minimize the original objective function f (x) as well as the degree of constraint violation G(x). At each generation, pop is sorted according to the degree of constraint violation in a descending order. Only the first half of pop are evolved by using Krohling and dos Santos Coelho's PSO [18].

In order to compensate the convergence speed and supply more valuable information to adjust the particles' trajectories, the pBest is updated by Different evolution. DE-based search process motivates the particles to search for new regions including some lesser explored regions and enhance the particles capability to explore the vast search space [19].

4 Simulation Result

The case study is a typical residential building. A micro-CHP with 3kW capacity is considered for the building. The water storage capacity is 80L. The building's hot water demand is shown in Fig.2. The building loads include must run electrical appliances and thermal loads. Fig.3 shows the total electrical demand by the must run appliances and the price of electricity supplied to terminal loads [8, 20]. Which the must run include lights, cook, fridge, computers, washing machine, dryer, dish washer and pool pump.

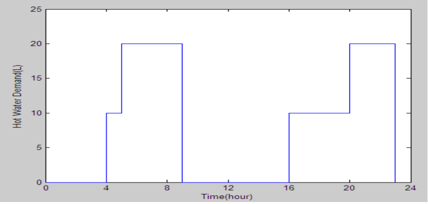


Fig. 2. Hot water demand in building

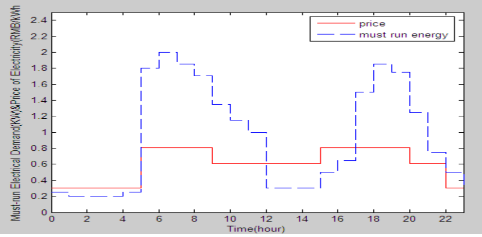


Fig. 3. Total electrical demand and the price of electricity

In this case the price of natural gas is 2.05RMB/m3, and the price of electricity fed into grid is 0.457 RMB/kWh [20]. Table I shows assumed parameters in solving(1).

Table 1. Assume values for parameters

Parameter	Value	Unit
$T_{in}^{\min}, T_{in}^{\max}\ T_{st}^{\min}, T_{st}^{\max}\ T_{st}^{cold}$	22,24,72,74,10	°C
C_{water}	$11.61*10^{-4}$	$\frac{kWh}{L°C}$
$P_{CHP}^{\min}, P_{CHP}^{\max}\ H_{CHP}^{\min}, H_{CHP}^{\max}\ P_{ch}^{\max}, P_{dch}^{\max}$	0.3,3,0.5,5,3,3	kW
P_{rr}	2.5	$\frac{kW}{h}$
$\eta_e, \eta_{th}\ \eta_{ch}, \eta_{dch}$	30,50,0.9,0.9	%
$SOC_{\min}, SOC_{\max}$	0.3,1	p.u.
G_{ref}	$92.59*10^{-3}$	$\frac{m^3}{h}$

The PSO-DE algorithm is programmed using the C++ programming language using a PC with an Intel dual core processor 2.6GHz on a Windows XP operation system. A visual C++ compiler was used in this work.

The proposed algorithm takes 5 trials to get the final best cost. And in each trial, the population size and maximum iteration take 300 and 5000 respectively. The best result with minimum cost is shown in Table II.

Table 2. Five trials of algorithm

Generation	Fitness	Vilolation
93	19.21	0.0
80	15.89	0.0
0	17.90	0.0
132	17.48	0.0
72	16.81	0.0

The simulation results show the algorithm can achieve optimum result under kinds of constraints. As been shown in Fig. 4, the building's temperature have been set within desired temperature (ie.,22-24), the hot water temperature also been set to comfort temperature (ie.,74-76) in the demand time (ie.,4:00-9:00,16:00-22).

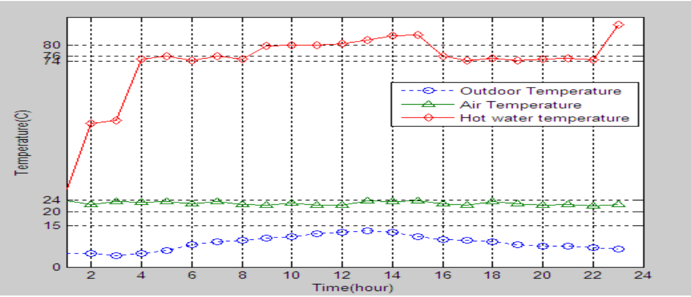


Fig. 4. Hot water temperature and building temperature

The micro-CHP output power, battery output power and its SOC is depicted in Fig. 5. The positive and negative values represent battery charging and discharging respectively.

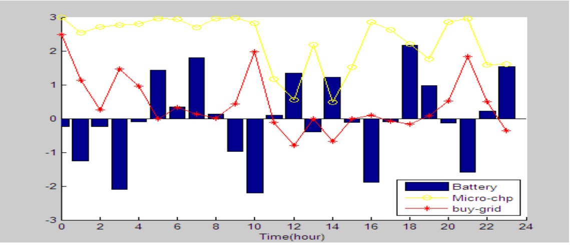


Fig. 5. Micro-CHP& Buy_grid& Battery

As expected, the battery coordinates with the micro-CHP output power to achieve economic operation of micro grid. The battery charges and discharges during low price and peak price hours, respectively. While the micro-CHP operation at its maximum capacity of peak load price hours.

Also ,as shown in the Fig.4 it achieve optimum cost operation by buying minimum electrical power during peak hours and selling its extra electrical power to the main grid at off-peak price hours.

5 Conclusion

This paper pioneers a problem of a residential smart home user equipped with kinds of appliances. We developed an optimal control algorithm for the smart home. The objective of the optimal control algorithm is to reduce the total electricity cost over a billing period (i.e., a day).This proposed PSO-DE algorithm can be used in home energy management systems and help in realization of optimum cost operation. As future study, it is suggested to look into issues such as optimization under price uncertain environment.

Acknowledgments. This work was supported in part by the Natural Science Foundation of China (61075064, 61034004, 61005090), Program for New Century Excellent Talents in University of Ministry of Education of China. Ministry of Education (NCET-10-0633), the Fundamental Research Funds for the Central Universities, and the Research Fund of State Key Lab. of Management and Control for Complex systems.

References

1. Mansouri, I., Newborough, M., Probert, D.: Energy consumption in UK households: impact of domestic electrical appliances. Appl. Energy 54(3), 211–285 (1996)
2. Hamidi, V., Li, F., Robinson, F.: Demand response in the UK's domestic sector. Electr. Power Syst. Res. 79(12), 1722–1726 (2009)

3. Katz, J.S.: Educating the smart grid. Presented at the IEEE En- ergy 2030 Conf., Atlanta, GA (November 17-18, 2008)
4. Colson, C.M., Nehrir, M.H.: A review of challenges to real-time power management of microgrids. Presented at the 2009 Power Energy Soc. Gen. Meet., Calgary, AB, Canada, PESGM2009-001250 (2009)
5. Mohsenian-Radand, A., Leon-Garcia, A.: Optimal residential load control with price prediction in real-time electricity pricing environments. IEEE Trans. Smart Grid 1(2), 120–133 (2010)
6. Conejo, A.J., Morales, J.M., Baringo, L.: Real-time demand response model. IEEE Trans. Smart Grid 1(2), 236–242 (2010)
7. Li, N., Chen, L., Low, S.H.: Optimal demand response based on utility maximization in power networks. In: Proc. IEEE Power Energy Soc. Gen. Meet. (July 2011)
8. Pedrasa, M.A.A., Spooner, T.D., MacGill, I.F.: Coordinated scheduling of residential distributed energy resources to optimize smart home energy services. IEEE Trans. Smart Grid 1(2), 134–143 (2010)
9. Mohsenian-Rad, A., Wong, V.W.S., Jatskevich, J., Schober, R., Leon-Garcia, A.: Autonomous demand-side management based on game- theoretic energy consumption scheduling for the future smart grid. IEEE Trans. Smart Grid 1(3), 320–331 (2010)
10. Sane, H., Guay, M.: Minimax dynamic optimization over a finite- time horizon for building demand control. In: Proc. Amer. Control Conf., pp. 1469–1474 (June 2008)
11. Samadi, P., Mohsenian-Rad, A., Schober, R., Wong, V.W.S., Jatskevich, J.: Optimal real-time pricing algorithm based on utility maximization for smart grid. In: Proc. IEEE Int. Conf. Smart Grid Commun., pp. 415–420 (2010)
12. Wi, Y.-M., Lee, J.-U., Joo, S.-K.: Electric Vehicle Charging Method for Smart Homes/Buildings with a Photovoltaic System. IEEE Transactions on Consumer Electronics 59(2) (May 2013)
13. Tasdighi, M., Ghasemi, H., Rahimi-Kian, A.: Residential Microgrid Scheduling Based on Smart Meters Data and Temperature Dependent Thermal Load Modeling. IEEE Trans. Smart Grid (2013)
14. Price, K., Storn, R., Lampinen, J.: Differential Evolution: A Practical Approach to Global Optimization. Springer-Verlag (2005) ISBN:3-540-20950-6
15. Neri, F., Tirronen, V.: Recent advances in differential evolution: a survey and experimental analysis. Artificial Intelligence Review 33(1-2), 61–106 (2010)
16. Mohsenian-Rad, A.-H., Wong, V.W.S., Jatskevich, J., Schober, R.: Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid. In: Proc. IEEE Conf. Innov. Smart Grid Technol., Gaithersburg, MD, USA (2010)
17. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2002)
18. Krohling, R.A., dos Santos Coelho, L.: Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems. IEEE Transactions on Systems, Man, Cybernetics Part B: Cybernetics 36(6), 1407–1416 (2006)
19. Liu, H., Cai, Z., Wang, Y.: Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Applied Soft Computing 10, 629–640 (2010)
20. Guan, X., Xu, Z., Jia, Q.-S.: Energy-Efficient Buildings Facilitated by Microgrid. IEEE Trans. Smart Grid 1(3) (December 2010)

A Multiobjective Large Neighborhood Search for a Vehicle Routing Problem

Liangjun Ke and Laipeng Zhai

The State Key Laboratory for Manufacturing Systems Engineering,
Xian Jiaotong University, Xi'an, 710049, China
keljxjtu@xjtu.edu.cn

Abstract. In this paper, a multiobjective adaptive large neighborhood search is proposed for a vehicle routing problem (VRP) of which the objectives are the total travel time and the cumulative time, i.e., the total arrival time at all customers. It hybrids destroy-repair operators with local search for generating new solutions. An adaptive probabilistic rule based on Pareto dominance is proposed to select a combination of destroy-repair operator. The effectiveness of the proposed algorithm is supported by the experimental study.

Keywords: multiobjective optimization, vehicle routing problem, adaptive large neighborhood search.

1 Introduction

Vehicle routing problem (VRP) is one of the most important combinatorial optimization problems. In this problem, a set of customers are dispersed in a graph. Each customer is associated with a demand. vehicles are scheduled to serve these customers so as to achieve one or more optimal objectives whilst the route of each vehicle must satisfy specific requirements. The most common studied objectives are the total travel time, the number of vehicles, makespan, balance, and others [1].

Recently, cumulative VRP becomes a hot topic [2, 3]. It aims at minimizing the cumulative time, that is, the total arrival time of all customers. This problem was extended from the delivery man problem [4, 5], and can be used to model many problems such as the routing schedule during the disaster aids [2].

Although many researchers considered VRP with only one single objective, VRP is multi-objective in nature [1]. Multiobjective VRP (MVRP) has attracted great research interests [1]. A lot of approaches have been used to deal with various MVRPs. A popular approach is the scalar approach, which transforms a multiobjective problem into a single objective problem by weighted sum method or other methods, then solves it by a single objective heuristic or exact algorithm [6, 7]. Based on the concept of Pareto dominance, Pareto methods are also widely used [8–10]. Other approaches, e.g., genetic algorithm [11], lexicographic method [12] and ant colony optimization [13], etc, were also adopted. Since VRP is

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 301–308, 2014.
© Springer International Publishing Switzerland 2014

a NP-hard problem, metaheuristics are widely adopted for finding satisfactory solutions within acceptable computational time.

In this paper, we consider an MVRP. In the MVRP, there are n customers. Customer i, staying at node i has a demand d_i. A fleet of vehicles start from node 0 and each node $n+1$ to serve these customers. The travel time between every two nodes $i, j (i, j \in \{0, 1, \cdots, n, n+1\})$ is w_{ij}. A feasible solution consists of a set of paths such that each customer is served by one path, and the total demand of a path is not more than the vehicle capacity. The objectives are the total travel time and the cumulative time at all customers. Intuitively, a solution with smaller total travel time may have smaller cumulative time. Nevertheless, these two objectives may be not always compatible [2]. Therefore this problem can not be reduced to a single objective problem. By optimizing this problem, one can obtain a solution for minimizing the total travel time or the cumulative time in a single run. Moreover, one can obtain a set of tradeoff solutions for decision making.

To deal with this problem, we propose a new algorithm extended from adaptive large neighborhood search (ALNS). ALNS has been proven to be a powerful metaheuristic for many variants of single objective VRP. It was firstly proposed by Ropke and Pisinger [14]. It is closely related to the large neighborhood search developed by Shaw [15]. ALNS generates new solutions by using destroy and repair operators. Local search is optionally adopted for exploiting the neighborhood of those solutions. An adaptive probabilistic rule is used to schedule destroy and repair operators according to their weights, which are renewed based on the previous search experience.

The remainder of this paper is structured as follows. Section 2 describes the details of the proposed multiobjective ALNS algorithm. The experimental results are presented in section 3. Finally, the main results are concluded in section 4.

2 Multiobjective Adaptive Large Neighborhood Search

Our algorithm, deonted as MALNS, evolves a population of N solutions over time where N is population size, and employs an external archive EA to store the nondominated solutions. It works as follows. At first, a population of solutions are initialized. After that, starting from each individual in the population, a new solution is generated by a combination of destroy-repair operator. A combination is probabilistically chosen from a set of candidate combinations, depending on their weights. Subsequently, local search is adopted to improve each new solution. By using the obtained solutions, our algorithm updates the population, the external archive EA, and the weights of the combinations of destroy-repair operators. It terminates when a stopping condition is satisfied.

2.1 The Single Objective Function

The single objective function is used to evaluate the quality of a move in destroy-repair operators or local search. Two things are considered to define the single objective function: 1) it is a weighted aggregation of the objectives in MVRP; 2) to obtain a wide and evenly nondominated front, it is desirable to guide the destroy-repair operator or local search to explore different directions. Therefore, the weight vector (or search direction) should be carefully assigned.

Formally, the single objective function is given as follows.

$$\mathcal{F}(x, \lambda) = \lambda^1 \frac{f_1(x)}{f_1^*} + \lambda^2 \frac{f_2(x)}{f_2^*} \tag{1}$$

where $\lambda = (\lambda^1, \lambda^2)$ is a weight vector, $\lambda^1 + \lambda^2 = 1, \lambda^1, \lambda^2 \geq 0$. Since the difference between f_1 and f_2 is very big, scale transformation is carried out in (1). f_1^* and f_2^* are the minimal values of these two objectives respectively. In practice, they are approximated by the minimal values obtained so far. Each weight vector is randomly selected.

2.2 Destroy and Repair Operators

Many different destroy and repair operators have been proposed in the literature. An interesting survey is available in [14]. MALNS heavily depends on simpler destroy-repair operators for exploring new competitive search areas, and uses local search for exploitation. The destroy operators we used are random removal, worst removal, relatedness removal, and cluster removal [14,16]. These operators remove some customers from the routes. The set of unvisited customers, called request bank [14], is denoted as U. During the running, removed customers are saved in U. The maximal size of U is denoted as u. Parameter u significantly affects the behavior of MALNS. With a larger value, the operators are able to explore larger search area. The repair operators consist of the basic greedy insertion heuristic and regret insertion heuristic.

2.3 Selecting a Combination of Destroy-Repair Operator

There are eight combinations of destroy-repair operators. During the search, a roulette-wheel selection method is employed to select one combination every time. Each combination c_i is associated with a weight w_i which is used to measure how well combination c_i has performed in past iterations. At the beginning of the algorithm, the initial value of w_i is set to 1. The combination c_i is selected with probability $p_i = w_i / \sum_{j=1}^{8} w_j$.

2.4 Local Search

Local search plays an very important role in the design of metaheuristics for VRP [17]. A local search operator iteratively improves a solution by exploring its neighborhood in terms of the single objective function given in (1). Given a feasible solution to VRP represented by a set of routes $x = \{R_1, \ldots, R_l, \ldots, R_v\}$, where R_l is the set of customers serviced by route (or vehicle) l. Its neighborhood is denoted as $N(x)$.

The following components are critical in the implementation of a local search operator [17]. The first is the starting solution. The second is the mechanism to generate neighboring solutions of a given solution. The third is the acceptance criterion. Two popular acceptance strategies are first-accept (FA) and best-accept (BA). The FA strategy chooses the first neighboring solution that satisfies the pre-specified acceptance criterion (e.g., the objective variation after a move). The BA strategy checks all neighboring solutions which satisfy a criterion and chooses the best among them. The fourth is the condition when to stop the local search operator.

The 2-opt, exchange, cross and relocation operators [17] are adopted to generate neighborhood. FA strategy is used. These operators are invoked one by one. Local search will be ended when no more improvement can be achieved.

Unlike 2-opt, exchange, cross and relocation operators are inter-route operators. To speed up these inter-route operators, we first propose the concept of neighborhood of routes based on polar angle. Polar angle is the basic tool in the famous sweep algorithm [18]. For a route R_l, its polar angle is defined as the polar angle of its center of gravity. Two routes are said to be close if their polar angles are close. For a route, only the N closest neighboring routes are permitted to inter-change. When to select a move, a local search operator only checks its neighborhood for each route. By interchanging with the routes out of smaller neighbor, it is more likely to find better solution with smaller travel time or cumulative time.

2.5 Update of Weights

In MALNS, a combination c_i is associated with a score, denoted by φ_i. At each iteration, φ_i is initialized to 0. After an iteration, φ_i of the chosen combination c_i will be renewed based on the quality of the solutions constructed at the iteration. In detail, starting from each solution (in current population) x_s, a new solution x_n is generated by a combination c_i and improved by local search. If x_n is nondominated by solutions found so far, the score of c_i is increased by 15; If x_n dominates x_s, the score of c_i is increased by 10; If x_n is nondominated by x_s, the score of c_i is increased by 5; Otherwise, no score is obtained. Formally, it is updated as follows:

$$\varphi_i = \begin{cases} \varphi_i + 15 & \text{if the new solution } x_n \text{ is nondominated} \\ & \quad \text{by solutions found so far} \\ \varphi_i + 10 & \text{if the new solution } x_n \text{ dominates} \\ & \text{its starting solution } x_s \\ \varphi_i + 5 & \text{if the new solution } x_n \text{ is nondominated} \\ & \quad \text{by its starting solution } x_s \\ \varphi_i & \text{otherwise} \end{cases} \tag{2}$$

Every iteration, each weight w_i is updated based on the scores obtained.

$$w_i = (1-\rho)w_i + \rho\frac{\varphi_i}{\max(Freq_i, 1)} \tag{3}$$

where $Freq_i$ denotes the times the combination c_i has been applied in the past iteration. ρ is a parameter which controls the forgotten rate of the past experience. ρ is set to 0.05. MALNS re-initializes the weights to 1 once no new nondominated solutions can be found during consecutive 50 iterations.

2.6 Population Initialization

N solutions are constructed by the initial procedure. At first, each customer is inserted in request bank, and then each customer is inserted by the regret insertion heuristic. Note that the single objective function is given by (1). To construct the the lth solution ($l \in \{1, \cdots, N\}$), the weight vector is $((l-1)/(N-1), (N-l)/(N-1))$.

2.7 Update of Population

At each iteration, we only accept a new generated solution of which request bank is empty and the number of routes is $|R|$. The population is updated by using nondominance ranking and crowding distance in [19]. From the last population and new generated solutions, a set of fronts are obtained by nondominance ranking. The first front F_1 consists of the nondominated solutions. The second front is the set of solutions which are only dominated by the first front, and so on. Crowding distance is the average side length of the cuboid formed by the objective values of these nearest neighboring solutions [19].

2.8 Update of External Archive

Once a solution x is accepted, the external archive EA is renewed as follows: If no vector in EA dominates $F(x)$, $F(x)$ will be added to EA. At the mean time, all the vectors dominated by $F(x)$ will be removed from EA.

3 Experimental Results

MALNS was coded in C++ and tested on a PC with Pentium 4, 2.4G CPU, and 4GB RAM. It was tested on 20 large-scale instances with 240 to 483 customers in [20]. The travel time between every two nodes is their Euclidean distance. All travel time is rounded to double precision [2,3].

Based on the preliminary test, the population size was set to 30. As done in [3], the maximal size u of request bank was randomly chosen from [10,60]. For each instance, MALNS was test the same times in [3] (i.e., 5) independently and stopped when a given time limit was achieved. In our experiment, the time limit was chosen as follows. At first the computational time in [3] was transformed, then the transformed time T was set. For example, the computational time of GWKC1 in [3] is 1038, then the transformed time T is 865, since our CPU is 1.2 times as fast as the one of [3].

3.1 Performance Metrics

It is widely accepted that, given two nondominated fronts, the better front has the following properties: it is closer to Pareto-optimal front; it distributes more evenly and widely in objective space [21]. Many performance metrics have been proposed to evaluate the solution quality. Among them, hypervolume is a very nice and popular metric [22], therefore we adopted it in this paper. The Hypervolume value of a nondominated front is the volume of the area in the objective space which is bounded by a reference point and dominated by the front itself. Let ζ be all the nondominated solution sets found in a test, the reference point was chosen as $(\max\{f_1(x)|x \in \zeta\}, \max\{f_2(x)|x \in \zeta\})$. Larger hypervolume value indicates better solution quality.

In order to pictorially illustrate the nondominated fronts, a statistical tool, called summary-attainment surface [23], is employed. Summary-attainment surface refers to the union of all tightest points in the objective space obtained by an algorithm. If an algorithm is tested l times, there will be l summary-attainment surface. The first, $l/2$th, lth summary-attainment surface is called the best, median, worst summary-attainment surface respectively.

3.2 Comparison with a Weighted Sum ALNS

To study MALNS, we implemented a weighted sum ALNS which works as follows: the same procedure of MALNS with only one single weight vector is performed N times ($N = 30$). In the lth time ($l = 1, \cdots, N$), only weight vector $((l-1)/(N-1), (N-l)/(N-1))$ is used and the computational time is T/N.

As seen from Fig. 1, MALNS provides better hypervolume than the weighted sum ALNS. According to the summary-attainment surfaces shown in Fig. 2, MALNS can provide wider nonodimianted front. We also note that the weighted sum ALNS performs better in some central parts. The reason may be that the computational resource in the weighted sum ALNS is biased to search some specific areas.

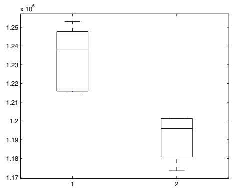


Fig. 1. The hypervolume values obtained by MALNS and weighted sum ALNS (shown from left to right) are tested for GWKC20.

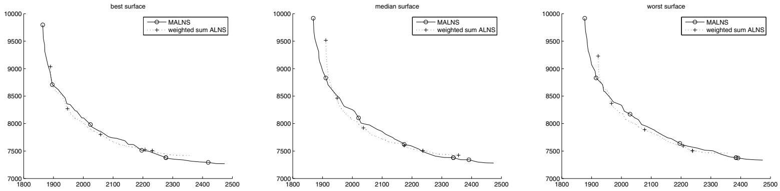


Fig. 2. The best, median, and worst summary-attainment surfaces obtained by MALNS and weighted sum ALNS for GWKC20

4 Conclusion

This paper presented a multi-objective adaptive large neighborhood search for an MVRP of which the objectives are the total travel time and the cumulative time. Although the total travel time and the cumulative time has been separately studied before, this paper investigated these objectives together. It selects a combination of destroy-repair operator for generating new solutions and improves them by local search. According to the yielded solutions, the preference of each combination is renewed based on the Pareto dominance. Compared with a weighted sum ALNS, the proposed algorithm performs better.

References

1. Jozefowiez, N., Semet, F., Talbi, E.: Multiobjective vehicle routing problems. European Journal of Operational Research 189, 293–309 (2008)
2. Ngueveu, S.U., Prins, C., Wolfler-Calvo, R.: An effective memetic algorithm for the cumulative capacitated vehicle routing problem. Computers and Operations Research 37, 1877–1885 (2010)
3. Ribeiro, G., Laporte, G.: an adaptive large variable neighborhood search heuristic for cumulative capacitated vehicle routing problem. Computers and Operations Research 39(3), 728–735 (2012)
4. Lucena, A.: Time-dependent traveling salesman problem:The deliveryman case. Networks 20, 753–763 (1990)
5. Fischetti, M., Laporte, G., Martello, S.: The delivery man problem and cumulative matroids. Operations Research 41, 1055–1064 (1993)
6. Bowerman, R., Hall, B., Calamai, P.: A multi-objective optimization approach to urban school bus routing: Formulation and solution method. Transportation Research Part A 29, 123–197 (1995)
7. Riera-Ledesma, J., Salazar-Gonzalez, J.: The biobjective travelling purchaser problem. European Journal of Operational Research 160, 599–613 (2005)
8. Geiger, M.: Genetic algorithms for multiple objective vehicle routing. In: Meta-Heuristic International Conference 2001 (MIC 2001), pp. 348–353 (2001)
9. Lacomme, P., Prins, C., Sevaux, M.: A genetic algorithm for a bi-objective capacitated arc routing problem. Computers and Operations Research 33, 3473–3493 (2006)

10. Jozefowiez, N., Semet, F., Talbi, E.G.: The bi-objective covering tour problem. Computers and Operations Research 34, 1929–1942 (2007)
11. Ombuki, B., Ross, B., Hanshar, F.: Multi-objective genetic algorithm for vehicle routing problem with time windows. Applied Intelligence 24, 17–30 (2006)
12. Current, J., Schilling, D.: The median tour and maximal covering tour problems: Formulations and heuristics. European Journal of Operational Research 73, 114–126 (1994)
13. Doerner, K., Focke, A., Gutjahr, W.: Multicriteria tourplanning for mobile healthcare facilities in a developing country. European Journal of Operational Research 179, 1078–1096 (2007)
14. Ropke, S., Pisinger, D.: A unified heuristic for a large class of vehicle routing problems with backhauls. European Journal of Operational Research 171, 750–775 (2006)
15. Shaw, P.: A new local search algorithm providing high quality solutions to vehicle routing problems. In: Technical report, University of Strathclyde, Glasgow (1997)
16. Ropke, S., Pisinger, D.: An adaptive large neighborhoodsearch heuristic for the pickup and delivery problem with time windows. Transportation Science 40, 455–472 (2006)
17. Braysy, O., Gendreau, M.: Vehicle routing problem with time windows, part I: route construction and local search algorithms. Transportation Science 39(1), 104–118 (2005)
18. Gillett, B., Miller, L.: A heuristic algorithm for the vehicle dispatch problem. Operations Research 22, 340–349 (1974)
19. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA?II. IEEE Trans. Evolutionary Computation 6(2), 182–197 (2002)
20. Golden, B.L., Wasil, E.A., Kelly, J.P., Chao, I.M.: Metaheuristics in vehicle routing. In: Fleet Management and Logistics, pp. 33–56. Springer, Heidelberg (1998)
21. Coello, C., Veldhuizen, D.V., Lamont, G.: Evolutionary Algorithms for Solving Multi-objective Problems. Kluwer (2002)
22. Fonseca, C.M., Paquete, L., Lopez-Ibanez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006), pp. 1157–1163. IEEE Press, Piscataway (2006)
23. Knowles, J.: A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In: Proceedings of the Fifth International Conference on Intelligent Systems Design and Applications (ISDAV), pp. 552–557. IEEE Computer Society (2005)

A Self-adaptive Interior Penalty Based Differential Evolution Algorithm for Constrained Optimization

Cui Chenggang*, Yang Xiaofei, and Gao Tingyu

Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai, China
cuicg@sari.ac.cn

Abstract. A self-adaptive interior penalty method is proposed for the constrained optimization problems by using interior penalty method to handle constraints. A set of interior penalty rules are designed to evaluate feasible solutions and infeasible solutions separately. A self-adaptive penalty factor method is proposed to prevent the interior penalty method from being sensitive to the values of penalty factor and to minimize the interior penalty function value of the optimal solution. As an instance of implementation, a different evolution algorithm is improved by means of the method proposed in this paper, based on which 10 benchmark problems are tested. The numerical solution results indicate that the performance of the method is better than four existing state-of-the-art techniques.

Keywords: Constrained optimization, Evolutionary algorithm, Interior penalty method, Differential evolution.

1 Introduction

Evolutionary algorithms (EAs) have been widely used to solve constrained optimization problems (COPs). However, EAs are normally used as “blind heuristics” in the sense of lacking an explicit mechanism to bias the search in constrained search spaces [1]. Several researchers have proposed different mechanisms to incorporate constraints into the fitness function of an EA [2]. Penalty functions are the most common approaches used to handle constraints with EAs [3]. There are two basic types of penalty functions: exterior penalty functions, which penalize infeasible solutions, and interior penalty functions, which penalize feasible solutions. Compared to interior penalty functions, exterior penalty functions are more common in EAs. The main reason is that there is no need to start with a feasible solution in exterior penalty functions. Another category of constraint handling techniques involves the preference of feasible solutions over infeasible solutions [4]. In these methods, a heuristic rule that feasible solutions are preferred over infeasible ones is used to process infeasible solutions. Multiobjective optimization techniques have also been used in the solution of constrained single objective optimization problems [5]. These techniques can be classified based on the way they transform the COP into a multiobjective optimization problem.

* Correspondig author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 309–318, 2014.
© Springer International Publishing Switzerland 2014

In this paper, a set of interior penalty (IP) based selection rules is proposed to balance the avoidance of the constraint boundaries and the minimization of the objective function in the search process of Differential Evolution (DE) algorithms. Feasible solutions closed to the constraint boundaries are penalized to balance the conflict aims of reducing the objective function and approaching the constraint boundaries; infeasible solutions are evaluated by constraint violations to reach feasible region quickly. Three elements are employed to make these rules more effective in DEs: (1) Logarithmic penalty functions are used to make DEs yield a rapid convergence; (2) Penalty factors are updated according to the types of constraints which are determined by the Spearman's rank-order correlation coefficients; (3) Equality constraints are handled by an adaptive relaxing rule. In this paper, a self-adapt interior penalty based differential evolution algorithm is implemented as an example of this constraint handling approach. Finally, the efficiency and effectiveness of the proposed method are evaluated on 10 benchmark problems.

2 Interior Penalty Based Selection Rules

2.1 Statement of the Problem

Generally, a COP can be expressed as follows:

$$\begin{aligned} &\min f(\boldsymbol{x}) \quad , \quad \boldsymbol{x} = (x_1,\ldots,x_n) \in R^n \\ &\text{s.t.} \quad g_j(\boldsymbol{x}) \le 0 \quad , j \in \{1,\ldots,m\} \quad , \\ &\qquad\ \ h_j(\boldsymbol{x}) = 0 \quad , j = q+1,\ldots,m \quad . \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is the decision vector, $f(\boldsymbol{x})$ is the objective function, q is the number of inequality constraints, and m-q is the number of equality constraints. Let $S \subset R^n$ define the search space bounded by the parametric constraints $\underline{x}_i \le x_i \le \overline{x}_i$, $i \in \{1,2,\ldots,n\}$, where $\underline{x}_\mathrm{i}$ and $\overline{x}_i$ are the lower bound and the upper bound of x_i, respectively.

2.2 Interior Penalty Method

The interior penalty method is very popular in the traditional mathematical programming techniques, motivated by minimizing a composite function that reflects the original objective function as well as the influence of the constraints [6].

Given the constrained optimization problem (1), the interior penalty function can be formulated as follows:

$$\phi(x, r(t)) = f(x) + r(t)B(x) \quad . \tag{2}$$

$B(\boldsymbol{x})$ is a penalty term that is nonnegative and approaches ∞ as the constraint boundaries are approached from the interior. $r(t)$ is a penalty factor.

The interior penalty method replaces a COP with a sequence of unconstrained optimization problems, defined as:

$$\lim_{t\to\infty} \min \phi(x, r(t)) . \tag{3}$$

Solving problem (3) sequentially for a monotonously decreasing sequence $\{r(t)\}$ such that $\lim_{t\to\infty} r(t) = 0$ gives a sequence $\{\boldsymbol{x}(r(t))\}$ yielding $\lim_{t\to\infty} f(x(r(t))) = f(x^*)$, where x^* is the optimal solution of problem (1).

2.3 Implementation of the IP Based Selection Rules

In order to balance the avoidance of the constraint boundaries and the minimization of the objective function in the search process, a set of interior penalty based selection rules is proposed to improve the efficiency of search and avoid the violations.

The IP rules can be formulated as follows:

> 1) Between two feasible solutions, the one with a better interior penalty function value is preferred.2) If both solutions candidates are infeasible, the one with a smaller constraint violation is preferred.3) A feasible solution is always preferred to an infeasible one.

For the IP based selection rules, there are three important properties:

1) The interior penalty term is defined only if a solution is feasible. Thus, the interior penalty function cannot handle infeasible solutions. Therefore, a preference of feasible solutions to infeasible ones is used in rule (3). In this way, feasible solutions and infeasible solutions can be evaluated by different methods.

2) In the interior penalty method, feasible solutions are penalized in order to avoid the boundaries of feasible region. Rule 1) penalizes feasible solutions approaching to the boundaries of feasible region. In this way, the search process of EAs will avoid the boundaries of feasible region.

3) The objective function is completely disregarded in rule 2). Therefore, the entire search effort is directed toward finding a feasible solution. This rule is especially suitable for highly constrained problems wherein finding a feasible solution may be extremely difficult [3]. In this way, the penalty rules can be applied to highly constrained problems without initial feasible solutions.

3 Self-adapt Interior Penalty

3.1 Form of Interior Penalty Function

The logarithmic penalty function is used as the interior penalty function in this paper since it has a superlinear convergence in the traditional mathematical programming [7]. The penalty term $B(\boldsymbol{x})$ in Eq.(2) can be formulated as follows:

$$B(x) = -\sum_{i=1}^{m} \ln\left(-v_i(x)\right) , \tag{4}$$

where $v_i(\boldsymbol{x})$ is the normalized constraint value, defined as:

$$v_i(x) = \frac{g_i(x)}{|\min g_i(x)|} \quad . \tag{5}$$

The scaling factor $|\min g_i(\boldsymbol{x})|$ for each constraint is taken as the minimal value of constraint value $g_i(\boldsymbol{x})$ in the search process.

3.2 Self-adapt Penalty Factor

The candidate solutions are selected based on interior penalty function value in the IP based selection rules. Therefore, the optimal solution must be the one with minimum internal penalty value of all the candidates. According this rule, we proposed a self-adapt interior penalty method as follows.

Given the constrained optimization problem (1), the penalty factor r must make the penalty value of the optimal solution is less than the one of any other solution.

The Penalty factor selection rules can be expressed as the following formula:

$$f(x^*) + r(t)B(x^*) \le f(x_i) + r(t)B(x_i) \,, \tag{6}$$

where x_i is any candidate, x^* is the optimal solution.

The Eq.(6) can be converted to:

$$(B(x^*) - B(x_i))r(t) \le f(x_i) - f(x^*) \;. \tag{7}$$

Considering three conditions as follows:

(1) $B(x^*)$-$B(x_i)$=0

The left of Eq.(7) is 0 in this condition. Therefore, for any $r(t)$, the formula $f(x_i) \ge f(x^*)$is established, i.e. Eq. (7) is established. This condition is not considered. This condition is not considered.

(2) $B(x^*)$-$B(x_i)$<0

The Eq.(7) can be convert to:

$$r(t) \ge \frac{f(x^*) - f(x_i)}{B(x_i) - B(x^*)} \,. \tag{8}$$

The right of Eq.(8) is less than or equal to 0 since $f(x_i) \ge f(x^*)$. Therefore, for any $r(t) \ge 0$, Eq.(8) is established. This condition is also not considered.

(3) $B(x^*)$-$B(x_i)$>0

We can get the upper bounder of $r(t)$ by Eq.(7) in this condition:

$$r(t) \le \frac{f(x^*) - f(x_i)}{B(x_i) - B(x^*)} \,. \tag{9}$$

We can get the follows since the optimal solution x^* satisfies all the constraints, i.e. $\forall j,\ g_j(x^*) \le 0$:

$B(x^*)=-\sum_{j=1}^{q}\ln|g_j(x^*)-\varepsilon_j|=-\sum_{j=1}^{q}\ln(\varepsilon_j-g_j(x^*))\le-\sum_{j=1}^{q}\ln(\varepsilon_j)$ Let $B_d=-\sum_{j=1}^{q}\ln(\varepsilon_j)$, then:

$$B_d-B(x_i)\ge B(x^*)-B(x_i)>0\quad . \tag{10}$$

Alternative $B(x^*)$ with B_d in Eq.(10). We can get an upper bounder of $r(t)$:

$$r(t)\le\min_{i\in N}\frac{f(x_i)-f(x^*)}{B_d-B(x_i)}\quad . \tag{11}$$

Further, the optimal solution can't be obtained before the original constrained optimization problem solved. However, we can use the best feasible solution in the current population instead of the optimal solution in the search process, i.e.

$$r(t)\le\min_{i\in N}\frac{f(x_i)-f(\overline{x}^*)}{B_d-B(x_i)}\quad , \tag{12}$$

where x^* is the feasible solution with minimum objective in the current population. $r(t)\le\min_{i\in N}\frac{f(x_i)-f(\overline{x}^*)}{B_d-B(x_i)}\le\min_{i\in N}\frac{f(x_i)-f(x^*)}{B_d-B(x_i)}$ is established since $f(\overline{x}^*)\le f(x^*)$.

Therefore, the penalty factor obtained by Eq. (12) satisfies Eq.(6). We can get an appropriate penalty factor without the optimal solution.

According to the above analysis, we can obtain the penalty factor by Eq.(12).

There may not be a feasible solution when the algorithm starts. We use a larger penalty factor in the early search process to ensure evolutionary algorithm can quickly find a feasible solution.

4 Self-adaptive Interior Penalty Based Differential Evolution

To illustrate validity of the interior penalty based selection rules, we introduce them to Differential Evolution algorithm called a self-adaptive interior penalty based differential evolution algorithm. The DE algorithm proposed by Storn and Price [8] is a heuristic method for real parameter optimization problems.

Let $\boldsymbol{x}_t^i$ denote an individual in the population of the DE algorithm and NP the size of the population, where i indicates the index of the individual, j the index of the variable, and t the current generation. A new mutated individual $v^i_{j,t+1}$ is generated according to the following equation:

$$v^i_{j,t+1}=x^{d_3}_{j,t}+\eta(x^{d_1}_{j,t}-x^{d_2}_{j,t}), \tag{13}$$

where the random indexes d_1, d_2, $d_3\in[0, \text{NP}]$ are mutually different integers and also different from the running index i, and $\eta\in(0,\ 1]$ is called the scaling factor or the amplification factor.

According to Eq. (13), a crossover operator is used to generate the trial individual $u^i_{j,t+1}$ based on the original individual $x^{d_3}_{j,t}$ and the new individual $v^i_{j,t+1}$.

$$u_{j,t+1}^{i}=\begin{cases} v_{j,t+1}^{i}, & \text{if } \mathrm{Rand}[0,1)\le \mathrm{CR} \text{ or } j=\mathrm{randint}(1,D), \\ x_{j,t}^{i}, & \text{otherwise,} \end{cases} \tag{14}$$

where Rand[0, 1) is a function that returns a real number between 0 and 1, randint(min, max) is a function that returns an integer between min and max, $CR \in [0, 1]$ is a crossover factor. The probability of the mutated individuals being preserved in the next generation is determined by the crossover factor *CR*.

A selection operator is used to choose an individual for the next generation (t+1) according to the following rule:

$$x_{t+1}^{i}=\begin{cases} u_{t+1}^{i}, & \text{if } u_{t+1}^{i} \text{ is better than } x_{t}^{i}, \\ x_{t}^{i}, & \text{otherwise,} \end{cases} \tag{15}$$

where u^{i}_{t+1} and x_t^{i} are compared by the IP based selection.

In this way, an individual will replace the one with a lower IP with respect to it; an individual will replace the one with the same IP depending on different conditions, where an infeasible individual will replace the one with a larger violation of the maximal non-common satisfied constraint and a feasible individual will replace the one with a worse objective with respect to it, respectively.

The capability of finding the global minimum and a fast convergence speed of DE are both highly sensitive to the control parameters *CR* and η [9]. Therefore, a self-adaptive approach is developed to adjust these parameters based on the success rate φ_t, where φ_t is defined by the percentage of original individuals replaced by trial individuals in the population at every generation, through the following updating law:

$$\eta_{t}=\begin{cases} \eta_{l}+\mathrm{rand}_{1}\eta_{u}, & \phi_{t}\le 0.5, \\ \eta_{t-1}, & \text{otherwise,} \end{cases} \tag{16}$$

$$\mathrm{CR}_{t}=\begin{cases} \mathrm{rand}_{2}, & \phi_{t}\le 0.5, \\ \mathrm{CR}_{t-1}, & \text{otherwise,} \end{cases} \tag{17}$$

where η_t and CR_t are the scaling factor η and the crossover factor *CR* at generation *t*, respectively; rand_1 and rand_2 are uniformly distributed random numbers in [0, 1]; η_l=0.1, η_u=0.9. The updating of η_t and CR_t is conducted before the mutation is performed. Eqs. (14) and (15) ensure that $\eta_t \in [0.1, 1] \subset (0, 1]$, $\mathrm{CR}_t \in [0, 1]$, $\forall t$.

The pseudo code of the DE with the IP based selection rules is shown as follows, the rules keep the operators of DE algorithms unchanged.

```
Begin
t=0;
Create NP random solutions for the initial population;
Evaluate all individuals;
For t=1 to MAX_GENERATION Do
   For i=1 to NP Do
      Select randomly d1•d2•d3;
```

```
        If (Rand[0, 1]•CR or j=randint(1, D)) Then
            $u^{i}_{j,t+1} = v^{i}_{j,t+1}$;
        Else
            $u^{i}_{j,t+1} = x^{i}_{j,t}$;
        End If
    End For
    Compare $u^{i}_{j,t+1}$ and $x^{i}_{j,t}$ by the IP based selected rules;
    If $u^{i}_{t+1}$ is better than $x^{i}_{t}$ Then
        $x^{i}_{t+1} = u^{i}_{t+1}$;
    Else
        $x^{i}_{t+1} = x^{i}_{t}$;
    End If
    t=t+1;
    update interior penalty factior;
End For
End
```

5 Experiments and Results

We performed the self-adapt interior penalty based differential evolution algorithm (SIPDE) algorithm 30 independent runs for ten benchmark problem described in Runarsson and Yao [9]. Equality constraints were transformed into inequalities using a tolerance value of 0.0001. The parameters were set the same as those of Mezura-Montes *et al.* [10]: NP=60, MAX_ GENERATIONS=5800. The control parameters CR and η were adjusted using a self-adaptive method. We compared our approach against four state-of-the-art approaches: the stochastic ranking (SR) algorithm [9], the simple multimembered evolution strategy (SMES) algorithm [11], the adaptive tradeoff model evolution strategy (ATMES) algorithm [12], and the constraint handling differential evolution (CHDE) algorithm [10]. The best, mean, worst results, and the standard deviations obtained by each approach are shown in Table 1.

5.1 Statement of the Problem

As described in Table 1, our approach was able to find the global optimum in ten benchmark problems. For problems g01, g03, g04, g05, g06, g07, g08, g09 and g10, the optimal solutions were consistently found in all 30 runs. For problems g02, the optimal solutions were not consistently found since the ratio of the size of the feasible region to the size of the search space was very large. Furthermore, feasible solutions were continuously found for all the benchmark problems in 30 runs. These results reveal that SIPDE has the substantial capability to deal with various kinds of COPs.

Table 1. Comparison of the best, the mean, the worst solutions, and the standard deviations found by our SIPDE against SR, SMES, ATMES, and CHDE

Prob	optimal	stat	methods				
			SR	SMES	ATMES	CHDE	SIPDE
g01		best	**-15.000**	**-15.000**	**-15.000**	**-15.000**	**-15.000**
	-15.000	mean	**-15.000**	**-15.000**	**-15.000**	-14.792	**-15.000**
		worst	**-15.000**	**-15.000**	**-15.000**	-12.743	**-15.000**
g02		best	-0.803515	-0.803601	-0.803388	**-0.803619**	**-0.803619**
	-0.803619	mean	-0.781975	-0.785238	-0.790148	-0.746236	-0.801758
		worst	-0.726288	-0.751322	-0.756986	-0.302179	-0.780843
g03		best	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	1.000	mean	**1.000**	**1.000**	**1.000**	0.640326	**1.000**
		worst	**1.000**	**1.000**	**1.000**	0.029601	**1.000**
g04		best	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**
	-30665.539	mean	**-30665.539**	**-30665.539**	**-30665.539**	-30592.154	**-30665.539**
		worst	**-30665.539**	**-30665.539**	**-30665.539**	-29986.214	**-30665.539**
g05		best	**5126.497**	5126.599	**5126.498**	**5126.497**	**5126.498**
	5126.497	mean	5128.881	5174.492	5127.648	5218.729	**5126.498**
		worst	5142.472	5304.167	5135.256	5502.410	**5126.498**
g06		best	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**
	-6961.814	mean	-6875.940	-6961.284	**-6961.814**	-6367.575	**-6961.814**
		worst	-6350.262	-6952.482	**-6961.814**	-2236.950	**-6961.814**
g07		best	24.307	24.327	24.306	24.306	**24.306**
	24.306	mean	24.374	24.475	24.316	104.599	**24.306**
		worst	24.642	24.843	24.359	1120.541	**24.306**
g08		best	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
	-0.095825	mean	**-0.095825**	**-0.095825**	**-0.095825**	-0.091292	**-0.095825**
		worst	**-0.095825**	**-0.095825**	**-0.095825**	-0.027188	**-0.095825**
g09		best	**680.630**	680.632	**680.630**	**680.630**	**680.630**
	680.630	mean	680.656	680.643	680.639	692.472	**680.630**
		worst	680.763	680.719	680.673	839.783	**680.630**
g10		best	7054.316	7051.903	7052.253	**7049.248**	**7049.248**
	7049.248	mean	7559.192	7253.047	7250.437	8442.657	**7049.255**
		worst	8835.655	7638.366	7560.224	15580.370	**7049.399**

5.2 Comparison with Four State-of-the-Art Approaches

The performance of SIPDE was compared in detail with four state-of-the-art techniques using the selected performance metrics (Table 1). For benchmark problems g01, g03, g04, and g08, RFDDE, SR, SMES, and ATMES consistently found the optimal solutions in all 30 runs. For problem g06, the optimal solutions were consistently found by SIPDE and ATMES in all 30 runs. For problem g05, SR, SMES, and ATMES found better ‘mean’ and ‘worst’ results than SIPDE. However, SIPDE was also able to find the optimal solution in 30 runs and the ‘mean’ results were very close to the optimal solution. For all the other 4 problems, SIPDE found better ‘best’, ‘mean’, and ‘worst’ results than SR, SMES, and ATMES. As against CHDE, our approach found “similar” best results in all the problems, and furthermore located better ‘mean’ and ‘worst’ results in all the problems.

In summary, we can conclude that SIPDE outperforms or has similar performances to SR, SMES, ATMES, and CHDE in all the problems.

6 Conclusion

In order to combine constraints into the evaluation of feasible solutions, a set of interior penalty rules for handling COPs was proposed in this paper. In these rules, interior penalty functions are used to evaluate feasible solutions and constraint violations are used to evaluate infeasible solutions. Three elements are proposed to make these rules effective in an EA: (1) a logarithmic penalty function is used to make the algorithm convergence quickly; (2) the penalty factors are updated according to the type of constraints which determined by a Spearman's rank-order correlation coefficient; (3) the equalities are handled by an adaptive relax method. Furthermore, the interior penalty rules are implemented based on a DE, namely, SIPDE. Finally, the experiment results show that the proposed approach is competitive with four other state-of-the-art techniques.

References

1. Mezura-Montes, E., Coello, C.: Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm and Evolutionary Computation 1(4), 173–194 (2011)
2. Kramer, O.: A review of constraint-handling techniques for evolution strategies. Applied Computational Intelligence and Soft Computing 1, 1–11 (2010)
3. Tessema, B., Yen, G.: An adaptive penalty formulation for constrained evolutionary optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 39(3), 565–578 (2009)
4. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186(2-4), 311–338 (2000)
5. Mezura-Montes, E.: Constraint-handling in evolutionary optimization. Springer, Heidelberg (2009)
6. Wright, M.: The interior-point revolution in optimization: history, recent developments, and lasting consequences. Bulletin of the American Mathematical Society 42(1), 39–56 (2005)
7. Wright, M.: The interior-point revolution in constrained optimization. High-Performance Algorithms and Software in Nonlinear Optimization, 359–381 (1998)
8. Storn, R., Price, K.: Differential Evolution–A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization 11(4), 341–359 (1997)
9. Runarsson, T., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation 4(3), 284–294 (2000)

10. Mezura-Montes, E., Coello Coello, C.A., Tun-Morales, E.I.: Simple feasibility rules and differential evolution for constrained optimization. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972, pp. 707–716. Springer, Heidelberg (2004)
11. Mezura-Montes, E., Coello, C.: A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Transactions on Evolutionary Computation 9(1), 1–17 (2005)
12. Wang, Y., Zixing, C.: An Adaptive Tradeoff Model for Constrained Evolutionary Optimization. IEEE Transactions on Evolutionary Computation 12(1), 80–92 (2008)

A Novel Hybrid Algorithm for Mean-CVaR Portfolio Selection with Real-World Constraints

Quande Qin[1,2], Li Li[1], and Shi Cheng[3,4]

[1] Department of Management Science, Shenzhen University, Shenzhen, China
[2] Research Institute of Business Analytics & Supply Chain Management,
Shenzhen University, Shenzhen, China
[3] Division of Computer Science, University of Nottingham Ningbo, China
[4] International Doctoral Innovation Centre, University of Nottingham Ningbo, China
qinquande@gmail.com, llii318@163.com, shi.cheng@nottingham.edu.cn

Abstract. In this paper, we employ the Conditional Value at Risk (CVaR) to measure the portfolio risk, and propose a mean-CVaR portfolio selection model. In addition, some real-world constraints are considered. The constructed model is a non-linear discrete optimization problem and difficult to solve by the classic optimization techniques. A novel hybrid algorithm based particle swarm optimization (PSO) and artificial bee colony (ABC) is designed for this problem. The hybrid algorithm introduces the ABC operator into PSO. A numerical example is given to illustrate the modeling idea of the paper and the effectiveness of the proposed hybrid algorithm.

Keywords: Conditional Value at Risk, CVaR, Hybrid algorithm, Portfolio selection.

1 Introduction

Portfolio selection is concerned with the allocation of a limited capital to a combination of securities in order to trade off the conflicting objectives of high profit and low risk [13,17]. Since the introduction of mean-variance (MV) model developed by Markowitz, variance has become the most popular risk measure in portfolio selection. Variance considers high returns as equally undesirable as low returns because high returns will also contribute to the extreme of variance. Both theory and practice indicate the variance is not a good risk measure. Some alternative risk measures have been proposed [11,18]. Value at Risk (VaR) is widely used by financial institution. However, it has its limitations, such as it is not a coherent risk measure [1]. Rockafellar and Uryasev [15] proposed the Conditional Value at Risk (CVaR), which is the conditional expectation of losses above the VaR.

In practice, problem of portfolio selection has some real-world constraints, which exacerbates the complexity. For example, it assumes that there exists a perfect market with no tax or transaction cost. In the present study, we will consider transaction cost, and floor and ceiling constraints. In addition, the least

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 319–327, 2014.
© Springer International Publishing Switzerland 2014

unit of trading is 100 shares in stock market of China, and shares must be subscribed a round lot. The modeling of such constraints involves the introduction of integer variables. We employ CVaR to measure the risk of portfolio, and a Mean-CVaR (MC) portfolio selection model with real-world constraints is proposed. In view of the difficulty to solve this model using classical optimization techniques, a hybrid meta-heuristics algorithm based Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) is designed to handle this problem. The hybrid algorithm introduces the ABC operator into PSO. The added ABC operator is used to evolve personal experience of the particles. The hybrid approach elegantly combines the exploitation ability of PSO with the exploration ability of ABC.

The rest of the paper is organized as follows. Section 2 presents the backgrounds including PSO, ABC and CVaR. Section 3 the proposed MC portfolio selection model with real-world constraints. A hybrid algorithm based on PSO and ABC is provided in Section 4. In Section 5, a numerical example is given. The conclusions are drawn in Section 6.

2 Backgrounds

2.1 Particle Swarm Optimization

PSO was originally developed to emulate the flocking behavior of birds and fish schooling [5,9]. Each individual, called a particle, in the PSO population represents a potential solution of the optimization problem [2,19]. The population of PSO is referred to as a swarm, which consists of a number of particles. Particle i at iteration t is associated with a velocity vector $\boldsymbol{v}_i^t = [v_{i1}^t, v_{i2}^t, \cdots, v_{iD}^t]$ and a position vector $\boldsymbol{x}_i^t = [x_{i1}^t, x_{i2}^t, \cdots, x_{iD}^t]$ where $i \in \{1, 2, \cdots, NP\}$, NP is the population size. $x_{id} \in [l_d, u_d]$, $d \in \{1, 2, \cdots, D\}$, where D is the number of dimensions, and l_d and u_d are the lower and upper bounds of the dth dimension of search space, respectively. Each particle flies through space with a velocity. The new velocities and the positions of the particles for the next iterations are updated using the following two equations [3–5,9]:

$$v_{id}^{t+1} = wv_{id}^t + c_1 r_1 (pbest_{id}^t - x_{id}^t) + c_2 r_2 (gbest_d^t - x_{id}^t) \tag{1}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{2}$$

where w is the inertia weight; $\boldsymbol{pbest}_i = [pbest_{i1}, pbest_{i2}, \cdots, pbest_{iD}]$ is the best position has been found by particle i, $\boldsymbol{gbest}_i = [gbest_{i1}, gbest_{i2}, \cdots, gbest_{iD}]$ is the historically best position has been found by the whole swarm so far; c_1 and c_1 are acceleration coefficients. The inertia weight w is used to trade off the exploration and exploitation; r_1 and r_2 represent two independently random numbers uniformly distributed on $[0, 1]$.

2.2 Artificial Bee Colony

ABC algorithm was proposed by simulating waggle dance and intelligent foraging behaviors of honeybee colonies [7]. In the ABC algorithm, there are two components: the foraging artificial bees and the food source [8]. The position of the a food source, $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \cdots, x_{iD}]$, represents a possible solution and the nectar amount of a food source corresponds to the fitness of the associated solution. The colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts [14].

The ABC algorithm consists of four phases: initialization, employed bee, onlooker bee and scout bee. In the initialization phase of the ABC, SN food source positions are randomly produced with the search space. After producing food sources and assigning them to the employed bees. In the employed bee phase of ABC, each employed bee tries to find a better quality food source based on $\boldsymbol{x}_i$. The new food source, denoted as $\boldsymbol{u}_i = [u_{i1}, u_{i2}, \cdots, u_{iD}]$, is calculated from the equation below.

$$u_{ij} = x_{ij} + \phi(x_{ij} - x_{sj}) \tag{3}$$

where $i \in \{1, 2, \cdots, SN\}$, where SN denotes the number of food source; j is a randomly generated integer number in the range $[1, D]$, ϕ is a randomly number uniformly distributed in the range $[-1, 1]$, and s is the index of a randomly chosen solution. ABC changes each position in only one dimension at each iteration. The source position $\boldsymbol{x}_i$ in the employed bee's memory will be replaced by the new candidate food source position $\boldsymbol{u}_i$ if the new position has a better fitness value. Each onlooker bee chooses one of the proposed food sources depending on the probability value p_i associated with the fitness value, where

$$p_i = fit_i / \sum_{j=1}^{SN} fit_j \tag{4}$$

where fit_i is the fitness of the food source i. After the food source is selected, a new candidate food source can be expressed by Eq. (3). If a food source, $\boldsymbol{x}_i$, cannot be improved for a predetermined number of cycles, referred to as *limit*, this food source is abandoned. Then, the scout produces a new food source randomly to replace $\boldsymbol{x}_i$.

2.3 Conditional Value at Risk

Let $L(x, y)$ be the loss function with weight vector x and the return rate vector y. Let $p(r)$ be the density function of the return rate vector y. Then $L(x, y)$ is random variable dependent on x. The probability of $L(x, y)$ not exceeding a threshold α is given by

$$\psi(x, \alpha) = \int_{L(x,y) \leq \alpha} p(y) dy \tag{5}$$

The VaR of the loss associated with x and a specified probability level β in $(0, 1)$ is the value

$$VaR_\beta(x) = \min\{\alpha \in R^m : \psi(x, \alpha) \geq \beta\} \tag{6}$$

As an improved risk measure, CVaR, is the expected portfolio return, conditioned on the portfolio returns being lower than VaR. It is defined as the Eq. (7). Compared with VaR, CVaR has some superior mathematical properties.

$$\begin{aligned} CVaR_\beta(x) &= E[L(x,y)|L(x,y)) \geq VaR_\beta(x)] \\ &= (1-\beta)^{-1} \int_{L(x,y)\geq VaR_\beta(x)} L(x,y)p(y)dy \end{aligned} \tag{7}$$

CVaR can be obtained by the following equation based on reference [15]

$$F_\beta(x,\alpha) = \alpha + (1-\beta)^{-1} \int_{y\in R^M} [L(x,y)-\alpha]^+ p(y)dy \tag{8}$$

where $(a)^+$ is defined as $\max(a,0)$.

3 The Proposed Portfolio Selection Model

In this section, we discuss the MC portfolio selection model. Assume there n risky asset and one risk-free asset in a financial market for trading. An investor hopes to allocate his/her initial wealth m_0. For notational convenience, we first introduce the following notations:

- r_i: the return of risky asset i.
- r_f: the return of risk-free asset.
- t_i: the transaction cost of risky asset i;
- $s(x)$: the total return of the portfolio.
- p_i: the price of risky asset i each round lot;
- k_i: the round lot of risky asset i invested;
- σ_i: the highest limits on risky asset i;
- ε_i: the lowest limits on risky asset i;
- λ the acceptable return of the portfolio.

The capital invested in risk assets is $\sum_{i=1}^n k_i p_i$ and the remaining capital $m_0 - \sum_{i=1}^n k_i p_i$ invested in the risk-free asset. Obviously, it holds that$\sum_{i=1}^n k_i p_i \leq m_0$. The transaction cost are consider, and it denotes as $\sum_{i=1}^n t_i k_i$. Thus, the total return $s(x)$ of the portfolio can be described as follows:

$$\begin{aligned} s(x) &= \sum_{i=1}^n k_i p_i r_i + r_f(m_0 - \sum_{i=0}^n k_i p_i) - \sum_{i=1}^n t_i k_i \\ &= r_f m_0 + \sum_{i=1}^n [k_i p_i (r_i - r_f) - t_i k_i] \end{aligned} \tag{9}$$

The intention of the proposed model is to minimize the CVaR in the case of the return of the portfolio is equal or greater than λ.

$$\min \boldsymbol{z} = CVaR \tag{10}$$

$$s.t. \begin{cases} \varepsilon_i \leq x_i \leq \sigma_i & i = 1, 2, \cdots, n \\ s(x)/m_0 \geq \lambda & \\ \sum_{i=1}^n k_i p_i \leq m_0 & \\ k_i \geq 0, & \text{integer}, \ i = 1, 2, \cdots, n \end{cases}$$

where $x = (k_1 p_1/m_0, k_2 p_2/m_0, \cdots, k_n p_n/m_0)$ is the weight vector. In practice, asset i is chosen to be invested and the weight lies in $[\varepsilon_i, \sigma_i]$, where $0 \leq \varepsilon_i \leq \sigma_i \leq 1$. The first constraint is called floor and ceiling constraints. The second constraint is used to ensure the return of the portfolio.

4 A Hybrid Algorithm Based on PSO and ABC

Due to the simple concept and efficiency of converging to reasonable solution fast, PSO has been successfully applied to a wide range of real-world problems. Despite the competitive performance of PSO, researchers have noted a major problem associated with the PSO is its premature convergence when solving complex problems [12]. ABC algorithm is good at exploration but poor at exploitation [20]. From the analysis of the merits and demerits of PSO and ABC, it is intuitive that hybridizing the PSO and ABC is a potential way to design an effective algorithm.

Generally, the locality of personal best position in PSO algorithm is distant from the global optimum. Once the swarm aggregates to such position, little opportunity is afforded for the swarm to explore for other solution and find the global optimum. This leads to the swarm suffer from premature convergence easily, especially when solving complicated multimodal problems. Thus, the evolution of the personal experience will promote the exploration of the personal experience space, which could potentially enhance PSO's performance. ABC has better ability to explore, which is beneficial to global search, but poor ability of exploitation. In this paper, we utilize the ABC operator to evolve the personal best position when the personal best position stagnated. It is expected that the proposed hybrid algorithm, PSOABC, combines the merits of PSO and ABC, and have capabilities of escaping from local optima and converge fast.

In PSOABC algorithm, we use PSO in the main loop. When the fitness of $\boldsymbol{pbest}_i$, denoted as $fit(\boldsymbol{pbest}_i)$, has not improved within a predefined number of successive iterations, denoted as k, it is considered to be stagnated and trapped into local optima. The setting of k is set to 3 in this paper. We only use the employed bee operator in ABC algorithm to evolve $\boldsymbol{pbest}_i$ in this work. The pseudo-code of the PSOABC algorithm is described in Algorithm 1. When $\boldsymbol{pbest}_i$ stagnated, we can use the employed bees operator to evolve $\boldsymbol{pbest}_i$. The mathematical expressions of this ABC operator described as follows:

$$z_{ij} = pbest_{ij} + \phi(pbest_{ij} - pbest_{sj}) \tag{11}$$

where s are randomly selected integers from the index of all solution with $s \neq i$. j is a randomly selected dimension number. ϕ is a randomly number uniformly distributed within the interval $[-1, 1]$.

5 Numerical Example

The portfolio selection model constructed is a non-linear discrete optimization problem. The proposed hybrid algorithm based on PSO and ABC is suitable

Algorithm 1. The pseudo-code of PSOABC algorithm

```
Initialization: set up all parameters;
Set the maximum iteration number $FEs$; $t = 1$, $Stop = 0$;
Evaluate the fitness of the swarm and determine $\boldsymbol{pbest}_i$ and $\boldsymbol{gbest}$ ;
while the stopping criteria is not satisfied do
    for $i = 1 : NP$ do
        for $d = 1 : D$ do
            $v_{id}^{t+1} = wv_{id}^t + c_1r_1(pbest_{id}^t - x_{id}^t) + c_2r_2(gbest_d^t - x_{id}^t)$;
            $x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$;
        $i = i + 1$;
        Evaluate the fitness of the particle $i$; Update $\boldsymbol{pbest}_i$ and $\boldsymbol{gbest}$ ;
        if $fit(\boldsymbol{pbest}_i^t) - fit(\boldsymbol{pbest}_i^{t-1}) = 0$ then
            $Stop(i) = Stop(i) + 1$ ;
        else
            $Stop(i) = 0$;
    for $i = 1 : NP$ do
        if $Stop(i) \geq k$ then
            $z_{ij} = pbest_{ij} + \phi(pbest_{ij} - pbest_{sj})$;
        if $fit(\boldsymbol{z}_i < fit(\boldsymbol{pbest}_i)$ then
            $\boldsymbol{pbest}_i = \boldsymbol{z}_i$ ;
    $t = t + 1$
```

for real-valued problems. Kitayama *et al.* utilized penalty function approach handle the discrete decision variables [10]. In this approach, the discrete decision variables are handled as the continuous ones by penalizing at the intervals. The penalty function is given as the following the Eq. (12).

$$\phi(x) = \sum_{i=1}^{n} \frac{1}{2} \left[\sin \frac{2\pi\{x_{m+i}^c - 0.25(d_{i,j+1} + 3d_{i,j})\}}{d_{i,j+1} - d_{i,j}} + 1 \right] \tag{12}$$

where $d_{i,j}$ and $d_{i,j+1}$ represents the discrete decision variables. x_{m+i}^c is the continuous decision variables between $d_{i,j}$ and $d_{i,j+1}$.

We select 20 stocks from Chinese security market, as shown in Table 1. The symbol of $m(\%)$ in Table 1 denotes the expected return. The requirement of selecting the average yield is greater than 0. This paper selected raw data for the weekend's closing price.

Assuming the investor has 500 million investment funds. According to the tax and commission in Chinese securities market, the transaction cost rate is set to 0.4%. The minimum invest weigh of each stock is 0, and the maximum weight is 10%. The risk-free return rate is equal to 4.14% based on one-year deposit rate in China, and λ is 4.5%.

Experimental results among genetic algorithm (GA), PSO-w [16], basic ABC [6] and PSOABC are compared. For a fair comparison, the population size is

Table 1. Stocks selected and expected return rate

Ticker	m(%)	Ticker	m(%)
000002	0.45	600631	0.25
000039	0.46	600642	0.5
600058	0.77	600649	0.18
600098	0.63	600663	0.1
600100	0.12	600688	0.26
600115	0.35	600690	0.09
600183	0.4	600776	0.22
000541	0.26	600811	0.3
000581	0.53	600812	0.29
600600	0.37	600887	0.18

Table 2. Experimental results comparison

Algorithm	$\beta = 90\%$		$\beta = 95\%$		$\beta = 99\%$	
	Mean	SD	Mean	SD	Mean	SD
GA	0.0454	0.0019	0.0527	0.0064	0.0737	0.0042
PSO-w	0.0428	0.0014	0.0519	0.0044	0.0743	0.0057
ABC	0.0412	0.0015	0.0479	0.0027	0.0632	0.0024
PSOABC	**0.0336**	0.0009	**0.0343**	0.0016	**0.0443**	0.0013

set to 40 for all algorithms, the maximum iteration is 3500. The selection rate, crossover rate and mutation rate is set to 0.9, 0.7 and 0.03, respectively. Other parameter settings in each algorithm are used according to their original references. All algorithms run 30 times independently. The experimental results are shown in the Table 2. In Table 2, "Mean" indicate the mean values of CVaR, and "SD" stands for the standard deviation. From Table 2, it can be seen that PSOABC has a good performance and is a good alternative for the proposed portfolio selection model.

6 Conclusions

In this work, we proposed a MC portfolio selection model. In this model, the portfolio risk is measured by CVaR and some real-world constraints are added. Note that the round lot, which involves the introduction of integer variables, is considered. We have proposed a novel hybrid algorithm to solve the portfolio selection problem. The proposed algorithm introduces the ABC operator to PSO in order to balance exploration and exploitation. A penalty function is adopted to transform the discrete portfolio selection model into a continuous one. A numerical example is given to illustrate the modeling idea of the paper, and the experimental results show that the proposed hybrid algorithm outperforms is highly competitive for this portfolio problem.

Acknowledgment. This work is partially supported by Natural Science Foundation of China under grant NO.71240015, 60975080, 61273367, 51305216, Natural Science Foundation of Guangdong Province under grant No.S2011010001337, Foundation for Distinguished Young Talents in Higher Education of Guangdong, China, under grant 2012WYM_0116 and the MOE Youth Foundation Project of Humanities and Social Sciences at Universities in China under grant 13YJC630123, and Ningbo Science & Technology Bureau (Science and Technology Project No.2012B10055). This work was carried out at the International Doctoral Innovation Centre (IDIC). The authors acknowledge the financial support from Ningbo Education Bureau, Ningbo Science and Technology Bureau, China's MOST and The University of Nottingham.

References

1. Artzner, P., Delbaen, F., Eber, J.M., Heath, D.: Coherent measures of risk. Mathematical Finance 9(3), 203–228 (1999)
2. Chen, X., Li, Y.: A modified pso structure resulting in high exploration ability with convergence guaranteed. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 37(5), 1271–1289 (2007)
3. Cheng, S.: Population Diversity in Particle Swarm Optimization: Definition, Observation, Control, and Application. Ph.D. thesis, Department of Electrical Engineering and Electronics, University of Liverpool (2013)
4. Cheng, S., Shi, Y., Qin, Q.: Population diversity of particle swarm optimizer solving single and multi-objective problems. International Journal of Swarm Intelligence Research (IJSIR) 3(4), 23–60 (2012)
5. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43 (1995)
6. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Tech. rep., Erciyes University, Engineering Faculty, Computer Engineering Department (October 2005)
7. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 39(3), 459–471 (2007)
8. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing 8(1), 687–697 (2008)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks (ICNN), pp. 1942–1948 (1995)
10. Kitayama, S., Arakawa, M., Yamazaki, K.: Penalty function approach for the mixed discrete nonlinear problems by particle swarm optimization. Structural and Multidisciplinary Optimization 32(3), 191–202 (2006)
11. Konno, H., Yamazaki, H.: Mean-absolute deviation portfolio optimization model and its applications to tokyo stock market. Management Science 37(5), 519–531 (1991)
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. Markowitz, H.: Portfolio Selection. The Journal of Finance 7(1), 77–91 (1952)

14. Qin, Q., Cheng, S., Li, L., Shi, Y.: Artificial bee colony algorithm: A survey. CAAI Transactions on Intelligent Systems 9(2), 127–135 (2014)
15. Rockafellar, R.T., Uryasev, S.: Optimization of conditional value-at-risk. Journal of Risk 2(3), 21–41 (2000)
16. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of the 1998 Congress on Evolutionary Computation (CEC1998), pp. 69–73 (1998)
17. Yoshimoto, A.: The mean-variance approach to portfolio optimization subject to transaction costs. Journal of the Operations Research Society of Japan 39(1), 99–117 (1996)
18. Young, M.R.: A minimax portfolio selection rule with linear programming solution. Management Science 44(5), 673–683 (1998)
19. Zhang, G., Li, Y.: Orthogonal experimental design method used in particle swarm optimization for multimodal problems. In: The Sixth International Conference on Advanced Computational Intelligence (ICACI 2013), pp. 183–188 (October 2013)
20. Zhu, G., Kwong, S.: Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied Mathematics and Computation 217(7), 3166–3173 (2010)

A Modified Multi-Objective Optimization Based on Brain Storm Optimization Algorithm

Lixia Xie, Yali Wu

Xi'an University of Technology, Xi'an Shaanxi
710048, China

Abstract. In recent years, many evolutionary algorithms and population-based algorithms have been developed for solving multi-objective optimization problems. In this paper, A new Multi-objective optimization algorithm-Modified Multiobjective Brain Storm Optimization (MMBSO) algorithm is proposed. The clustering strategy acts directly in the objective space instead of in the solution space and suggests potential Pareto-dominance areas in the next iteration. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise (DBSCAN) clustering and Differential Evolution (DE) mutations are used to improve the performance of MBSO. A group of multi-objective problems with different characteristics were tested to validate the usefulness and effectiveness of the proposed algorithm. Experimental results show that MMBSO is a very promising algorithm for solving these tested multi-objective problems.

Keywords: Brain Storm Algorithm, Clustering Technique, Multi-objective Optimization, Pareto-dominance.

1 Introduction

Many real world problems are commonly looked at from a variety of perspectives, and therefore are represented as multiple objectives which usually conflict with each other. These problems are called Multi-objective problems, which have gained much attention in the study of sciences, economic, engineering, etc. The optimum solution for a multi-objective optimization problem is not unique but a set of candidate solutions. In the candidate solution set, no solution is better than any other one with regards to all objectives. This set is named as Pareto-optimal set, and the associated objective vectors form the trade-off surface, also called Pareto-front, in the objective space.

During the last decades, a number of evolutionary algorithms and population-based methods have been successfully used to solve multi-objective optimization problems. For example, there are Multiple Objective Genetic Algorithm (MOGA) [1], Nondominated Sorting Genetic Algorithm (NSGA, NSGA II)[2][3] , Strength Pareto Evolutionary Algorithm (SPEA, SPEA II) [4][5], Multi-objective Particle Swarm Optimization (MOPSO) [6], to name just a few. Most of the above algorithms can improve the convergence and distribution of the Pareto-front more or less.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 328–339, 2014.
© Springer International Publishing Switzerland 2014

Human beings, as one kind of social animals, are the most intelligent in the world. When we face a difficult problem which every single person cannot solve, group person, especially with different background, get together to brain storm, the problem can usually be solved with high probability. Being inspired by this human idea generation process, Shi [6] proposed a novel optimization algorithm - Brain Storm Optimization (BSO) algorithm. The simulation results on two single-objective benchmark functions validated the effectiveness and usefulness of the BSO to solve optimization problems. In [8], two novel component designs were proposed to modify the BSO algorithm and it has significantly enhanced the performance of BSO. In [9] and [10], a multi-objective optimization algorithm based on the brainstorming process was developed. Simulation results illustrated that it can be a good optimizer for solving multi-objective optimization problems.

In this paper, a modified Multi-objective BSO (MMBSO) algorithm with clustering strategy in the objective space is proposed to solve multi-objective optimization problems. Instead of action on the population and on the obtained Pareto front, in the MMBSO, the clustering strategy acts directly on the objective vectors in the objective space. Then this operation gives a feedback to the decision space to decide which candidate solution should survive. The novel using of the clustering technique, especially for the multi-objective optimization problems with high dimensional decision vectors could reduce computational burden. Clustering and mutation, the main operators of BSO were analyzed by using a Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise (DBSCAN) clustering and Differential Evolution (DE) mutation which is different from the previous operator. Then the different dimensions of bench functions that named ZDT [3] were tested. The simulation results showed that MMBSO would be a promising algorithm in solving multi-objective optimization problems.

The remaining paper is organized as follows. Section 2 briefly reviews the related works about the BSO and the MOP. In Section 3, the Modified Multi-objective BSO (MMBSO) is introduced and described in detail. Section 4 contains the simulation results and discussion. Finally, Section 5 provides the conclusions and some possible paths for future research.

2 Related Work

2.1 Multi-Objective Optimization Problem (MOP)

Without loss of generality, all of the multi-objective optimization problems can be formulated as minimization optimization problems. Let us consider a multi-objective optimization problem:

$$\text{Minimize } \mathbf{F}(\mathbf{X}) = (f_1(\mathbf{X}), f_2(\mathbf{X}), \cdots, f_M(\mathbf{X})) \quad (1)$$

Where $\mathbf{X} = (x_1, \cdots, x_D) \in \Re^D$ is called the decision vector in the D dimensional search space and $\mathbf{F} \in \Omega^M$ is the objective vector with M objectives in the M dimensional objective space. The basic concepts of a minimization MOP can be described in [11].Two goals of a multi-objective optimization are the convergence to the true Pareto-optimal set, and the maintenance of diversity of solutions in the Pareto front set. Many performance metrics have been suggested to measure the performance of multi-objective optimization algorithms. In this paper, we use the metric Υ and metric Δ , which were defined by Deb et al. in [3] , to measure the performance of the MBSO algorithm.

2.2 Brainstorm Optimization Algorithm

The BSO algorithm is designed based on the brainstorming process [12] . In the brainstorming process, the generation of the idea obeys the Osborn's original four rules [12] . The people in the brainstorming group will need to be open-minded as much as possible and therefore generate more diverse ideas. Any judgment or criticism must be held back until at least the end of one round of the brainstorming process, which means no idea will be ignored. The algorithm is described as follows. In the initialization, N potential individuals were randomly generated. During the evolutionary process, BSO generally uses the clustering technique, mutation operator and selection operator to create new ideas based on the current ideas, so as to improve the ideas generation by generation to approach the problem solution. In the clustering technique, BSO uses a k-means clustering [11] . In the mutation operator, BSO creates N new individuals one by one based on the current ideas. To create a new individual, BSO first determines whether to create the new individual based on one selected cluster or based on two selected clusters. After the cluster(s) have been selected, BSO then determines whether create the new idea based on the cluster center(s) or random idea(s) of the cluster(s). No matter to use the cluster center or to use random idea of the cluster, we can regard the selected based idea as $X_{selected}$ which can be expressed as $X_{selected} = (x^1_{selected}, x^2_{selected}, \cdots, x^d_{selected})$, then applying a mutation of the $X_{selected}$ to get the new idea X_{new} which can be expressed as $X_{new} = (x^1_{new}, x^2_{new}, \cdots, x^d_{new})$. After the new idea X_{new} has been created, BSO evaluates X_{new} and replaces $X_{selected}$ if X_{new} has a better fitness than $X_{selected}$. The procedure of the BSO algorithm is shown in [6].

In fact, there have been several recent proposals to extend BSO to handle optimization problems. For example, A Modified Brain Storm Optimization[8],Predator–Prey Brain Storm Optimization for DC Brushless Motor [15],Brain Storm Optimization [13],Solution Clustering Analysis in Brain Storm Optimization Algorithm [14], Brain Storm Optimization Algorithm for Multi-objective Optimization Problems[9], and Multi-objective Optimization Based on Brain Storm Optimization Algorithm(MBSO)[10]. In [10], The BSO is used to solve multi-objective problems in which the k-means cluster and Gaussian mutation was used. Besides, Cauchy mutation was also utilized.

3 Modified Multi-Objective Brain Storm Optimization Algorithm (MMBSO)

The paper makes improvements about clustering and mutation operations for the paper [10]. DBSCAN clustering and differential mutation was used to improve the original algorithm. Also a probability of generating a random individual is added to increase the diversity of algorithm.

3.1 Clustering Technique

In the Multi-objective Brain Storm Optimization Algorithm, the k-means cluster algorithm was used in the clustering technique, but the k-means cluster algorithm has two disadvantages: First, it must specify the cluster center in advance, thus it could have been the idea of the same class was assigned to a different class and then makes the clustering lost its original role in the algorithm. And the second is that it must determine the number of cluster, and the number of cluster is fixed which is not changed with the idea of change in each iteration. In MMBSO, the clustering technique is implemented by a clustering method based on density named DBSCAN(A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise)[19] .We can image that when the similar ideas beyond a certain number, they can be put together as a class, then the ideas generated in each iteration can be fully used and thus makes different number of clusters according to the different ideas in each iteration. There are a lot of density-based clustering algorithms currently, in this article, in order to make the algorithm simple we use the simplest and most classical density-based clustering algorithm which is named DBSCAN. The procedure of it is shown in [19].

3.2 Generation Progress

The generation progress of MMBSO contains the mutation and selection operator which can be referred as follows:

- Mutation Operator:

The mutation operator plays an important role in the generation progress. Gaussian and Cauchy mutation are typical mutation operators which were used in [10]. In solving some multimodal or multi-objectives optimization problems, Gaussian mutation as one of the main operators in the algorithm may lead to a result with a slow convergence to a good near-optimum. In MMBSO, Differential Mutation is used to improve the performance of the algorithm.

As to BSO, a new idea X_{new} is created by adding Gaussian random noise to a based idea $X_{selected}$. It also can be known that the noise is large in early evolutionary phase and gradually become smaller during the running by the control of the logarithmic sigmoid transfer function according to [8]. Such a time varying noise strategy is consistent with the commonly intuition that large noises are needed in the early phase for global search while small noises are needed in the late phase for local fine-tuning.

In this paper, we propose to use the Differential Mutation to produce the noise value. The Differential Mutation is based on such a consideration. In the human being's brainstorming process, we can image that at the beginning of the process, everyone's idea would be much different. When they create new ideas based on the current ideas, they should take the differences of the current ideas into consideration. For example, when creating a new idea X_{new} based on a current idea $X_{selected}$, two distinct random ideas X_a which can be expressed as $X_a = (x_a^1, x_a^2, \cdots, x_a^d)$ and X_b which can be expressed as $X_b = (x_b^1, x_b^2, \cdots, x_b^d)$ from all the current ideas are taken to represent the idea difference, and the X_{new} is created as:

$$x_{new}^d = x_{selected}^d + rand(0,1)_d \times (x_a^d - x_b^d) \quad (2)$$

Where $rand(0,1)_d$ is a random number between (0, 1).

Using Eq. (2) to create new ideas, there are two advantages. Firstly, the computational burden of (2) is much lighter than that of the mutation of BSO that involves logarithmic sigmoid transfer function, Gaussian distribution function, random function, addition, subtraction, multiplication, and division, while (2) involves random function, multiplication, and subtraction for making up the noise value. Secondly, Eq. (2) can match the search environment of the evolutionary process. Be consistent with the brainstorming process for human being in solving problem, the ideas are much different from each other in the beginning, therefore the term ($x_a^d - x_b^d$) in (2) is larger and the new created ideas can keep the diversity in the early phase. In the late phase of the brainstorming process, the people may reach a consensus and the idea difference may be smaller. In this condition, the term ($x_a^d - x_b^d$) in (2) is also smaller to help refine the ideas. Therefore, Eq. (2) may be good at balance the global search and local search abilities according to the search information during the evolutionary process.

- Selection Operator:

It is also quite important to decide whether any newly generated solution should survive to the next generation. The selection based on Pareto dominance is utilized in this paper.

3.3 Update the Pareto Set

The Pareto Set is updated by the new non-dominated solutions. In this step, each new non-dominated solution obtained in the current iteration will be compared with all members in the Pareto Set. If the size of the Pareto Set exceeds the maximum size limit, it is truncated using the diversity consideration. In this paper, the circular crowded sorting operator [18] is adopted to guide the points toward a uniformly spread-out Pareto-optimal front.

3.4 The Procedure of the MMBSO

The MMBSO contains clustering, mutation, selection, and updating the Pareto set, which have been described above. The whole procedure of the MMBSO is shown in Fig. 1.

In the process of MMBSO algorithm, population size N imitate the number of ideas generated during the course of each round; randomly selects an elite cluster imitate the process of generating new ideas which excited by a single thought. Randomly selects two clusters was imitated by the idea of two people from different clusters inspired by the process of generating new ideas. Select an individual from the archive set was used to keep a few good ideas which imitate to pick up several good ideas in the brain storm process. The clustering technique is used to classify the idea of people with different backgrounds and ideas. And in each cluster, the cluster center acts as a facilitator of the problem to solve problems with better idea. The mutation operator is used to generate new ideas on the basis for the existing ideas. Also different from MBSO, a probability of generating a random individual is added to increase the diversity of algorithm.

4 Experiments and Discussions

In this section the MMBSO will be tested. Without loss of generality, all the multi-objective optimization problems tested in this paper are minimization problems.

4.1 Test Problems

In order to evaluate the performance of MMBSO, the ZDT test functions [3] are used in this paper. The ZDT suite is comprised of six problems, each one presenting a specific characteristic that generally cause difficulties to major evolutionary optimization strategies [20]. The bi-objective test functions used to examine the effect of the introduced MMBSO are the ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6. Test problem ZDT1 and ZDT3 have convex Pareto fronts while other test problems have non-convex Pareto fronts; ZDT3 also possesses a disconnected Pareto front; the Pareto front of ZDT6 is non-uniformly spaced; and ZDT4 as a complex multimode problem is difficult to find the global Pareto front. The information of these test functions in detail can be seen in [3].

4.2 Parameter Settings

During the test, a lot of parameters are used to test the algorithm. Finally a set of parameter that is relatively good for these test functions is used. In all the simulation runs, the population size is set to be 200 and the maximum size of the Pareto set is fixed at 100. After conducting a series of experiments, the pre-determined probability values P1 is set to be 0.99, P2 and P3 are set to 0.8, P4 and P5 are set to 0.2. For the DBSCAN clustering, MinPts is set to 7 while ε is set according to [21]. All of the algorithms are implemented in MATLAB using a real-number representation for decision variables. For each experiment, 30 independent runs were conducted to collect statistical results. Each test problem will be run with different dimension, 5, 10, 20, and 30, respectively.

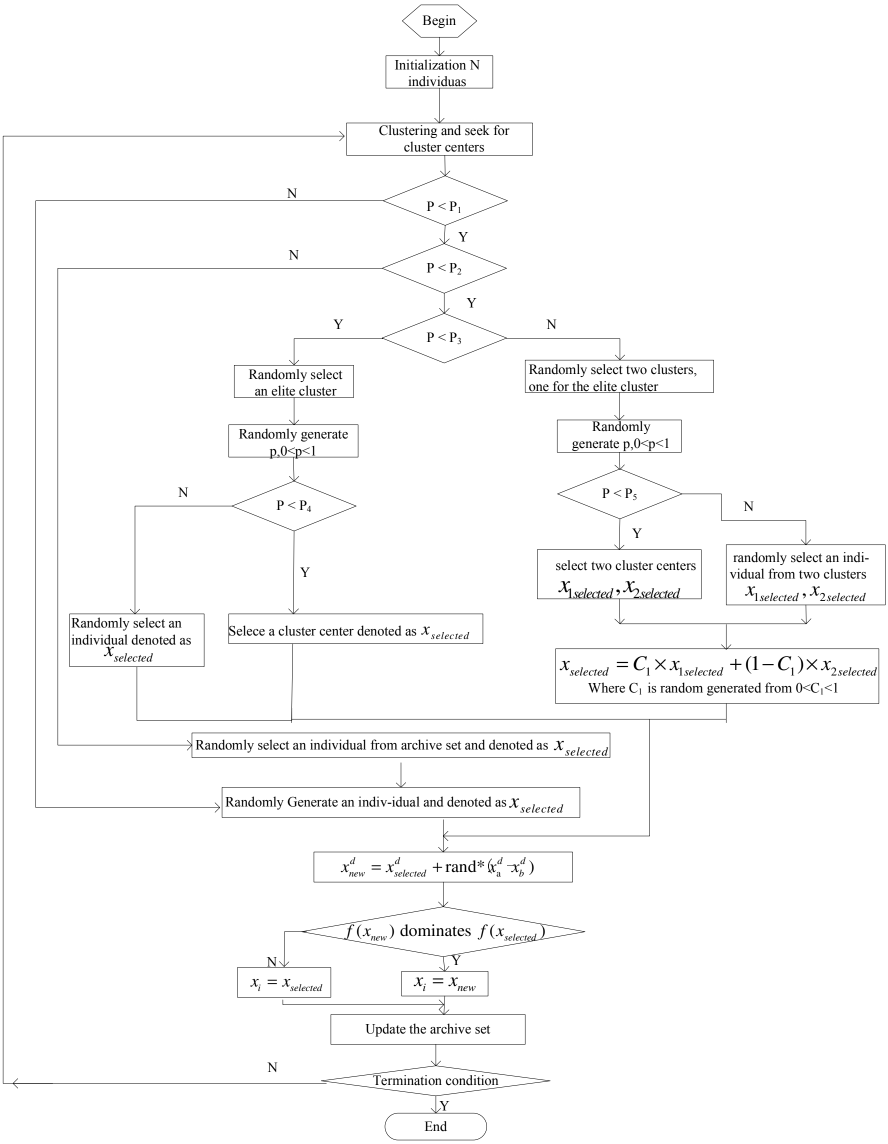

Begin
Initialization N individuas
Clustering and seek for cluster centers
$P < P_1$
N
Y
$P < P_2$
N
Y
$P < P_3$
Y
N
Randomly select an elite cluster
Randomly select two clusters, one for the elite cluster
Randomly generate p,0<p<1
Randomly generate p,0<p<1
$P < P_4$
N
Y
$P < P_5$
Y
N
select two cluster centers $x_{1selected}, x_{2selected}$
randomly select an individual from two clusters $x_{1selected}, x_{2selected}$
Randomly select an individual denoted as $x_{selected}$
Selece a cluster center denoted as $x_{selected}$
$x_{selected} = C_1 \times x_{1selected} + (1 - C_1) \times x_{2selected}$
Where C_1 is random generated from $0<C_1<1$
Randomly select an individual from archive set and denoted as $x_{selected}$
Randomly Generate an indiv-idual and denoted as $x_{selected}$
$x_{new}^d = x_{selected}^d + \text{rand}*(x_a^d - x_b^d)$
$f(x_{new})$ dominates $f(x_{selected})$
N
Y
$x_i = x_{selected}$
$x_i = x_{new}$
Update the archive set
Termination condition
N
Y
End

Fig. 1. The procedure of the MMBSO

4.3 Results

In all simulation runs, the metric Υ [3]and metric Δ [3]will be calculated and recorded for all the test problems. Table 1 compares the best and mean values of the convergence metric Υ obtained using MMBSO (denoted as DE), MBSO-G (MBSO with Gaussian mutation)[10] and MBSO-C (MBSO with Cauchy mutation)[10]. The diversity metric Δ about the test problems are listed in Table 2. The best results are marked with italics and bold.

Table 1. The comparisons of best and mean value of γ between MMBSO,MBSO-G and MBSO-C

Dimension	Algorithm	ZDT1		ZDT2		ZDT3		ZDT4		ZDT6	
		best	mean	best	mean	best	mean	best	mean	best	mean
5	DE	***0.0010***	***0.0011***	***0.6806e-003***	***0.7904e-003***	*0.0010*	*0.0012*	0.0062	0.1898	***0.0037***	***0.0040***
	G	0.0011	0.0016	0.0007	0.0008	0.0012	0.0015	***0.0009***	***0.0019***	0.0040	0.0048
	C	0.0011	0.0017	0.0009	0.0011	0.0012	0.0014	0.0010	0.0048	0.0039	0.0050
10	DE	***0.0009***	***0.0011***	***0.6803e-003***	***0.7888e-003***	***0.0010***	***0.0012***	0.0015	1.9661	***0.0036***	***0.0041***
	G	0.0050	0.0079	0.0014	0.0050	0.0024	0.0033	0.0029	2.4179	0.0046	0.0079
	C	0.0031	0.0062	0.0016	0.0029	0.0018	0.0027	***0.0011***	***0.1335***	0.0046	0.0072
20	DE	***0.0010***	***0.0011***	***0.7047e-003***	***0.8018e-003***	***0.0011***	***0.0012***	***0.0016***	6.7146	***0.0037***	***0.0040***
	G	0.0328	0.0472	0.0294	0.0498	0.0168	0.0248	2.8367	18.768	0.0323	0.0412
	C	0.0193	0.0312	0.0186	0.0311	0.0111	0.0150	1.4905	***4.4188***	0.0151	0.0241
30	DE	***0.0010***	***0.0011***	***0.0007***	***0.0008***	***0.0011***	***0.0012***	***2.9322***	***13.8379***	***0.0037***	***0.0040***
	G	0.1060	0.1347	0.1006	0.1308	0.0863	0.1136	10.3624	38.2581	0.0928	0.1385
	C	0.0695	0.0912	0.0725	0.0905	0.0443	0.0589	6.4966	15.2905	0.0580	0.0813

Table 2. The comparisons of best and mean value of Δ between MMBSO,MBSO-G and MBSO-C

Dimension	Algorithm	ZDT1		ZDT2		ZDT3		ZDT4		ZDT6	
		best	mean	best	mean	best	mean	best	mean	best	mean
5	DE	***0.1074***	***0.1266***	***0.1031***	***0.1228***	***0.4128***	***0.4180***	0.6160	1.0436	***0.5294***	***0.5422***
	G	0.3478	0.4187	0.3363	0.4016	0.5667	0.6013	0.3550	0.5058	0.6635	0.7021
	C	0.3325	0.4066	0.3422	0.4126	0.5016	0.5763	***0.3444***	***0.4786***	0.6627	0.7002
10	DE	***0.1087***	***0.1259***	***0.1031***	***0.1213***	***0.4113***	***0.4195***	***0.1413***	1.2555	***0.5325***	***0.5473***
	G	0.3765	0.4370	0.3475	0.4258	0.5420	0.6054	0.5159	0.8372	0.6751	0.7050
	C	0.3882	0.4542	0.3682	0.4115	0.5357	0.5931	0.4330	***0.7865***	0.6633	0.6989
20	DE	***0.1098***	***0.1275***	***0.0955***	***0.1208***	***0.4117***	***0.4173***	***0.1138***	1.2331	***0.5339***	***0.5443***
	G	0.4238	0.4789	0.3975	0.4869	0.5381	0.5868	0.9342	0.9748	0.6823	0.7103
	C	0.4215	0.4717	0.4371	0.4855	0.5073	0.5795	0.8660	***0.9569***	0.6724	0.7038
30	DE	***0.1008***	***0.1257***	***0.0997***	***0.1253***	***0.4126***	***0.4188***	1.2958	1.3968	***0.5322***	***0.5436***
	G	0.4823	0.5340	0.4834	0.5494	0.6026	0.6364	0.9619	0.9890	0.7080	0.7452
	C	0.5105	0.5529	0.4898	0.5588	0.5708	0.6293	***0.8362***	***0.9699***	0.6969	0.7425

In the Table 1 and 2, DE is represents MMBSO,G is MBSO-G and C is MBSO-C. The result from table 1 ~ 2 showed that MMBSO has better convergence and diversity than MBSO-G and MBSO-C except ZDT4. For the ZDT4, it is a complex multimode problem which is difficult to find the global Pareto front. DBSCAN which used in MMBSO uses the current ideas to get the clustering result, so it is limited the ZDT4 find the global optimum to some extent while k-means cluster can made a slightly good performance for ZDT4.

To further verify the performance of the algorithm, the hypervolume(HV)[22] which can evaluate convergence and diversity at the same time is added to illustrate the performance of the algorithm. The Table 3 compares the best and mean values of the ratio of HV that get by the solution set of algorithm and the corresponding true solution on the Pareto front. The ratio is between 0-1. If the ratio equals 1, then the solution obtained by the algorithm is on the true Pareto frontier. The more the ratio is close to 1, the solution obtained by algorithm is closer to the true Pareto frontier. The best results are marked with italics and bold.

Table 3. The comparisons of best and mean value of reaching the target HV value between MMBSO,MBSO-G and MBSO-C

Dime-nsion	Algo-rithm	ZDT1		ZDT2		ZDT3		ZDT4		ZDT6	
		best	mean	best	mean	best	mean	best	mean	best	mean
5	DE	0.9945	***0.9945***	0.9892	***0.9891***	***0.9975***	***0.9974***	***0.9988***	***0.9944***	***0.9945***	***0.9943***
	G	0.9908	0.9891	0.9811	0.9795	0.9958	0.9951	0.9952	*0.9915*	0.9938	0.9909
	C	***0.9976***	0.9910	***0.9986***	0.9874	1.5494	1.3569	0.9961	0.8582	0.9989	0.9939
10	DE	***0.9945***	***0.9944***	***0.9931***	***0.9890***	***0.9975***	***0.9974***	***0.9998***	***0.9962***	***0.9944***	***0.9938***
	G	0.9861	0.9831	0.9773	0.9700	0.9922	0.9904	0.9890	0.7215	0.9869	0.9825
	C	0.9939	0.9779	0.9924	0.9476	0.9939	0.9774	*0.9797*	*0.7009*	0.9942	0.9802
20	DE	***0.9950***	***0.9944***	***0.9956***	***0.9888***	***0.9975***	***0.9973***	***0.9999***	***0.9983***	***0.9942***	***0.9933***
	G	0.9729	0.9579	0.9493	0.9146	0.9706	0.9553	0.7565	0.4163	0.9786	0.9518
	C	0.9720	0.8955	0.9450	0.8078	1.1744	1.1016	*0.9775*	*0.7821*	0.9577	0.9130
30	DE	***0.9945***	***0.9943***	***0.9891***	***0.9889***	***0.9974***	***0.9973***	***0.9991***	***0.9989***	***0.9943***	***0.9930***
	G	0.9393	0.9069	0.8898	0.8203	0.9333	0.8824	0.7124	0.3065	0.9455	0.8990
	C	0.9298	0.7910	0.8709	0.6943	0.9847	0.9403	0.9732	0.7143	0.8725	0.7973

In the Table 3, DE is represents MMBSO,G is MBSO-G and C is MBSO-C.As can be seen from Table 3, the results obtained by MMBSO are better than MBSO-G and MBSO-C for the dimensions 5, 10, 20 and 30. Most of the obtained solution reached by MMBSO is more than 0.99, which illustrates the algorithm has good performance.

The Fig. 2-5 shows the simulation result. In the figure, the fine line represents the true Pareto edge and the point represents the solution get by the algorithm.

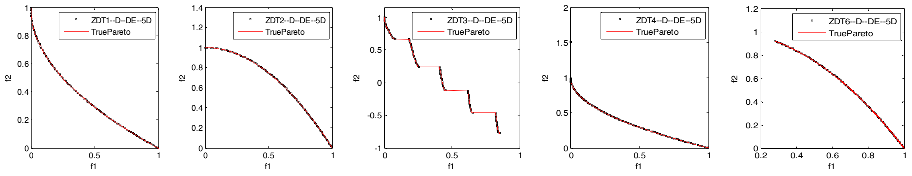


Fig. 2. Pareto-optimal front of ZDT1, 2, 3, 4 and 6 obtained by the MMBSO (5 dimension)

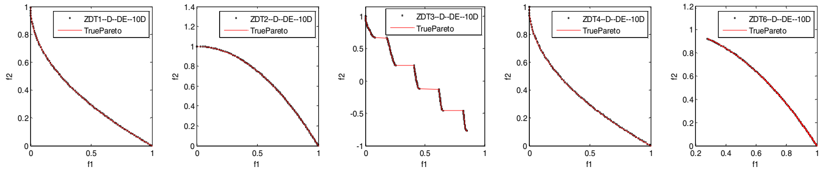


Fig. 3. Pareto-optimal front of ZDT1, 2, 3, 4 and 6 obtained by the MMBSO (10 dimension)

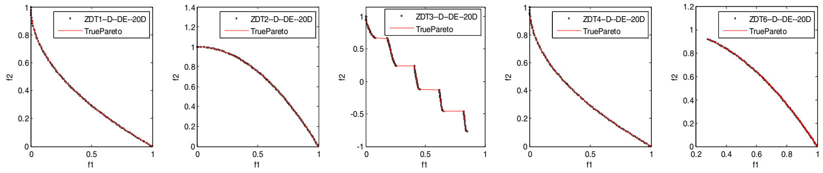


Fig. 4. Pareto-optimal front of ZDT1, 2, 3, 4 and 6 obtained by the MMBSO (20 dimension)

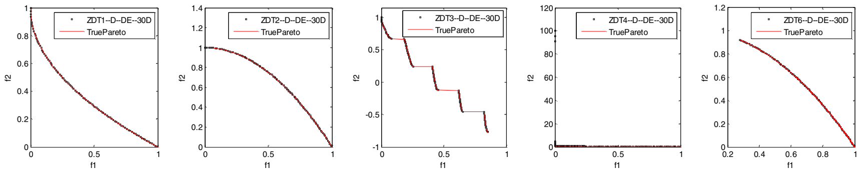


Fig. 5. Pareto-optimal front of ZDT1, 2, 3, 4 and 6 obtained by the MMBSO (30 dimension)

The result from table 1~2, also with the Fig. 2-6 show that MMBSO does better than MBSO on all functions except some result of ZDT4. This may be due to that DBSCAN clustering can make good use of the ideas generated in each iteration and get a reasonable clustering result according to the current idea, thus get a good current optimal solution. In addition, Differential Evolution mutation can match the search environment to provide suitable noise to create better ideas around the global optimal region to refine the solution for high accuracy. In MBSO, the new created idea was disturbed based on the current idea by Gaussian noise. However, this noise may be coarse. In the contrast, MMBSO uses the difference between two ideas as the disturbed noise. This way, the disturbed noise can be within a comparable order of magnitude with the current ideas. With the combination of DBSCAN clustering and Differential Evolution mutation, MMBSO get a good convergence and the diversity.

5 Conclusions and Discussion

In this paper, a modified multi-objective brain storm optimization algorithms, called MMBSO, was introduced. In the MMBSO, A clustering operator named DBSCAN has been utilized to cluster the different individuals and a different mutation operator is utilized to generate new individuals in the generation progress of the algorithm.

The results of the MMBSO have been evaluated according to three performance measures. From the simulation results, it was observed that MMBSO can be a good optimizer for solving multi-objective optimization problems. Although MMBSO performs better than the MBSO on most test problems in this study, more problems need to be tested to fully confirm this observation.

The results in Table 1 and 2 showed that MMBSO has a bad preference in solving ZDT4. In order to improve the shortcoming of the MMBSO, adaptive and mixing mutations based on niching techniques should be investigated. Another interesting area is to exploit the MMBSO for solving multi-objective optimization problems with constraints , many-objective problems and the other new test functions.

References

1. Fonseca, C.M., Fleming, P.J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Proceedings of the Fifth International Conference on Genetic Algorithms. University of Illinois at Urbana-Champaign, pp. 416–423. Morgan Kaufmann Publishers (1993)
2. Srinivas, N.: Deb. K.: Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation 2(3), 221–248 (1994)
3. 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)
4. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)
5. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, pp. 95–100 (2001)
6. Coello, C.A.C., Pulido, G., Lechuga, M.: Handling multi-objective with particle swarm optimization. IEEE Transactions on Evolutionary Computation 8(3), 256–279 (2004)
7. Shi, Y.: Brain Storm Optimization Algorithm. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) ICSI 2011, Part I. LNCS, vol. 6728, pp. 303–309. Springer, Heidelberg (2011)
8. Zhan, Z., Zhang, J., Shi, Y., Liu, H.: A Modified Brain Storm Optiization. In: IEEE World Congress on Computational Intelligence, pp. 10–15 (2012)
9. Xue, J., Wu, Y., Shi, Y., Cheng, S.: Brain Storm Optimization Algorithm for Multi-objective Optimization Problems. In: Tan, Y., Shi, Y., Ji, Z. (eds.) ICSI 2012, Part I. LNCS, vol. 7331, pp. 513–519. Springer, Heidelberg (2012)
10. Shi, Y., Xue, J., Wu, Y.: Multi-objective Optimization Based on Brain Storm Optimization Algorithm. Journal of Swarm Intelligence Research (IJSIR) 4(3) (2013)

11. Coello, C.A.C., Becerra, R.L.: Evolutionary Multiobjective Optimization using a Cultural Algorithm. In: Proceedings of IEEE Swarm Intelligence Symposium (SIS 2003), pp. 6–13 (2003)
12. Smith, R.: The 7 Levels of Change, 2nd edn. Tapeslry Press (2002)
13. Zhan, Z., Chen, W., Lin, Y., Gong, Y., Li, Y., Zhang, J.: Parameter Investigation in Brain Storm Optimization. In: IEEE Symposium on Swarm Intelligence (SIS), pp. 103–110 (2013)
14. Cheng, S., Shi, Y., Qin, Q., Gao, S.: Solution Clustering Analysis in Brain Storm Optimization Algorithm. In: IEEE Symposium on Swarm Intelligence (SIS), pp. 111–118 (2013)
15. Duan, H., Li, S., Shi, Y.: Predator–Prey Brain Storm Optimization for DC Brushless Motor. IEEE Transactions on Magnetics 49(10), 5336–5340 (2013)
16. Xu, D., Wunsch II, D.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks 16(3), 645–677 (2005)
17. Jain, A.K.: Data clustering: 50 years beyond K-means. Journal of Pattern Recognition Letters 31, 651–666 (2010)
18. Luo, C., Chen, M., Zhang, C.: Improved NSGA-II algorithm with circular crowded sorting. Control and Decision 25(2), 227–232 (2010)
19. Ester, M., Kriegel, H., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, KDD 1996 (1996)
20. Adra, S.F., Dodd, T.J., Griffin, I.A., Fleming, P.J.: Convergence Acceleration Operator for Multiobjective Optimization. IEEE Transactions on Evolutionary Computation 13(4), 825–847 (2009)
21. Daszykowski, M., Walczak, B., Massart, D.L.: Looking for Natural Patterns in Data. Part 1: Density Based Approach, Chemmon Intell. Lab. Syst. 56, 83–92 (2001)
22. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)

Modified Brain Storm Optimization Algorithm for Multimodal Optimization

Xiaoping Guo, Yali Wu, and Lixia Xie

Xi'an University of Technology, Xi'an Shaanxi, China
710048

Abstract. Multimodal optimization is one of the most challenging tasks for optimization. The difference between multimodal optimization and single objective optimization problem is that the former needs to find both multiple global and local optima at the same time. A novel swarm intelligent method, Self-adaptive Brain Storm Optimization (SBSO) algorithm, is proposed to solve multimodal optimization problems in this paper. In order to obtain potential multiple global and local optima, a max-fitness grouping cluster method is used to divide the ideas into different sub-groups. And different sub-groups can help to find the different optima during the search process. Moreover, the self-adaptive parameter control is applied to adjust the exploration and exploitation of the proposed algorithm. Several multimodal benchmark functions are used to evaluate the effectiveness and efficiency. Compared with the other competing algorithms reported in the literature, the new algorithm can provide better solutions and show good performance.

Keywords: Brain Storm Algorithm , Max-fitness Grouping Cluster, Self-adaptive Parameter Control, Multimodal Optimization.

1 Introduction

Multimodal optimization is a most challenging task in the area of optimization. Unlike the single objective optimization problems, multimodal optimization needs to provide multiple optimal solutions simultaneously. Since in the practical optimization problems, it is very common that the best solution cannot be realized at times due to the physical constraints. If multiple satisfaction solutions are known, the implementation can be quickly switched to an alternative solution while the system performance is still maintaining. As the name suggests, multimodal optimization requires optimization algorithms to find multiple optimal solutions (both local and global) and not just one single optimum as is done in a typical optimization study.

Evolutionary algorithms (EAs), due to their population-based approach, provide a population of possible solutions processed at every iteration. If multiple solutions can be preserved over all these iterations, we can have multiple good solutions at termination of the algorithm. Although evolutionary algorithm shows the potential power on multimodal optimization, two difficulties may be faced in the multimodal optimization algorithm. The first is how to locate multiple global and local optima

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 340–351, 2014.
© Springer International Publishing Switzerland 2014

during the evolutionary process. And the second is that how to maintain the identified optima until the end of the search.

With the development of computer science and technology, numerous techniques have been developed for locating multiple optima (global or local). Multipopulation based method can be incorporated into a standard EA to promote and maintain formation of multiple stable subpopulations within a single population to locate multiple optimal or suboptimal solutions. One of the multipopulation techniques is commonly referred as "niching" methods. R. Thomsen [1] uses crowding metric to force new individuals entering a population to replace similar individuals. But the algorithm suffers from higher computational complexity. And the performance relies on prior knowledge of some niching parameters. Fitness Euclidean distance ratio PSO (FERPSO) [2] and speciation-based PSO (SPSO) [3] are two commonly effective niching PSO algorithms. It is designed only for locating all global optima, while ignoring local optima. In the literature [4], Thomsen proposed a Crowding DE (CDE) to solve multimodal problems. It is generally difficult to select suitable trial vector generation strategies and control parameters for CDE that can generate satisfactory performance over all test functions. Most of existing niching methods have difficulties to be overcome before they can be applied successfully to real-world multimodal problems [5]. In recent years, the clustering technique, as another multipopulation method, is used successfully to solve multi-modal optimization by more and more scholars. Yin and Germay [8] proposed an Adaptive Clustering algorithm (ACA) to avoid the a priori estimation of σ_{share} . ACA adopts the identified cluster instead of sharing the fitness function, but this method introduces two additional variables at the same time, which need to set a reasonable maximum and minimum value as the radius of the cluster. Hanagandi and Nikolaou [10] also use the clustering method to find the global optima in a genetic search framework. However, the main difficulty of all the methods lies in how to define the area for each sub-region in the search space and how to determine the number of sub-populations.

Brain Storm Optimization (BSO) algorithm, inspired by human idea generation process, is proposed by Shi in [11] to solve single objective optimization problem. But two features make it perform well in multimodal problems. One is the clustering operator that divides all the ideas generated in the current generation into some different groups, which is possible to maintain multi optimal solutions. And the other is the creating operator that creates new idea by learning from the self-group or other group, which can maintain the diversity of each group. While the classical clustering method such as k-means cannot solve the multimodal problem well. So in this paper, a new clustering method named Max-fitness Clustering Method (MCM) is cooperated with BSO for multimodal problem. And the self-adaptive parameter control is used for the creating operator. The clustering method enables the algorithm to assign individuals to different promising sub-regions. And the self-adaptive parameter control methods are effective in maintaining the diversity of the population.

The rest of the paper is organized as follows. Section II briefly reviews the related works about BSO. The improved algorithm is described in detail in section III. The parameter setting and results are given in Section IV. And the conclusion and further research are detailed finally in Section V.

2 Brain Strom Optimization

In any swarm intelligence algorithm, each individual cooperatively and collectively move toward the better and better areas in the solution space. When Human being face a difficult problem which every single person cannot solve, group person, especially with different background, get together to brain storm, the problem can usually be solved with high probability. Being inspired by this human idea generation process, in 2011, Shi proposed a novel algorithm named Brain Storm Optimization(BSO). In the brainstorming process, the generation of the idea obeys the Osborn's original four rules (Smith, 2002). The people in the brainstorming group will need to be open-minded as much as possible and therefore generate more diverse ideas. Any judgment or criticism must be held back until at least the end of one round of the brainstorming process, which means no idea will be ignored. The algorithm is given in Fig.1 and is described as follows:

In the initialization, N potential ideas were randomly generated. BSO uses a k-means clustering [12] as its clustering technique. In the selection operator, BSO creates N new ideas one by one based on the current ideas. To create a new idea, BSO first determines whether to create the new idea based on one selected cluster or based on two selected clusters. After the cluster(s) have been selected, BSO then determines the selected ideas whether create the new idea based on the cluster center(s) or random idea(s) of the cluster(s).

```
Algorithm BSO
Begin
Randomly generate $N$ ideas ($X_i, 1 \le i \le N$) and evaluate their fitness;
While (Not stop) Do
  Cluster the $N$ ideas into $M$ clusters; //According to the positions
  Record the best idea in each cluster as the cluster center,
   //Probability of replacing a randomly selected cluster center,0.2
  If(random(0,1)< $p\_replace$)
    Randomly selected a cluster and replace the cluster center
    with a randomly generated idea;
  End of If
  For($i$=1 to $N$)
     //Probability of generating new idea based on one cluster ,0.8
    If(random(0,1)< $p\_one$)
      Randomly select a cluster $j$ with a probability $p_j$;
      //Probability of using the cluster center, $p\_one\_center$=0.4
      If (random(0,1)< $p\_one\_center$)
```

```
          Add random values to the selected cluster center
          to generate a new idea $Y_i$;
        Else
          Add random values to a random idea of the selected cluster
          to generate a new idea $Y_i$;
        End of If
      Else//Generating new idea based on two clusters
        Randomly select two clusters $j_1$ and $j_2$;
        //Probability of using the cluster center, $p\_two\_center=0.5$
        If (random(0,1)< $p\_two\_center$)
          Combine the two selected cluster centers and add with
          random values to generate a new idea $Y_i$;
        Else
          Combine two random ideas from the two selected clusters
          and add with random values to generate a new idea $Y_i$;
        End of If
      End of If
      Evaluate the idea $Y_i$ and replace $X_i$ if $Y_i$ has better fitness than $X_i$;
    End of For
  End of While
End
```

Fig. 1. Pseudo-code of the BSO algorithm

If the selected idea $X_{selected}=(x^1_{selected},x^2_{selected},...,x^d_{selected})$ is gotten, a mutation of the $X_{selected}$ is applied to get the new idea expressed as $X_{new}=(x^1_{new},x^2_{new},...,x^d_{new})$. After the new idea X_{new} has been created, BSO evaluates X_{new} and replaces $X_{selected}$ if X_{new} has a better fitness than $X_{selected}$.

In the mutation process, the Gaussian mutation will be used as random values which are added to generate new ideas; it can be represented as follows:

$$x^d_{new}=x^d_{selected}+\xi * N(\mu,\sigma) \tag{1}$$

$$\xi=\mathrm{logsig}((0.5 * max\_iternation - current\_iteration)/K) * \mathrm{rand}() \tag{2}$$

In the equation (1) and (2), $x^d_{selected}$ is the d-dimensional of the idea selected to generate new idea; x^d_{new} is the d-dimensional of the idea newly generated; $N(\mu,\sigma)$ is the Gaussian random function with mean μ and σ; ξ is a coefficient that weights the contribution of the Gaussian mutation; $\log sig()$ is a logarithmic sigmoid transfer function; $max\_iternation$ and $current\_iteration$ are the maximum iteration number and the current iteration number, K is for changing $\log sig()$ function's slope, and rand() is a random value within (0,1).

3 Self-adapted Brain Strom Optimization

In the multimodal optimization problem, the challenging issue is how to find all the global optimum, local optimum while maintaining the diversity of the population. The experimental results reported in [6]-[10] show that clustering operation is an ideal technique. Although many common evolutionary algorithms (EAs) are used to solve multimodal problems, most of them introduce a variety of clustering strategies into the evolutionary process. Different from other EAs, BSO adopts the clustering operation as their converging process, which makes it very suitable to solve multimodal problem. As we all know, different clustering strategies show different advantages. The K-means method in traditional BSO is a typical clustering based on distance. The biggest drawback is that the choice of initial cluster center has a great influence on its clustering result. Once the initial cluster center is not chosen well, it may have a bad influence on its clustering result. On the other hand, the preservation mechanism of traditional BSO is just for comparison of the fitness values of simple problems. For multimodal problem, this mechanism is not able to maintain all individuals which have found extreme point. In this paper we improved the classical BSO to solve multimodal optimization in two aspects. Firstly, a new clustering operation named Max-fitness Clustering method (MCM) replaces the classical k-means clustering method to assign individuals in different promising subregions. Secondly, a self-adaptive parameter control technique is used to ensure retention of individual diversity and convergence. Detailed explanation will be showed as the followings subsections.

3.1 Max-fitness Clustering Method

The K-means clustering method in traditional BSO overly depends on the selection of the initial cluster centers, which cannot solve multimodal optimization. The maximum clustering method is proposed to solve the different clustering center selection in this paper. The operation produces are listed as follows. Firstly, the largest individual fitness value is selected as the first category center from an individual original population. Secondly, the nearest to the center of each individual is emptied. Finally, the process is repeated for the remaining individuals until all of them are classified. The difference between the clustering and other clustering methods is that each category center is the best individual of all the remaining individuals, so each clustering center has a larger probability that is distributed in the extreme point, which makes the individual learning more direction. This clustering algorithm makes full use of the information about each individual, and the solution space is effectively combined with the target space. A schematic illustration of the clustering partition is shown in Fig 2. The replacing operator of BSO is given as line 4&5 in Fig.1.

```
Algorithm  Max-fitness clustering method
Step 1 Initialize a population $P$ of $Np$ ideas $\{X_i | i=1,2,...,Np\}$
     in the search region randomly
Step2 Find the best(fitness value) idea as seed $X$
Step3 Combine $M-1$ ideas of the population $P$, which are
      nearest to $X$, with $X$ to form a subpopulation. Any
      tie will be broken randomly.
Step4  Eliminate these $M$ ideas from $P$
Step5 Execute Step 1- Step 3 repeatedly until the population
$P$ is divided
     into $P/M$ subpopulation
```

Fig. 2. Pseudo-code of the max-fitness clustering

3.2 Self-adaptive Parameter Control

According to Eq.(1), a new idea is created by adding Gaussian random noise to the selected idea. By this way, all ideas in the group will quickly converge to the direction along the previous best idea. In this paper, a self-adaptive parameter control is used to make the algorithm quickly converge to different optima by different clusters. The crossover probability Cr is updated dynamically according to the evolution process. This is similar to the adaptation strategy used in JADE [13].

In the mutation operation, the new idea $u_{i,j}$ is generated by making use of a binomial crossover operation on the last iteration idea $x_{i,j}$ and the after mutation idea $v_{i,j}$.

$$u_{i,j} = \begin{cases} v_{i,j} & if(rand_j(0,1) \le \mathrm{C_r}) or (j = j_{rand}) \\ x_{i,j} & otherwise \end{cases} \tag{3}$$

Where $i = 1,2,...,Np$, $j = 1,2,...,n\_d$, j_{rand} is a randomly chosen integer from $\{1,2,,...,n\_d\}$, $rand_j(0,1)$ is a random value within (0,1). Due to the use of j_{rand} , $u_{i,j}$ is guaranteed to differ from $x_{i,j}$. At each generation, the crossover rate Cr_i of each idea is independently generated according to a normal distribution of mean C_{rm} and standard deviation 0.1

$$Cr_i = randn(C_{rm}, 0.1) \tag{4}$$

C_{rm} is updated as follow:

$$C_{rm} = mean(S_{cr}) \tag{5}$$

Where S_{cr} is the set of all successful crossover rates at previous generation. The initial value of C_{rm} is 0.5 and $S_{cr} = \phi$.

The selection operation is conducted by comparing u_i and the closest individual x_s in population. The better one will enter the next generation. For the maximization problem:

$$x_s = \begin{cases} u_i & if \quad f(u_i) > \mathrm{f}(x_s) \\ x_s & \qquad otherwise \end{cases} \tag{6}$$

A schematic illustration of the self-adaptive parameter control is shown in algorithm 2.

```
Algorithm2 self-adaptive parameter control
For $i = 1 : Np$
Step 1 Use the Eq.3 to produce new idea ;
Step 2 Evaluate offspring using fitness function;
Step 3 According to Eq.(6) to update idea and save the respective
      crossover probability in $Scr$
End for
Step 4 Update $Crm = mean(Scr)$ .
```

Fig. 3. Pseudo-code of the self-adaptive parameter control

4 Experimental Studies

4.1 Parameter Setting

A set of multimodal optimization benchmark functions [14] is used to evaluate the ability of the proposed algorithm in this paper. A level of accuracy (typically $0 < \varepsilon < 1$) indicates the error between the fitness value and the given extreme point. The extreme point will be considered been found if the error is less than ε . In order to compare with other algorithms fairly, the parameters will follow the guideline given in [15] and [16] and apply these parameters to all compared algorithms. The level of accuracy (ε), niching radius (γ), population size and maximal number of function evaluations (FES) allowed are listed in Table II. From the table 2, we can see that a larger number of optima require a large population size and more function evaluations.

Table 1. Parameter setting of different benchmark functions

Test function	Population size	M	ε	γ	No. of function evaluations
F1	50	5	0.05	0.5	10,000
F2	50	5	0.05	0.5	10,000
F3	50	5	0.05	0.5	10,000
F4	50	5	0.000001	0.01	10,000
F5	50	5	0.000001	0.01	10,000
F6	50	5	0.000001	0.01	10,000
F7	50	5	0.000001	0.01	10,000
F8	50	5	0.0005	0.5	10,000
F9	50	5	0.000001	0.5	10,000
F10	250	25	0.00001	0.5	10,000
F11	250	10	0.05	0.5	100,000
F12	100	10	0.0001	0.2	20,000
F13	500	50	0.001	0.2	200,000
F14	1000	100	0.001	0.2	400,000

4.2 Experimental Result

To give a clearer view, SBSO is compared with the original BSO. The distributions of population of the two algorithms at different iterations are plotted in Fig. 4.（the function of F6） and Fig. 5.（the function of F7）.

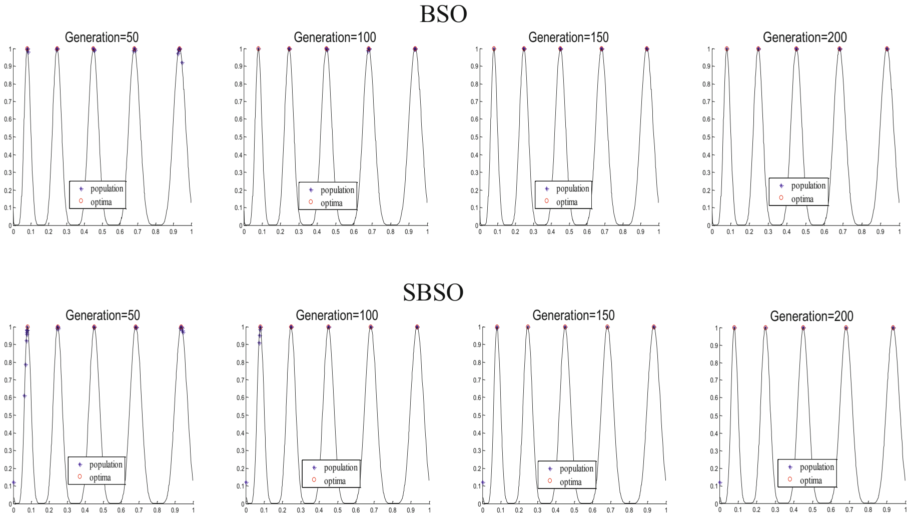


Fig. 4. Distributions of population of BSO and SBSO on different stages（F6）

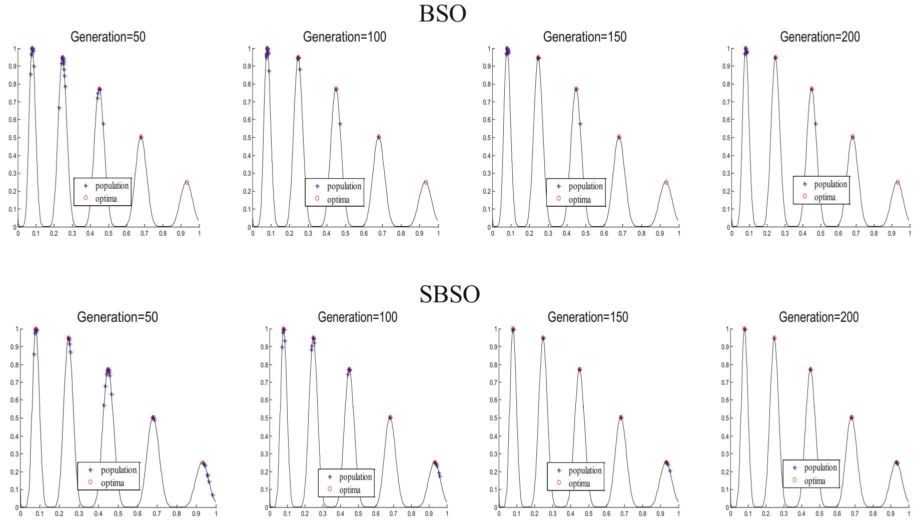


Fig. 5. Distributions of population of BSO and SBSO on different stages（F7）

From the above figure, we can clearly see the original BSO have accurately found all the extreme points of function F6 which has the same extreme points, but for the function like F7 which has different extreme points, only the SBSO can find and save all the extreme points. With the number of iterations increasing, the SBSO can converge to the extreme points. Therefore, it presents that the effectiveness of SBSO in dealing with multimodal optimization problem. In order to further demonstrate the superior performance of this algorithm, we compare the performance indicators of the proposed algorithm with the other algorithms in one of the latest multimodal works [17]. As the same as the literature [17], we adapts the following indicator as our assessment criteria.

Peak Accuracy: For each desired $peak_i,(i=1,2,...,\#\,peaks)$ to be located, the closet idea X in the population is taken and absolute difference in objective values is calculated. If the objective value of idea X is denoted by $f(X)$,the peak accuracy is calculated using

$$peak \quad accuracy = \sum_{i=1}^{\#\,peaks} \frac{\left| f\left(peak_i\right) - f\left(X\right) \right|}{\#\,peaks}$$

On all the benchmark functions, all algorithms are run until all known peaks are found or maximum number of FEs is exhausted. The proposed algorithm is compared with following standard multimodal algorithms which are generally cited in literature [17]:

1) CDE: Crowding DE;
2) SDE: Speciation-based DE
3) FER-PSO: Fitness –Euclidean distance Ratio PSO
4) SPSO: Speciation-based PSO

5) r3pso: a local best PSO with a ring topology•each member interacts with its immediate member on its left and right.
6) r2psolhc: no overlapping neighborhoods, hence acting as multiple local hill climbers, more suitable for finding global as well as local optima.
7) r3psolhc: Basically multiple PSOs search in parallel, like local hill climbers. This variant is more appropriate if the goal of optimization is to find global ad well as local optima.

The simulation results are presented in Table 3.

Table 2. Peak Accuracy comparisons of different algorithms

Fun	SBSO	CED[18]	SDE[19]	FER-PSO[14]	SPSO[14]	r3pso[14]	r2psolhc[14]	r3psolhc[14]
F1	4.33e-05(3)	9.46e-08 (2)	*1.23e-08(1)*	5.24e-02(5)	8.74e-02(7)	3.54e-02(4)	9.78e-02(8)	8.68e-02(6)
F2	1.78e-05(3)	8.76e-06 (2)	*3.43e-07(1)*	9.65e-04(4)	9.45e-02(8)	1.43e-02(5)	5.23e-02(7)	4.54e-02(6)
F3	5.49e-04(3)	9.76e-05 (2)	*5.73e-05(1)*	41999,.34e-02(4)	9.54e-01(8)	7.93e-02(6)	5.52e-01(7)	7.64e-02(5)
F4	*2.63e-10(1)*	7.43e-05 (8)	9.53e-07(7)	5.65e-07(6)	3.12e-07(5)	2.24e-07(4)	5.43e-09（2）	9.43e-08(3)
F5	8.24e-09(3)	9.43e-06 (8)	4.03e-09(2)	8.34e-09(4)	*2.16e-09(1)*	9.61e-07(7)	8.58e-07(6)	5.36e-07(5)
F6	*7.79e-10(1)*	5.39e-05 (8)	8.27e-07(7)	5.45e-09(2)	9.58e-08(4)	8.79e-08(3)	8.05e-07(6)	7.54e-08(5)
F7	1.23e-06(7)	8.97e-05 (8)	4.51e-07(3)	7.41e-07(4)	*2.98e-07(1)*	9.29e-07(5)	3.43e-07(2)	9.88e-07(6)
F8	*4.53e-06(1)*	4.27e-02 (7)	8.57e-04(2)	8.69e-04(3)	5.21e-02(8)	5.63e-03(4)	6.12e-03(5)	9.19e-03(6)
F9	*2.26e-08(1)*	3.42e-04 (7)	5.33e-08(2)	7.38e-08(3)	3.58e-04(8)	6.92e-05(5)	4.33e-05(4)	8.28e-05(6)
F10	*5.91e-08(1)*	3.96e-03 (7)	9.92e-02(8)	5.50e-06(2)	9.77e-04(6)	8.31e-05(3)	9.43e-05(4)	6.87e-04(5)
F12	*1.10e-005(1)*	5.24e-04 (7)	8.33e-04(8)	4.53e-04(5)	1.23e-04(2)	3.87e-04(4)	2.54e-04(3)	4.56e-04(6)
F13	*7.93e-004(1)*	9.87e-04 (2)	9.85e-03(8)	9.69e-03 (7)	8.45e-03 (5)	8.25e-03(4)	8.52e-03 (6)	6.32e-03 (3)
F14	*1.4e-003(1)*	9.23e-02 (2)	7.89e-01(5)	5.95e-01 (3)	7.86e-01 (4)	8.45e-01 (7)	9.68e-01 (8)	8.12e-01(6)
Total ranks	27	70	52	56	67	61	63	68

Please note that function F11 is a 2-D inverted Shubert function, which is not reported in[19]. Thus, the simulation results of function F11 is not included in Table III. Above the table, the numbers in brackets present the rank of Peak Accuracy, which is obtained by the different algorithms dealing with the same function. The smaller the rank, the higher the accuracy. The last line of the table is the total rank (i.e. summation of all the individual ranks). The lower the total rank, the better the performance of the algorithm. From the result, in terms of peak accuracy, we can see that the proposed algorithms show better performance. Besides, it also indicates that the proposed algorithm present a good exploitative behavior in convergence to different global and local optima.

5 Conclusion

In this paper, there are two parts being proposed to modify the BSO algorithm in multimodal optimization. The clustering strategy drives populations to search the different sub-regions to obtain the potential multiple global and local optima. At the same time, it also reduces the calculation of complexity in the proposed algorithm. Self-adaptive parameter control maintains population diversity by allowing competition to limit resources among similar idea in subpopulation. Our future works will focus on testing the performance of the algorithm on much more massive multimodal problems with high dimensionality and constraints. In some degree, the subpopulation size M affects the performance of the algorithm, so how to design an adaptive strategy to control M through the process is the future work.

References

1. Thomsen, R.: Multimodal optimization using crowding-based differential evolution. In: Proc. IEEE Congr. Evol. Comput., pp. 1382–1389 (June 2004)
2. Wang, H.F., Moon, I., Yang, S.X., Wang, D.W.: A memetic particle swarm optimization algorithm for multimodal optimization problems. IEEE Trans. Cyber. 43(2), 634–647 (2013)
3. Parrott, D., Li, X.: Locating and tracking multiple dynamic optima by a particle swarm model using speciation. IEEE Trans. Evol. Comput. 10(4), 440–458 (2006)
4. Wang, Y.J., Zhang, J.S., Zhang, G.Y.: A dynamic clustering based differential evolution algorithm for global optimization. Journal of Operational Research 183(1), 56–73 (2007)
5. Li, X.: Niching without niching parameters: Particle swarm optimization using a ring topology. IEEE Transaction on Evolutionary Computation 14(1), 150–169 (2010)
6. Rigling, B., Moore, F.: Exploitation of subpopulations in evolutionary strategies for improved numerical optimization. In: Proc. 11th Midwest Artif. Intell. Cogn. Sci. Conf., pp. 80–88 (1999)
7. Rumpler, J., Moore, F.: Automatic selection of subpopulations and minimal spanning distances for improved numerical optimization. In: Proc. Congr. Evol. Comput., pp. 38–43 (2001)
8. Zaharie, D.: A multi-population differential evolution algorithm for multimodal optimization. In: Proc. 10th Mendel Int. Conf. Soft Comput., pp. 17–22 (June 2004)
9. Hendershot, Z.: A differential evolution algorithm for automatically discovering multiple global optima in multidimensional discontinuous spaces. In: Proc. 15th Midwest Artif. Intell. Cogn. Sci. Conf., pp. 92–97 (April 2004)
10. Zaharie, D.: Extensions of differential evolution algorithms for multimodal optimization. In: Proc. SYNASC, pp. 523–534 (2004)
11. Shi, Y.: Brain storm optimization algorithm. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) ICSI 2011, Part I. LNCS, vol. 6728, pp. 303–309. Springer, Heidelberg (2011a)
12. Zhan, Z.H., Shi, Y., Zhang, J.: A modified brain storm optimization. In: IEEE World Congr. Comput. Intell. (Jule10-15, 2012)
13. Zhang, J., Sanderson, A.C.: JADE: Adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. 13(5), 945–958 (2009)
14. Li, X.: Niching without niching parameters: Particle swarm optimization using a ring topology. IEEE Trans. Evol. Comput. 14(1), 150–169 (2010)

15. Qu, B.Y., Suganthan, P.N., Liang, J.J.: Differential evolution with neighborhood mutation for multimodal optimization. IEEE Trans. Evol. Comput. 16(5), 601–614 (2012)
16. Qu, B.Y., Suganthan, P.N., Das, S.: A distance-based locally informed particle swarm model for multi-modal optimization. IEEE Trans. Evol. Comput. 17(3), 387–402 (2013)
17. Roya, S., Islama, S.M., Dasb, S., Ghosha, S.: Multimodal optimization by artificial weed colonies enhanced with localized group search optimizers. Appl. Soft Comput. 13(1), 27–46 (2013)
18. Thomsen, R.: Multimodal optimization using crowing-based differential evolution. In: Proc. IEEE Congr. Evol. Comput., pp. 1382–1389 (June 2004)
19. Li, X.: Efficient differential evolution using speciation for multimodal function optimization. In: Proc. Conf. Genetic Evol. Comput., pp. 873–880 (2005)

Classification of Electroencephalogram Signals Using Wavelet Transform and Particle Swarm Optimization

Nasser Omer Ba-Karait[1,2], Siti Mariyam Shamsuddin[1,2], and Rubita Sudirman[3]

[1] UTM Big Data Centre, Universiti Teknologi Malaysia
[2] Faculty of Computing
Universiti Teknologi Malaysia, 81310 Skudai, Johor Bahru, Malaysia
[3] Faculty of Electrical Engineering
Universiti Teknologi Malaysia, 81310 Skudai, Johor Bahru, Malaysia
bakarait@yahoo.com, mariyam@utm.my, rubita@fke.utm.my

Abstract. The electroencephalogram (EEG) is a signal measuring activities of the brain. Therefore, it contains useful information for diagnosis of epilepsy. However, it is a very time consuming and costly task to handle these subtle details by a human observer. In this paper, particle swarm optimization (PSO) was proposed to automate the process of seizure detection in EEG signals. Initially, the EEG signals have been analysed using discrete wavelet transform (DWT) for features extraction. Then, the PSO algorithm has been trained to recognize the epileptic signals in EEG data. The results demonstrate the effectiveness of the proposed method in terms of classification accuracy and stability. A comparison with other methods in the literature confirms the superiority of the PSO.

Keywords: Particle swarm optimization, machine learning, discrete wavelet transform, EEG, epileptic seizure.

1 Introduction

The electroencephalogram (EEG) is a highly complex signal recording the brain's neural activities through changes in electrical potentials at multiple locations over the scalp. It conveys valuable clinical information about the state of the brain. Therefore in recent decades, the EEG signals (EEGs) have been used intensively to study brain function and neurological disorders. For this reason, the EEG has long been an important clinical tool in diagnosing, monitoring and managing of neurological disorders, especially those related to epilepsy [1, 2]. Epileptic seizures are caused by temporary electrical disturbance of the brain. The occurrence of a seizure seems unpredictable and its course of action is still very poorly understood. Research is therefore needed to gain a better understanding of the mechanisms causing epileptic disorders. Careful analysis of EEGs could provide valuable insight into this widespread brain disorder [3, 4].

A continuous EEG recording is required to study the epileptic seizures for pre-surgical evaluation. It provides essential information for locating the brain regions

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 352–362, 2014.
© Springer International Publishing Switzerland 2014

that generate epileptic activity [5, 6]. Clearly, the analysis of the recorded EEG based on visual inspection is a very time consuming and costly task. In some cases, the seizures are uncontrollable. Recently, methods have started being developed to treat medically resistant epilepsy through delivering a local therapy to the affected regions of brain. Automatic seizures detection forms an integral part of such methods [6, 7].

It is therefore worthwhile to propose an effective algorithm for EEG changes recognition. In literature, most methods deal with this problem based on well-known classification techniques [8-12]. However, classification can be seen in multidimensional space as optimization problem where a class is identified by a centroid. In this case, particle swarm optimization (PSO) can be employed effectively to optimize coordinates of the centroids [13, 14]. To the best of our knowledge, the PSO has not been used for EEGs classification. However in many works, the classifier of EEGs is trained and/or its parameters are optimized by PSO [15, 16]. Also, it was employed to estimate the locations of sources of electrical activity, e.g. epileptic, in the brain based on the scalp EEGs [17-19]. Other EEGs issues have been addressed by PSO such as feature selection [20, 21] and optimal selection of Electrode Channels [22, 23]. In this research, the PSO algorithm is studied to evaluate its performance in detecting the epileptic seizures in EEGs using discrete wavelet transform (DWT) for feature extraction.

2 Particle Swarm Optimization

Particle swarm optimization (PSO) algorithm was originally designed by Kennedy and Eberhart [24] in 1995. The idea was inspired by the social behavior of flocking organisms, as a bird flock. PSO uses a population (called swarm) of candidate solutions (called particles) to probe promising regions of the search space. Each particle moves in the search space with a velocity that is dynamically adjusted according to its own flying experience and its companions' flying experience and it retains the best position it ever encountered in memory. The best position ever encountered by all particles of the swarm is also communicated to all particles.

The popular form of PSO algorithm is defined as:

$$v_{i,d}(t+1) \leftarrow w * v_{i,d}(t) + c_1 r_1 (p_{i,d}(t) - x_{i,d}(t)) + c_2 r_2 (p_{g,d}(t) - x_{i,d}(t)) \tag{1}$$

$$x_{i,d}(t+1) \leftarrow x_{i,d}(t) + v_{i,d}(t+1) \tag{2}$$

where $v_{\mathrm{i,d}}$ is the velocity of particle i along dimension d, $x_{\mathrm{i,d}}$ is the position of particle i in d, c_1 is a weight applied to the cognitive learning portion, and c_2 is a similar weight applied to the influence of the social learning portion. r_1 and r_2 are separately generated random numbers in the range between zero and one. $p_{\mathrm{i,d}}$ is the previous best location of particle i; it is called the personal best (*pbest*). $P_{\mathrm{g,d}}$ is the best location found by the entire population; it is called the global best (*gbest*). w is the inertia weight [25, 26].

Velocity values must be within a range defined by two parameters -$v_{\max}$ and $v_{\max}$. A PSO with an inertia weight in the range (0.9, 0.4) has the better performance on average. To get a better searching pattern between global exploration and local exploitation, researchers recommended decreasing w linearly over time from a maximal value $w_{\max}$ to a minimal value $w_{\min}$ [26-28].

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{t_{\max}} * t \tag{3}$$

where, $t_{\max}$ is the maximum number of iterations allowed and t is the current iteration number.

3 Materials and Methods

3.1 EEG Data

The current study used the publicly available EEG data described by Andrzejak *et al.* [29]. The complete dataset contains five different sets (denoted A-E), each containing 100 single channel EEG segments of 23.6 second duration. Segments in sets A and B were recorded from five healthy volunteers. They were relaxed in an awake state with eyes open (set A) and closed (set B). The EEG archive of presurgical diagnoses was used to originate sets C, D and E by selecting EEGs from five patients. Signals in sets C and D were measured in seizure free intervals from within the epileptogenic zone and opposite the epileptogenic zone of the brain, respectively. Set E were obtained from within the epileptogenic zone during seizure activity. Fig. 1 shows typical EEG segments, one from each category.

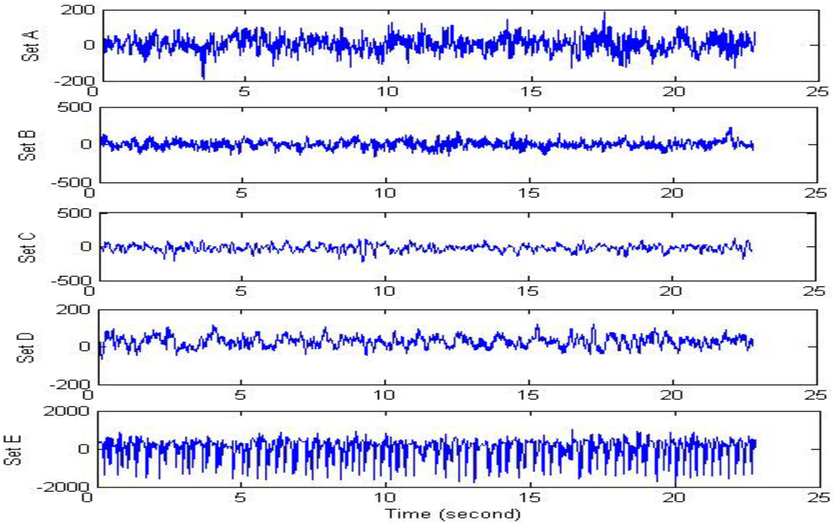


Fig. 1. Samples of five different sets of EEG data

3.2 Discrete Wavelet Transform: Feature Extraction

Discrete wavelet transform (DWT) has been particularly successful in the area of epileptic seizure detection due to its ability to capture transient features and localize them in both time and frequency domains accurately [30]. The DWT analyses a signal *s*(n) at different frequency bands by decomposing the signal into an approximation and detail information using two sets of functions known as scaling functions and wavelet functions, which are associated with low-pass *g*(n) and high-pass *h*(n) filters, respectively. The DWT decomposition process is described in Fig. 2.

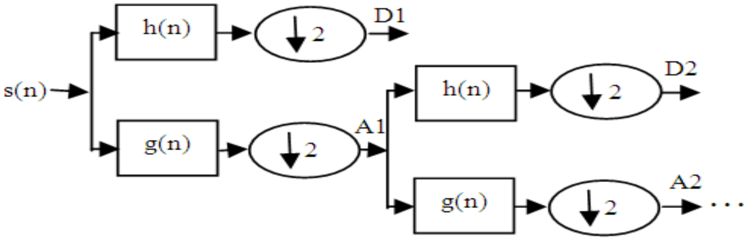


Fig. 2. Sub-band decomposition of DWT

When DWT is used to analyse the signals, two important aspects should be considered: the number of decomposition levels and the type of wavelet. The decomposition level number is selected based on the dominant frequency components of the signal. According to Subasi [31], the levels are selected such that those parts of the signal that correlate well with the frequencies required for the signal classification are retained in the wavelet coefficients. Therefore, level 4 wavelet decomposition was selected for the present study. Accordingly, the EEGs have been decomposed into the details D1-D4 and one final approximation, A4. The smoothing feature of the Daubechies wavelet of order 2 (db2) made it more suitable to detect changes in EEGs [32]. In this research, db2 has been used to compute the wavelet coefficients of the EEGs.

The computed coefficients of discrete wavelet provide a compact representation that shows the energy distribution of the signal in time and frequency. In order to decrease dimensionality of the extracted feature vectors further, statistics over the set of the wavelet coefficients are used [32]. The following statistical features were used to represent the time-frequency distribution of the EEGs: Maximum, Minimum, Mean, and Standard deviation of the wavelet coefficients in each sub-band.

3.3 PSO for EEG Classification

The PSO algorithm has been introduced in this study to classify EEGs for diagnosis purposes. In PSO for classification, each class is identified by a centroid. Therefore, for a dataset $Z = (z_1, z_2, \ldots, z_p, \ldots, z_{Np})$ with N_c classes, where z_p is a pattern with N_d features, and N_P is the number of patterns in Z. The PSO is applied to optimize positions of N_c centroids in an N_d-dimensional space. Fig. 3 highlights model of encoding PSO for EEGs classification. It includes three main steps: particle encoding, defining the fitness function and optimization process.

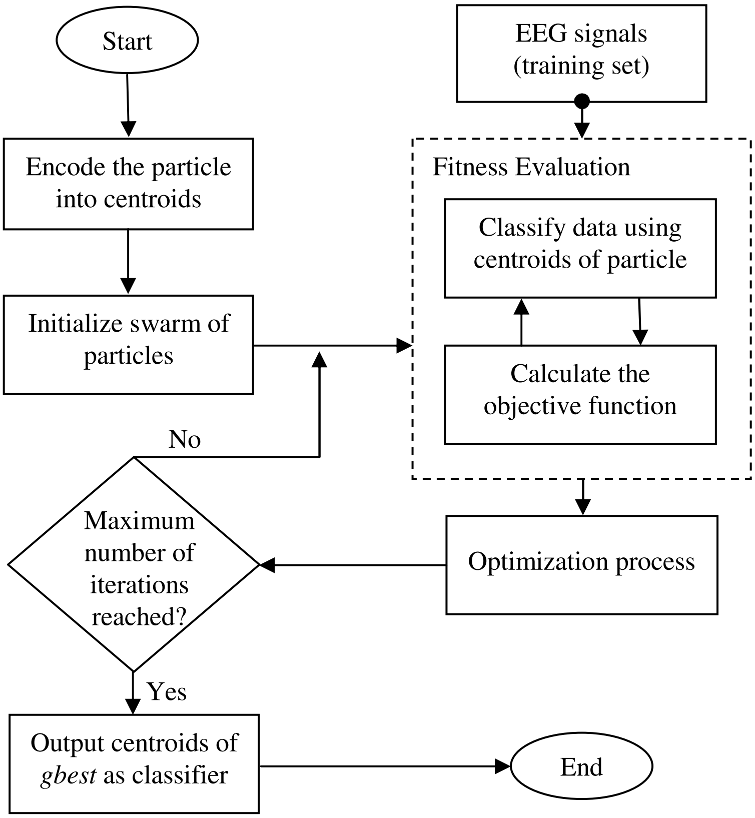

Start
EEG signals (training set)
Encode the particle into centroids
Fitness Evaluation
Classify data using centroids of particle
Initialize swarm of particles
Calculate the objective function
No
Maximum number of iterations reached?
Optimization process
Yes
Output centroids of *gbest* as classifier
End

Fig. 3. PSO-based classification model for EEG signals

Each particle is represented by N_c centroids. Consequently, the position of a particle i is encoded as ($x_i(1)$, …, $x_i(j)$…, $x_i(N_c)$) where $x_i(j)$ refers to the j^{th} centroid of the i^{th} particle. The position of the j^{th} centroid is constituted by N_d real numbers representing its N_d coordinates in the problem space:

$$x_i(j) = \{x_{i,1}(j), x_{i,2}(j), \ldots, x_{i,N_d}(j)\} \tag{4}$$

As above, the velocity of each particle i is encoded as ($v_i(1)$, …, $v_i(j)$…, $v_i(N_c)$) where the velocity of the j^{th} centroid, $v_i(j)$ is made up of N_d real numbers representing its N_d velocity components in the problem space:

$$v_i(j) = \{v_{i,1}(j), v_{i,2}(j), \ldots, v_{i,N_d}(j)\} \tag{5}$$

In classification problem, the objective is to assign any pattern to its correct class. Therefore, the performance of a classification algorithm is evaluated by its accuracy, defined as the percentage of patterns correctly assigned to their classes. This study uses accuracy measure as a fitness function to evaluate the quality of solutions. The fitness of the i^{th} particle is computed based on the dataset portion Z_D (training set) as in Eq. 6.

$$\mathit{Accuracy}(i, z_D) = \frac{\sum_{p=1}^{|Z_D|} \mathit{Assess}(z_p, i)}{|Z_D|} \tag{6}$$

$$\mathit{Assess}(z_p, i) = \begin{cases} 1, & \mathit{if\ Classify}(z_p, i) = z_p.c \\ 0, & \mathit{otherwise} \end{cases} \tag{7}$$

where z_p is a pattern in Z_D, $z_p.c$ is the class of z_p and $\mathit{Classify}(z_p,i)$ returns the class assigned to z_p by the particle i according to the nearest centroid based on Euclidean distance.

With the above premises, optimization mechanism of PSO algorithm is used to update coordinates of the centroids toward the best solution as summarized in Alg.1.

```
Alg.1. PSO for classification
  1. Initialize each particle $i$ to contain $N_c$ centroids
  2. For $t$=1 to $t_{max}$
     a. For each particle $i$
        i. Calculate fitness value using Eq. 6.
       ii. Update the personal best solution, pbest
     b. Update the global best solution, gbest
     c. For each particle $i$
        i. Update the centroids using Eq. 1 and Eq. 2.
     d. Update the inertia weight, $w$ using Eq. 3.
```

4 Experimental Results

4.1 Performance Measures

In medical diagnosis tasks, the common performance measures are sensitivity, specificity and classification accuracy. Sensitivity is defined as the percentage of correctly detected epileptic EEG patterns to the total number of patterns in epileptic EEG. On the other hand, specificity is defined as the percentage of correctly detected normal EEG patterns to the total number of patterns in normal EEG. Finally, the percentage of all correctly classified patterns to the total number of patterns in both normal and seizure EEG dataset represents the accuracy. Formally, the performance of a diagnostic system is measured as

$$Sensitivity = \frac{TP}{TP + FN} \tag{8}$$

$$Specificity = \frac{TN}{TN + FP} \tag{9}$$

where TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives respectively.

Accuracy: Eq.6 is calculated for testing set Z_{T} using the final *gbest*; *Accuracy* (*gbest*, Z_{T}).

4.2 Results and Discussion

The EEG dataset used consists of three categories of signals: healthy (sets A and B), seizure-free (sets C and D) and seizure (set E). Therefore, the three sets: A, D, and E of the above-described dataset are used to analyse the performance of PSO. Sets A and D are gathered to form the normal class against set E which represents the epileptic class. This is similar to real medical applications in which the EEG segments are classified into non-seizures and seizures.

In each set of EEG data, there are 100 EEGs of 4096 samples. In this research, each signal is further divided by a rectangular window composed of 256 samples. Therefore, the dataset of the considered EEG problem was formed of 4800 patterns; i.e., each set has 1600 vectors. Consequentially, the epileptic class contains 1600 patterns, while the number of patterns in the normal class is 3200. The DWT coefficients at the fourth level (D1-D4 and A4) were computed for each pattern. The statistical features that were calculated over the set of wavelet coefficients reduce the dimensionality of feature vector to 20.

It is common to partition the dataset into two separate sets: a training set and a testing set. Additionally, *k*-fold cross validation is often used by the researchers to evaluate the behavior of the algorithm in the bias associated with the random sampling of the training data. In this study, the EEG dataset (sets A, D and E) was randomly divided into training-testing as 50-50%, 70-30%, and with a 10-fold cross validation. The values of the PSO parameters are as follows: $v_{\max}$=0.05, c_1=2.0, c_2=2.0, $w_{\max}$=0.9, $w_{\min}$=0.4. 50 particles were trained for 1000 iterations to evolve two centroids for the normal and epileptic classes. The centroids produced are then used to classify the patterns in the testing set in order to assess the effectiveness of the proposed method.

Table 1 presents the results achieved by the PSO algorithm with respect to the sensitivity, specificity and accuracy. The results are reported in terms of average, and standard deviation (SD) of ten runs for each partition of the dataset. As can be seen from Table 1, the PSO on average classified the EEGs of training-test datasets partitions: 50-50%, 70-30%, and 10-fold cross validation with accuracies of 96.91%, 97.08%, and 96.53% respectively. The results using all training-test datasets partitions are depicted in Fig. 4. These overall results illustrate that the PSO has good performance and stable behaviour for EEGs classification with accuracy of 96.84%, and standard deviation of 0.90.

Table 1. Sensitivity, specificity, and accuracy of the PSO algorithm on EEG signals

Training-testing dataset partitions (%)	Performance measures (%)			
		Sensitivity	Specificity	Accuracy
50-50	Average	96.50	97.12	96.91
	SD	1.04	0.57	0.30
70-30	Average	96.49	97.38	97.08
	SD	0.83	0.63	0.37
10-fold cross validation	Average	95.75	96.92	96.53
	SD	3.77	2.35	1.45

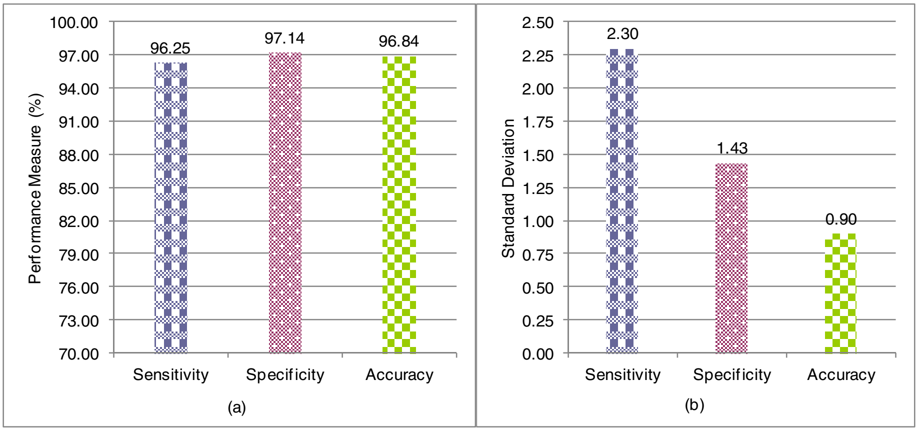


Fig. 4. Overall results of PSO on EEG signals: average (a), and standard deviation (b)

Table 2 illustrates a comparative study of the proposed algorithm with other studies in the literature. For a feasible comparison, only the studies that used the same EEG dataset with the three mentioned sets (A, D and E), and DWT for features extraction are considered. It can be concluded from this comparison that the PSO algorithm showed a promising performance compared to other methods, with a difference in accuracy varies from 0.17% to 6.46%. This proves its ability to compete with well-known classification techniques. In fact, the results reveal that combination of PSO and DWT can produce an efficient automated system for diagnosing epileptic seizures in EEGs.

Table 2. Comparison of the PSO accuracy on EEG signals with methods in the literature

Study	Method	Accuracy
Übeyli [8]	Mixture of expert	93.17
Übeyli [9]	Combined neural network	94.83
Hsu and Yu [10]	Genetic algorithm, support vector machine	90.38
Guo et al. [11]	Genetic programming, k-Nearest Neighbor	93.50
Orhan et al. [12]	K-means clustering, artificial neural network	96.67
This study	Particle swarm optimization	96.84

5 Conclusion

In the present study, discrete wavelet transform and particle swarm optimization have been hybridized to process EEGs for automatic diagnosis of epileptic seizures. The DWT is used to extract the features of signals for the PSO which separates epileptic signals from others in the EEG data. The ability of the proposed method was tested on EEG recordings with healthy, seizure-free, and seizure data. The results indicate that the PSO has very good performance in discriminating the EEGs compared to algorithms reported in the literature. Therefore, the proposed system could be a powerful tool to assist experts in facilitating the analysis of a patient's information and reducing the time and effort required to make accurate decisions on their patients.

Acknowledgment. This work is supported by Research University Grant (Vot 03H72) and Long Term Research Grant (LRGS/TD/2011/UTM/ICT/03/03). The authors would like to thanks Research Management Centre (RMC), Universiti Teknologi Malaysia (UTM) for the support in R & D, Soft Computing Research Group (SCRG) for the inspiration in making this study a success. The authors would also like to thank the anonymous reviewers who have contributed enormously to this work.

References

1. Adeli, H., Zhou, Z., Dadmehr, N.: Analysis of EEG records in an epileptic patient using wavelet transform. Journal of Neuroscience Methods 123, 69–87 (2003)
2. Übeyli, E.D.: Least squares support vector machine employing model-based methods coefficients for analysis of EEG signals. Expert Systems with Applications 37, 233–239 (2010)
3. Subasi, A.: EEG signal classification using wavelet feature extraction and a mixture of expert model. Expert Systems with Applications 32, 1084–1093 (2007)
4. Nigam, V.P., Graupe, D.: A neural-network-based detection of epilepsy. Neurological Research 26, 55–60 (2004)
5. Ocak, H.: Optimal classification of epileptic seizures in EEG using wavelet analysis and genetic algorithm. Signal Processing 88, 1858–1867 (2008)
6. Patnaik, L.M., Manyam, O.K.: Epileptic EEG detection using neural networks and post-classification. Computer Methods and Programs in Biomedicine 91, 100–109 (2008)

7. Gardner, A.B.: A novelty detection approach to seizure analysis from intracranial EEG. PhD Thesis, Georgia Institute of Technology. Georgia, United States (2004)
8. Übeyli, E.D.: Wavelet/mixture of experts network structure for EEG signals classification. Expert Systems with Applications 34, 1954–1962 (2008)
9. Übeyli, E.D.: Combined neural network model employing wavelet coefficients for EEG signals classification. *Digital Signal Processing 19, 297–308 (2009)*
10. Hsu, K.-C., Yu, S.-N.: Detection of seizures in EEG using subband nonlinear parameters and genetic algorithm. *Computers in Biology and Medicine 40, 823–830 (2010)*
11. Guo, L., Rivero, D., Dorado, J., Munteanu, C.R., Pazos, A.: Automatic feature extraction using genetic programming: An application to epileptic EEG classification. *Expert Systems with Applications 38, 10425–10436 (2011)*
12. Orhan, U., Hekim, M., Ozer, M.: EEG signals classification using the K-means clustering and a multilayer perceptron neural network model. Expert Systems with Applications 38, 13475–13481 (2011)
13. Sousa, T., Silva, A., Neves, A.: Particle Swarm based Data Mining Algorithms for classification tasks. Parallel Computing 30, 767–783 (2004)
14. De Falco, I., Cioppa, A.D., Tarantino, E.: Facing classification problems with Particle Swarm Optimization. Applied Soft Computing 7, 652–658 (2007)
15. Hema, C.R., Paulraj, M.P., Nagarajan, R., Yaacob, S., Adom, A.H.: Application of particle swarm optimization for EEG signal classification. Biomedical Soft Computing and Human Sciences 13, 79–84 (2008)
16. Chai, R., Ling, S., Hunter, G., Tran, Y., Nguyen, H.: Brain Computer Interface Classifier for Wheelchair Commands using Neural Network with Fuzzy Particle Swarm Optimization. IEEE Journal of Biomedical and Health Informatics (in Press)
17. Qiu, L., Li, Y., Yao, D.: A feasibility study of EEG dipole source localization using particle swarm optimization. In: 2005 IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, UK, vol. 1, pp. 720–726 (2005)
18. Xu, P., Tian, Y., Lei, X., Yao, D.: Neuroelectric source imaging using 3SCO: A space coding algorithm based on particle swarm optimization and l_0 norm constraint. *NeuroImage 51, 183–205 (2010)*
19. Shirvany, Y., Mahmood, Q., Edelvik, F., Jakobsson, S., Hedstrom, A., Persson, M.: Particle Swarm Optimization Applied to EEG Source Localization of Somatosensory Evoked Potentials. IEEE Transactions on Neural Systems and Rehabilitation Engineering 22, 11–20 (2014)
20. Nakamura, T., Ito, S., Mitsukura, Y., Setokawa, H.: A Method for Evaluating the Degree of Human's Preference Based on EEG Analysis. In: Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, pp. 732–735 (2009)
21. Zhiping, H., Guangming, C., Cheng, C., He, X., Jiacai, Z.: A new EEG feature selection method for self-paced brain-computer interface. In: 10th International Conference on Intelligent Systems Design and Applications, pp. 845–849. IEEE, Cairo (2010)
22. Jin, J., Wang, X., Zhang, J.: Optimal selection of EEG electrodes via DPSO algorithm. In: 7th World Congress on Intelligent Control and Automation, pp. 5095–5099. IEEE, Chongqing (2008)
23. Kim, J.-Y., Park, S.-M., Ko, K.-E., Sim, K.-B.: A Binary PSO-Based Optimal EEG Channel Selection Method for a Motor Imagery Based BCI System. In: Lee, G., Howard, D., Ślęzak, D., Hong, Y.S. (eds.) ICHIT 2012. CCIS, vol. 310, pp. 245–252. Springer, Heidelberg (2012)

24. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948 (1995)
25. Ghosh, S., Das, S., Kundu, D., Suresh, K., Abraham, A.: Inter-particle communication and search-dynamics of lbest particle swarm optimizers: An analysis. Information Sciences 182, 156–168 (2012)
26. Samal, N.R., Konar, A., Das, S., Abraham, A.: A closed loop stability analysis and parameter selection of the Particle Swarm Optimization dynamics for faster convergence. In: IEEE Congress on Evolutionary Computation, Singapore, pp. 1769–1776 (2007)
27. Eberhart, R.C., Shi, Y.: Particle swarm optimization: developments, applications and resources. In: 2001 Congress on Evolutionary Computation, Seoul, Korea, pp. 81–86 (2001)
28. Lin, C.-L., Mimori, A., Chen, Y.-W.: Hybrid Particle Swarm Optimization and Its Application to Multimodal 3D Medical Image Registration. Computational Intelligence and Neuroscience 2012, 7 (2012)
29. Andrzejak, R.G., Lehnertz, K., Mormann, F., Rieke, C., David, P., Elger, C.E.: Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. Physical Review E 64, 061907 (2001)
30. Subasi, A.: Automatic detection of epileptic seizure using dynamic fuzzy neural networks. Expert Systems with Applications 31, 320–328 (2006)
31. Subasi, A.: Epileptic seizure detection using dynamic wavelet network. Expert Systems with Applications 29, 343–355 (2005)
32. Güler, İ., Übeyli, E.D.: Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients. Journal of Neuroscience Methods 148, 113–121 (2005)

FOREX Rate Prediction Using Chaos, Neural Network and Particle Swarm Optimization

Dadabada Pradeepkumar[1,2] and Vadlamani Ravi[1,*]

[1] Center of Excellence in CRM and Analytics,
Institute for Development and Research in Banking Technology,
Hyderabad-500057, India
[2] SCIS, University of Hyderabad, Hyderabad-500046, India
dpradeepphd@gmail.com, rav_padma@yahoo.com

Abstract. This paper presents two two-stage intelligent hybrid FOREX Rate prediction models comprising chaos, Neural Network (NN) and PSO. In these models, Stage-1 obtains initial predictions and Stage-2 fine tunes them. The exchange rates data of US Dollar versus Japanese Yen (JPY), British Pound (GBP), and Euro (EUR) are used to test the effectiveness of hybrid models. We conclude that the proposed intelligent hybrid models yield better predictions compared to the baseline neural networks and PSO in terms of MSE and MAPE.

Keywords: FOREX Rate Prediction, Hybrid model, Chaos, MLP, GRNN, GMDH, PSO.

1 Introduction

The accurate prediction of Foreign Exchange (FOREX) rates helps a country to make its economy stronger [1]. The prediction process is carried out using time series analysis. A time series is said to be chaotic time series if and only if it is non-linear, deterministic and sensitive to initial conditions [2]. The exchange rates are inherently noisy, non-stationary and deterministically chaotic [34]. Chaotic time series prediction involves the prediction of chaotic system behavior in future based on information of current and past states of that system and is always a complex problem.

In order to predict today's value accurately one has to ask the following questions: (i) How many steps of relevant past values are to be considered in the process of prediction? (ii) How many such past values are needed? The answer to the first question is lag and to the second one is embedding dimension.

The following two studies form background to the current work. Ravi et al. [20] presented a forecasting model using a number of computational intelligent techniques. According to them, a variable Y_t is predicted using a vector of lagged variables $Y_{t-1}, Y_{t-2} ... Y_{t-m}$ where m is count of considered lagged variables. Then, Hadvandi et al. [21] presented the PSO-based autoregression time series model

[*] Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 363–375, 2014.
© Springer International Publishing Switzerland 2014

to forecast gold price, where only last two days' gold price is considered to predict today's gold price. They employed PSO to estimate optimal coefficients of the model.

The drawbacks of these studies are: (i) Both studies considered sequential lagged variables without using any scientific method to determine the optimal lag. (ii) Both studies took arbitrary count of lagged variables which is not a scientific approach. (iii) Both studies did not check the presence of chaos in dataset. (iv) Hadvandi et al. [21] modeled only characteristic information of gold price time series but they did not model residual informaton. Hence the model needs to be extended.

The proposed two-stage prediction models have the following features:

1. The models check for presence of chaos at both stages.
2. The methodology determines both optimal lag and optimal embedding dimension scientifically as opposed to guessing them arbitraily.
3. They model both characteristic information and residual information in order to yield better predictions.

The remainder of this paper is organized as follows: A review of literature is presented in Section 2. In section 3, we presented overview of proposed hybrid models. In section 4, experimental methodology is presented. Section 5 discusses the obtained results. The paper is then concluded in Section 6.

2 Literature Survey

It is known that combining many forecasting models yields better estimates than single model [35,36,37,38] in general and in time series [6] in particular. Many previous researches had presented various hybrid FOREX rate prediction models. In this direction, Ni and Yin [7] proposed a hybrid of various regressive neural networks and trading indicators moving average convergence/divergence, relative strength index and genetic algorithm, Zhang [8] hybridized ARIMA and MLP models, Zhang and Wan [9] proposed a statistical fuzzy interval neural network, Donate et. al. [10] proposed a weighted cross-validation evolutionary artificial neural network (EANN) ensemble, Gheyas et. al. [11] proposed novel neural network ensemble, Yu et. al. [12] proposed a multistage nonlinear radial basis function (RBF) neural network ensemble, Sermpins et.al. [13] proposed hybrid neural network architecture of Particle Swarm Optimization and Adaptive Radial Basis Function (ARBF-PSO), Chang [14] proposed hybrid (PSOBPN) that is composed of particle swarm optimization and back propagation network (BPN), Huang et. al. [15] implemented a two-stage chaos and Support Vector Machines (SVMs), Aladag et. al. [16] proposed a time invariant fuzzy time series forecasting method based on PSO, Rout et. al. [17] proposed a hybrid prediction model by combining an adaptive ARMA and Differential Evolution (DE) based training of its feed-forward and feed-back parameters, Chen and Leung [18] proposed a hybrid comprising a time series model and GRNN in tandem and Ince and Trafalis [19] proposed a hybrid two-stage model consisting of ARIMA, VAR and SVR and NN to predict foreign exchange rates. All of these researches presented that hybrid forecasting models yielded better predictions than standalone forecasting models. However comparing all of them is out of the scope of the paper.

3 Proposed Models

3.1 Notations

Let l_1 and m_1 be lag and embedding dimensions that are used in Stage-1; l_2 and m_2 be lag and embedding dimensions that are used in Stage-2 respectively; $e(t)$ be error at time t and $\dot{\mathrm{e}}(t)$ be predicted error at time t; $\alpha_0, \alpha_1, \alpha_2, ...$ be coefficients to be optimized. Finally, let $\dot{\mathrm{y}}(t)$ be the predicted value of Stage-1 at time t and $\ddot{\mathrm{y}}(t)$ be the final predicted value at time t; $f(.)$ be a non-linear function for obtaining predictions using Multi-Layer Perceptron(MLP)/General Regression Neural Network (GRNN)/ Group Method for Data Handling(GMDH).

3.2 Hybrid Model-1

The proposed Hybrid Model-1 consists of two stages. Each stage, in turn, works in both training and test phases. The detailed description is as follows. Let $Y = \{y(1), y(2), ..., y(k), y(k+1), ..., y(n)\}$ be a dataset of n observations at times $1, 2, ..., n$ respectively. First, check for the presence of chaos in Y and rebuild Y using l_1 and m_1, if chaos is present. Later, divide this dataset as Training set $Y_1 = \{y(1), y(2), ..., y(k)\}$ and Test set $Y_2 = \{y(k+1), y(k+2), ..., y(n)\}$. The two-stage prediction proceeds as follows:

I. Stage-1: Using NN (MLP/GRNN/GMDH)

A. Training Phase

1. Input Y_1 to NN.
2. Train NN and obtain initial training set predictions using eq.(1) and errors using eq. (2):

$$\dot{y_1}(t) = f(y(t-l_1), y(t-2l_1), y(t-3l_1), ..., y(t-m_1l_1))$$
$$t = l_1m_1+1, l_1m_1+2, ..., k \qquad (1)$$

$$e(t) = y(t) - \dot{y_1}(t); t = l_1m_1+1, l_1m_1+2, ..., k \qquad (2)$$

B. Test Phase

1. Input Y_2 to trained NN and obtain initial test set predictions using eq.(3) and errors using eq.(4) :

$$\dot{y_2}(t) = f(y(t-l_1), y(t-2l_1), y(t-3l_1), ..., y(t-m_1l_1))$$
$$t = k+1, k+2, ..., n \qquad (3)$$

$$e(t) = y(t) - \dot{y_2}(t); t = k+1, k+2, ..., n \qquad (4)$$

After Stage-1 ends, Initial training set predictions
$\dot{Y}_1$ ={ $\dot{y}_1(l_1m_1+1)$,$\dot{y}_1(l_1m_1+2)$,...,$\dot{y}_1(k)$} and Initial test set predictions $\dot{Y}_2$= {$\dot{y}_2(k+1)$,$\dot{y}_2(k+2)$,...,$\dot{y}_2(n)$} are obtained.

II. Stage-2: Using PSO-Based Autoregression model
  A. Training Phase
    1. Check $e(t)$ for the presence of chaos. Rebuild $e(t)$ using l_2 and m_2. Obtain ė(t) using PSO-based Autoregression model as in eq. (5), if chaos present; otherwise, apply Polynomial Regression to obtain ė(t).

$$\dot{e}(t) = \alpha_0 + \alpha_1 e(t-l_2) + \alpha_2 e(t-2l_2) + ... + \alpha_{m_2} e(t-m_2l_2)$$
$$t = l_1m_1 + l_2m_2 + 1, l_1m_1 + l_2m_2 + 2, ..., kl_2 \quad (5)$$

    2. Compute final training set predictions using eq. (6):

$$\ddot{y_1}(t) = \dot{y_1}(t) + \dot{e}(t); t = l_1m_1 + l_2m_2 + 1, l_1m_1 + l_2m_2 + 2, ..., k \quad (6)$$

  B. Test Phase
    1. Obtain ė(t) using PSO-based Autoregression model as in eq. (7), if chaos present; otherwise, use Polynomial Regression to obtain ė(t).

$$\dot{e}(t) = \alpha_0 + \alpha_1 e(t-l_2) + \alpha_2 e(t-2l_2) + ... + \alpha_{m_2} e(t-m_2l_2)$$
$$t = k+1, k+2, ..., n \quad (7)$$

    2. Compute final test set predictions using eq. (8):

$$\ddot{y_2}(t) = \dot{y_2}(t) + \dot{e}(t); t = k+1, k+2, ..., n \quad (8)$$

After Stage-2 ends,Final training set predictions
$\ddot{Y}_1$={ $\ddot{y}_1(l_1m_1+l_2m_2+1)$,$\ddot{y}_1(l_1m_1+l_2m_2+2)$,...,$\ddot{y}_1(k)$} and Final test set predictions $\ddot{Y}_2$={ $\ddot{y}_2(k+1)$,$\ddot{y}_2(k+2)$,...,$\ddot{y}_2(n)$} are obtained.

3.3 Hybrid Model-2

The proposed Hybrid Model-2 also consists of two stages. In this hybrid, PSO is invoked in Stage-1 and NN/PR is invoked in Stage-2 (if chaos is present/absent). Accordingly, in this hybrid, in place of eq. (1) and eq. (3), eq. (9) and eq. (10) are used to obtain initial predictions and in place of eq. (5) and eq. (7), eq. (11) and eq. (12) are used to obtain final predictions.

$$\dot{y_1}(t) = \alpha_0 + \alpha_1 y(t-l_1) + \alpha_2 y(t-2l_1) + ... + \alpha_{m_1} y(t-m_1l_1)$$
$$t = l_1m_1 + 1, l_1m_1 + 2, ..., k \quad (9)$$

$$\dot{y_2}(t) = \alpha_0 + \alpha_1 y(t-l_1) + \alpha_2 y(t-2l_1) + ... + \alpha_{m_1} y(t-m_1l_1)$$
$$t = k+1, k+2, ..., n \quad (10)$$

$$\begin{aligned} \dot{e}(t) &= f(e(t-l_2), e(t-2l_2), ..., e(t-m_2l_2)) \\ t &= l_1m_1 + l_2m_2 + 1, l_1m_1 + l_2m_2 + 2, ..., k \end{aligned} \tag{11}$$

$$\begin{aligned} \dot{e}(t) &= f(e(t-l_2), e(t-2l_2), ..., e(t-m_2l_2)) \\ t &= k+1, k+2, ..., n \end{aligned} \tag{12}$$

4 Experimental Design

The foreign exchange data used in our study are obtained from US Federal Reserve System(http://www.federalreserve.gov/releases/h10/hist/). The sets of data collected are of daily US dollar exchange rates with respect to three currencies- JPY, GBP and EUR. The daily data of USD-JPY and USD-GBP from 1st January 1993 to 31st December 2013 (6036 observations each) and USD-EUR from 3rd January 2000 to 31st December 2013 (3772 observations), are used as datasets. From both USD-JPY and USD-GBP datasets,80% of dataset is used as training set (4829 observations) and 20% of dataset is used as test set (1207 observations) and from USD-EUR dataset,80% of dataset is used as training set (3018 observations) and 20% of dataset is used as test set (754 observations).

In the proposed hybrid models, Saida's Method [22,23] implemented in MATLAB is used for checking the presence of chaos. Akaike Information Criterion (AIC) [24,31] available in Gretl tool is used to obtain optimal lag. Cao's Method [25,32] implemented in MATLAB is used to obtain minimum embedding dimension. Various Neural Networks (MLP/GRNN/GMDH) available in NeuroShell tool [26,27,28,29,33] are used to obtain predictions. PSO [30] implemented in Java is used to obtain coefficients of the autoregression model. Finally, Polynomial Regression available in Microsoft Excel is used to obtain predictions in the abscence of chaos.

While conducting experiments over the datasets, different user-defined parameters are tweaked in order to obtain the best performance from the techniques. While training on USD-JPY data set using MLP, the learning rate is 0.6,the momentum rate 0.9 and number of hidden nodes is 10, and using GRNN, the smoothing factor is 0.1144531 and using PSO, number of particles is 50, dimensions are 11, inertia is 0.8, iterations are 40000 and $c_1 = c_2 = 2$ are tweaked. Similarly, while training on USD-GBP data set using MLP, the learning rate is 0.5,the momentum rate 0.7 and number of hidden nodes is 30, and using GRNN, the smoothing factor is 0.0915294 and using PSO, number of particles is 60, dimensions are 16, inertia is 0.6, iterations are 40000 and $c_1 = c_2 = 2$ are tweaked.Similarly, while training on USD-EUR data set using MLP, the learning rate is 0.6,the momentum rate 0.8 and number of hidden nodes is 20, and using GRNN, the smoothing factor is 0.0179688 and using PSO, number of particles is 60, dimensions are 16, inertia is 0.8, iterations are 40000 and $c_1 = c_2 = 2$ are tweaked. Since the performance of machine learning techniques in general and Neural Networks techniques, in particular, is, by and large dataset dependent, we need to tweak parameter values in order to get best results for that Neural Network architecture in a given dataset.Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) are used as performance measures as in (13) and (14) :

$$MSE = \frac{\sum_{t=1}^{k}(y(t) - \dot{y}(t))^2}{k} \tag{13}$$

$$MAPE = \frac{100}{k}\sum_{t=1}^{k}\left|\frac{y(t) - \dot{y}(t)}{y(t)}\right| \tag{14}$$

where k is number of forecasting observations, $y(t)$ is actual observation at time t, ẏ(t) is predicted value at time t.

5 Results and Discussion

In tables 1, 2 and 3, the MSE and MAPE values for both training and test sets after modeling chaos using stand-alone models are presented. In Tables 4, 5 and 6, the MSE and MAPE values for both training and test sets after modeling chaos using hybrid models are presented along with the optimal lag and the optimal embedding dimension. The experiments #1,#2 and #3 are variants of Hybrid Model-1 and #4 is Hybrid Model-2. From Tables 4, 5 and 6, it is observed that the GRNN, GMDH and PSO were extremely effective in modeling the chaos present in the datasets in Stage-1 whereas MLP was ineffective in doing so. The observation is corroborated by the figures 1-6 where the winning hybrid variant and corresponding neural network in stand-alone mode (Stage-1) behaved pretty closely. Further, it is observed that Stage-2 fine tunes the predictions only when chaos is completely modeled in Stage-1.

Moreover, GMDH turned out to be the best technique in Stage-1, for obtaining initial predictions of USD-JPY and USD-GBP datasets, while PSO-based autoregression model turned out to be the best in the case of USD-EUR dataset. The best hybrid models in tables 4,5 and 6 are highlighted in boldface. The variant of Hybrid Model-1, involving GMDH, turned out to be the best hybrid model in the case of both USD-JPY and USD-GBP. whereas the Hybrid Model-2 yielded the best result in the case of USD-EUR. Figures 1-6 depict the predictions of proposed hybrid models for both training and test sets of all datasets along with actual values, MLP, GRNN, GMDH and PSO predictions. It can be easily observed that the proposed hybrid models outperformed the standalone MLP, GRNN, GMDH and PSO.

Table 1. Results of Stand-alone models for USD-JPY data

Model	Training set MSE (MAPE)	Test set MSE (MAPE)
MLP	486.5760 (15.6530)	10.7855 (3.0728)
GRNN	5.7430 (1.6291)	33.4999 (5.7749)
GMDH	0.6581 (0.5064)	0.3560 (0.4793)
PSO	2.4686 (1.0316)	2.0915 (1.3491)

Table 2. Results of Stand-alone models for USD-GBP data

Model	Training set MSE (MAPE)	Test set MSE (MAPE)
MLP	0.0417 (8.7604)	0.0027 (2.6912)
GRNN	0.0001765 (0.6146)	0.000540 (1.1062)
GMDH	0.0001024(0.4324)	0.0000856 (0.4530)
PSO	2.5353 (0.7056)	2.1005 (0.7044)

Table 3. Results of Stand-alone models for USD-EUR data

Model	Training set MSE (MAPE)	Test set MSE (MAPE)
MLP	0.0527 (17.4361)	0.0024 (2.9137)
GRNN	0.00004707 (4.4992)	0.00010758 (0.5918)
GMDH	0.00006252 (0.4880)	0.000064089 (0.4473)
PSO	0.00006741 (0.5115)	0.000073068 (0.4842)

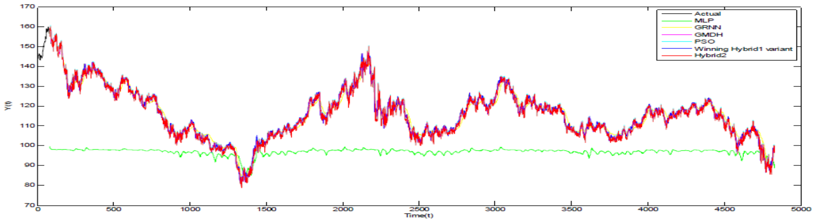


Fig. 1. Predictions of Training set of USD-JPY

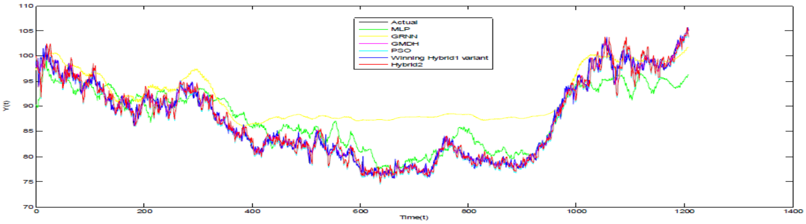


Fig. 2. Predictions of Test set of USD-JPY

Table 4. Results of proposed hybrid models for USD-JPY data

Experi -ment #	Stage #	Technique / Chaos paramters	Training set MSE (MAPE)	Test set MSE (MAPE)
#1	Stage -1 (Chaos present)	MLP ($l_1 = 4, m_1 = 20$)	0.7093 (0.5374)	0.5211 (0.6147)
	Stage-2 (Chaos present)	PSO-based Model ($l_2 = 10, m_2 = 11$)		
#2	Stage-1 (Chaos present)	GRNN ($l_1 = 4, m_1 = 20$)	5.8905 (1.6450)	5.0712 (7.9670)
	Stage-2 (Chaos absent)	2nd degree polynomial regression		
		3rd degree polynomial regression	5.9475 (1.6660)	2.9160 (4.4905)
		4th degree polynomial regression	7.0218 (1.8343)	1.8247 (3.4829)
#3	Stage-1 (Chaos present)	**GMDH** ($l_1 = 4, m_1 = 20$)	0.7314 (0.5487)	0.3552 (0.4757)
	Stage-2 (Chaos absent)	2nd degree polynomial regression		
		3rd degree polynomial regression	**0.7314 (0.5487)**	**0.3548* (0.4757)**
		4th degree polynomial regression	0.7348 (0.5501)	0.3556 (0.4759)
#4	Stage-1 (Chaos present)	PSO-based Model ($l_1 = 4, m_1 = 20$)	2.6083 (1.0825)	1.4666 (1.0676)
	Stage-2 (Chaos absent)	2nd degree polynomial regression		
		3rd degree polynomial regression	3.1846 (0.5147)	1.3127 (0.9866)
		4th degree polynomial regression	3.8139 (1.3870)	0.00007616 (0.4975)

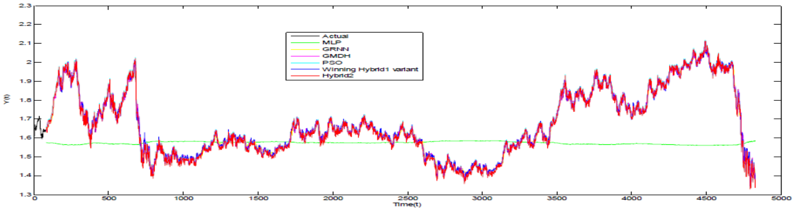


Fig. 3. Predictions of Training set of USD-GBP

Table 5. Results of proposed hybrid models for USD-GBP data

Experi -ment #	Stage #	Technique / Chaos paramters	Training set MSE (MAPE)	Test set MSE (MAPE)
#1	Stage -1 (Chaos present)	MLP ($l_1 = 5, m_1 = 16$)	0.0002266 (0.6717)	0.0001967 (0.6933)
	Stage-2 (Chaos present)	PSO-based Model ($l_2 = 5, m_2 = 16$)		
#2	Stage-1 (Chaos present)	GRNN ($l_1 = 5, m_1 = 16$)	0.0008580 (0.6335)	0.0008841 (1.5039)
	Stage-2 (Chaos absent)	2nd degree polynomial regression		
		3rd degree polynomial regression	0.0001883 (1.6660)	0.001000 (1.6830)
		4th degree polynomial regression	0.0001883 (0.6409)	0.001800 (2.3914)
#3	Stage-1 (Chaos present)	**GMDH** ($l_1 = 5, m_1 = 16$)	**0.0001097 (0.4595)**	**0.000094* (0.4773)**
	Stage-2 (Chaos absent)	**2nd degree polynomial regression**		
		3rd degree polynomial regression	0.0001118 (0.4623)	0.000094 (0.4773)
		4th degree polynomial regression	0.0001097 (0.4595)	0.0001029 (0.4959)
#4	Stage-1 (Chaos present)	PSO-based Model ($l_1 = 5, m_1 = 16$)	0.0002641 (0.7346)	0.0002058 (0.7013)
	Stage-2 (Chaos absent)	2nd degree polynomial regression		
		3rd degree polynomial regression	0.0002550 (0.7114)	0.0002027 (0.9866)
		4th degree polynomial regression	0.0002550 (0.7114)	1.3352 (0.9993)

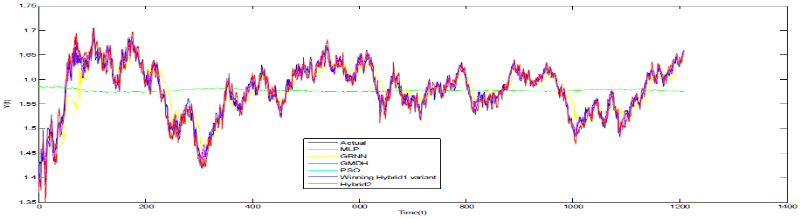


Fig. 4. Predictions of Test set of USD-GBP

Table 6. Results of proposed hybrid models for USD-EUR data

Experi -ment #	Stage #	Technique / Chaos paramters	Training set MSE (MAPE)	Test set MSE (MAPE)
#1	Stage -1 (Chaos present)	MLP ($l_1 = 1, m_1 = 10$)	0.0002328 (0.9916)	0.0001922 (0.7970)
	Stage-2 (Chaos present)	PSO-based Model ($l_2 = 10, m_2 = 16$)		
#2	Stage-1 (Chaos present)	GRNN ($l_1 = 1, m_1 = 10$)	0.0000522 (0.4807)	0.0001245 (0.6445)
	Stage-2 (Chaos absent)	2nd degree polynomial regression		
		3rd degree polynomial regression	0.0000540 (0.4888)	0.0001388 (0.6843)
		4th degree polynomial regression	0.0000574 (0.5069)	0.0001468 (0.7084)
#3	Stage-1 (Chaos present)	GMDH ($l_1 = 1, m_1 = 10$)	0.0000697 (0.5226)	0.0000744 (0.4899)
	Stage-2 (Chaos absent)	2nd degree polynomial regression		
		3rd degree polynomial regression	0.0000697 (0.5226)	0.0000749 (0.4941)
		4th degree polynomial regression	0.0000697 (0.5226)	0.0000749 (0.4941)
#4	Stage-1 (Chaos present)	**PSO-based Model** **($l_1 = 1, m_1 = 10$)**	**0.0000684 (0.5147)**	**0.0000736* (0.4870)**
	Stage-2 (Chaos absent)	**2nd degree polynomial regression**		
		3rd degree polynomial regression	0.0000684 (0.5147)	0.0000736 (0.4870)
		4th degree polynomial regression	0.0000674 (0.7114)	0.00007616 (0.4975)

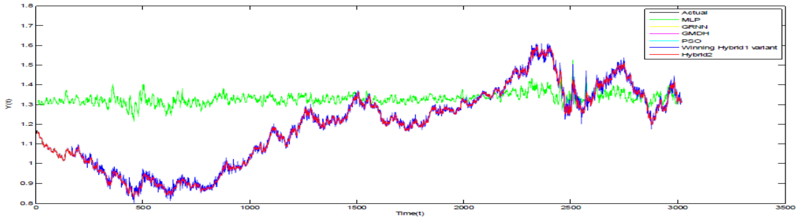


Fig. 5. Predictions of Training set of USD-EUR

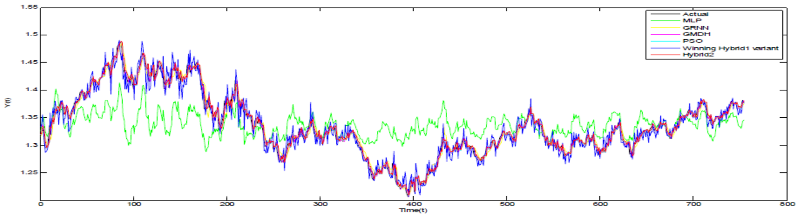


Fig. 6. Predictions of Test set of USD-EUR

6 Conclusion

For predicting FOREX rates, the paper proposes two 2-stage hybrid models comprising chaos theory, various neural network architectures viz. MLP, GRNN and GMDH and PSO or Polynomial Regression. The results of the hybrids in terms of MSE and MAPE on test datasets indicate that the proposed hybrid models outperformed the stand-alone forecasting models: MLP, GRNN, GMDH and PSO. This is the significant outcome of this study. And also, systematic modeling of chaos present in the datasets along with the application of powerful neural networks and PSO for prediction is the single most advantage of the current research. Future directions include applying Multi-objective-PSO and other competing techniques.

References

1. Hoag, A.J., Hoag, J.H.: Introductory Economics. 4th edn. World Scientific Publishing Co. Ptc. Ltd., Singapore (2006)
2. Dhanya, C.T., Nagesh Kumar, D.: Nonlinear ensemble prediction of chaotic daily rainfall. Advances in Water resources 33, 327–347 (2010)
3. Yu, L., Wang, S., Lai, K.K.: Foreign-Exchange-Rate with Artificial Neural Networks. Springer, USA (2007)
4. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. Intl. J. of Forecasting 14, 35–62 (1998)
5. Ozkan, F.: Comparing the forecasting performance of neural network and purchasing power parity: The case of Turkey. Elsevier, Economic Modelling 31, 752–758 (2013)
6. Temizel, T.T., Casey, M.C.: A comparitive study of autoregressive neural network hybrids. Neural Networks 18, 781–789 (2005)
7. Ni, H., Yin, H.: Exchange rate prediction using hybrid neural networks trading indicators. Neurocomputing 72, 2815–2823 (2009)
8. Zhang, G.P.: Time sereis forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50, 159–175 (2003)
9. Zhang, Y.Q., Wan, X.: Statistical fuzzy interval neural networks for currency exchange rate time series prediction. Applied Soft Computing 7, 1149–1156 (2007)
10. Donate, J.P., Cortez, P., Sanchez, G.G., Migue, A.S.: Time sereis forecasting using a weighted cross-validation evolutionary artificial neural network ensemble. NeuroComputing 109, 27–32 (2013)

11. Gheyas, I.A., Smith, L.S.: A novel neural network ensemble architecture for time sereis forecasting. NeuroComputing 74, 3855–3864 (2011)
12. Yu, L., Lai, K.K., Wang, S.: Multistage RBF neural network ensemble learning for exchange rates forecasting. NeuroComputing 71, 3295–3302 (2008)
13. Sermpins, K., Theofilatos, K., Karanthanpoulos, A., Georgopoulos, E.F., Dunis, C.: Forecasting foreign exchange rates with adaptive neural networks using radiabasis functions and particle swarm optimization. European Journal of Operational Research 225, 528–540 (2013)
14. Chang, J.F., Hsieh, P.Y.: Particle Swarm Optimization based on BackPropagation Network Forecasting Exchange Rates. ICIC International 7(12), 6837–6847 (2011)
15. Huang, S.C., Chuang, P.J., Wu, C.F., Lai, H.J.: Chaos-based support vector regressions for exchange rate forecasting. Expert Systems with Applications 37, 8590–8598 (2010)
16. Aladag, C.H., Yolcu, U., Egrioglu, E., Dalar, A.Z.: A new time invariant fuzzy time series forecasting method based on particle swarm optimization. Applied Soft Computing 12, 3291–3299 (2012)
17. Rout, M., Majhi, B., Majhi, R., Panda, G.: Forecasting of currency exchange rates using an adaptive ARMA model with differential based evolution. Journal of King Saud University-Computer and Information Sciences 26(1), 7–18 (2014)
18. Chen, A.S., Leung, M.T.: Regression neural network for error correction in foreign exchange forecasting and trading. Computers and Operations Research 31(7), 1049–1068 (2004)
19. Ince, H., Trafils, T.B.: A Hybrid model for exchange rate prediction. CDecision Support Systems 42, 1054–1062 (2004)
20. Ravi, V., Lal, R., Raj Kiran, N.: Foreign Exchange Rate Prediction using computational Intelligence methods. Int. J. of Computer Information Systems and Industrial Management Applications 4, 659–670 (2012)
21. Hadavandi, E., Ghanbari, A., Naghneh, S.A.: Developing a Time Series Model Based On Particle Swarm Optimization for Gold Price Forecasting. In: IEEE Third Int. Conf. on Business Intelligence and Financial Engineering (2010)
22. Saida, A.B.: Using the Lyapunov exponent as a practical test for noisy chaos (Working paper), http://ssrn.com/abstract=970074
23. http://www.mathworks.in/matlabcentral/fileexchange/22667-chaos-test
24. Akaike, H.: A new Look at the Statistical Model Identification. IEEE Transactions on Automatic Control AC-19(6) (1974)
25. Cao, L.: Practical Method for determining the minimum embedding dimension of a scalar time series. Physica D 110, 43–50 (1997)
26. Rumelhart, G.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation (1). MIT Press, Cambridge (1986)
27. Specht, D.F.: A General Regression Neural Network. IEEE Transactions on Neural Networks 2(6), 568–576 (1991)
28. Ivakhnenko, A.G.: The GMDH: A rival of stochastic approximation. Sov. Autom. Control 3(43) (1968)
29. Farlow, S.J.: Self-Organizing Methods in Modeling: GMDH type Algorithm, Bazel, Marcel Dekker Inc. Newyork (1984)
30. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proc of IEEE International Conference on Neural Network, Perth, Australia, pp. 1942–1948 (1995)
31. http://gretl.sourceforge.net/
32. http://www.mathworks.in/matlabcentral/fileexchange/36935-minimum-embedding-dimension/content/cao_deneme.m

33. http://www.neuroshell.com/
34. Yao, J., Tan, C.L.: A case study on using neural networks to perform technical forecasting of forex. Neurocomputing 12(4), 79–98 (2000)
35. Bates, J.M., Granger, C.W.J.: The combination of forecasts. Oper. Res. Quart. 20, 451–468 (1969)
36. Clemen, R.: combining forecasts: a review and annotated bibliography with discussion. Int. J. Forecast 5, 559–608 (1989)
37. Makridakis, S., Anderson, A., Carbone, R., Fildes, R., Hibdon, M., Lewandowski, R., Newton, J., Parzen, E., Winkler, R.: The accuracy of extraploation (time series) methods: results of a forecasting competition. J. Forecast. 1, 111–153 (1982)
38. Pelikan, E., De Groot, C., Wurtz, D.: Power consumption in West-Bohemia: improved forecasts decorrelating connectionist networks. Neural Network World 2, 710–712 (1992)

Path Planning Using Neighborhood Based Crowding Differential Evolution

Boyang Qu[1,3], Yanping Xu[1], Dongyun Wang[1], Hui Song[2], and Zhigang Shang[2]

[1] School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou, China
[2] School of Electrical Engineering, Zhengzhou University, Zhengzhou, China
[3] School of Information Engineering, Zhengzhou University, Zhengzhou, China
qby1984@hotmail.com, 120828633@qq.com, wdy1964@aliyun.com, hsong320@163.com, zhigang_shang@zzu.edu.cn

Abstract. Path planning problems are known as one of the most important techniques used in robot navigation. The task of path planning is to find several short and collision-free paths. Various optimization algorithms have used to handle path planning problems. Neighborhood based crowding differential evolution (NCDE) is an effective multi-modal optimization algorithm. It is able to locate multiple optima in a single run. In this paper, Bezier curve concept and NCDE are used to solve path planning problems. It is compared with several other methods and the results show that NCDE is able to generate satisfactory solutions. It can provide several alternative optimal paths in one single run for all the tested problems.

Keywords: Evolutionary Computation, Path Planning, Different Evolution, Constrained Optimization.

1 Introduction

In recent years, due to the development of artificial intelligence and electrical integration technologies, robots have been used in various fields such as lunar exploration, navigation, underground exploration, rescue, etc. Mobile robot is an integrated system which combines different functions. Path planning is one of the key technologies used in mobile robot. The performance of the robot highly depends on the path planning method used. Therefore, developing efficient path planning method has attracted many researchers' attention. The aim of path planning is to generate a collision-free path between an initial location and a desired destination in an environment which is full of obstacles. The planned path refers to the optimal or suboptimal trajectory under certain specific conditions, such as the shortest path or the safest path. A good strategy of robot path planning can make a robot fulfill a desired task safely and effectively.

In an environment full of obstacles, the mobile robot must arrange its trajectory to avoid obstacles and find the shortest path when it travels form starting point to target point [1]. In literature, many algorithms have been used to solve the path planning

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 376–383, 2014.
© Springer International Publishing Switzerland 2014

problems, such as self-adjusting fuzzy control algorithm [2], genetic algorithms [3] and [4], ant colony optimization [5] and particle swarm optimization [6]. However, only a few works use differential evolution (DE) to solve this problem [7] and [8]. DE is one of the most powerful stochastic real-parameter optimization algorithms in current use. It is effective in solving single global optimization problems. However, the canonical DE is not suitable to solve multimodal problems or locate multiple peaks in one run. Therefore, a newly developed niching DE algorithm called Neighborhood based Crowding Differential Evolution (NCDE) is used to handle path planning problem in this paper. This algorithm is able to generate multiple optimal paths simultaneously.

The remainder of this paper is organized as follows: Section 2 introduces the definition of Bezier curves and how to use it to solve path planning problem. Section 3 presents the novel DE algorithm which is used to optimize the path. The experimental preparation and simulation results are presented and discussed in section 4 and 5 respectively. Finally, section 6 concludes the paper.

2 Bezier Curves

For designing automobile bodies, French engineer Pierre Bezier invented the Bezier Curves in 1962, which is a new parameter curve [9]. Bezier Curves have become an essential tool in many areas, especially that it has been widely used in computer graphics and animation. Bezier curve is suitable to describe the path because of its space properties. Through controlling the anchor points, different Bezier curves can be obtained. The path planning problem can be transformed into an optimization problem with limited control points to be optimized [10].

2.1 The Definition and Properties of Bezier Curves

In Bezier Curves, *n+1* vertices can define polynomials of degree *n*, and a Bezier Curve of basis polynomial of degree *n* can be expressed using the following equation [11]:

$$P(t) = \sum_{i}^{n} P_i B_{i,n}(t) \qquad t \in [0\ 1]. \tag{1}$$

where P_i represents the coordinates of i_{th} vertex, while $B_{i,n(t)}$ stands for a Bernstein polynomial, which is given as:

$$B_{i,n}(t) = C_n^{\ i} t^i (1-t)^{n-i} \quad (i = 0,1,...,n). \tag{2}$$

$C_n^{\ i}$ is the binomial coefficient. The parameter equation of three times Bezier curve could be obtained by combining formula (1) and (2) as follows:

$$P(t) = P_0(1-t)^3 + 3P_1 t(1-t)^2 + 3P_2 t^2 (1-t) + P_3 t^3. \tag{3}$$

From formula (3), we can observe that the three times Bezier curve starts at t=0 and ends at t=1.

The properties of Bezier curves can be described as follows [12] and [13]:

1. Bezier curve is decided by four points. The curve goes through the first point and the last point. However, the shape of the curve is determined by the two other points.
2. The curve is a straight line if and only if all the control points are in a straight line.
3. The start/end of the curve is tangent to the first/last section of the Bezier polygon (The lines are used to connect the Bezier curve points, and they start from the first point and end at the last point. The lines formed the Bezier polygon.). this property can be described using the following formula:

$$P'(0) = 3\times(P_1 - P_0),\ \ P'(n) = 3\times(P_n - P_{n-1}). \tag{4}$$

Where *P'(0)* is the first derivative of the start point and *P'(n)* is the first derivative of the end point.

3 DE and Neighborhood Based Differential Evolution

3.1 DE

Differential Evolution (DE) is a stochastic search technique developed by Storn and Price in 1995 [14]. It is a simple but efficient search algorithm. It starts with a randomly initialized population. The optimal solution is obtained through a cycle of three stages known as mutation, crossover and selection. These three stages are applied to every solution member x_i in each generation to create a new solution. In each generation, DE employs the mutation operation to produce a mutant vector *vp* associated with each parent *xp* at each generation.

After the mutation phase, the binary (or uniform) crossover operation is applied to each pair of the generated mutant vector and its corresponding parent vector. The process can be represented as:

$$u_{p,i} = \begin{cases} v_{p,i} & \text{if } rand_i \le CR \\ x_{p,i} & otherwise. \end{cases} \tag{5}$$

Where $u_{p,i}$ is the offspring vector. The crossover rate *CR* is a user-specified constant within the range [0,1], which determines what parameter dimensions of $u_{p,i}$ are copied from.

F and *CR* are the two most important control parameters in the DE. These two parameters can significantly influence the optimization performance of the DE. Therefore, to successfully solve an optimization problem, it is generally required to perform a trial-and-error search for the most appropriate values for these parameters [14].

Different from traditional evolutionary algorithms, DE employs the difference of the parameter vectors to explore the objective function landscape. DE has been proved to be one of most powerful evolutionary algorithm for solving the real-valued optimization problems.

3.2 NCDE

The standard DE searches for a global optimum in a D-dimensional hyperspace. It is efficient when searching for single global solution. However, it is often desirable to locate multiple optimal solutions in real world problems. Taking path planning problem as an example, it is always better to find several alternative optimal paths for the decision makers to choose, as the prime objective for different decision makers can be different. Therefore, the neighborhood based crowding differential evolution (NCDE) is applied to solve the path planning problem. Different from canonical DE, vector generations are limited to a number of similar individuals as measured by Euclidean distance. In this way, individuals are evolved towards its nearest optimal point and the possibility of between niche difference vector generation is reduced [15]. The neighborhood concept facilitates multiple convergences, because it allows a higher exploitation of the areas which pilot the moves.

4 Experiment Preparation

4.1 Experiment Setup

For the experiment, Matlab R2008a is used as the programming language and the computer configurations are Intel Pentium® 4 CPU 3.00 GHZ, 4 GB of memory. The DE parameters used are list as below:

$$\textit{Population size}=30,\ F=0.5,\ CR=0.5$$

4.2 Cost Function

In order to obtain a satisfying path, an objective function for the path planning problem need to be defined. Security and length of the path are the two most important criteria for path planning problems. Therefore, these two criteria are used to form the objective function in this work.

(1) Security

Considering that the path cannot intersect with the obstacles, a punishment function f_{safe} can be designed as below:

$$f_{safe}=\begin{cases} 0 & \text{if } d_{\min}>D_{safe} \\ d_{\min} & \text{if } 0\le d_{\min}\le D_{safe}. \end{cases} \tag{6}$$

where d_{min} is the minimum distance between the path and all obstacles. D_{safe} is a predefined security distance.

(2) Length of the path

The length of the curve should be as short as possible. The length function is defined as:

$$f_{len}=L=\int_0^1 \sqrt{(x'(t))^2+(y'(t))^2}\,dt. \tag{7}$$

Where $x(t)$, $y(t)$ are the coordinates of points.

(3) The overall objective function
Considering the two objectives above weight coefficient *a,* the final cost function can be defined as:

$$f = f_{len} + af_{safe}. \tag{8}$$

where *a* is the weight factor to balance the two objectives and it is chosen as 1000 in this experiment.

5 Simulation Results

To assess the performance of the proposed algorithm, four predefined path planning problems are used. The results are plotted in Fig.1. The green circles in the figure mean the dangerous areas around the obstacles. The red path describes the best path while the black paths illustrate the possible paths for every problem. Four algorithms are tested on these problems with two Bezier curves (*n*=2, *D*=4 *n*=8):

1. PSO+: Classical Particle Swarm Optimizer with crossover operator
2. DMS-PSO+: Dynamic Multi-Swarm Optimizer with crossover operator
3. DE: Classical Differential Evolution
4. NCDE: Neighborhood Based Crowding Differential Evolution

The max fitness evaluation is 20000 for all algorithms. The population size of PSO with crossover is set to 30 and the number of sub-swarms of DMS-PSO with

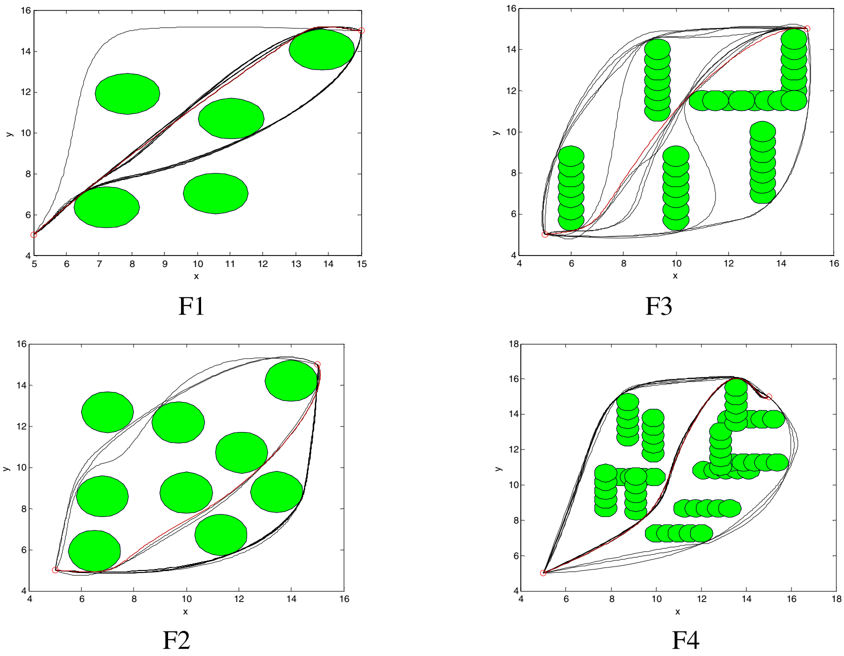


Fig. 1. Landscapes of the test problems

crossover is set to 10 (with 3 particles in each sub-swarm). All performances are calculated and averaged over 25 runs with the random initialization. The results are presented in Table 1. And to evaluate the difference of the two algorithms, Nonparametric statistical method ttest is used. *h*=1 indicates a rejection of the null hypothesis at the 5% significance level. *h*=0 indicates a failure to reject the null hypothesis at the 5% significance level.

In Fig.1, F1, F2, F3 and F4 show that the distributions of the solutions achieved by NCDE for each test problem, where the red line describes the best path. We can observer that it is easy for NCDE to find the best suitable path for every problem as well as other alternative satisfying paths which are not around the best solution at the same time.

Table 1. Results for F1-F4

Problems		PSO+	DMS-PSO+	DE	NCDE
F1	Mean	14.7698	14.7465	14.8774	**14.7237**
	Std.	0.0016	0.0006	0.1717	0.0000
	Min	14.7007	14.7290	14.6842	14.7000
	Max	14.8418	14.7797	15.9563	14.7404
	h	1	1	1	0
F2	Mean	15.9863	15.8101	16.5603	**15.6768**
	Std.	0.5050	0.1726	0.9181	0.1010
	Min	15.6221	15.0865	15.2764	15.3144
	Max	17.3518	17.2887	17.5268	16.7618
	h	1	0	1	0
F3	Mean	17.0764	16.8388	17.4209	**16.6135**
	Std.	0.2402	0.2563	0.1594	0.1398
	Min	16.4046	16.3139	16.2838	16.3900
	Max	18.1084	17.8895	17.8122	17.0463
	h	1	1	1	0
F4	Mean	15.6119	15.1971	16.5413	**14.9887**
	Std.	0.5745	0.4369	0.1743	0.2086
	Min	14.8479	14.8024	14.8028	14.7138
	Max	16.6730	16.6179	17.6099	16.6336
	h	1	1	1	0

In [16], compared with PSO and DMS-PSO, PSO+ and DMS-PSO+ perform better respectively. For every problem, every algorithm is compared with the best algorithm with ttest. From the table, some conclusions could be made as follows:

1. DMS-PSO+ performs better than PSO+.
2. NCDE outperforms DE on all the four problems.
3. NCDE performs best among all algorithms on mean value.

4. Except for problem 2, NCDE is better than the other three algorithms obviously which can be seen from the result of ttest. The result of ttest on problem 2 is no difference between NCDE and DMS-PSO+, which means that the result of NCDE accepts DMS-PSO+ at the 5% significance level. However, from the mean value we could know that NCDE has a smaller value which means NCDE has a stable ability to find good solutions during the process of search compared with DMS-PSO+.

In the process of searching, NCDE has a better local search ability, which makes multi-paths possible, so various satisfied paths could be acquired for every problem in this task. From the above four points we can know that, NCDE performs betters.

6 Conclusion

In this work, Bezier curves and neighborhood based crowding differential evolution algorithm are used to tackle path planning problem. To assess the performance of the neighborhood based crowding differential evolution algorithm in solving path planning problem, four different path problems are tested. The experiments show that neighborhood based crowding differential evolution is effective in solving all four problems. In future work, dynamic environment and constraints will be added to increase the complexity of the path planning problems. High order Bezier curves will also be used to improve the quality of the solutions.

Acknowledgments. This work was supported in part by National Natural Science Foundation of China (61305080, U1304602), Postdoctoral Science Foundation of China (Grants 20100480859, 2014M552013), Specialized Research Fund for the Doctoral Program of Higher Education (20114101110005), Scientific and Technological Project of Henan Province (132102210521, 122300410264), and Key Foundation of Henan Educational Committee (14A410001).

References

1. Chakraborty, J., Konar, A., Chakraborty, A.K., Jain, L.C.: Distributed Cooperative Multi–Robot Path Planning Using Differential Evolution. In: 2008 IEEE Congress on Evolutionary Computation, CEC 2008, Hong Kong, China, pp. 718–725 (2008)
2. Wu, C.D., Zhang, Y., Li, M.X.: A Rough Set GA-based Hybrid Method for Robot Path Planning. Journal of International Automation and Computing, 29–34 (2006)
3. Sugihara, K., Smith, J.: Genetic algorithms for adaptive motion planning of an autonomous mobile robot. In: Proc. of IEEE Intl. Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, USA, pp. 138–143 (1997)
4. ALtaharwa, S.A., Alweshah, M.: A mobile robot path planning using genetic algorithm in static environment. Journal of Computer Science 4, 341–344 (2008)
5. Bell, J.E., McMullen, P.R.: Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics, 41–48 (2004)

6. Saska, M., Macas, M., Preucil, L., Lhotska, L.: Robot path planning using particle swarm optimization of ferguson splines. In: IEEE Conference on Emerging Technologies and Factory Automation, Prague, pp. 833–839 (2006)
7. Jayasree, C., Amit, K., Jain, L.C., Chakraborty, U.K.: Cooperative multi-robot path planning using differential evolution. Journal of Intelligent and Fuzzy Systems, 13–27 (2009)
8. Mo, H.W., Li, Z.Z.: Bio-geography based differential evolution for robot path planning. In: 2012 IEEE International Conference on Information and Automation, ICIA 2012, pp. 1–6. Inner Mongolia, China (2012)
9. Bashir, Z.A., Hawary, M.E.: Short-term Load Forecasting using Artificial Neural Network based on Particle Swarm Optimization Algorithm. In: 2007 Canadian Conference on Electrical and Computer Engineering, CCECD, Canadian, pp. 272–275 (2007)
10. Ho, Y.J., Liu, J.S.: Collision-free Curvature-bounded Smooth Path Planning using Composite Bezier Curve based on Voronoi Diagram. In: Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA, pp. 463–468 (2009)
11. Yang, L.Q.: Path Planning Algorithm for Mobile Robot Obstacle Avoidance Adopting Bezier Curve Based on Genetic Algorithm. In: Control and Decision Conference, pp. 3286–3289 (2008)
12. Liu, H.G., Qin, G.L.: A Bezier Curve Based on The Military Arrow Mark Realized. Ordnance of Sichuan 30, 67–68 (2009)
13. Gao, S., Zhang, Z.Y., Cao, C.G.: Particle Swarm Algorithm for The Shortest Bezier Curve. In: International Workshop on Intelligent Systems and Applications, pp. 1–4 (2009)
14. Storn, R., Price, K.: Differential Evolution—a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Journal of Global Optimization 11, 22–25 (1995)
15. Qu, B.Y., Suganthan, P.N., Liang, J.J.: Differential Evolution with Neighborhood Mutation for Multimodal Optimization. IEEE Transactions on Evolutionary Computation, 601–614 (2012)
16. Liang, J.-J., Song, H., Qu, B.-Y., Mao, X.-B.: Path Planning Based on Dynamic Multi-Swarm Particle Swarm Optimizer with Crossover. In: Huang, D.-S., Ma, J., Jo, K.-H., Gromiha, M.M. (eds.) ICIC 2012. LNCS, vol. 7390, pp. 159–166. Springer, Heidelberg (2012)

Neural Network Based on Dynamic Multi-swarm Particle Swarm Optimizer for Ultra-Short-Term Load Forecasting

Jane Jing Liang[1], Hui Song[1], Boyang Qu[1,2], Wei Liu[3], and Alex Kai Qin[4]

[1] School of Electrical Engineering, Zhengzhou Univerisity, China
[2] School of Electric and Information Engineering, Zhongyuan University of Technology, China
[3] State Grid Henan Economic Research Institute, Zhengzhou, China
[4] School of Computer Science and Information Technology RMIT University, Melbourne 3001, Victoria, Australia
LIANGJING@zzu.edu.cn, qby1984@hotmail.com, liuwei830610@163.com, {hsong320,kai.qin}@rmit.edu.au

Abstract. Ultra-Short-Term Load Forecasting plays an important role in Power Load Forecasting. Back Propagation Neural Network(BPNN) has become one of the most commonly used methods in Power System Ultra-Short-Term Load Forecasting for its ability of computing complex samples and training large-scale samples. However, traditional BPNN algorithm needs to set up a large amount of network training parameters, and it is easy to be trapped into local optima. A new algorithm which is Neural Network based on Dynamic Multi-Swarm Particle Swarm Optimizer (DMSPSO-NN) is proposed for Ultra-Short-Term Load Forecasting in this paper. DMSPSO-NN overcomes the shortage of traditional BPNN and has a good global search and higher accuracy which shows that it is suitable to be used for Ultra-Short-Term Load Forecasting.

Keywords: Ultra-Short-Term Load Forecasting; Back Propagation Neural Network; Dynamic Multi-Swarm Particle Swarm Optimizer

1 Introduction

Power load forecasting is an indispensable part for managing and researching power system, and it can make the full use of electricity and ease the conflict between supply and demand based on the analysis of the existing electric energy [1]. Power system load forecasting method based on electric power, economic, social and meteorological factors and so on. According to the time length of the prediction, power load forecasting can be classified as ultra-short-term load forecasting, short-term load forecasting, medium long-term load forecasting and long-term load forecasting. In terms of power system dispatching and management, ultra-short-term load forecasting which varies from an hour to a week is the most important. Accurate ultra-short-term load forecasting is very important in maintaining ultra-short-term analysis for electric power, power exchange, trading evaluation as well as the analysis of the network function, security and trend, the safety strategy of reducing load and so on[2].

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 384–391, 2014.
© Springer International Publishing Switzerland 2014

In recent years, there are various methods to solve this problem, such as Expert Systems(ES)[3], Support Vector Machines(SVM)[4][5], Back Propagation Neural Network(BPNN)[6][7] and so on. The main drawback of the ES is that it learns nothing from the environment and has ambiguous relationship between rules, low efficiency and adaptability. The main disadvantage of SVM is difficult to achieve large-scale training samples and solve multi-classification problems. BPNN is widely used in power load forecasting in recent years, to calculate large-scale complex training samples as well as to slow the speed of convergence during training process.

With the development of Evolutionary Algorithm (EA), some researchers have found its advantage in handling large-scale, non-differentiable and complex multi-mode problem without any information about optimized problems for its global convergence ability and strong robustness. EA can optimize the weight, structure and learning rules of NN by searching for optimal solutions in search space with the help of evolutionary strategy, genetic algorithm or evolutionary programming. Genetic Algorithm (GA) is the most widely used in EA because it can deal with many complex problems. However, GA is easy to be trapped into local optimum and the process is difficult to control when GA is used to train NN.

Recently, the emergence of Swarm Intelligence such as Particle Swarm Optimizer (PSO) has overcome the drawbacks of EA. Compared with other algorithms, Particle Swarm Optimizer is simple, easy to realize, and need less parameter to adjust, making it an effective optimal tool. So PSO has been used to optimize NN widely such as power load forecasting [8], Fault Diagnosis of Power Transformer [9], Reservoir Parameter Dynamic Prediction [10], Modeling and Simulation of Screw Axis [11]and so on[12]. However, the traditional PSO can’t maintain the diversity of particles and is difficult to reach global optimum. So Dynamic Multi-Swarm Particle Swarm Optimizer (DMSPSO) which not only overcomes the drawback of the Particle Swarm Optimizer, but also has strong global search ability, is proposed to optimize NN in this paper. The result shows DMSPSO is easy to find global optimum when it is used to optimize NN.

The rest of this paper is organized as follows. Section II gives a brief introduction about the basic Particle Swarm Optimizer and describes the search process of the Dynamic Multi-Swarm Particle Swarm Optimizer. The Back Propagation Neural Network and Neural Network Based on Dynamic Multi-Swarm Particle Swarm Optimizer model employed in this work are described in detail in Section III. Section IV introduces the experimental setup and presents the results. Conclusions and future work are given in Section V.

2 Dynamic Multi-swarm Particle Swarm Optimizer

Particle Swarm Optimizer was proposed by Kennedy and Eberhart in 1995[13][14]. The basic idea of PSO simulates the behavior of flying birds. In PSO, each bird is regarded as a potential solution in search space which is called a “particle”. There exists a fitness value in each particle obtained by the fitness function. Each particle adjusts the distance and its flying direction according to its velocity. The model of PSO is shown as:

$$V_i^d = \omega * V_i^d + c_1 * rand1_i^d * (pbest_i^d - X_i^d) + c_2 * rand2_i^d * (gbest^d - x_i^d) \tag{1}$$

$$X_i^d = X_i^d + V_i^d \tag{2}$$

Where, ω is the inertia weight and the range is [0.4 0.9]; c_1 and c_2 are balance factors which are set 2.05 in general; *rand* is a random number in [0, 1].The weakness of standard PSO is premature and easy to be trapped into local optimum.

Dynamic Multi-Swarm Particle Swarm Optimizer (DMS-PSO) is developed from local Particle Swarm Optimizer, where neighborhood structure is used in a small population [15][16]. In order to increase the distribution of population and accelerate the speed of convergence, the entire population is divided into sub-swarms equally in DMS-PSO and each sub- swarm searches in space with its own particles. The population is regrouped randomly every *L* generation (*L* is called regroup period), then the population starts the search with new topology structure. Due to this method, the information obtained from sub-swarms exchanges among them, and the diversity of the population is also increased. The updating formula is given as follows:

$$\begin{aligned} & V_i^d \leftarrow \omega * V_i^d + c_1 * rand1_i^d * (pbest_i^d - X_i^d) + c_2 * rand2_i^d * (lbest_k^d - X_i^d) \\ & V_i^d = \min(V_{\max}^d, \max(-V_{\max}^d, V_i^d)) \\ & X_i^d \leftarrow X_i^d + V_i^d \end{aligned} \tag{3}$$

where, V_i^d represents the velocity of i^{th} particle in dimension *d*; X_i^d is the position of i^{th} particle in dimension *d*. $lbest_k^d$ is the position of local optimum in dimension *d* of k^{th} sub-swarm; $pbest_i^d$ is the best personal position in dimension *d* of i^{th} particle.

3 Back Propagation Neural Network Based on Dynamic Multi-swarm Particle Swarm Optimizer

3.1 Back Propagation Neural Network

Back Propagation Neural Network can solve learning problems for connecting weights between hidden units in multi-layers network, so it has become one of the most important modal of Artificial Neural Network. The BPNN is made up of three layers: input layer, hidden layer and output layer. In order to get satisfied forecast, backward transmission error and error correction methods are used to adjust the network parameters (weights and threshold)[17].

Input layer neurons are responsible for receiving input information from the outside world, and transmitting to the middle layer neurons (which is hidden layer). The hidden layer is the internal information processing layer, which is responsible for transforming information. According to the requirements of changed information, hidden layer can be designed for single hidden layer or several hidden layers[18][19]. The structure of the BP Neural Network is shown as follows:

Where, $x_0, x_1, ..., x_j ..., x_n$ is the input value of BPNN, $o_1, o_2, ..., o_k ..., o_l$ is predictive value, v_{ij} and w_{jk} $(i = 1, 2, ..., n, j = 1, 2, ..., m, k = 1, 2, ..., l)$ are the input and output weights of BPNN respectively. If the input node is *n* and the output node is *l*, BPNN expresses the mapping relationship from *n* independent input variables to *l* independent output variables. BPNN acquires associative memory and ability to predict through training the network.

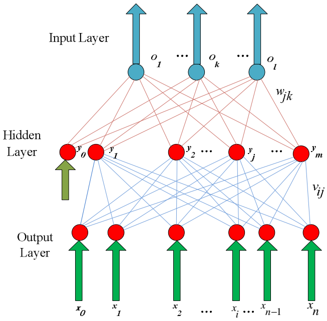


Fig. 1. Structure of BPNN

3.2 The Process of Neural Network Based on Dynamic Multi-swarm Particle Swarm Optimizer

Dynamic Multi-Swarm Particle Swarm Optimizer is used to optimize the input weights, input thresholds, output weights output thresholds of NN. The process of DMSPSO which is used to train the parameter of NN includes the following steps:

a. Ensure the parameters of NN

According to the input and output(*x*, *o*) determine the number of input layer nodes *n*, hidden layer nodes *m* and output layer nodes *l*. Then the total dimension can be calculated through the following formula:

$$D=n*m+m+m*l+l \tag{4}$$

Sigmoid is used as the excitation function.

$$f(x) = \frac{1}{1+e^{-x}} \tag{5}$$

At the same time, we normalize the original data input and output samples in order to increase the learning efficiency of NN.

$$\begin{aligned} x_k^{new} &= \frac{0.1 + 0.8 \times (x_k^{old} - \min x_k^{old})}{\max x_k^{old} - \min x_k^{old}} \\ o_k^{new} &= \frac{0.1 + 0.8 \times (o_k^{old} - \min o_k^{old})}{\max o_k^{old} - \min o_k^{old}} \end{aligned} \tag{6}$$

where, K=1, 2,…m is the number of samples. x_k^{old} and y_k^{old}, x_k^{new} and y_k^{new} represent the input and output of the network which are unprocessed and processed respectively.

b. Initialize the parameters

Initialize the weight between input layer and hidden layer v_{ij} and between hidden layer and output layer w_{jk}, hidden threshold a and output threshold b, regroup period L, number of Sub-warms P, population of every sub-swarm ps and every particle's velocity.

c. Calculate fitness value

Firstly, calculate the output of hidden layer H according to the input data x, weights v_{ij} between input and hidden layer and threshold a.

$$H_j = f(\sum_{i=1}^{n} v_{ij}x_i - a_j) \quad j = 1,2,...,m \tag{7}$$

m is the nodes of hidden layer and f is the excitation function of hidden layer.

Secondly, according to the output of hidden layer H, weights w_{jk} between hidden layer and output layer, threshold b calculate o which is the predicted output of NN.

$$o_k = \sum_{j=1}^{m} H_j w_{jk} - a_k \qquad k = 1,2,...,l \tag{8}$$

Then, according to predicted output o and expected output p calculate predicted error e of neural network.

$$e_t^k = P_t^k - o_t^k \qquad k = 1,2,...,l \ \ t = 1,2,...ps * P \tag{9}$$

The fitness function is set as:

$$fit(t) = \sum_{k=1}^{l} abs(e_t^k) \tag{10}$$

Calculate every particle's fitness value by (10), search for each particle's best position achieved so far.

d. Get sub-swarm

Divide the whole population into sub-swarms equally and get the local optimum $lbest_P$ of every sub-swarm according to the idea of DMSPSO.

e. Update

Update every particle's position and velocity, and then enter into *b*.

f. Judge the times of loop

If the iteration of regroup is satisfied, all the sub-swarms will be regrouped again.

g. If iteration ends, stop, else return *e*.

4 Experimental Results

This data of experiment is got through the system of monitoring and analyzing key power industry. Two models which are BPNN and DMSPSO-NN are used to predict the power load about one day which is based on the data achieved 29 day previously.

According to the recorded load data of October, November, December, precious 29 days in December, related days (related days of everyday are 20 days) and related sampling points(related points for every sampling point are 42 points) of these 29 days are regarded as train data to predict the load of 30th, December. The prediction of each algorithm contains the whole day's prediction and every point's prediction during one day. They are defined as follows:

BPNN1: The data of the whole day is regarded as a whole prediction output (the dimension of the output is 96) and the predicted mode is BPNN.

DMSPSO-NN1: The data of the whole day is regarded as a whole prediction output and the predicted mode is DMSPSO-NN.

BPNN2: Every point in one day is regarded as a whole prediction output (the dimension of the output is 1) and the predicted mode is BPNN.

DMSPSO-BPNN2: Every point in one day is regarded as a whole prediction output and the predicted mode is DMSPSO-NN.

The parameters used in this task are set as follows:

The structure of NN1: 62 input nodes, 20 hidden nodes, 96 output nodes

The structure of NN2: 62 input nodes, 20 hidden nodes, 1output node

Dimension: 1281

Population size: 30

Number of sub-swarms:5

MaxFES (Maximum Fitness Evaluation): 40000

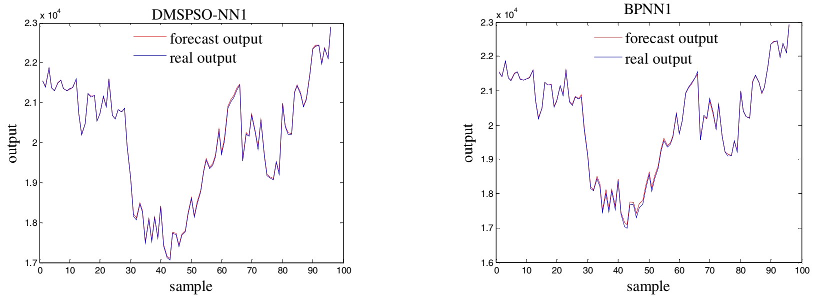


Fig. 2. The whole day as a sample to be optimized

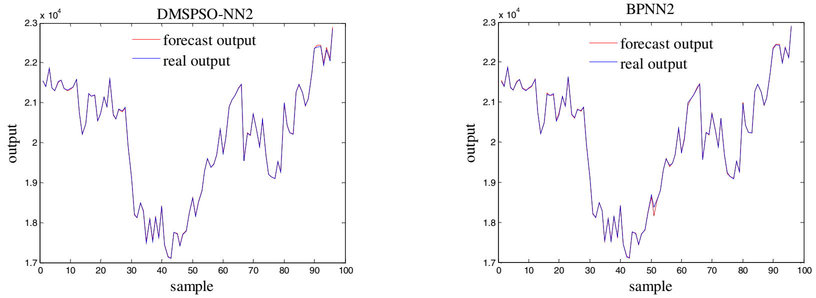


Fig. 3. Every point in one day as a sample to be optimized

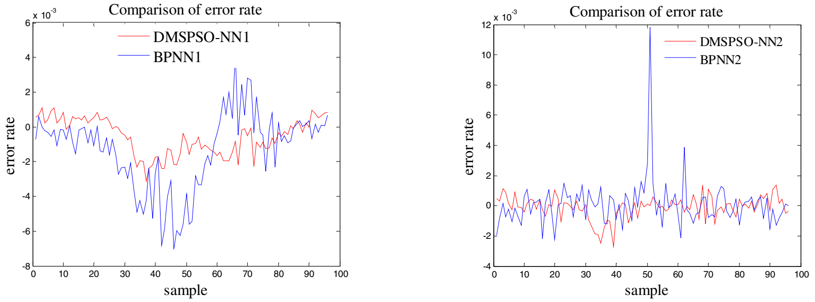


Fig. 4. Comparison of error rate under different conditions

Table 1. The error rate which is below 0.3%

	BPNN1	DMSPSO-NN1	BPNN2	DMSPSO-NN2
Error rate	0.9796	1	0.9892	1

Some conclusions could be made from the result as follows:

- DMSPSO-NN performs better than BPNN which shows that dynamic sub-swarm has improve the diversity of the solutions;
- To predict every point has a better result than regard the whole day as a sample;
- The accuracy of DMSPSO-NN is higher than PSO.

From these three points, we could know that DMSPSO has a better global search ability which helps it avoid being trapped into local optimum. After combining the NN with the DMS-PSO, the prediction is much more better which shows that DMSPSO-NN is suitable for Ultra-Short-Term Load Forecasting.

5 Conclusions

In this paper, an improved PSO(DMSPSO) is employed to optimize NN. What's more, two methods are used to test the property of DMSPSO-NN. The result shows that DMSPSO-NN has better global search ability when it is used in Ultra-Short-Term Load Forecasting problem. The result of error rate also makes us know that when every point is regared as training sample, the result is much more better. In the future, more algorithms will be used to predict load, and the more better and fast algorithm will be used for online forcast.

Acknowledgment. This research is partially supported by The Second Batch Project of Science and Technology of Henan Electric Power Company in 2013 (5217L0135029)and National Natural Science Foundation of China (61305080) and Postdoctoral Science Foundation of China (20100480859) and Specialized Research Fund for the Doctoral Program of Higher Education (20114101110005) and Postdoctoral Science Foundation Grand (2014M552013).

References

1. Gross, G., Galiana, F.D.: Short-term load forecasting. Proceedings of the IEEE 75(12), 1558–1573 (1987)
2. Abdel-Aal, R.E.: Improving electric load forecasts using network committees. Electric Power Systems Research 74(1), 83–94 (2005)
3. Hsu, Y.Y.: Fuzzy expert systems: an application to short-term load forecasting. IEE Proceedings C: Generation Transmission and Distribution 139(6), 471–477 (1992)
4. Francis, E., Tay, H.: Application of support vector machines in financial time series forecasting. Omega 29(4), 309–317 (2001)
5. Rüping, S.: Incremental learning with support vector machines. In: Proceedings-IEEE International Conference on Data Mining, ICDM, pp. 641–642 (2001)
6. Luo, X., Zhou, Y.H., Zhou, H.: Forecasting the daily load based on ANN. In: Control theory and Application, pp. 1–4 (2007)
7. Kim, C.I., Yu, I.K.: Kohonen neural network and transform based approach to short-term load forecasting. Elect. Elecr. Power Syst. Res. 63(3), 169–176 (2002)
8. Hu, J., Zeng, X.: A hybrid PSO-BP algorithm and its application. In: 2010 Sixth International Conference on Natural Computation (ICNC), vol. 5, pp. 2520–2523. IEEE (2010)
9. Li, H., Yang, D., Ren, Z.: Based on PSO-BP network algorithm for fault diagnosis of power transformer. In: 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE), vol. 4, pp. 484–487. IEEE (2010)
10. Zhang, L., Ma, J., Wang, Y.: PSO-BP neural network in reservoir parameter dynamic prediction. In: 2011 Seventh International Conference on Computational Intelligence and Security (CIS), pp. 123–126. IEEE (2011)
11. Zhang, P.Y., Sheng, Y.L., Wan, L.L.: Modeling and simulation of screw axis based on PSO-BP neural network and orthogonal experiment. In: Second International Symposium on Computational Intelligence and Design, ISCID 2009, pp. 272–275. IEEE (2009)
12. Ren, J., Yang, S.: An Improved PSO-BP Network Model. In: 2010 International Symposium on Information Science and Engineering (ISISE), pp. 426–429. IEEE (2010)
13. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
14. Shi, Y., Eberhart, R.C.: A Modified Particle Swarm Optimizer. In: The IEEE Congress on Evolutionary Computation, pp. 69–73 (1998)
15. Liang, J.J., Suganthan, P.N.: Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search. In: IEEE Congress on Evolutionary Computation, vol. 1, pp. 522–528 (2005)
16. Liang, J.J., Suganthan, P.N.: Dynamic Multi-Swarm Particle Swarm Optimizer. In: IEEE International Swarm Intelligence Symposium, pp. 124-129 (2005)
17. Gavrilas, M., Ciutea, I., Tanasa, C.: Short-term Load Forecasting with Artificial Neural Network Models. Proceedings of IEE CIRED 25(12), 28–31 (2001)
18. Wang, Q., Zhou, B., Li, Z.: Forecasting of short-term load based on fuzzy clustering and improved BP algorithm. In: International Conference on Electrical and Control Engineering (ICECE), pp. 4519–4522. IEEE (2011)
19. Yao, S.J., Song, Y.H., Zhang, L.Z., Cheng, X.Y.: Wavelet transform and neural networks for short-term electrical load forecasting. Energy Conversion and Management 41(18), 1975–1988 (2000)

Dynamic Differential Evolution for Emergency Evacuation Optimization

Shuzhen Wan

School of Computer and Information Technology,
Three gorges University of ChinaYichang, China
wanshuzhen@163.com

Abstract. Emergency evacuation in public places has become the hot area of research in recent years. Emergency evacuation route assignment is one of the complex dynamic optimization problems in emergency evaluation. This paper proposed the modified dynamic differential evolution algorithm and studied the emergency evacuation, then applied the multi-strategy dynamic differential evolution for emergency evacuation route assignment in public places. We use the Wuhan Sport Center in Wuhan China as the experiment scenario to test the performance of the proposed algorithm. The results show that the proposed algorithm can effectively solve the complex emergency evacuation route assignment problem.

Keywords: dynamic differential evolution, emergency evacuation assignment, dynamic optimization.

1 Introduction

Evolutionary Algorithms (EAs) have been applied to solve dynamic optimization problems. Many real world optimization problems are dynamic optimization problems (DOPs)[1-3]. Dynamic optimization algorithms are different from the traditional one which focuses on the static conditions, while the former one admits that both the problems and the solutions may be changed in time.

More and more scholars pay special attention to the study of applying the EAs to solve dynamic optimization problems in recent years. Several approaches have been developed in EAs to address DOPs, such as maintaining diversity during the run via random immigrants [4, 5], using memory to store and reuse useful information[3, 4, 6], multi-population approach[1, 7, 8], increasing diversity after a change[2, 9, 10].

Although all the above approaches are effectively for solving some dynamic optimization problems, there are some points of criticism, Such as: how to trace the different changing optimums in searching space. We use the prediction based multi-strategy differential evolution algorithm [11]to meet the challenges. We use a hybrid method that combines population core based multi-population strategy and prediction strategy and new local search scheme to enhance differential evolution (DE) performance for solving DOPs. The population core based multi-population strategy

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 392–400, 2014.
© Springer International Publishing Switzerland 2014

is useful to maintain the diversity of the population by using the multi-population and population core concept. The prediction strategy is helpful to rapidly adapt to the dynamic environment by using the prediction area. The local search scheme is useful to improve the searching accuracy by suing the new chaotic local search method. Experimental results on the moving peaks benchmark show that the proposed schemes enhance the performance of DE in the dynamic environments. In this paper, we apply the proposed algorithm to solve the real world dynamic optimization problems-emergency evacuation route assignment optimization problem.

Emergency evacuation is an important issue for the large public spaces. There have been a lot of literatures focus on the problem of emergency evacuation[12, 13] Emergency evacuation is the study of how to evacuate the people from dangerous locations to safe places in hurry.[3] Evacuation planning is a very complex problem which needs to satisfy the consideration of many aspects. From the perspective operation research, evacuation planning is a dynamic optimization problem.

Studies on evacuation in buildings mainly focus on the simulation [13] some other researches focus on the evacuation of the traffic network [14].

This paper applied the proposed dynamic differential evolution algorithm to solve the emergency evacuation route assignment optimization problem and achieved good results comparing with other algorithms.

The remaining sections of this paper are organized as follows: Section II describes the proposed algorithm. Section III details the emergency evacuation route assignment model. Section IV presents the applying of the proposed algorithm on the emergency evacuation route assignment. Section V introduces the experimental study and discussion based on the experimental results. Finally, Section VI draws conclusions.

2 Prediction based Multi-Strategy Differential Evolution

In order to enhance the performance of differential evolution for dynamic optimization problems, we use the population core based multi-population strategy to maintain the population diversity and avoid the premature convergence. And the new local search scheme is employed in our algorithm to improve the exploitation precision. Moreover, the prediction strategy is utilized to trace the optima when the environment changes. Some of the details of the three schemes will be given as follows. This algorithm is also detailed in our previous work [11].

2.1 Multi-Population Strategy

The multi-population method can be used to maintain multiple populations on different optima when dealing with DOPs. To make this method work, the search space is divided into several local search spaces, and each local search space can contain more than one peak. Each subpopulation covers one local search space and searches the local optimum in it. However, there are some important issues to be considered such as how to divide the local search space, how to decide the number of the subpopulation, how to generate the subpopulations. In this paper, we use the hierarchical clustering method to achieve the goal of dividing subpopulations. The population core concept is used also to maintain the diversity of populations after the dividing.

2.2 Tent Map Based Local Search

Because of the ergodicity and randomicity, a chaotic system changes randomly, but eventually goes through every state if the time duration is long enough. This characteristic of chaotic systems can be utilized to build up a search operator for optimizing objective functions. To improve the local search ability, we introduce the tent map based local search method in the proposed algorithm. The tent map based local search scheme is described in [11].

2.3 Prediction Strategy

Tracing the new optima in the changed environment is the most important issue for the algorithms which are applied to deal with DOPs. The changes of environments might be very complex, and the width, the height and the location of the optima can all be changed. So, how to find the optima accurately and quickly have become the challenges for all the dynamic algorithms. We propose the prediction strategy to predict the location of the new optima before the environments are changed. The prediction strategy is based on the idea that we create the prediction areas before the changes occur, if the prediction areas just cover the areas containing the new optima, the algorithm will quickly find the optima, if not, the searching speed can also be improved due to the prediction areas are nearer to the new optima, and the probability of finding the optima will be increased. The details of the prediction scheme are shown in [11].

3 Evacuation Route Assignment Model

In this paper, we utilize the model [15] as the base of our evacuation route assignment model. In literature [15], three optimization objects are presented: minimizing the evacuation time, minimizing the total travel distance of all evacuees and minimizing the congestion during the evacuation process. The evacuation time is the most important in the three objects because if all evacuees can be evacuated within the set time, the evacuation route assignment will be effective. Thus, in this model, we select the evacuation time as the single optimization object, as the same, we take the congestion as the constraints. If the congestion degree during the evacuation process exceeds a threshold value which is defined according to the extent of the evacuation environment can afford, the evacuation route assignment proves infeasible. This object is defined as follows:

$$\begin{aligned} &\min T = \max\{t_{D1}, t_{D2}, \dots t_{Dr}, \dots t_{Dn}\}, r \in M \\ &s.t. f_{\max} < f_{pre} \end{aligned} \tag{1}$$

Where, $f_{\max}$ is the max congestion of the evacuation passageway in the system during the evacuation, f_{pre} is threshold of the congestion. The congestion in the evacuation passageway during the evacuation is changed with time, so the evacuation route assignment is the dynamic optimization problem.

4 Modified Dynamic DE for Evacuation Route Assignment

4.1 Frame of the Proposed Algorithm for Evacuation Route Assignment

In this paper, we apply the prediction based multi-strategy differential evolution algorithm [11]to solve the evacuation route assignment optimization problem. We employ the k-shortest paths algorithm [16] to initialize the evacuation routes. The frame of the proposed algorithm for evacuation route assignment is detailed as follows:

Step 1. Initialize the evacuation routes with k-shortest paths algorithm.

Step 2. Initialize the population randomly in the searching space and the archive for prediction strategy

Step 3. Use the clustering method to create the subpopulations

Step 4. Apply DE operators (mutation, crossover and selection) on each individual in each subpopulation.

Step 5. Apply the diversity control operator to maintain the population diversity.

Step 6. If optimization environments are changed, Set the prediction zone then go back to Step 3. Else, goto next step.

Step 7. If a termination condition is met then goto end, else go back to Step 4.

4.2 Frame of the Proposed Algorithm for Evacuation Route Assignment

The goal of the evacuation route assignment is to construct an effective and safe evacuation scheme to guide the evacuees evacuating from the dangerous orderly. Considering the complexity of the evacuation route assignment, we adopt the sorting based precoding method to encode each individual.

Firstly, because the evacuees often select the shortest paths while evacuating, so we apply k-shortest paths algorithm to initialize the paths from the starting point to the destination of evacuation in the evacuation zone.

Next, encoding each route with an integer and then we assign the coded paths to the individuals randomly. The individual of the population is coded as follows:

$$\begin{aligned} &individual = \left\{ \lambda_1 \lambda_{2\dots} \lambda_{N_1} \mid \lambda_{N_1+1} \dots \lambda_{N_2} \mid \dots \right. \\ &\left. \mid \lambda_{N_{n-1}+1\dots} \lambda_{N_n} \right\}, \lambda_1 \lambda_{2\dots} \lambda_{N_n} \in N \end{aligned} \tag{2}$$

where λ_{N_i} is the evacuation passageway assigned to i individual. Each evacuee can randomly select these paths to evacuate, all the evacuation paths selected by evacuees can be constructed the evacuation routes assignment scheme.

4.3 Algorithm Design

For the evacuation route assignment, the dynamic DE[11]should be modified to adopt this optimization problem. The mutation operation and the crossover operation of DE are modified as follows.

For the mutation operation, firstly, we set a mutation factor F (value range 0~1), then mutate the individual A with the mutation factor F , thus, the evacuation paths change. The change of evacuation paths is to reselect the evacuation path whose start point is corresponding to variables of the individual A and being calculated by the k-shortest paths algorithm. After the muting, we can obtain the mutated individual A' .

For the crossover operation, we perform the crossover operation on each variable of the mutated individual A' with the crossover rate (CR). The variable will be retained if that variable is selected, otherwise, it will be replaced by the variable selected from the individual A . The process will continue till the crossover operation performs on all variables. Where, the variables are referred to the evacuation paths. The mutation operation and crossover operation are shown in Fig. 1 and Fig. 2.

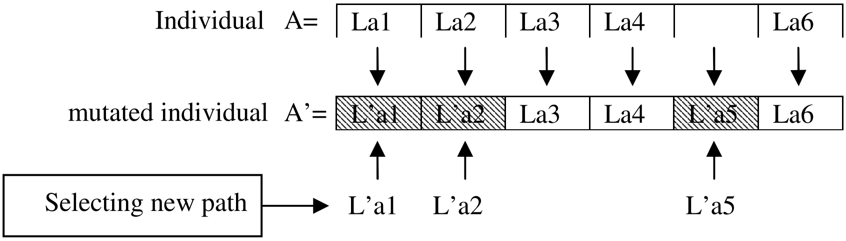


Fig. 1. Mutation operation for evacuation route assignment with $F = 0.5$

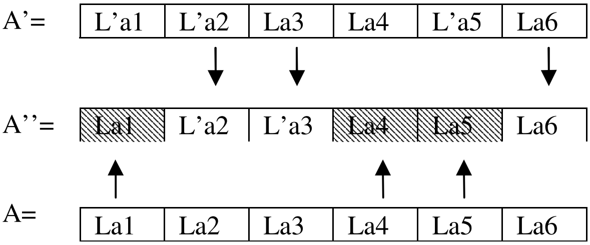


Fig. 2. Crossover operation for evacuation route assignment with $CR = 0.5$

5 Experimental Study and Discussion

5.1 Experimental Settings

The settings for the proposed dynamic evolution algorithm are shown in Table 1.

Table 1. Settings for the proposed algorithm

Parameter	*Setting*
Population size	100
Maximum subpopulation size	15
Mutation factor(F)	0.5
Crossover rate(CR)	0.9
Radius of the subpopulation core r_{core}	5.0
Number of optional paths for each start point	50
Threshold of congestion degree f_{pre}	0.75

The Wuhan Sport Center in Wuhan city China is taken as the experimental area to test the performance of the proposed algorithm. Wuhan Sport Center can hold about 60000 people, and there are 42 grandstands to accommodate the spectators. Suppose in a massive activity, the spectators should be evacuated to the safe area as quickly as possible for some reasons such as fire disaster and horrible attack. The number of evacuees is about 24727. The spectator is assigned to each grandstand randomly according to the number limitation of each grandstand.

The evacuation network of this stadium which contains 158 nodes and 224 arcs is converted by its structure (Fig. 3).The original locations of the spectators are the 42 grandstands, and the exits are the 5 ticket entrances, final destinations of evacuation in this scenario, as it is shown below.

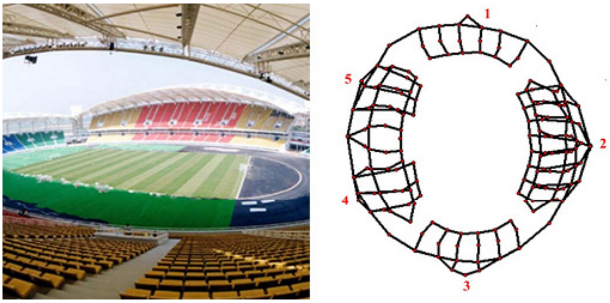


Fig. 3. Experiment area and evacuation network

In order to measure the performance of the proposed algorithm, we compare our proposed algorithm with DynDE[17]which is the classic dynamic differential evolution. In the experiments, each algorithm runs independently for 20 times with the 200 initialized paths for each start point. Though, the object is the minimum evacuation time, we also calculate the congestion degree and the total length of all evacuation paths.

5.2 Results and Analysis

The results are shown in Table 2.

Table 2. Results of the two algorithms for the evacuation route assignment

Algorithm		*Evacuation time(T)*	*Congestion*	*Total length of evacuation paths (L)*
Proposed algorithm	*Avg _best*	721.101	0.2731	34240.3
	Avg _mean	735.476	0.4353	34266.2
	Avg _worst	767.150	0.5252	34301.5
	STD	9.2346	0.0863	36.07
DynDE	*Avg _best*	732.119†	0.2907†	34304.8†
	Avg _mean	740.167†	0.4410	34329.6
	Avg _worst	770.234†	0.5723†	34378.3†
	STD	15.3028†	0.2057†	33.482‡

†indicates Proposed algorithm is significantly better than DynDE by the Wilcoxon signed-rank test at α = 0. 05. ‡means that DynDE is significantly better than our algorithm.

The convergence curve of the evacuation time and congestion is shown in Figure 4.

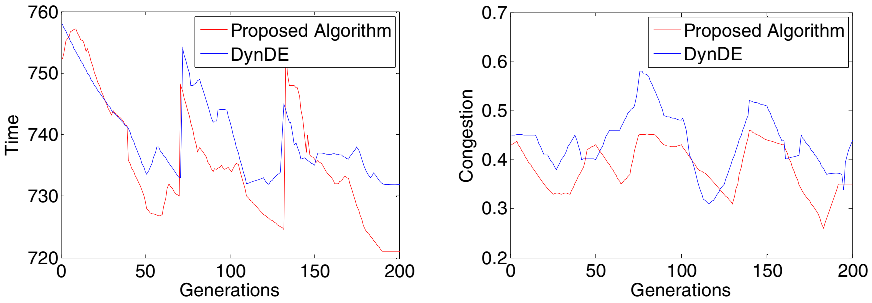


Fig. 4. The convergence curve of the evacuation time (left) and the convergence curve of the congestion (right)

From Table 2 can be seen that the proposed algorithm is superior to DynDE for the evacuation route assignment optimization. The satisfactory results are achieved by the proposed algorithm. It has the shorter evacuation time than DynDE with the better congestion. The total length of evacuation paths achieved by the proposed in the optimization process is shorter than the DynDE’s.

It is clearly seen from Figure 4 that the proposed algorithm outperforms the DynDE not only on the evacuation time and but also on the evacuation congestion.

6 Conclusions

Making a feasible and effective evacuation plan in the emergency situation is an important issue. In this paper, a prediction based multi-strategy differential evolution is adopted to solve this evacuation route assignment optimization problem based on the evacuation model. The experimental results of the evacuation scenario of a large public activity in Wuhan Sport Center in Wuhan City of China show that the proposed algorithm can solve the real world complex dynamic optimization problems such as the evacuation route assignment optimization problem. The analyses of the results imply that the congestion value restricts the performance of an evacuation plan. Thus, the congestion should be given a high priority.

Acknowledgments. This work was partially supported by the Scientific Research Foundation for Introduced Excellent Scholars, China Three Gorges University (Grant No. KJ2012B055).

References

1. Blackwell, T., Branke, J.: Multiswarms, exclusion, and anti-convergence in dynamic environments. IEEE Transactions on Evolutionary Computation 10(4), 459–472 (2006)
2. Branke, J., Saliho, U. G., Lu, E., Uyar, C.S.: Towards an analysis of dynamic environments. In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 1433–1440 (2005)
3. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, USA, pp. 1875–1882 (1999)
4. Mavrovouniotis, M., Yang, S.: Memory-based immigrants for ant colony optimization in changing environments. In: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, EvoApplications, pp. 324–333. Springer Verlag (2011)
5. Grefenstett, J.J.: Genetic algorithms for changing environments. In: Proceedings of the 2nd Conference on Parallel Problem Solving from Nature, Brussels, Belg, pp. 137–144 (1992)
6. Yang, S.: Explicit Memory Schemes for Evolutionary Algorithms in Dynamic Environments. In: Evolutionary Computation in Dynamic and Uncertain Environments, vol. 51, pp. 3–28 (2007)
7. Novoa-Hernández, P., Corona, C., Pelta, D.: Efficient multi-swarm PSO algorithms for dynamic environments. Memetic Computing 3(3), 163–174 (2011)
8. Mendes, R., Mohais, A.S.: DynDE: a differential evolution for dynamic optimization problems. In: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, pp. 2808–2815. IEEE Press (2005)
9. Blackwell, T.M., Bentley, P.: Dynamic Search with Charged Swarms. In: Proceedings of the 2002 Genetic and Evolutionary Computation Conference, pp. 19–26. Morgan Kaufman (2002)
10. Cobb, H.G.: An Investigation into the Use of Hypermutation as anAdaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments Naval Res. Lab., Washington, DC: Tech.Rep. AIC-90-001 (1990)

11. Shuzhen, W., Shengwu, X., Yi, L.: Prediction based multi-strategy differential evolution algorithm for dynamic environments. In: 2012 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE Press, Brisbane (2012)
12. Lämmel, G., Grether, D., Nagel, K.: The representation and implementation of time-dependent inundation in large-scale microscopic evacuation simulations. Transportation Research Part C: Emerging Technologies 18(1), 84–98 (2010)
13. Zheng, X., Zhong, T., Liu, M.: Modeling crowd evacuation of a building based on seven methodological approaches. Build Environ. 44(3), 437–445 (2009)
14. Yamada, T.: A network flow approach to a city emergency evacuation planning. Int. J. Syst. Sci. 27(10), 931–936 (1996)
15. Qiuping, L., Zhixiang, F., Qingquan, L., Xinlu, Z.: Multiobjective evacuation route assignment model based on genetic algorithm. In: 18th International Conference on Geoinformatics, Beijing, pp. 1–5. (2010)
16. Eppstein, D.: Finding the k shortest paths. In: Proceedings of 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, pp. 154–165, (1994)
17. Mendes, R., Mohais, A.S.: DynDE: a differential evolution for dynamic optimization problems. In: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, vol. 3, pp. 2808–2815. IEEE Press (2005)

Centralized Charging Strategies of Plug-in Electric Vehicles on Spot Pricing Based on a Hybrid PSO

Jiabao Wang[1], Qi Kang[1], Hongjun Tian[1], Lei Wang[1,2], and Qidi Wu[1]

[1] Department of Control Science and Engineering, Tongji University, Shanghai, China
wangjiabao0316@163.com, qkang@tongji.edu.cn
[2] Shanghai Key Laboratory of Financial Information Technology, Shanghai, China
wanglei@tongji.edu.cn

Abstract. This work proposes an efficient charging regulation strategy based on optimal charging priority and location of plug-in electric vehicles (PEVs). It employs a hybrid particle swarm optimization for optimal charging priority and location of PEVs in distribution networks, with the objectives of minimization of charging cost, power loss reduction and voltage profile improvement. The algorithm is executed on IEEE 30-bus test system. The results are compared with those that are gained by executing sample genetic algorithm (SGA) with diverse parameters on the same system. The results indicate the effectiveness and promising application of the proposed methodology.

Keywords: PEVs, spot pricing, centralized charging strategy, HPSO, optimal charging priority and location.

1 Introduction

Nowadays, more and more vehicles are on roads, thereby increasing the consumption of fossil fuel. Consequently, the environment is being seriously polluted. Under this circumstance, many countries have proposed their energy policies with objectives of economic effectiveness improvement, achievement of energy security, and environment pollution reduction, which promotes electrification of transportation and, especially, the rapid development of plug-in electric vehicle (PEV) industry [1].

However, PEVs, a new kind of power load, would exert a tremendous influence on the daily residential load curve of distribution network if they widely connected to the power grid for battery charging [2]. Due to the uncertainty of their charging behaviors, uncoordinated random charging of PEVs may lead to unforeseen effect on normal operation of distribution system, such as aggravating the load peak and off-peak difference in network, etc. Meanwhile, taking the spot pricing into consideration, the owners of PEVs may afford much higher cost for battery charging. Therefore, the appropriate dispatch of PEVs in a distribution system will be a challenging demand side management (DSM) [3]. Fortunately, PEVs are more flexible than traditional load, because majority of PEVs owners usually return home early in the evening and have no request for the special time that their vehicle will be charged, as long as the

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 401–411, 2014.
© Springer International Publishing Switzerland 2014

batteries are full by the next morning [4]. According to the statistical data of National Household Travel Survey (NHTS), more than 90% of vehicles are parked at home between 9 P.M and 6 A.M [4]. Taking this opportunity into account, several centralized charging strategies of PEVs have been researched for utilizing less expensive electricity and shifting the PEVs load to off-peak hours. For example, under a spot pricing based on electricity market environment, a demand side response based charging strategy is proposed, and a dynamic estimation interpolation based algorithm is designed to optimize the mathematical model which is established taking into account the valley-filling effect of supply side and the users' cost [1]. Wu et al. [4] proposes a novel minimization of charging cost based heuristic approach by analyses of PEVs travel pattern and spot pricing. The results show that the strategy can lower the peak-valley difference and save users' cost effectively, but they ignore the influence of PEVs charging behavior on power qualities, such as power loss, voltage fluctuation, etc. Deilami et al. [5] proposes a real-time smart load management control strategy which is developed for the coordination of PEVs charging based on real-time minimization of total cost of generating the energy plus the associated grid energy losses. The results indicate that the approach can reduce the power losses and improve the voltage profile by considering the maximum sensitivities selection optimization based priority charging. Lan et al. [6] presents a nonlinear electric vehicle (EV) battery model, and a dynamic programming-based algorithm for optimizing an EV's charging schedule with given electricity price and driving pattern. But only one EV's charging schedule is researched here.

In this paper, taking advantage of the flexibility of the PEVs, we arrange them to charge at the relatively inexpensive electricity which occurs during off-peak hours at night. An efficient charging regulation based on optimal charging priority and location of PEVs is proposed under a spot pricing based electricity market environment. And then, PSO-GA [7], a hybrid particle swarm optimization by incorporating genetic algorithm, hereinafter to be referred as HPSO, is used for optimal charging priority and location of PEVs in distribution networks system. We take power quality and economy objectives into account to define the optimization objectives in this paper, including power losses, voltage profile and charging cost. The proposed approach is executed on the IEEE 30-bus test system.

2 Problem Formulation

Under a spot pricing based electricity market environment, uncoordinated charging of many PEVs may exert negative impacts on the security and economy of power system operation, such as power losses increment, overload, voltage fluctuation aggravation and charging cost increment. To minimize such impacts, we may proceed as follows:

1) Different locations for PEVs charging may have different influences on power quality. Appropriate locations for PEVs charging benefit power quality improvement.

2) In order to abate the vacancy of electric power, PEVs are scheduled for charging during the off-peak period of the day.

3) The charging cost should be as low as possible. Because different periods of the day correspond to different electricity price under a spot pricing based electricity market environment, if all PEVs owners had a preference for the exact time of which electricity price is lowest, a new peak demand would occur. Hence, maximum power consumption should be set for every time slot in order to avoid overload. Under this circumstance, PEVs charging priority that determined a sequence of PEV choosing charging slots impacts on daily residential load curve and total cost of electricity heavily.

2.1 Charging Rule

In this paper, the maximum demand level has been defined as the maximal value of residential load during a scheduling period. The power for PEV charging is constrained by:

$$P_i \leq \mathrm{P}_i^{\mathrm{limit}}, i = 1,2,\ldots\ldots,\mathrm{T} \ . \tag{1}$$

$$\mathrm{P}_i^{\mathrm{limit}} = \mathrm{P}_{\mathrm{max}} - \mathrm{P}_i^{\mathrm{load}} \ . \tag{2}$$

where i and T are the time slot number and total number of slots, $\mathrm{P}_{\mathrm{max}}$ is the maximum residential load demand level with PEVs being charged, $\mathrm{P}_i^{\mathrm{load}}$ is the total residential power consumption at the ith time slot without PEVs plug in, $\mathrm{P}_i^{\mathrm{limit}}$ is the maximum permissible power consumption for PEVs charging at ith time slot, P_i is the total power consumption for PEVs charging at ith time slot.

If the PEVs charging priority and locations are known, load scheduling is transformed into a PEVs charging rule. The basic idea is to charge each vehicle in the time slots where the lowest electricity price occur and power consumption meet the Eq. (1).

The flowchart of PEVs charging rule is shown as Fig.1. And the relevant parameters can be define as follow: $Priority_j$ is the charging priority number of jth PEV , Bus_j is the charging location number of jth PEV , $\mathrm{Duration}_j$ is the number of time slots which jth PEV need for charging , n is the total number of PEVs, Price_i is the electricity price at ith time slot, $\mathrm{P}_i^{\mathrm{limit}}$ is the maximum permissible power consumption for PEVs charging at ith time slot, T is the total number of time slots, ρ_k is the serial number of PEV whose priority rank is k, τ_l is the serial number of slot whose price rank is l, P is the rated power of PEV, $Power_j^s$ is the total power consumption of PEVs at ith time slot and sth charging node, N is the number of locations, ε_j is the set of time slots where jth PEV is being charged.

2.2 Objective Outline

The objective of problem model includes minimization of charging cost, power loss reduction, voltage profile improvement. Therefore, a three-fold objective function is given by

$$\text{OBJ}: \; = \text{f} = \min\left(P_{loss} + \sigma V_{devi} + \gamma Cost\right) \; . \tag{3}$$

where P_{loss} is active power loss, σ and γ are the non-negative weighting factor used to indicate the relative important of three items, (here σ=20, γ=0.01), V_{devi} denotes load bus voltage deviations from 1.0 per unit, $Cost$ is the total electricity cost.

P_{loss} can be obtained with power flow calculation and is represented as $P_{loss} = \sum_{i=1}^{\mathrm{T}} \sum_{j=1}^{\mathrm{L}} |I_{ij}|^2 * R_j$, where T is the number of slots, L is the number of lines in the power system, $I_{ij}$is the current of jth line at ith time slot, R_j is the resistance of jth line.

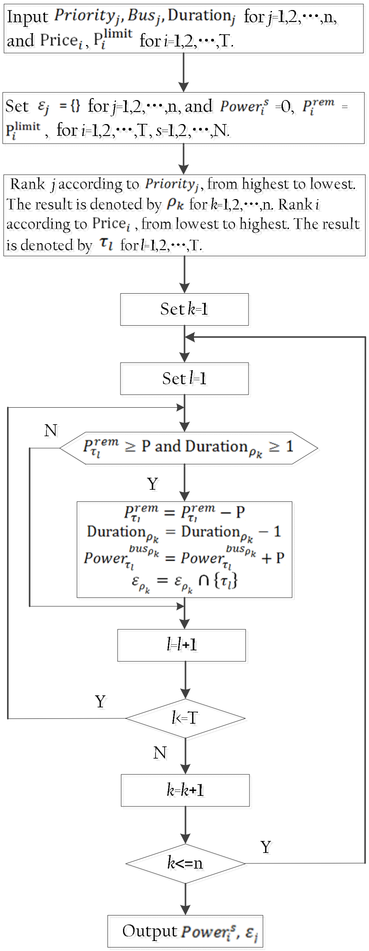

Input $Priority_j$, Bus_j, Duration$_j$ for j=1,2,⋯,n, and Price$_i$, $\mathrm{P}_i^{\mathrm{limit}}$ for i=1,2,⋯,T.
Set ε_j ={} for j=1,2,⋯,n, and $Power_i^s$ =0, P_i^{rem} = $\mathrm{P}_i^{\mathrm{limit}}$, for i=1,2,⋯,T, s=1,2,⋯,N.
Rank j according to $Priority_j$, from highest to lowest. The result is denoted by ρ_k for k=1,2,⋯,n. Rank i according to Price$_i$, from lowest to highest. The result is denoted by τ_l for l=1,2,⋯,T.
Set k=1
Set l=1
N
$P_{\tau_l}^{rem} \geq \mathrm{P}$ and Duration$_{\rho_k} \geq 1$
Y
$P_{\tau_l}^{rem} = P_{\tau_l}^{rem} - \mathrm{P}$
Duration$_{\rho_k}$ = Duration$_{\rho_k} - 1$
$Power_{\tau_l}^{bus_{\rho_k}} = Power_{\tau_l}^{bus_{\rho_k}} + \mathrm{P}$
$\varepsilon_{\rho_k} = \varepsilon_{\rho_k} \cap \{\tau_l\}$
l=l+1
Y
k=T
N
k=k+1
Y
k<=n
Output $Power_i^s$, ε_j

Fig. 1. The flowchart of PEVs charging rule

V_{devi} can be denoted as $V_{devi} = \sum_{i=1}^{\mathrm{T}} \sum_{j=1}^{\mathrm{N}} \left| V_{ij} - 1.0 \right| (\mathrm{p.u.})$, where T plays the same role in P_{loss}, N is the number of buses in power system, and V_{ij} is node voltage $(\mathrm{p.u.})$ of jth line at ith time slot [8].

$Cost$ is denoted as $Cost = \sum_{i=1}^{\mathrm{T}} \mathrm{P}_i * \mathrm{price}_i * \Delta\mathrm{T}$, where T plays the same role in P_{loss}, P_i is the power consumption of PEVs at ith time slot, price_i is the electricity price at ith time slot, $\Delta\mathrm{T}$ is time span of a time slot.

3 Scheduling Algorithms

3.1 HPSO: An Improved Particle Swarm Optimization

Particle swarm optimization (PSO) was proposed by Kennedy and Eberhart in 1995. In [9], dynamic weights are integrated into a standard PSO to improve its global and local convergence.

HPSO [7] discards the method with which particles update their positions by tracking the individual and group optimal position in PSO. Instead, it introduces the crossover and mutation operation of a genetic algorithm (GA) into PSO. Its new particles are refreshed by crossover and mutation operators according to optimal solutions of entire population and individual.

The steps of HPSO are as follows:

Step 1. Set the optimal position of individual ($p_{best,i}$) as the initial position of ith particle (i=1, 2,…, M), where M is the population size. Select the best one among $\{p_{best,1}, \cdots, p_{best,\mathrm{M}}\}$ as optimal position of population (g_{best}).

Step 2. Calculate the objective function values of all members.

Step 3. Update the $p_{best,i}$ and g_{best} according to the fitness of all members.

Step 4. Execute a crossover operation with $p_{best,i}$ and g_{best}. Update the position of the ith particle by executing a crossover operation with $p_{best,i}$ and g_{best}, respectively.

Step 5. Perform a mutation operation on each particle in the swarm.

Step 6. If the termination condition is met, stop. Otherwise, return to Step 2.

4 Simulation and Result

4.1 Simulation Data

In order to explain the problem more specifically and clearly, some assumptions are given and listed below:

1) All information of EVs and control signals generated by aggregators can be delivered immediately between EVs and aggregators [6].

2) PEV battery capacities typically range from a few kWh to over 50 kWh [5]. The capacity and rated power of PEV can be defined as 50kWh and 10kW, respectively.

3) The number of charging time slots of PEV is assumed to follow approximately Gaussian distribution whose mean value and standard deviation equal to 20 and 5 respectively.

4) 1000 PEVs are scheduled as a whole, thereby they can be treated as a PEV set, hereinafter still referred to as "PEV". And all vehicle batteries in PEV have same state of charge (SOC).

In this paper, PEVs are dispatched for charging during off-peak hours from 9 P.M. to 7 A.M., because most vehicles are vacant and the electricity price is generally low during this period. The daily residential load curve [5] shows in Fig.2. Furthermore, other relevant data are given as follow:

1) The total time for charging dispatch is segmented into 40 time slots, where each time slot has a duration of 15 minutes.

2) Data of spot pricing of electricity is released by Long Island, New York on Jan. 1, 2010 [2]. The value of electricity price ($/MW) during off-peak hours from 9 P.M. to 7 A.M is given as {71.50, 71.16, 63.46, 58.86, 62.67, 39.84 44.78, 53.03, 65.34, 57.82}.

3) The total number of PEVs which participate in scheduling is 50.

4) The distribution system used for simulation and analysis of PEVs ordered charging strategy in this paper is the IEEE 30-bus test system [10].

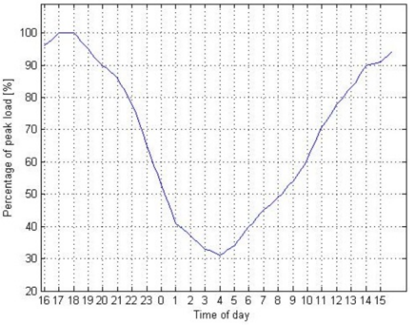


Fig. 2. Daily residential load curve

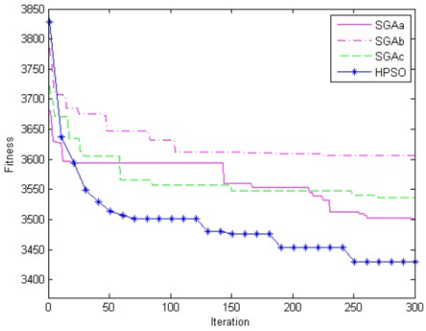


Fig. 3. Optimal fitness dynamics of system

IEEE 30-bus test system [11] is adopted for simulation and analysis in this work. In this paper, 10 buses, i.e., buses 3-12, are chosen as PEVs charging nodes. The HPSO-based method scheme can be described as follows, which is used as a solver of this optimization problem of charging priority and location of PEVs.

Step 1. Initialization

Set the time count *t*=0, dimension D (here D=50), maximum iteration number Iter_{max}, and population size M. The current position of all members is $X(0) = \{X_1^0, \dots, X_i^0, \dots, X_\text{M}^0\}$, where

$$X_i^0 = \begin{bmatrix} x_{i,1}^0 \; x_{i,2}^0 \; \cdots \; x_{i,D}^0 \\ y_{i,1}^0 \; y_{i,2}^0 \; \cdots \; y_{i,D}^0 \end{bmatrix} \quad (i=1,2,\dots, \text{M}).$$

X_i^t represents the coding scheme of ith particle at tth iteration, $x_{i,j}^t$(*j*=1,2,…, D) is the charging priority index of jth PEV of ith particle at tth iteration, $y_{i,j}^t$(*j*=1,2,…, D) is the charging location (bus index) of jth PEV of ith particle at tth iteration. The first line of X_i^0 is a random permutation of integers 1~50 and $y_{i,j}^0$ is a random integer in {3,4,…,12}.

PEVs charging rule and power flow calculation are applied to all members, by the results of which the fitness of value $f(X_i^0)$ of X_i^0 is obtained. For each individual, set $p_{best,i}= X_i^0$ and $Pbest_i = f(X_i^0)$, i=1,2,..., M. Select the best one among $\{p_{best,1}, \cdots, p_{best,\mathrm{M}}\}$ as g_{best}, and set $Gbest = f(g_{best})$.

Step 2. Update the time counter.

$t = t + 1$

Step 3. Execute crossover operation.

(1) Execute a crossover operation with $p_{best,i}$

Update the position of X_i^t (i=1,2,..., M) by executing crossover operation with $p_{best,i}$. If $f(X_i^t)$< Pbest$_i$, then update individual best as $Pbest_i = f(X_i^t)$ and set $p_{best,i}= X_i^t$.

(2) Execute a crossover operation with g_{best}

Update the position of X_i^t (i=1,2,..., M) by executing crossover operation with g_{best}. If $f(X_i^t)$< $Pbest_i$, then update individual best as $Pbest_i = f(X_i^t)$ and set $p_{best,i}= X_i^t$.

Step 4. Execute mutation operation

Execute a mutation operation by exchanging any two elements of first line of X_i^t and change corresponding element of second line using a random integer in {3,4,...,12}. If $f(X_i^t)$< $Pbest_i$, and then update individual best as $Pbest_i = f(X_i^t)$ and set $p_{best,i}= X_i^t$.

Step 5. Carry out fitness evaluation and update individual best and population best.

For all members, PEVs charging rule and power flow calculation are applied, and then evaluates the fitness of every member X_i^t (i=1,2,..., M) according to Eq.(3). If $f(X_i^t)$< $Pbest_i$, and then update individual best as $Pbest_i = f(X_i^t)$ and set $p_{best,i}= X_i^t$. Select the best one among $\{p_{best,1}, \cdots, p_{best,\mathrm{M}}\}$ as $\mathrm{g_{best}}$, and set $Gbest = f(g_{best})$.

Step 6. Check the stopping criteria

If $t \le \mathrm{Iter_{max}}$, then go to **Step 2**; Else, stop the algorithm.

4.2 Parameters Setting

The parameters setting of SGA and HPSO are given in Table 1. In SGA•a population of M solutions is maintained and two probability-based operations, i.e., crossover operator and mutation operator are employed. Whether the two operators work depends on crossover rate $\mathrm{Rate_c}$ and mutation rate $\mathrm{Rate_m}$, respectively. Meanwhile, the SGAs with different crossover and mutation parameters are denoted as SGAa, SGAb and SGAc.

Table 1. The parameters setting of SGA and HPSO

SGA	a	M=20;	$\mathrm{Iter_{max}}$=300;	$\mathrm{Rate_c}$=0.8;	$\mathrm{Rate_m}$=0.05.
	b	M=20;	$\mathrm{Iter_{max}}$=300;	$\mathrm{Rate_c}$=0.9;	$\mathrm{Rate_m}$=0.005.
	c	M=20;	$\mathrm{Iter_{max}}$=300;	$\mathrm{Rate_c}$=0.7;	$\mathrm{Rate_m}$=0.025.
HPSO		M=20;	$\mathrm{Iter_{max}}$=300.		

By executing two types of algorithms 20 times separately, the Monte Carlo simulation results are shown in Table 2, where *iteration* represents the number of iterations. The evolutionary trajectory of four algorithms is shown in Fig.3.

The simulation results show that the HPSO outperforms SGA on the metrics in P_{loss}, Cost, and fitness value. Further, HPSO have better global search ability. However, the superiority of HPSO in searching the metric V_{devi} is not outstanding.

The proposed PEVs charging rule is applied to the optimal individual sought out by HPSO, and yields the power consumption of every time slot. Fig.4 shows the impact of HPSO-based coordinated PEVs charging on total system power demand. Compared with the results of uncoordinated PEVs charging, HPSO-based coordinated PEVs charging strategy effectively shift electricity use from on-peak to off-peak period.

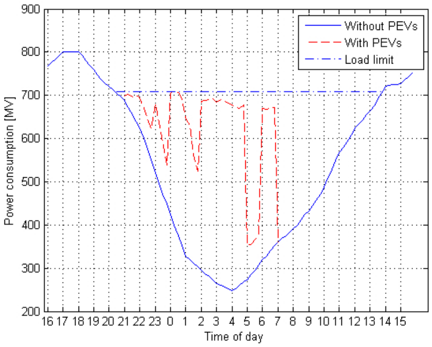


Fig. 4. The impact of HPSO-based coordinated PEVs charging on total system power demand

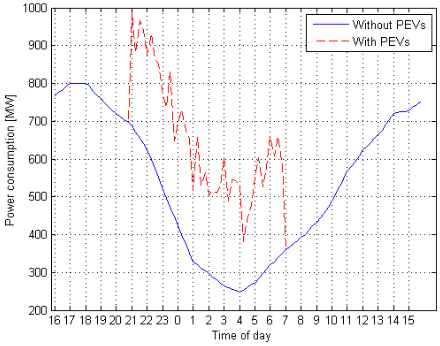


Fig. 5. The impact of uncoordinated PEVs charging on total system power demand

The impact of uncoordinated PEVs charging on total system power demand is shown in Fig.5, in which many peak power demands appear. With the increase of the gap between peak load and valley load in power system, it is uneconomic to install units considering the capacity of peak load. Besides, frequent start and stop are detrimental to the life of generators.

The impact of HPSO-based coordinated PEVs charging is further compared with Min-Cost Load Scheduling (MCLS) approach [4]. The basic idea of MCLS is that each vehicle is charged in the time slots where the lowest electricity price occurs. Fig.6 shows that the impact of MCLS-based PEVs charging on total system power demand. The minimum electricity cost can be achieved by carrying out the MCLS approach, but the impact of PEVs charging on system power loss and voltage deviation is ignored by MCLS approach. Besides, as we can see in Fig.6, there are still three peak loads and large gap between peak load and valley load. HPSO-based voltage (p.u.) of 30 buses at worst time slot is compared with MCSL-based voltage of 30 buses at random time slot and stochastic charging nodes, as shown in Fig.7. The HPSO-based coordinated PEVs charging improve the security and reliability of networked distribution network by minimizing voltage deviations, overloads, and power losses that would otherwise be impaired by MCSL-based PEVs charging.

Table 2. Simulation Results

Metric		SGAa	SGAb	SGAc	HPSO
$P_{loss}(MW)$	Optimum value	1374.8	1453.7	1402.4	1331.7
	Worst value	1480.5	1518.5	1461.6	1363.5
	Median value	1439.0	1474.1	1455.6	1336.8
	Mean value	1432.2	1478.1	1437.9	1344.5
	Standard deviation	41.7753	21.5690	26.3989	12.3578
$Cost(\$ \times 10^3)$	Optimum value	132.9018	133.6457	133.6895	133.9468
	Worst value	136.2426	136.6502	136.5378	135.3387
	Median value	135.4281	134.9455	135.1636	135.2335
	Mean value	134.8389	135.2212	135.1725	134.8535
	Standard deviation	1.3417	1.1264	0.9367	0.5565
V_{devi}	Optimum value	38.1942	40.3422	38.9862	37.3199
	Worst value	40.8019	41.5776	40.5185	38.0455
	Median value	39.2931	40.5403	40.1329	37.6670
	Mean value	39.6493	40.6993	39.8711	37.6834
	Standard deviation	0.9853	0.4502	0.5426	0.2360
$fitness$	Optimum value	3501.9	3606.8	3536.9	3429.5
	Worst value	3648.9	3716.5	3628.3	3477.3
	Median value	3585.0	3630.4	3606.7	3440.1
	Mean value	3573.5	3644.3	3587.0	3457.0
	Standard deviation	43.3275	38.2886	37.4149	19.320
$iteration$	Minimum value	172	82	123	252
	Maximum value	281	275	287	286
	Median value	269	245	270	274
	Mean value	249.8	218.6	244.8	269.4
	Standard deviation	39.6353	45.9125	37.3197	12.8623

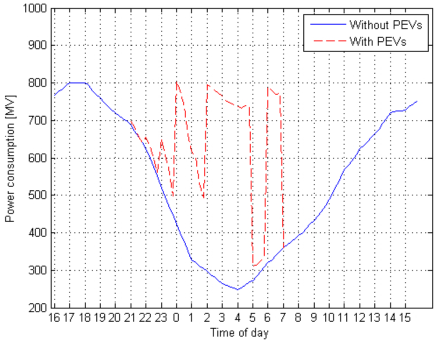


Fig. 6. The impact of MCSL-based coordinated PEVs charging on total system power demand

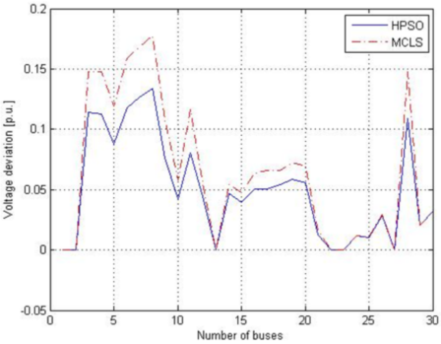


Fig. 7. Voltage deviation at 30 buses

5 Conclusion

This paper presents a centralized charging model of PEVs under a spot pricing-based electricity market environment. All PEVs taking part in centralized charging scheduling are equally divided into multiply, and an efficient charging rule which is based on charging priority and location of PEVs is proposed in the model. Further, in this paper, HPSO is employed to optimally determine the PEVs charging priority and location to be plugged in distribution system. Combining the HPSO with Newton method of power flow calculation can solve the problem of optimal charging priority and location of PEVs well with the objective of minimization of charging cost, power loss reduction and voltage profile improvement. The results are compared with those are obtained using SGA, and validate the superiority and effectiveness of the approach.

Under the intelligent centralized charging strategy, PEV owners are able to pay the electricity bills only in line with the number of charging time slots they use. The electricity costs resulting from the use of the proposed approach are slightly higher than those by MCLS algorithm whose objective is the maximization of energy trading profits, but it effectively reduces the gap between peak and valley load and simultaneously improves the power quality. In this case, a discount electricity price scheme can be introduced by power supplier in order to encourage PEV owners to take part in centralized charging mechanism.

The further research will be oriented to dispatching PEVs for charging under the assumption of stochastic vehicles' arrival, with the same objectives as discussed in this work.

Acknowledgments. This work was supported in part by the Natural Science Foundation of China (71371142, 61005090, 61034004), the Program for New Century Excellent Talents in University of Ministry of Education of China. Ministry of Education (NCET-10-0633), the Fundamental Research Funds for the Central Universities, and the Research Fund of State Key Lab. of Management and Control for Complex systems.

References

1. Zhao, J., Wen, F., Yang, A., Xin, J.: Impacts of electric vehicles on power systems as well as the associated dispatching and control problem. Automation of Electric Power Systems 14, 2–9 (2011)
2. Zou, W., Wu, F., Liu, Z.: Centralized charging strategies of plug-in hybrid electric vehicles under electricity markets based on spot pricing. Dianli Xitong Zidonghua (Automation of Electric Power Systems) 35(14), 62–67 (2011)
3. Masoum, A.S., Deilami, S., Moses, P.S., Masoum, M.A.S., Abu-Siada, A.: Smart load management of plug-in electric vehicles in distribution and residential networks with charging stations for peak shaving and loss minimisation considering voltage regulation. Generation, Transmission & Distribution, IET 5(8), 877–888 (2011)

4. Wu, D., Aliprantis, D.C., Ying, L.: Load scheduling and dispatch for aggregators of plug-in electric vehicles. IEEE Transactions on Smart Grid 3(1), 368–376 (2012)
5. Deilami, S., Masoum, A.S., Moses, P.S., Masoum, M.A.: Real-time coordination of plug-in electric vehicle charging in smart grids to minimize power losses and improve voltage profile. IEEE Transactions on Smart Grid 2(3), 456–467 (2011)
6. Lan, T., Hu, J., Kang, Q., Si, C., Wang, L., Wu, Q.: Optimal control of an electric vehicle's charging schedule under electricity markets. Neural Computing and Applications 23(7-8), 1865–1872 (2013)
7. Min, X.I.E.: An Improved Hybrid Particle Swarm Optimization Algorithm for TSP. Journal of Taiyuan University of Technology 4, 023 (2013)
8. Kang, Q., Lan, T., Yan, Y., Wang, L., Wu, Q.: Group search optimizer based optimal location and capacity of distributed generations. Neurocomputing 78(1), 55–63 (2012)
9. Jian-Hua, L., Rong-Hua, Y., Shui-Hua, S.: The analysis of binary particle swarm optimization. Journal of Nanjing University (Natural Sciences) 5, 003 (2011)
10. AlRashidi, M.R., El-Hawary, M.E.: Hybrid particle swarm optimization approach for solving the discrete OPF problem considering the valve loading effects. IEEE Transactions on Power Systems 22(4), 2030–2038 (2007)
11. AlRashidi, M.R., El-Hawary, M.E.: Hybrid particle swarm optimization approach for solving the discrete OPF problem considering the valve loading effects. IEEE Transactions on Power Systems 22(4), 2030–2038 (2007)

A New Multi-region Modified Wind Driven Optimization Algorithm with Collision Avoidance for Dynamic Environments

Abdennour Boulesnane[1,2] and Souham Meshoul[1]

[1] Computer Science Department, Constantine 2 University, Algeria
[2] MISC Laboratory, Constantine, Algeria
abdennour.boulesnane@gmail.com,
souham.meshoul@univ-constantine2.dz

Abstract. This paper describes a new approach to deal with dynamic optimization that uses a multi-population. Its main features include the use of a modified wind driven optimization algorithm that aims to foster impact of pressure on velocities of particles. Moreover, a concept of multi-region inspired from meteorology has been introduced along with a new collision avoidance technique to maintain good diversity while preventing collision between sub-populations. The method has been assessed using Moving Peaks Benchmark and compared to state of the art methods. Preliminary results are very encouraging and show viability of the method.

Keywords: Dynamic optimization, Swarm intelligence, Wind driven optimization, collision, multiple population methods, Moving Peaks Benchmark.

1 Introduction

In everyday life and in almost all domains, each type of optimization problem has features that make it different from the others. However, these problems usually have a common property that is their dynamic nature. Such problems are difficult to solve because the challenge is not only to locate global optima, but also to track them in environments that change over time. Therefore, a crucial requirement a dynamic optimization algorithm should fulfill is to achieve a balance between exploitation and exploration of the search space to handle optimization over time. This requires fostering diversity while ensuring very fast convergence to global optima throughout the search process because the time between two successive changes may be insufficient to converge and to follow optima at the same time. Moreover, dynamic optimization is faced to the challenge to solve both issues of outdated memory due to changes in environment and diversity loss due to traps of local optima. Outdated memory problem is usually solved by clearing the memory when a change is detected however the matter is what to do with the knowledge acquired once a change in the environment occurs: should it be reused for next changes or discarded? In [1], a study showed that the reuse of information lead to faster adaptation to changes, and thus, to

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 412–421, 2014.
© Springer International Publishing Switzerland 2014

better solutions. Many algorithms have been proposed to address dynamic optimization problems (DOPs). A comprehensive survey can be found in [2]. More particularly, using multiple populations has been shown to be a suitable way to keep up with changes over time [7,10].

Recently, a new swarm based metaheuristic inspired from atmospheric motion has been proposed in [3] and termed as Wind Driven Optimization (WDO). WDO's model is based on the definition of trajectories of small air parcels within the earth atmosphere according to the Newton's second law of motion. WDO has been designed for static optimization and applied to electromagnetics optimization problems. In this paper, we propose investigating the potential of WDO to solve dynamic optimization. Intuitively, the motivation can be explained by the fact that the movement of air from high pressure zones to low pressure zones at velocities proportional to pressure gradient force would lead to simulation models that can be used in dynamic environments. Taking inspiration from WDO, a modified version is developed and used within a multi-population framework with the aim to adaptively detect promising regions in the search space. Therefore, we refer to the developed approach as Multi-Region Modified WDO (MR-MWDO). Furthermore, in order to maintain several sub-populations on several peaks (promising regions) and to avoid collisions between sub-populations a new strategy called Collision Avoidance Technique is introduced.

The rest of the paper is organized into four sections. Section 2 presents the WDO algorithm. The proposed Multi-Region Modified WDO is described in section 3. Section 4 is devoted to the experimental study. Finally, conclusions and perspectives are given in section 5.

2 Wind Driven Optimization

Recently, a new approach to deal with multi-dimensional and multi-modal optimization problems has been proposed by Bayraktar [3] and termed as Wind Driven Optimization. As the name suggests, WDO is inspired by the earth's atmosphere in the Troposphere layer and more specifically by the contribution of wind in the equalization of horizontal imbalances in the air pressure. In his study, Bayraktar [3] used the physical equations that govern atmospheric motion. This later is generally described by the movement of air which is a consequence of pressure gradient due to temperature differences. It is observed that wind blows from a high pressure zone to low pressure zone with a velocity proportional to the pressure gradient force. For in-depth insight into the physical model, the reader can refer to [4]. Starting from the Lagrangian model that describes atmospheric motion, Bayraktar derived WDO as an iterative metaheuristic. WDO's dynamics is similar to that of Particle Swarm Optimization (PSO). Particles in WDO refer to small air parcels that are assumed dimensionless and weightless for simplification. The trajectories of these parcels are defined according to the Newton's second law of motion. Like PSO, these air parcels are described by a position and a velocity that refer to a candidate solution and the amount of position displacement respectively. The pressure at each air parcel is used as information about the related solution quality. Updating positions and velocities of air parcels is governed by the following equations where the variable *i* refers to the particle and the variable *t* to iteration [3].

$$u_{t+1}^{i} = (1-\alpha)u_{t}^{i} - gx_{t}^{i} + \left(RT\left|\frac{1}{\mathrm{r}} - 1\right|(x_{opt} - x_{t}^{i})\right) + \left(\frac{\mathrm{c}u_{t}^{other\ dim}}{r}\right) \tag{1}$$

$$x_{t+1}^{i} = x_{t}^{i} + u_{t+1}^{i} \tag{2}$$

Where u_t^i and u_{t+1}^i are the current and the new velocity of the air parcel respectively, x_{opt} is the global best position, x_t^i and x_{t+1}^i are the current and the new positions of the air parcel, parameters α, g, R, and T are related respectively to the friction coefficient, gravity, universal gas constant and temperature in the physical model. The variable r represents the rank of the air parcel where all air parcels are ranked in descending order based on their pressure. An in-depth description WDO is available at [3].

3 The Proposed Algorithm for Dynamic Optimization

For sake of clarity, we first describe the modifications brought to WDO to properly handle optimization in dynamic environments then we present the proposed MR-MWDO for dynamic optimization.

3.1 Modified WDO Algorithm

In nature, atmospheric pressure is influenced by several factors such as temperature, humidity and altitude among others. These factors contribute in raising and lowering the air pressure according to the spatial distribution and topography on the surface of the earth. This leads to the formation of high pressure regions and low pressure regions and the gradient of the pressure force gives rise to air motion. As shown in equation (1) of WDO model, the influence of pressure on velocity of air parcels is not expressed in terms of actual values of pressure but is implicitly represented in the third and fourth terms by the rank of the particle among other particles based on their fitness values. By another side, it is known that altitude is inversely proportional to pressure. Moreover, one natural way to establish analogy between optimization and atmospheric models is to relate altitude to fitness. Therefore, the matter is how to express in a convenient manner the relationship between pressure and fitness. In [5], Chao et al. proposed a Tropical Cyclone-based Method (TCM), for solving global optimization problems with box constraints. Chao and al expressed the relationship between the pressure p^i of a particle *i* and its fitness $f(x_i)$ by the following equation:

$$p^{i} = exp\left(-n * \frac{f(x^{i}) - f(x^{worst})}{\sum_{\mathrm{k=1}}^{\mathrm{m}}(f(x^{k}) - f(x^{worst}))}\right) \tag{3}$$

Where $f(x^{worst})$ represents the fitness of the worst position, n is the problem dimension and m the number of particles. According to equation (3), a particle with a higher fitness possesses a lower pressure, and the pressure is scaled to be 1 at x^{worst}. Therefore, in the modified WDO we propose to use the value of pressure as given by equation (3) instead by the rank of the particle to better reflect the influence of pressure on velocities. In other words, parameter r in equation (1) is replaced by parameter p^i.

3.2 Features for Handling Dynamic Optimization

3.2.1 Concepts of Multi-region

Multi-population methods maintain multiple sub-populations concurrently. Each sub-population may handle a separate area of the search space. Each of them may also handle a specific task. For example, some sub-populations may focus on searching for the global optimum while some others may concentrate on tracking any possible changes. These two types of populations then may communicate with each other to bias the search [6].

In meteorology, a region is set as a low-pressure area (respectively a high-pressure area) if the atmospheric pressure is lower (respectively higher) than that of surrounding locations. Wind blows in an attempt to equalize horizontal imbalances in the air pressure. Meteorologists use the value of pressure at sea level as a threshold to decide the type of region: if the pressure values in a region are lower than 1013.25, then the region is low pressure else it is high pressure. The idea behind MR-MWDO is then classifying regions of a search space into low and high pressure regions could be used to guide the search process and better identify promising areas that may include optima.

In the proposed Multi-Region MWDO algorithm, this principle is implemented through the use of two different types of sub-populations with different numbers of particles. The aim is to better control the level of diversity inside and outside sub-populations knowing that after occurrence of a change in dynamic environment, the new optimum can be located near or very far compared to the previous optimum. The first type is a sub-population of observer particles (SOP), which role is exploring and calculating pressure at various search areas. The second type is a sub-population of air particles (SAP) that navigate within the search space according to MWDO equations. For example, in the case of a maximization problem, a new low pressure region is detected (a low pressure region in this case represents a promising area) if the best value of pressure in SOP is less than or equal to a threshold value and all SAP are converged. The threshold value is the median of all pressure values of all particles.

When a new low pressure region is found, a new SAP is generated within this region. Accordingly, a radius r_{region} is assigned to each SAP in order to avoid the re-exploration of already visited regions by SOP.

3.2.2 Collision Avoidance Technique

In order to ensure that each promising region (each peak) is explored by only one sub-population, researchers proposed to delimit regions using a radius. For example, Blackwell et al [7] proposed the technique of exclusion which consists in using a simple competition between the swarms that are close to each other. The swarm with the best function value is kept (best solution found so far) whereas the other will be expelled and reinitialized. Two swarms collide if and only if their attractors (best solutions found so far by each swarm) are located within an exclusion radius r_{excl} of each others [7]. Within the same spirit, we introduce a new technique called Collision Avoidance Technique to prevent collision between sub-populations and therefore maintain several sub-populations on several peaks. For each sub-population, the distance to the closest other sub-population is recorded. To determine whether this proximity between these sub-populations is desirable or not, the midpoint of the recorded distance is considered. Figure 1 shows two cases where proximity between subpopulations is desirable and one case where it is undesirable.

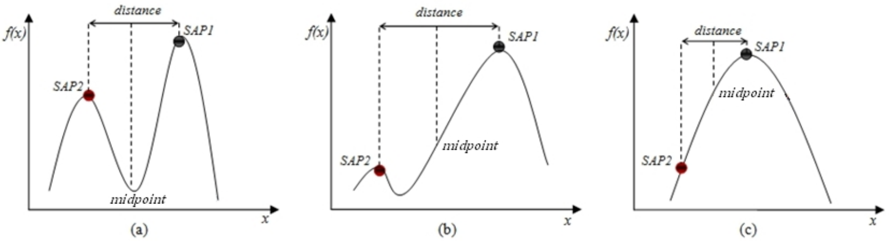


Fig. 1. Proximity between sub-populations: plots (a) and (b) show cases of desirable proximities between two SAPs related to different peaks. In plot (c), the proximity between the two SAPs is undesirable (both SAPs are related to the same peak).

From Figure 1, we can identify two major cases depending on the relative fitness of the midpoint compared to the best solutions in the two SAPs; the first case is when this value is less than the best solutions found by SAPs Figure 1.a. In this case, both SAPs will be kept. While in the second case, if this value is better than one of them as in Figure 1.b and 1.c, a new position of the midpoint must be calculated iteratively between the previous midpoint and the weakest SAP's solution. At each iteration, if the fitness of the new midpoint is worse than the weakest SAP's solution the iterative process is stopped and both SAPs will be retained Figure 1.b. Otherwise, the process will pursue until the distance between the midpoint and the lowest SAP is less than r_{region}. At this stage, if the fitness of the midpoint remains better than the weakest SAP's solution, then this SAP is removed to avoid the collision as in Figure 1.c.

3.3 Outline of the proposed Multi-region MWDO Algorithm

The algorithm starts with single sub-population of n observer particles (SOP) with the aim to explore only the search space and to find promising regions. Then the algorithm goes through an iterative process as described on Figure 2. For each generated SAP_i the convergence test is performed in the following manner:

```
For each $SAP_i$
 If best value of pressure is better than threshold
    & not $Converged_i$ then
  $Converged_i$=true;
 Elseif best value of pressure is lower than threshold
         & $Converged_i$ then
  $Converged_i$=false;
 Endif
Endfor
```

To ensure a reasonable number of sub-populations in the search space without degrading the performance of the algorithm and performing unnecessary functions evaluations knowing that the number of evaluation should be limited because the time of the next change is unknown, two successive phases are applied. The first phase is to generate a new sub-population (SAP) when a new promising region is detected as seen previously, while in the second phase, a sub-population with low pressures or being in collision with another one is removed.

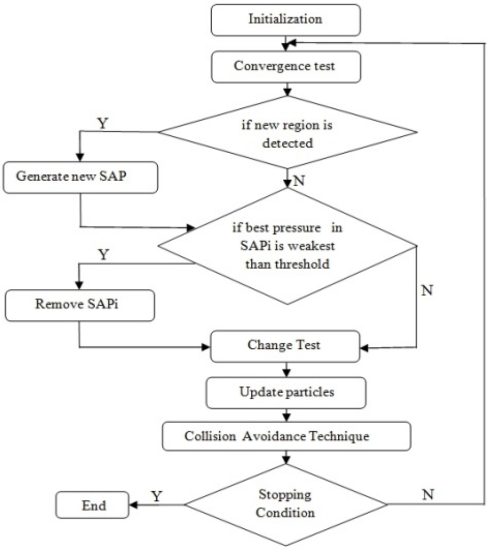


Fig. 2. Flow chart of proposed algorithm

The phase of change test is a significant step as it allows the algorithm to adapt to a new change in the environment by comparing, at each iteration, the best solutions found by the sub-populations (SAPs) to their best old achieved values. Then in the next phase, particles positions are updated using modified equations (1) and (2) for the sub-population of air particles (SAP) and random displacement for sub-population of observer particles (SOP). Once new positions are recorded, Collision Avoidance procedure is performed to keep several sub-populations on different peaks as described above. The algorithm evolves in this manner till a termination criterion is satisfied.

4 Experimental Results

In order to assess the performance of MR-MWDO, Moving Peaks Benchmark (MPB) has been used [8]. All experiments have been performed using the Scenario 2 of the MPB, proposed by Branke in [8]. This scenario has also been used by several authors and allows comparison of results obtained by different methods. The settings for this scenario are as follows: the search space has five dimensions $X^5 = [0{,}100]^5$ there are $p = 10$ peaks, the peak heights vary randomly in the interval $[30{,}70]$ and the peak width parameters vary randomly within $[1{,}12]$. The peaks change position every $K = 5000$ evaluations by a distance of $s = 1$ in a random direction, and their movements are uncorrelated (the MPB coefficient $\lambda = 0$). The algorithm was run for 100 consecutive changes, and each run was independently repeated 30 times, with different random seeds. The performance measure used to evaluate the algorithm is the offline error [9] given by the following equation:

$$oe = \frac{1}{Nc}\sum_{i=1}^{Nc}\left(\frac{1}{Ne(i)}\sum_{j=1}^{Ne(j)}(vbest_i - vbest_{ij})\right) \quad (4)$$

Where Nc is the total number of changes in the objective function, $Ne(i)$ is the number of evaluations performed during the i^{th} change, $vbest_i$ is the value of the global optimum in the i^{th} change and $vbest_{ij}$ is the value of the best solution obtained by the algorithm in the j^{th} evaluation of i^{th} change.

First experiment has been conducted to find suitable settings of the algorithm's parameters that consist in the size of a SOP (n), the size of a SAP (m), the radius r_{region} assigned to each SAP to avoid collisions and parameters used for the update of air particles' positions that is $[\alpha, g, c, RT]$ seen in the equation (1). These latter have been set to $(\alpha = 1, g = 0, c = 0.4, RT = 3)$. The results of this experiment are shown on tables 1 and 2.

Table 1. Offline error and Standard deviation for varying particles number. The data is for 30 runs of MPB (Scenario 2) and r_{region}=5.0.

$(m+n)$	Offline error (Std error)	$(m+n)$	Offline error (Std error)
$(5+5)$	6.19 ± 0.07	$(10+5)$	4.92 ± 0.09
$(5+10)$	2.26 ± 0.08	$(10+10)$	3.76 ± 0.10
$(5+20)$	1.78 ± 0.04	$(10+20)$	2.66 ± 0.08
$(5+30)$	1.92 ± 0.03	$(10+30)$	2.23 ± 0.06
$(5+50)$	2.13 ± 0.03	$(10+50)$	1.85 ± 0.03

Table 2. Offline error and Standard deviation depending on parameter r_{region}

r_{region}	Offline error (Std)	r_{region}	Offline error (Std)
0.5	1.91 ± 0.03	5.0	1.78 ± 0.04
1.0	1.88 ± 0.04	8.0	1.83 ± 0.04
3.0	1.79 ± 0.03	10.0	1.84 ± 0.04

As can be seen, the algorithm achieves the best performance with $(\boldsymbol{m}+\boldsymbol{n}) = (\mathbf{5}+\mathbf{20})$ and $\boldsymbol{r_{region}} = \mathbf{5.0}$. With these settings, the algorithm was found to converge quickly after each change while maintaining good diversity level within and outside sub-populations. Furthermore, it has been observed that sub-populations work within their regions delimited by $\boldsymbol{r_{region}}$ radius without collision problem.

4.1 Impact of the number of Sub-populations

The number used of sub-populations plays an important role in maintaining good diversity and occupying the promising areas that can include global optima after each change. However, using large numbers of sub-populations impacts negatively the performance of the algorithm by making unnecessary evaluations of the objective function. Figure 3 shows the total number of sub-populations during a single run of the 10 peaks MPB environment with $(m+n) = (5+20)$ and $r_{region} = 5.0$. As can be seen, the algorithm was able to maintain the appropriate number of sub-populations (SAP) on different peaks (10 peaks in this case), which is achieved at 35000 evaluations while improving the offline error.

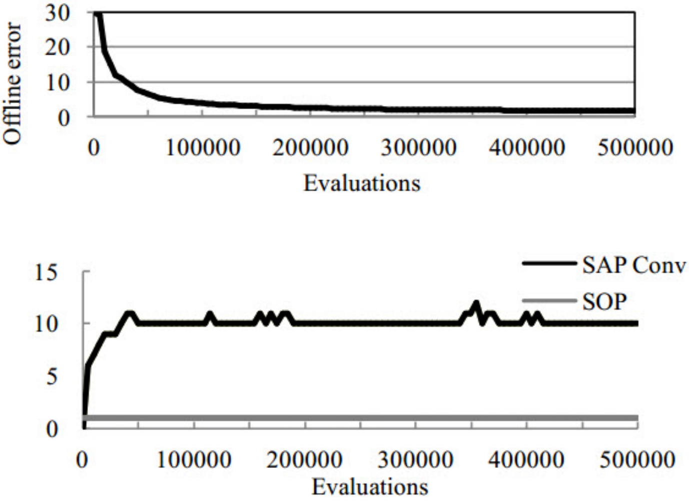


Fig. 3. Total number of sub-populations for a single instance of the 10 peaks MPB environment. Upper plot shows offline error, lower plot shows number of converged sub-populations (SAPs) and explorer sub-population (SOP).

4.2 Comparison with Other Algorithms

In order to compare MR-MWDO algorithm to other algorithms from the literature, an experiment was performed during which the offline error and its standard deviation were recorded for different shift severities (*s*) and different number of peaks (*p*) for all algorithms namely: HmSO [10], CPSO [11], mQSO [7], SPSO [12] and CPSOR [13]. Results are reported on tables 3 and 4. The shift length s is the severity of the problem dynamics. Whenever it increases, localization and tracking of peaks becomes more difficult.

Table 3. Offline error ± Standard error for different algorithms on the MPB problem with different shift severities

s	MR-MWDO	CPSOR	HmSO	CPSO	mQSO	SPSO
0	1.01 ±0.04	1.57 ±1.23	2.68 ±1.26	0.899 ±1.05	**0.601** ±0.439	0.807 ±0.972
1	1.78 ±0.04	**0.448** ±0.626	3.49 ±1.65	1.35 ±1.12	1.69 ±0.784	2.28 ±1.49
2	2.53 ±0.03	**0.588** ±0.654	3.88 ±1.66	1.31 ±1.15	1.84 ±0.723	2.81 ±1.54
3	3.57 ±0.03	**0.749** ±0.70	4.22 ±1.74	1.41 ±1.08	2.06 ±0.775	3.24 ±1.34

Table 4. Offline error ± Standard error for different algorithms on the MPB problem with different numbers of peaks

p	MR-MWDO	CPSOR	HmSO	CPSO	mQSO	SPSO
1	1.48 ±0.06	0.245 ±0.253	1.80 ±2.09	**1.4e − 4** ±5.9e − 4	3.15 ±1.21	1.86 ±1.45
5	1.46 ±0.06	**0.936** ±0.788	3.99 ±1.92	0.97 ±0.755	1.56 ±0.691	1.52 ±0.769
10	1.78 ±0.04	**0.448** ±0.626	3.49 ±1.65	1.35 ±1.12	1.69 ±0.784	2.28 ±1.49
100	2.60 ±0.03	2.19 ±1.11	3.50 ±0.953	**1.42** ±0.752	3.39 ±1.53	3.50 ±1.47

From the results shown above, we can see that MR-MWDO achieves better results than SPSO, HmSO and mQSO for all numbers of peaks except for number of peaks *p*=10 where mQSO achieves slightly better values with highest standard deviation. In general, MR-MWDO achieves intermediate results compared to the other algorithms. The best results are almost shared between CPSOR and CPSO using a hierarchical clustering method to locate and track multiple peaks. These results are very promising and show viability of the proposed approach.

5 Conclusion

In this paper, we described a Multi-Region Modified Wind Driven Optimization algorithm (MR-MWDO) to solve dynamic optimization problems. MR-MWDO employs a new technique called Multi-Region technique inspired from meteorology to detect adaptively promising regions in search space and track multiple peaks. A new Collision Avoidance Technique has been introduced as well. Sub-populations evolve according to a modified WDO algorithm. The method has been assessed using Moving Peak Benchmark and compared to state of the art methods. Obtained results are very promising and show viability of the method. As future work, we intend to foster algorithm search abilities by introducing prediction models to better track moving peaks.

References

1. Calderín, J.F., Masegosa, A.D., Suárez, A.R., Pelta, D.A.: Adaptation Schemes and Dynamic Optimization problems: A Basic Study on the Adaptive Hill Climbing Memetic Algorithm. In: Terrazas, G., Otero, F.E.B., Masegosa, A.D. (eds.) NICSO 2013. SCI, vol. 512, pp. 85–97. Springer, Heidelberg (2014)
2. Yang, S., Yao, X. (eds.): Evolutionary Computation for Dynamic Optimization Problems. SCI, vol. 490. Springer, Heidelberg (2013)
3. Bayraktar, Z., Komurcu, M., Bossard, J.A., Werner, D.H.: The Wind Driven Optimization Technique and its Application in Electromagnetics. IEEE Transactions on Antennas and Propagation 61(5), 2745–2757 (2013)
4. James, R.H.: An Introduction to Dynamic Meteorology, 4th edn., USA, vol. 88 (2004)
5. Chao, C.W., Fang, S.C., Liao, C.J.: A Tropical Cyclone-Based Method For Global Optimization. Journal of Industrial And Management Optimization 8, 103–115 (2012)
6. Nguyen, T.T.: Continuous Dynamic Optimisation Using Evolutionary Algorithms. PhD thesis, School of Computer Science, University of Birmingham (2011)
7. Blackwell, T., Branke, J.: Multiswarms, exclusion, and anti-convergence in dynamic environments. IEEE Trans. Evol. Comput. 10(4), 459–472 (2006)
8. Branke, J.: The moving peaks benchmark, `http://www.aifb.uni-karlsruhe.de/~jbr/MovPeaks/` (viewed November 8, 2008)
9. Branke, J., Schmeck, H.: Designing evolutionary algorithms for dynamic optimization problems. In: Advances in Evolutionary Computing: Theory and Applications, pp. 239–262 (2003)
10. Kamosi, M., Hashemi, A.B., Meybodi, M.R.: A hibernating multi-swarm optimization algorithm for dynamic environments. In: Proc. World Congr. on Nature and Biologically Inspired Computing, NaBIC 2010, pp. 363–369 (2010)
11. Yang, S., Li, C.: A clustering particle swarm optimizer for locating and tracking multi-ple optima in dynamic environments. IEEE Trans. Evol. Comput., 959–974 (2010)
12. Parrott, D., Li, X.: Locating and tracking multiple dynamic optima by a particle swarm model using speciation. IEEE Trans. Evol. Comput. 10(4), 440–458 (2006)
13. Li, C., Yang, S.: A general framework of multipopulation methods with clustering in undetectable dynamic environments. IEEE Trans. Evol. Comput. 16(4), 556–577 (2012)

Evaluating a Hybrid DE and BBO with Self Adaptation on ICSI 2014 Benchmark Problems [⋆]

Yu-Jun Zheng[1] and Xiao-Bei Wu[2]

[1] College of Computer Science & Technology, Zhejiang University of Technology,
Hangzhou 310023, China
[2] College of Electronics and Information Engineering,
Tongji University, Shanghai 201804, China
yujun.zheng@computer.org, xwu4@StateStreet.com

Abstract. The paper presents a new hybrid differential evolution (DE) and biogeography-based optimization (BBO) algorithm and tests its performance on the benchmark set for the ICSI 2014 Competition. The algorithm tends to perform more DE mutations in early search stage and more BBO migrations in later stage, in order to provide a good balance of exploration and exploitation. It also uses a trial-and-error method inspired by the self-adaptive DE (SaDE) to choose appropriate mutation/migration schemes during the search. Computational experiment shows that the algorithm outperforms DE, SaDE, and blended BBO on the benchmark set.

Keywords: single objective optimization, differential evolution (DE), biogeography-based optimization (BBO), self-adaptation.

1 Introduction

Evolutionary algorithms (EAs) are stochastic search methods drawing inspiration from biological evolution for optimization problems. Most EAs are initially proposed for solving single objective optimization problems, which are the basis of a wide range of real-world optimization problems.

The tradeoff between exploration and exploitation is the sticking point in search processes, having a great effect on convergence speed and accuracy of EAs [4]. Among various EAs, differential evolution (DE) [11] is a popular algorithm known for its very competitive exploration ability. Another relatively-new EA, biogeography-based optimization (BBO) [10], has shown strong exploitation ability on a variety of optimization problems.

The purpose of the paper is to present a hybrid algorithm that combines the DE's exploration ability with BBO's exploitation ability, and to evaluate the performance of the hybrid algorithm on the benchmark set for the ICSI 2014 Competition on Single Objective Optimization [12]. The hybrid method, named HSDB (Hybrid Self-adaptive DE and BBO), prefers to make more use of

[⋆] This work was supported by Natural Science Foundation (No. 61105073) of China.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 422–433, 2014.

© Springer International Publishing Switzerland 2014

DE mutations in exploration in early search stage, and is more likely to adopt BBO migrations in exploitation in later search stage. Since the performance of DE/BBO heavily depends on the mutation/migration operators, here we embed two DE mutation schemes and two BBO migration schemes in HSDB, and borrow the ideas from the SaDE algorithm [9] to dynamically choose the schemes during the search. We also compare the performance of HSDB with the DE, SaDE, and an improved BBO on the benchmark set.

The rest of this paper is as follows: Section 2 and Section 3 respectively introduce DE and BBO, Section 4 describes our HSDB algorithm, Section 5 presents the numerical experiments on the benchmark set, and Section 6 concludes.

2 Differential Evolution

DE is a population-based EA that simultaneously evolves a population P of floating-point solution vectors towards the global optimum to the given optimization problem. The most important operator in DE is mutation, which produces a mutant vector $\mathbf{v}_i$ for each individual $\mathbf{x}_i$ in the population $(1 \leq i \leq |P|)$. The most-widely used mutation scheme is the DE/rand/1/bin scheme that adds the weighted difference between two randomly selected vectors to a third one:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \ . \tag{1}$$

where F is the scaling factor typically in the range [0,1], r_1, r_2 and r_3 are three mutually exclusive random indexes in [1, $|P|$].

Afterwards, a trial vector $\mathbf{u}_i$ is generated by mixing the components of the mutant vector and the original one, where each dth component of $\mathbf{u}_i$ is determined as follows:

$$\mathbf{u}_i^d = \begin{cases} \mathbf{v}_i^d & \text{if } rand(0,1) < c_r \text{ or } d = r_i \\ \mathbf{x}_i^d & \text{else} \end{cases} \ . \tag{2}$$

where c_r is the crossover rate ranged in $(0, 1)$ and r_i is a random integer within [1, $|P|$] for each i, ensuring that the trial vector gets at least one component from the mutant vector.

In the last step of each iteration, the selection operator chooses the better one for the next generation by comparing the fitness of $\mathbf{u}_i$ with $\mathbf{x}_i$:

$$\mathbf{x}_i = \begin{cases} \mathbf{u}_i & \text{if } f(\mathbf{u}_i) > f(\mathbf{x}_i) \\ \mathbf{x}_i & \text{else} \end{cases} \ . \tag{3}$$

Nevertheless, there are other mutation schemes that can be implemented in DE. The following presents three other mutation schemes frequently used:

- DE/best/1:

$$\mathbf{v}_i = \mathbf{x}_{\text{best}} + F \cdot (\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) \ . \tag{4}$$

- DE/rand-to-best/2:

$$\mathbf{v}_i = \mathbf{x}_i + F \cdot (\mathbf{x}_{\text{best}} - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \ . \tag{5}$$

- DE/best/2:

$$\mathbf{v}_i = \mathbf{x}_{\text{best}} + F \cdot (\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) + F \cdot (\mathbf{x}_{r_3} - \mathbf{x}_{r_4}) \ . \tag{6}$$

The performance of DE heavily depends on its mutation scheme and control parameter settings. Unfortunately, due to the variety of problems, various conflicting conclusions have been drawn with regard to parameter settings in DE literatures [9]. To avoid manual tuning, Abbass [1] proposed a self-adaptive mechanism that encodes the crossover rate c_r into each solution and simultaneously evolves the parameter and the solution. Omran et al. [8] developed a similar mechanism for adapting the scaling factor F. Qin et al. [9] proposed the SaDE algorithm, which performs a trial-and-error search for the most appropriate mutation scheme during the search. Let K be the total number of mutation schemes in the pool, at the Gth generation, the probability of choosing the kth scheme is proportional to its success rate $s_k(G)$ within the previous LP generations:

$$s_k(G) = \frac{\sum_{g=G-LP}^{G-1} ns_k(g)}{\sum_{g=G-LP}^{G-1} ns_k(g) + \sum_{g=G-LP}^{G-1} nf_k(g)} \ . \tag{7}$$

where $ns_k(g)$ is the number of trial vectors generated by the kth strategy at the gth generation that successfully enter the next generation, and $nf_k(g)$ is the number of trial vectors that fail to do so. SaDE also uses a similar strategy for adapting the c_r values, and sets F as a Gaussian random number with mean 0.5 and standard deviation 0.3, denoted by $N(0.5, 0.3)$.

3 Biogeography-Based Optimization

BBO is also a population-based EA, which was proposed by Simon [10] based on the mathematics of island biogeography. A solution is analogous to a habitat the solution components are analogous to the habitat's suitability index variables (SIVs), and the solution fitness is analogous to the species richness or habitat suitability index (HSI) of the island. High HSI habitats tend to share their features with low HSI habitats, and low HSI habitats are likely to accept many new features from high HSI habitats. For example, in a simple linear migration model, the immigration rate λ_i and the emigration rate μ_i of each habitat $\mathbf{x}_i$ are calculated as follows:

$$\lambda_i = I \frac{f_{\max} - f_i}{f_{\max} - f_{\min}} \ . \tag{8}$$

$$\mu_i = E \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \ . \tag{9}$$

where $f_{\max}$ and $f_{\min}$ are the maximum and minimum fitness value of the population, and I and E are the maximum possible immigration rate and emigration rate which are typically set to 1. However, there are also many other nonlinear migration models can be used [6].

At each generation of BBO, each SIV of a habitat $\mathbf{x}_i$ has a probability of λ_i to be immigrated, and the migrating SIV comes from an emigrating habitat $\mathbf{x}_j$, the selection probability of which is proportional to μ_j of the islands.

BBO also has a mutation operator for setting a SIV to a random value in the search range, which is mainly for increasing solution diversity and hence improving exploration.

In [7] Ma and Simon developed the blended BBO (B-BBO), which replaces the clonal migration of BBO with the following blended migration:

$$X_i^d = \alpha X_i^d + (1 - \alpha) X_j^d \ . \tag{10}$$

where α is a real number between 0 and 1.

Gong et al. [5] proposed a hybrid migration operator by combining BBO migration and DE mutation, where each habitat component has a probability of c_r to be mutated by DE mutation, and a probability of $(1 - c_r)$ to be changed by BBO migration. Experiments show that the hybrid method outperforms both DE and BBO on a set of benchmark problems. Boussaïd et al. [2] proposed another approach combining BBO and DE, which evolves the population by alternately applying BBO and DE iterations, and at each iteration always selects a fitter one from the updated solution and its parent for the next iteration. In [3] the authors extended the approach for constrained optimization, where the original mutation operator of BBO is replaced by the DE mutation operator.

By default, BBO uses a global topology where any two habitats in the population have a chance to communicate with each other. Zheng et al. [13] equipped BBO with a local topology, and any habitat can only immigrate SIVs from its neighboring habitats. In consequence, the flow of information can be moderately pass through the neighborhood, and the algorithm can achieve a better balance between exploration and exploitation. In [14] Zheng et al. enhanced BBO with two new migration operators: global migration and local migration. The former migrates features from a neighbor X_{nb} and a non-neighbor X_{far}, while the latter only considers migration from a neighbor:

$$X_i^d = X_{far}^d + \alpha(X_{nb}^d - X_i^d) \ . \tag{11}$$

$$X_i^d = X_i^d + \alpha(X_{nb}^d - X_i^d) \ . \tag{12}$$

4 A Hybrid DE and BBO with Self Adaptation

The HSDB method differentiates from the hybrid algorithms of Gong et al. [5] and Boussaïd et al. [2] in that it does not combine DE mutation and BBO migration into an integrated operator. Instead, HSDB prefers to make more use of DE mutation in exploration in early search stage, and is more likely to adopt BBO migration in exploitation in later search stage. The key to realize this is a parameter named maturity index, denoted by ν, which increases with the generation: The higher (lower) the ν, the more likely the algorithm is to conduct migration (mutation). A simple approach is to linearly increase ν from an initial value $\nu_{\min}$ to a final value $\nu_{\max}$ as follows (where g is the current generation number and $g^{\max}$ is the total generation number of the algorithm):

$$\nu = \nu^{\min} + \frac{g}{g^{\max}}(\nu^{\max} - \nu^{\min}) \ . \tag{13}$$

At each generation, each habitat has a probability of ν to be modified by BBO migration and a probability of $(1 - \nu)$ by DE mutation. Moreover, we embed a set of DE mutation schemes (denoted by S_{DE}) and a set of BBO migration schemes (denoted by S_{BBO}) in HSDB, for each scheme record its success and failure numbers within a fixed number *LP* of previous generations, and calculate its success rate in a similar way as SaDE [9]. However, in HSDB, we calculate the success rates for DE mutation schemes and BBO migration schemes independently. That is, at beginning every scheme is assigned with an equal selection probability; at each generation $G \geq LP$, the probability of choosing the kth DE mutation scheme and that of the k'th BBO migration scheme are respectively updated as follows:

$$p_k(G) = \frac{s_k(G)}{\sum_{k \in S_{\text{DE}}} s_k(G)} \ . \tag{14}$$

$$p_{k'}(G) = \frac{s_{k'}(G)}{\sum_{k' \in S_{\text{BBO}}} s_{k'}(G)} \ . \tag{15}$$

Currently, in HSDB we embed two DE mutation schemes including DE/rand /1/bin (1) and DE/rand-to-best/2/bin (5), and two BBO migration schemes including the original clonal migration and the local migration (12). This is because the two DE mutation schemes exhibit good exploration ability and the two BBO migration schemes have more exploitation ability than others.

To support local migration, HSDB also implements a local topology to establish the neighborhood structure for the population. Given an expected average neighborhood size K, the local topology, represented by an adjacent 0-1 matrix *NB*, is randomly set by the procedure described in Algorithm 1. As we can see, for each solution in the population p, every other solution has a probability of $K/(|P| - 1)$ to be its neighbor. An advantage of this approach is that the neighborhood size K does not need to be limited to an integer. However, isolated solutions are not allowed. To increase diversity, HSDB will reset its local topology if it fails to find a new better known solution after *NP* generations, where *NP* is a control parameter typically ranges from 1 to 6.

For coherence, we replace the crossover rate c_r of DE by the immigration rate λ_i of each solution i. That is, the DE crossover operation (2) is replaced by the following equation in HSDB:

$$\mathbf{u}_i^d = \begin{cases} \mathbf{v}_i^d & \text{if } rand(0,1) < \lambda_i \text{ or } d = r_i \\ \mathbf{x}_i^d & \text{else} \end{cases} \ . \tag{16}$$

We also use the same parameter scheme for coefficient F in Eq. (1) and (5) and α in Eq. (12), all denoted by F in HSDB. We employ a Gaussian random

Algorithm 1. The procedure for setting the neighborhood topology.

```
Let $NB$ be a $|P| \times |P|$ all-zeros matrix;
Let $p = K/(|P| - 1)$;
for $i = 1$ to $|P|$ do
  for $j = 1$ to $|P|$ do
    if $i \neq j \wedge rand(0,1) < p$ then
      $NB(i,j) = NB(j,i) = 1$;
  if the $i$th solution has no neighbor then
    Randomly choose a $j$ other than $i$ and set $NB(i,j) = NB(j,i) = 1$;
```

number $N(F_\mu, F_\sigma)$ to approximate F, where F_μ and F_σ are typically set as 0.5 and 0.3 (as in SaDE [9]). Moreover, since the original BBO uses mutation mainly for exploration purpose, while here we concentrate on the exploitation ability of BBO, the HSDB method does not use the original BBO mutation scheme. Algorithm 2 describes the framework of HSDB.

Algorithm 2. The HSDB algorithm.

```
Randomly initialize a population $P$ of solutions to the problem;
Initialize the local topology of the population based on Algorithm 1;
while stop criterion is not satisfied do
  for $i = 1$ to $|P|$ do
    Compute $\lambda_i$ and $\mu_i$ for each solution $\mathbf{x}_i$;
    if $rand(0,1) < \nu$ then
      Select a DE mutation scheme $k$ with probability $\propto s_k(G)$;
    else
      Select a BBO migration scheme $k'$ with probability $\propto s_{k'}(G)$;
    Create a mutant vector $\mathbf{v}_i$ using the selected scheme;
    Create a trial vector $\mathbf{u}_i$ according to the crossover operation (16);
    if $\mathbf{u}_i$ is fitter than $\mathbf{x}_i$ then
      Replace $\mathbf{x}_i$ with $\mathbf{u}_i$ in $P$;
  Update the maturity index $\nu$;
  if $G \geq LP$ then
    Update the success rates of the schemes;
  if no new better solution has been found for $NG$ generations then
    Reset the local topology of the population;
return the best solution found so far.
```

5 Computational Experiment

5.1 Experiment Setup

We test the HSDB algorithm on the ICSI 2014 benchmark set, which consists of 30 high-dimensional problems, denoted as f_1–f_{30}, all shifted and rotated. Since all the test problem are considered as black box problems, we do not tune parameters for each problem; instead we use a general parameter setting of HSDB for the whole problem set as: the range of maturity index $\nu_{\min} = 0.05$, $\nu_{\max} = 0.95$, the learning period $LP = 30$, the limit of non-improved generations $NG = 3$, and the neighborhood size $K = 3$. The population size $|P|$ is set to 50.

The experiments are conducted on a computer of Intel Core i5-2430M processor and 4GB DDR3 memory. The HSDB algorithm has been implemented using Matlab R2013a. As requested by the ICSI 2014 competition session, the algorithm has been tested on each problem with dimensions 2, 10 and 30 respectively, the search space is $[-100, 100]^D$, the maximum number of function evaluations is set as $10000D$ (where D denotes the problem dimension), and 51 simulation runs have been performed on each problem instance. Any function value smaller than $2^{-52} \approx 2.22e - 16$ (the ϵ in Matlab) is considered as zero.

In our experimental environment, the mean time for executing the benchmark program of ICSI 2014 competition over 5 runs is $T1 = 48.62$s, and the mean time of HSDB on function 9 and $D = 30$ over 5 runs is $T2 = 116.02$s, and the time complexity is evaluated by the ratio $(T2 - T1)/T1 = 1.386$.

5.2 Experimental Results

Tables 1, 2 and 3 respectively present the computational results of HSDB on 2-D, 10-D and 30-D problems (where 'Std' denotes standard deviation). As we can see, on the simple 2-D problems, HSDB achieves a function error value less than ϵ on 21 problems. With the increase of dimension, the performance of HSDB decreases on most problems. This phenomenon, known as the "curse of dimensionality", is serious on several problems such as f_1 and f_2. However, an interesting finding is that the solution accuracy of the algorithm increases on some special problems including f_{12} and f_{26}.

5.3 Comparative Results

We have compared the performance of HSDB with DE/rand/1/bin scheme, SaDE, and B-BBO on the benchmark set, and conducted nonparametric Wilcoxon rank sum tests on the results of HSDB and that of the other three methods. Due to the page limits, here we only present the comparison results on 30-D problems in Table 4 (where an h value of 1 indicates that the performances of the HSDB and DE/SaDE are statistically different with 95% confidence, h value of 0 implies there is no statistical difference, the superscript $^+$ denotes HSDB has significant performance improvement over DE/SaDE and $^-$ vice versa).

As we can see, HSDB significantly outperforms DE on 28 problems, DE only outperforms HSDB on f_{26}, and there is no statistical difference between them on f_{28}. In comparison with SaDE, HSDB has significant performance improvement on 15 problems; SaDE outperforms HSDB on 6 problems, and there is no statistical difference between them on 9 problems. HSDB also outperforms B-BBO on 29 problems, and there is no statistical difference only on f_{28}. The results show that the overall performance of HSDB is better than the other three methods on the benchmark set.

Table 1. The experimental results on 2-D problems

ID	Max	Min	Mean	Median	Std
f_{1}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{2}	2.98E−27	0.00E+00	1.13E−28	0.00E+00	4.18E−28
f_{3}	1.67E+00	1.67E+00	1.67E+00	1.67E+00	1.12E−15
f_{4}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{5}	1.52E−17	8.00E−20	2.45E−18	1.53E−18	2.44E−18
f_{6}	1.97E−31	1.97E−31	1.97E−31	1.97E−31	0.00E+00
f_{7}	1.21E−17	0.00E+00	6.44E−18	9.65E−18	5.26E−18
f_{8}	8.88E−16	8.88E−16	8.88E−16	8.88E−16	9.96E−32
f_{9}	-4.00E+00	-4.00E+00	-4.00E+00	-4.00E+00	2.24E−15
f_{10}	-9.80E−01	-1.00E+00	-9.96E−01	-1.00E+00	5.86E−03
f_{11}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{12}	5.20E+01	4.86E+01	4.90E+01	4.86E+01	8.74E−01
f_{13}	1.94E−02	0.00E+00	7.00E−03	0.00E+00	9.15E−03
f_{14}	6.99E−02	1.00E−11	5.90E−03	9.41E−04	1.06E−02
f_{15}	3.60E−03	3.91E−08	7.92E−04	3.59E−04	9.40E−04
f_{16}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{17}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{18}	-8.38E+02	-8.38E+02	-8.38E+02	-8.38E+02	4.59E−13
f_{19}	1.35E−32	1.35E−32	1.35E−32	1.35E−32	1.11E−47
f_{20}	5.77E−23	0.00E+00	2.58E−24	1.77E−33	8.18E−24
f_{21}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{22}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{23}	8.97E+00	8.97E+00	8.97E+00	8.97E+00	7.64E−15
f_{24}	-4.53E+00	-4.53E+00	-4.53E+00	-4.53E+00	6.28E−15
f_{25}	-1.39E+00	-1.71E+00	-1.60E+00	-1.71E+00	1.54E−01
f_{26}	6.67E−01	5.71E−01	5.87E−01	5.71E−01	2.79E−02
f_{27}	-3.74E+07	-3.74E+07	-3.74E+07	-3.74E+07	4.01E+02
f_{28}	-4.63E+00	-5.83E+00	-5.22E+00	-5.25E+00	2.63E−01
f_{29}	2.00E+01	2.00E+01	2.00E+01	2.00E+01	8.92E−03
f_{30}	1.01E+00	2.67E−01	4.00E−01	2.70E−01	2.71E−01

Table 2. The experimental results on 10-D problems

ID	Max	Min	Mean	Median	Std
f_1	1.35E+02	5.03E−02	8.23E+00	2.28E+00	1.87E+01
f_2	2.64E+02	2.03E−01	3.26E+01	5.43E+00	5.33E+01
f_3	1.70E+02	1.70E+02	1.70E+02	1.70E+02	1.77E−02
f_4	2.96E+00	6.54E−04	3.45E−01	1.55E−01	4.67E−01
f_5	1.27E−01	1.38E−05	1.14E−02	8.59E−04	2.33E−02
f_6	9.57E+00	4.97E−01	5.02E+00	5.95E+00	2.40E+00
f_7	1.15E−03	1.00E−12	1.46E−04	1.96E−05	2.74E−04
f_8	1.16E+00	5.66E−13	1.71E−01	5.16E−03	3.71E−01
f_9	-1.95E+01	-2.00E+01	-1.98E+01	-1.99E+01	1.35E−01
f_{10}	-7.42E+00	-8.76E+00	-8.34E+00	-8.37E+00	2.85E−01
f_{11}	4.56E+00	0.00E+00	1.09E+00	1.51E−02	1.82E+00
f_{12}	5.40E−01	3.76E−01	4.52E−01	4.47E−01	4.17E−02
f_{13}	6.35E−01	1.55E−01	3.99E−01	4.02E−01	1.37E−01
f_{14}	9.91E−02	1.01E−02	5.17E−02	5.25E−02	2.19E−02
f_{15}	9.83E−02	1.74E−02	4.89E−02	4.84E−02	1.76E−02
f_{16}	4.01E−02	1.14E−08	6.05E−03	1.20E−03	9.83E−03
f_{17}	4.72E−03	3.06E−06	8.07E−04	2.72E−04	1.03E−03
f_{18}	-2.02E+03	-4.14E+03	-2.30E+03	-2.02E+03	6.89E+02
f_{19}	1.62E−03	5.20E−21	6.90E−04	9.55E−04	6.29E−04
f_{20}	3.68E−02	4.98E−06	5.61E−03	1.90E−03	8.75E−03
f_{21}	9.99E−02	9.99E−02	9.99E−02	9.99E−02	7.70E−11
f_{22}	1.08E−02	5.32E−31	8.48E−04	3.56E−05	1.90E−03
f_{23}	2.37E+01	1.55E+01	1.94E+01	1.90E+01	2.01E+00
f_{24}	-3.47E+01	-3.56E+01	-3.53E+01	-3.52E+01	2.40E−01
f_{25}	4.30E+01	4.29E+01	4.29E+01	4.29E+01	2.80E−03
f_{26}	7.59E−03	5.90E−03	6.72E−03	6.70E−03	4.87E−04
f_{27}	-6.34E+07	-1.60E+08	-1.02E+08	-1.07E+08	1.70E+07
f_{28}	-5.04E+00	-5.85E+00	-5.38E+00	-5.40E+00	1.72E−01
f_{29}	2.00E+01	2.00E+01	2.00E+01	2.00E+01	9.31E−04
f_{30}	1.01E+00	1.01E+00	1.01E+00	1.01E+00	1.86E−10

Table 3. The experimental results on 30-D problems

ID	Max	Min	Mean	Median	Std
f_1	1.69E+05	4.08E+03	7.67E+04	7.80E+04	4.15E+04
f_2	1.60E+04	1.34E+03	5.18E+03	3.71E+03	3.27E+03
f_3	4.53E+03	4.51E+03	4.52E+03	4.52E+03	5.11E+00
f_4	1.33E+01	1.59E+00	7.45E+00	7.31E+00	2.71E+00
f_5	7.87E−02	8.39E−03	3.77E−02	3.66E−02	1.72E−02
f_6	3.03E+01	2.82E+01	2.91E+01	2.90E+01	5.15E−01
f_7	1.01E−02	3.23E−03	7.37E−03	7.84E−03	1.72E−03
f_8	1.60E+00	1.86E−01	9.06E−01	9.91E−01	4.42E−01
f_9	-5.76E+01	-5.91E+01	-5.83E+01	-5.82E+01	3.35E−01
f_{10}	-1.99E+01	-2.79E+01	-2.48E+01	-2.54E+01	1.80E+00
f_{11}	8.90E−02	4.55E−04	2.98E−02	2.85E−02	1.82E−02
f_{12}	1.47E−02	1.22E−02	1.43E−02	1.44E−02	4.42E−04
f_{13}	4.02E+00	1.40E+00	3.03E+00	3.14E+00	6.88E−01
f_{14}	1.03E−01	2.53E−02	7.29E−02	7.43E−02	1.96E−02
f_{15}	1.70E−01	2.04E−02	9.48E−02	9.36E−02	3.40E−02
f_{16}	3.33E−01	8.08E−02	2.21E−01	2.24E−01	5.23E−02
f_{17}	3.93E+00	4.31E−01	2.02E+00	2.10E+00	8.17E−01
f_{18}	-3.69E+03	-6.06E+03	-5.79E+03	-6.06E+03	7.11E+02
f_{19}	8.67E−03	2.00E−03	5.05E−03	4.81E−03	1.77E−03
f_{20}	7.45E−01	9.08E−06	2.07E−01	1.11E−01	2.27E−01
f_{21}	3.00E−01	9.99E−02	2.13E−01	2.00E−01	5.30E−02
f_{22}	2.15E−01	1.01E−02	1.01E−01	9.68E−02	5.44E−02
f_{23}	5.68E+01	3.09E+01	4.43E+01	4.32E+01	7.31E+00
f_{24}	-1.13E+02	-1.16E+02	-1.15E+02	-1.15E+02	7.39E−01
f_{25}	1.13E+03	1.13E+03	1.13E+03	1.13E+03	2.06E+00
f_{26}	4.48E−04	4.12E−04	4.33E−04	4.35E−04	8.73E−06
f_{27}	-2.99E+08	-5.25E+08	-4.76E+08	-5.21E+08	8.72E+07
f_{28}	-5.24E+00	-5.81E+00	-5.48E+00	-5.48E+00	1.12E−01
f_{29}	2.00E+01	2.00E+01	2.00E+01	2.00E+01	1.53E−04
f_{30}	1.04E+00	1.01E+00	1.02E+00	1.01E+00	1.34E−02

Table 4. The Comparison of HSDB with DE and SaDE

	DE				SaDE				B-BBO			
ID	Median	Std	p-value	h	Median	Std	p-value	h	Median	Std	p-value	h
f_1	1.73E+06	1.10E+06	3.30E−18	1^+	1.98E+04	2.95E+04	2.12E−08	1^-	3.98E+06	9.84E+05	3.30E−18	1^+
f_2	8.28E+05	1.87E+05	3.30E−18	1^+	2.58E+04	8.72E+03	4.18E−18	1^+	5.72E+04	2.18E+04	3.30E−18	1^+
f_3	5.26E+03	7.97E+02	6.30E−17	1^+	4.51E+03	5.27E+00	1.24E−05	0	4.84E+03	1.30E+02	3.30E−18	1^+
f_4	3.72E+02	6.08E+01	3.30E−18	1^+	1.76E+02	3.06E+01	3.30E−18	1^+	1.36E+02	3.23E+01	3.30E−18	1^+
f_5	1.39E+00	9.06E−01	3.30E−18	1^+	2.31E−02	1.36E−02	2.41E−05	1^-	1.64E+00	3.71E-01	3.30E−18	1^+
f_6	3.77E+01	9.47E+00	3.72E−18	1^+	2.84E+01	4.02E−01	3.77E−08	1^-	6.91E+01	1.79E+01	3.30E−18	1^+
f_7	3.98E−02	3.87E−03	3.30E−18	1^+	8.74E−03	3.81E−03	2.06E−02	1^+	3.10E-02	1.38E-02	3.30E−18	1^+
f_8	2.29E+00	4.90E−01	1.24E−16	1^+	9.81E−01	6.12E−01	8.94E−01	0	3.37E+00	4.41E-01	3.30E−18	1^+
f_9	-5.61E+01	7.52E−01	3.30E−18	1^+	-5.84E+01	3.60E−01	9.70E−02	0	-5.58E+01	5.06E-01	3.30E−18	1^+
f_{10}	-1.03E+01	1.12E+00	3.30E−18	1^+	-1.84E+01	1.10E+00	4.70E−18	1^+	-2.42E+01	2.43E+00	2.66E−03	1^+
f_{11}	2.82E−01	2.92E−01	5.29E−18	1^+	1.21E−02	1.14E−02	8.30E−06	1^-	2.73E+00	8.30E-01	3.30E−18	1^+
f_{12}	1.53E−02	1.57E−04	3.30E−18	1^+	1.48E−02	1.87E−04	3.91E−12	1^+	1.46E-02	6.13E-04	1.44E−03	1^+
f_{13}	7.40E+00	4.03E−01	3.30E−18	1^+	5.16E+00	5.64E−01	3.50E−18	1^+	4.79E+00	5.59E-01	1.32E−16	1^+
f_{14}	1.40E−01	2.29E−02	1.79E−17	1^+	9.16E−02	1.97E−02	1.40E−05	1^+	8.37E-02	2.02E-02	4.94E−03	1^+
f_{15}	1.46E−01	4.37E−02	3.24E−08	1^+	1.19E−01	3.67E−02	4.42E−04	1^+	4.78E-01	8.24E-02	7.51E−18	1^+
f_{16}	5.63E−01	1.50E−01	3.72E−18	1^+	2.07E−01	8.51E−02	5.21E−01	0	1.01E+00	2.24E-01	3.30E−18	1^+
f_{17}	1.69E+02	2.91E+01	3.30E−18	1^+	1.19E+01	2.83E+00	3.30E−18	1^+	1.14E+01	3.15E+00	3.30E−18	1^+
f_{18}	-6.05E+03	1.66E+00	2.67E−12	1^+	-6.06E+03	7.60E+02	8.63E−09	1^+	-3.66E+03	1.91E+03	9.65E−16	1^+
f_{19}	2.85E−01	1.39E−01	3.30E−18	1^+	2.35E−02	1.53E−02	1.79E−17	1^+	1.90E-02	8.13E-03	3.30E−18	1^+
f_{20}	6.04E+02	2.07E+02	3.30E−18	1^+	7.82E+01	3.06E+01	3.30E−18	1^+	1.89E+01	7.50E+00	3.30E−18	1^+
f_{21}	7.78E−01	2.18E−01	5.38E−19	1^+	2.00E−01	6.16E−02	2.85E−01	0	8.00E-01	2.79E-01	9.80E−19	1^+
f_{22}	8.45E−01	6.23E−01	3.30E−18	1^+	1.05E−02	1.94E−02	4.61E−15	1^-	7.75E+00	2.32E+00	3.30E−18	1^+
f_{23}	9.28E+01	5.00E+00	3.30E−18	1^+	6.98E+01	4.62E+00	3.30E−18	1^+	6.69E+01	4.92E+00	4.70E−18	1^+
f_{24}	-1.10E+02	1.49E+00	4.18E−18	1^+	-1.14E+02	8.46E−01	3.63E−01	0	-1.10E+02	1.24E+00	3.30E−18	1^+
f_{25}	1.32E+03	7.89E+01	3.30E−18	1^+	1.13E+03	1.69E+00	4.64E−10	1^-	1.21E+03	2.57E+01	3.30E−18	1^+
f_{26}	3.98E−04	1.19E−06	3.30E−18	1^-	4.26E−04	1.58E−05	1.92E−01	0	6.26E-04	5.67E-05	3.30E−18	1^+
f_{27}	-4.64E+08	3.13E+07	5.36E−07	1^+	-5.19E+08	4.89E+07	3.29E−01	0	-2.69E+08	5.82E+07	3.30E−18	1^+
f_{28}	-5.49E+00	1.26E−01	7.18E−01	0	-5.51E+00	1.44E−01	3.25E−01	0	-5.49E+00	1.14E-01	8.72*E*-01	0
f_{29}	2.00E+01	1.26E−04	5.95E−03	1^+	2.00E+01	1.18E−04	2.77E−02	1^+	2.00E+01	1.57E-04	2.72*E*-02	1^+
f_{30}	1.04E+00	1.14E−02	8.40E−19	1^+	1.01E+00	1.34E−02	4.05E−04	1^+	1.04E+00	8.03E-03	6.07*E*-03	1^+

6 Conclusion

The paper presents a new hybrid DE and BBO algorithm, named HSDB, which uses a trial-and-error method to select among two DE mutation schemes and two BBO migration schemes, and balances the exploration and exploitation based on the maturity index parameter ν. Experiments show that HSDB outperforms DE, SaDE, and B-BBO on the benchmark set for the ICSI 2014 Competition. We are currently including more DE mutation and BBO migration schemes into HSDB and testing more effective method for tuning ν.

References

1. Abbass, H.: The self-adaptive pareto differential evolution algorithm. In: Proceedings of the 2002 Congress on Evolutionary Computation, vol. 1, pp. 831–836 (2002)
2. Boussaïd, I., Chatterjee, A., Siarry, P., Ahmed-Nacer, M.: Two-stage update biogeography-based optimization using differential evolution algorithm (DBBO). Comput. Oper. Res. 38(8), 1188–1198 (2011)
3. Boussaïd, I., Chatterjee, A., Siarry, P., Ahmed-Nacer, M.: Biogeography-based optimization for constrained optimization problems. Comput. Oper. Res. 39(12), 3293–3304 (2012)
4. Chen, J., Xin, B., Peng, Z., Dou, L., Zhang, J.: Optimal contraction theorem for exploration – exploitation tradeoff in search and optimization. IEEE Trans. Syst. Man Cybern. Part A 39(3), 680–691 (2009)
5. Gong, W., Cai, Z., Ling, C.X.: DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. Soft Comput 15(4), 645–665 (2010)
6. Ma, H.: An analysis of the equilibrium of migration models for biogeography-based optimization. Inform. Sci. 180(18), 3444–3464 (2010)
7. Ma, H., Simon, D.: Blended biogeography-based optimization for constrained optimization. Engin. Appl. Artif. Intell. 24(3), 517–525 (2011)
8. 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. LNCS (LNAI), vol. 3801, pp. 192–199. Springer, Heidelberg (2005)
9. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 1785–1791 (2005)
10. Simon, D.: Biogeography-based optimization. IEEE Trans. Evol. Comput. 12(6), 702–713 (2008)
11. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11(4), 341–359 (1997)
12. Tan, Y., Li, J., Zheng, Z.: ICSI 2014 competition on single objective optimization. Tech. rep., Peking University (2014), `http://www.ic-si.org/competition/ICSI.pdf`
13. Zheng, Y.J., Ling, H.F., Wu, X.B., Xue, J.Y.: Localized biogeography-based optimization. Soft Comput. (2014), doi:10.1007/s00500-013-1209-1
14. Zheng, Y.J., Ling, H.F., Xue, J.Y.: Ecogeography-based optimization: Enhancing biogeography-based optimization with ecogeographic barriers and differentiations. Comput. Oper. Res. 50, 115–127 (2014)

The Multiple Population Co-evolution PSO Algorithm

Xuan Xiao and Qianqian Zhang

Beijing Institute of Technology, Beijing, China
{1521494822,289314426}@qq.com

Abstract. In order to overcome the standard particle swarm optimization algorithm which is easily trapped in local minima and optimize the shortcoming of low precision, this paper proposed a way which can make multiple information exchange between particles come true: the multiple population co-evolution PSO algorithm. This paper proposes a multiple population co-evolutionary algorithm to achieve communication among populations, and then show the feasibility and effectiveness of this algorithm through experiments.

Keywords: Particle swarm, co-evolution, PSO multiple population.

1 Introduction

Simulating the behaviors of biological populations has become a research hotspot to solve calculating problems in the field of intelligence calculation. The theory t hat the swarm intelligence is the core has formed, and it has made revolutionary progress in a number of practical applications. Particle swarm optimization(PSO) i s an intelligent optimization algorithm, which is used to handle the problem of co ntinuous variables searching the search space, and it has been applied to many ar eas, such as function optimization, constrained optimization and neural networks. This paper proposes a multiple population co-evolutionary algorithm to achieve c ommunication among populations, and then show the feasibility and effectiveness of this algorithm through experiments.

2 The Description of Basic Particle Swarm Optimization

The basic concept of PSO comes from the study of the preying of birds developed by Kennedy and Eberhart [1, 2]. Imaging a scenario like this: a flock of birds randomly search food in an area, but no one knows where the food are and how far away their current location is from the food. The strategy of flying and searching is to follow the first bird in population. PSO get inspiration from this model and is used to solve optimization problems. Every possible solution is a bird which is called "particles" in search space, and all particles have been evaluated by fitness decided by fitness function. Each particle is used to describe an alternative solution in the solution space,

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 434–441, 2014.
© Springer International Publishing Switzerland 2014

and has a random velocity throughout the whole solution space. Each particle gets heuristic information from each other and guides the movement of the entire group by the exchange of information with other particles.

In the basic PSO algorithm, each particle represents a possible solution and all the particles forms swarm. Particles depend on their own historical information and swarm information in the search space to determine the velocity and direction of flying to find the optimal solution. Assuming that solving problems in *D*-dimension search space, swarm composed of *m* particles, $Swarm=\{x_1^{(k)},x_2^{(k)},\cdots,x_m^{(k)}\}$. At time *k*+1, the position vector is $x_i^{(k+1)}=(x_{i1}^{(k+1)},x_{i2}^{(k+1)},\cdots,x_{iD}^{(k+1)})$, *i*= 1,2, ...,m, which is the location of individuals in the search space, and it is also a possible solution of the problem. Corresponding to the individual position vector is its velocity vector $v_i^{(k+1)}=(v_{i1}^{(k+1)},v_{i2}^{(k+1)},\cdots,v_{iD}^{(k+1)})$, which describes the movement of particles of each dimension in search space. (Note: The superscript of variable represents the iteration cycle, for example, $x_{id}^{(k+n)}$ represents the *k*+*n* cycles; superscript of variable without parentheses represents power, asω^n represents the *n*-th power of ω.)

The neighborhood function of PSO generates a new location status according to each individual's own position vector, velocity vector, individual historical information, group information, and disturbance. In standard PSO algorithm, function calculating formulation of *i*-th particle at time *k*+1 in *d*-dimension is as follows:

$$\begin{cases} v_{id}^{(k+1)}=\omega v_{id}^{(k)}+c_1\cdot r_1\cdot(p_{id}^{(k)}-x_{id}^{(k)})+c_2\cdot r_2\cdot(p_{ld}^{(k)}-x_{id}^{(k)}) \\ x_{id}^{(k+1)}=x_{id}^{(k)}+v_{id}^{(k+1)} \end{cases} . \tag{1}$$

Standard PSO has few parameters which need to be adjusted, and we usually select them empirically:

Population size *m* is generally selected from 10 to 30, the number of particles is enough forthe general problem with small scale, it also reduces the complexity of calculation. The constant limit number of particle velocity *Vmax* determines the maximum moving distance of particles in an iteration cycle. Achieving maximum iterations or meeting the requirement of accuracy is generally selected as terminal conditions.

3 The PSO Algorithm Consensus Analysis and the Consensus Region Boundary

In the fields of biological evolution, consensus problems are especially reflected in self-organizing aggregation of biological systems, such as flocks of birds [3], schools of fish [4], mammals [5]. Jiang and Jin [6, 7] mainly investigate the stochastic convergence of PSO system from the perspective of stochastic process. In order to solve the consensus problem of PSO, we think of a way to improve the velocity and position updating formula of single particle according to the existing standard PSO algorithm, this paper improved PSO by consensus Protocol U:

$$U=v_i^{(k+1)}=\omega v_i^{(k)}+\eta\phi_{i1}^{(k)}\cdot(p_i^{(k)}-x_i^{(k)})+\eta\phi_{i2}^{(k)}\cdot(p_l^{(k)}-x_i^{(k)}) . \tag{2}$$

The new PSO model is further described as:

$$\begin{cases} U = \omega \cdot v_i^{(k)} + \eta \phi_{i1}^{(k)} \cdot (p_i^{(k)} - x_i^{(k)}) + \eta \phi_{i2}^{(k)} \cdot (p_l^{(k)} - x_i^{(k)}) \\ x_i^{(k+1)} = x_i^{(k)} + U \end{cases} . \tag{3}$$

and then,

$$x_i^{(k+1)} = (1 + \omega - \eta \phi_i^{(k)}) x_i^{(k)} - \omega \cdot x_i^{(k-1)} + \eta \phi_i^{(k)} \cdot p_i^{(k)} . \tag{4}$$

Particles in PSO achieve the optimization goal and consensus through mutual cooperation. The cooperative behavior is that the particle and its connected particles(neighbor) pass on information with each other, and change state according to certain strategy and the received neighbor information, resulting in a corresponding self-organizing behavior. Assuming that each particle in PSO population represents a node, then the interactions between particles form a sensing figure, which decodes relationship and interaction between particles and their neighbor.

In the improved consensus protocol U,

$$\begin{cases} 0 \langle \eta \langle 0.25 \\ \eta = 0.25 \\ \eta \rangle 0.25 \end{cases} . \tag{5}$$

The corresponding region for $0 < \eta < 0.25$, $0 \le \phi_i^{(k)} < \frac{1}{\eta}(1+\omega)$, $0 \le \phi_i^{(k)} < 4(1+\omega)$ is in complete consensus. The corresponding region for $4(1+\omega) < \phi_i^{(k)} < \frac{1}{\eta}(1+\omega)$ is inconsistent state area. When η=0.25 , $\phi_i^{(k)} = 4(1+\omega)$ is the critical value for consensus area; when $\eta > 0.25$, the corresponding region for $0 \le \phi_i^{(k)} < 4(1+\omega)$ is complete consensus area.

Based on the research of PSO consistency theory, $\phi_i^{(k)} = 4(1+\omega)$ are the critical value for consensus area. According to the boundary value, aiming at the shortcomings of the particle swarm algorithm itself, this paper puts forward an improved particle swarm optimization algorithm: particle swarm optimization algorithm of multiple population co-evolution.

When w = 0.79, select a few particle swarms , the values of c within these particle swarms are different from each other, and all values are in consensus area. Multiple particle swarms search solution space independently, this way can enhance global searching capability. The results of several test functions using the improved algorithm and the standard particle swarm algorithm demonstrate the effectiveness of the improved algorithm.

4 The Multiple Population Co-evolution PSO Algorithm

Experiment parameter:
Number of population: 3
Population size: 10
Particle neighborhood size: 2
Dimension: 2、10、30

The scope of the search space: [-100 100]^D

Maximum iterations: 10000*dimension

The number of goal: 1

Accuracy: 2.22e-16

Inertia weight: 0.8

Learning factor: for swarm 1, the learning factor is 1.5; for swarm 2 , the learning factor is 1.2; for swarm 3, the learning factor is 0.9.

experimental environment: MATLAB 7.11.0（R2010b）.

steps of algorithm:

Step1: Particle swarm initialization, including population number, population scale, the inertiaweight, learning factor, the initial position and velocity of the particle, etc.

The particle swarm learning factor selection strategy: the inertia weight of each particle group is 0.8, but learning factor is different, if the swarm is *i*, then the learning factor of each swarm is 1.8 -0.3**i* respectively.

Step2: Calculate the fitness of each particle of each group

Group update strategy:

Step3: Compare the fitness of each particle of each group with its previous experience fitness at the best position, if good, then use its current fitness value as the best fitness value of particles.

Step4: Compare the fitness of each particle in the particle group (fitness) with the pbest, if good, it will be updated as the particle group best fitness value.

Step5: Compare the best fitness value of each particle group, select the smallest value as the best fitness value for the particle swarm.

Step6: According to the speed and position update formula, update each particle's velocity and position.

Step7: Algorithm set termination conditions (usually good enough to adapt to the value or reaches the maximum iterations and precision) according to specific problems, if not, then return to step 2; if do, stop the iteration, output the optimal solution.

5 Simulation Experiment and Result Analysis

In simulation experiments, we compare the multiple population co-evolution PSO algorithm procedure with the result of the given example.m, both use 30-dimension in the experiments, the test time is five, using function 9 Weierstrass Function as test function. Run example.m and record the average time of five experiments as T1, run the multiple population co-evolution PSO algorithm program and record the average time of five experiments as T2.

T1=57.7936

T2=76.2999

(T2-T1)/T1=0.320214

For dimension 2、10、30, run the multiple population co-evolution PSO algorithm program 51 times respectively, results are saved in 20140418Bit_2d、20140418Bit_10d、20140418Bit_30d. Maximum, minimum, average, median, standard variance of fitness value are recorded in analysis_2d.csv、analysis_10d.csv、analysis_30d .csv respectively.

Table 1. For 2 dimension

function number	max	min	mean	median	standard deviation
1	0.27622	7.15E-09	0.017526	0.000323	0.052023
2	381.39	0.006224	25.446	6.3034	60.777
3	1.6667	1.6667	1.6667	1.6667	2.24E-16
4	38.605	0.00205	3.9485	0.69603	7.1389
5	4.05E-05	2.18E-12	3.47E-06	2.37E-07	7.69E-06
6	1.73E-21	1.97E-31	5.66E-23	1.29E-28	2.65E-22
7	1.31E-05	1.09E-13	1.59E-06	1.01E-07	3.45E-06
8	4.44E-15	8.88E-16	1.03E-15	8.88E-16	6.96E-16
9	-3.9804	-4	-3.9992	-4	0.002984
10	-0.96952	-1	-0.99485	-0.99821	0.006531
11	2.58E-13	0	6.20E-15	0	3.64E-14
12	54.063	48.585	49.496	49.213	1.0739
13	0.019432	1.61E-08	0.010733	0.019432	0.009166
14	0.007925	5.40E-08	0.000566	0.000182	0.0012
15	0.001914	2.74E-09	0.000241	5.14E-05	0.000388
16	3.45E-16	0	6.76E-18	0	4.83E-17
17	0	0	0	0	0
18	-601.09	-837.97	-771.73	-837.68	102.84
19	1.67E-14	1.35E-32	4.44E-16	1.63E-28	2.39E-15
20	0	0	0	0	0
21	0	0	0	0	0
22	2.59E-31	0	5.08E-33	0	3.62E-32
23	8.9715	8.9715	8.9715	8.9715	0
24	-4.2842	-4.5265	-4.5063	-4.5237	0.043573
25	-1.3921	-1.7107	-1.6329	-1.7106	0.13568
26	0.67181	0.57074	0.60386	0.59685	0.02904
27	-37200000	-37400000	-37300000	-37400000	40169
28	-5.4038	-5.8966	-5.7391	-5.7558	0.11493
29	20.006	19.992	20.001	20.001	0.002386
30	1.0097	0.26706	0.31886	0.26966	0.17126

From analysis_2d.csv we can see that when dimension is 2, the maximum、minimum、average、median and standard deviation of fitness value are shown in table 1.

From analysis_10d.csv we can see that when dimension is 10, the maximum、minimum、average、median and standard deviation of fitness value are shown in table 2.

Table 2. For 10 dimension

Function number	max	min	mean	median	Standard deviation
1	28464	10.482	4164.1	1115.3	6835.1
2	9820.6	206.77	3107.7	2103.8	2614.3
3	169.67	169.55	169.56	169.55	0.023359
4	255.7	10.773	78.632	74.071	51.557
5	0.10652	0.001738	0.025106	0.017069	0.024209
6	9.5293	0.090725	5.3674	6.1252	2.7106
7	0.027443	0.000625	0.008021	0.006973	0.005679
8	4.9899	0.002668	2.4819	2.5799	1.0874
9	-16.232	-19.288	-18.477	-18.623	0.6125
10	-6.3833	-8.7012	-7.9124	-7.9542	0.49787
11	14.881	9.27E-05	5.3959	4.6682	4.4417
12	0.55492	0.41675	0.4764	0.47755	0.029455
13	1.3251	0.077766	0.62293	0.57989	0.30117
14	0.12353	0.01847	0.058184	0.052282	0.025136
15	0.46897	0.015289	0.16643	0.12418	0.13413
16	0.49012	0.011708	0.21558	0.20158	0.10959
17	0.35762	0.000104	0.030355	0.015601	0.05335
18	-534.4	-3963.7	-1939.5	-2018.4	393.32
19	0.046597	0.001339	0.008284	0.005039	0.008649
20	0.020475	7.25E-17	0.001305	1.62E-06	0.004377
21	0.79987	0.099873	0.29399	0.29987	0.13916
22	0.047381	2.56E-06	0.003458	0.000649	0.007831
23	33.908	17.788	24.723	24.623	3.2122
24	-23.297	-34.026	-31.592	-31.87	1.9737
25	42.999	42.943	42.945	42.943	0.007724
26	0.010188	0.006475	0.008403	0.008328	0.000751
27	-101790000	-65600000.00	-28200000.00	-24800000.00	1.31E+07
28	-5.3948	-5.9232	-5.663	-5.642	0.13994
29	20.002	20	20	20	0.000393
30	1.0372	1.0097	1.0183	1.0097	0.012889

From analysis_30d.csv , we can see that when dimension is 30, the maximum、minimum、average、median and standard deviation of fitness value are shown in table 3.

Table 3. For 30 dimension

Function number	max	min	mean	median	Standard deviation
1	1.65E+05	837.96	40120	28993	35732
2	35919	1041.7	9241.1	7925.3	6736.7
3	4554.3	4510.3	4521.8	4520.2	8.5227
4	122.78	22.076	67.24	64.817	21.807
5	0.12922	0.004461	0.040408	0.034891	0.025361
6	33.922	25.335	29.054	28.94	1.1685
7	0.034895	0.008832	0.018335	0.017374	0.00532
8	4.4911	1.5532	3.0353	3.0689	0.69854
9	-49.45	-55.663	-53.521	-53.673	1.1596
10	-20.794	-26.507	-23.939	-24.092	1.2552
11	29.852	0.001599	1.4679	0.03437	4.9221
12	0.014924	0.013084	0.013988	0.014076	0.000392
13	2.9674	0.61538	1.9113	1.8738	0.49718
14	0.036961	0.009863	0.021115	0.02106	0.005493
15	0.48666	0.044493	0.45524	0.4659	0.060491
16	1.4644	0.34813	0.76603	0.73933	0.2357
17	21.569	1.3109	7.1059	6.8476	3.7275
18	-1813.4	-6055.2	-2495	-1908.9	1145.1
19	0.13058	0.006985	0.025035	0.016779	0.023687
20	15.096	6.43E-06	0.64335	0.081668	2.2122
21	1.1999	0.29987	0.7234	0.69987	0.19554
22	0.13694	0.003823	0.048059	0.041579	0.035101
23	50.619	28.644	39.607	39.267	5.4406
24	-101.22	-110.21	-105.56	-105.43	2.1965
25	1151	1124.1	1128.2	1127.2	4.0849
26	0.000504	0.000441	0.000465	0.000459	1.67E-05
27	-47290000	-290950000	-108730000	-101790000	4.88E+07
28	-5.4465	-5.8975	-5.7366	-5.76	0.11652
29	20.005	20.004	20.005	20.005	0.00016
30	1.0782	1.0372	1.0404	1.0372	0.011123

6 Conclusion

This paper proposes a multiple population co-evolution particle swarm optimization algorithm, through which the population information exchanges a lot than before and it is more quickly and efficiently to find the optimal solution. What's more, multiple populations and changes of the learning factor can guarantee the diversity of population, which can effectively improve the inherent defects of the particle swarm algorithm. The experiment results show that the algorithm has good optimization ability, and can get satisfactory results in less time.

References

1. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
2. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp.39–43 (1995)
3. Emlen, J.T.: Flocking behavior in birds. The Auk 69(2), 160–170 (1952)
4. Barlow: Behaviour of teleost fishes. Reviews in Fish Biology and Fisheries 4(1), 126–128 (1994)
5. Gueron, S., Levin, S.A.: R. D. I., The dynamics of herds: from individuals to aggregations. Journal of Theoretical Biology 182(1), 85–98 (1996)
6. Jiang, M., Luo, Y.P., Yang, S.Y.: Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. Information Processing Letters 102(1), 8–16 (2007)
7. Jin, X.L., Ma, L.H., Wu, T.J.: The analysis of pso convergence based on stochastic process. Acta Automatica Sinica 33(12), 1263–1268 (2007)

Fireworks Algorithm and Its Variants for Solving ICSI2014 Competition Problems

Shaoqiu Zheng, Lang Liu, Chao Yu, Junzhi Li, and Ying Tan*

Department of Machine Intelligence, School of EECS, Peking University, China
Key Laboratory of Machine Perception (MOE), Peking University, China
{zhengshaoqiu,langliu,chaoyu,ljz,ytan}@pku.edu.cn

Abstract. Firework algorithm (FWA) is a newly proposed swarm intelligence based optimization technique, which presents a different search manner by simulating the explosion of fireworks to search within the potential space till the terminal criterions are met. Since its introduction, a lot of improved work have been conducted, including the enhanced fireworks algorithm (EFWA), the dynamic search in FWA (dynFWA) and adaptive fireworks algorithm (AFWA). This paper is to use the FWA and its variants to take participate in the ICSI2014 competition, the performance among them are compared, and results on 2-, 10-, 30-dimensional benchmark functions are recorded.

Keywords: ICSI2014 competition, FWA, EFWA, dynFWA, AFWA.

1 Introduction

FWA is a population based swarm intelligence algorithm proposed by Tan and Zhu [16] in 2010. It takes the inspiration from the phenomenon that the fireworks explode and illuminate the local space around the fireworks in the night sky. Its proposed explosion search manner for each firework and cooperative strategy for allocating the resources among the fireworks swarm make it a novel and promising algorithm.

Assume that objective function f is a minimization problem with the form $\min_{x \in \Omega} f(x)$, and Ω is the feasible search region. The conventional FWA works as follows: At first, a fixed number of fireworks (N) are initialized within the feasible search range, and the quality of the fireworks' positions are evaluated, based on which the explosion amplitudes and explosion sparks number are calculated. Here, the principle idea for calculating them is that: the firework with smaller fitness will have larger number of explosion sparks and smaller explosion amplitude, while the firework with larger fitness will have smaller number of explosion sparks and bigger explosion amplitude. In addition, to increase the diversity of the population of the fireworks and explosion sparks, Gaussian mutation sparks are also introduced. After these operations of generating explosion and Gaussian mutation sparks, selection strategy is performed among the candidates set which includes fireworks, explosion sparks and Gaussian mutation

[*] Corresponding author.

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 442–451, 2014.
© Springer International Publishing Switzerland 2014

sparks, and a fixed number of (N) fireworks are selected for the next iteration. The algorithm continues the search until the termination criterions are reached.

Since its first presentation in [16], FWA has attracted a number of researchers to develop the conventional algorithm and apply the algorithm for optimization of real world problems. For the algorithm developments, it includes the single objective algorithm developments [13] [12] [18] [14] [11], multi-objective algorithm developments [21], hybrid version with other algorithms [20] [2] [4] and parallel implementation versions [3]. For the application, FWA has been applied for FIR and IIR digital filters design [4], the initialization of Non-negative Matrix Factorization (NMF) and iterative optimization of NMF [10], [8], [9], spam detection [5], finger-vein identification [19] and power system reconfiguration [7] [6]. Experimental results suggest that FWA is a promising swarm intelligence algorithm, which needs further research and developments.

Motivation and Synopsis: The original motivation of this paper is to let FWA and its variants to participate the competition in ICSI2014 competition, and the performance among some typical improved work are compared. The remainder of this paper is organized as follows: Section 2 briefly introduces the framework of conventional fireworks algorithm, and the FWA variants are presented in Section 3, Experiments are given in Section 4 and finally conclusions are drawn in Section 5.

2 The Conventional FWA

In FWA, it works with a population of fireworks which can generate the explosion sparks and Gaussian mutation sparks thus to maintain the fireworks swarm with global and local search abilities. After generating two kinds of sparks, the selection strategy is performed for the selection of fireworks to the next iteration. Algorithm 1 gives the framework of conventional FWA.

In FWA, to make a contract among the fireworks and balance between the exploration and exploitation capacities, the fireworks are designed to take different explosion amplitudes and explosion sparks number. Assume that the fireworks number is N, then for each firework, the explosion sparks number s_i and explosion amplitude A_i are calculated as following:

$$A_i = \hat{A} \cdot \frac{f\left(X_i\right) - y_{min} + \varepsilon}{\sum_{i=1}^{N}\left(f\left(X_i\right) - y_{min}\right) + \varepsilon}, \tag{1}$$

$$s_i = M_e \cdot \frac{y_{max} - f(X_i) + \varepsilon}{\sum_{i=1}^{N}(y_{max} - f(X_i)) + \varepsilon}, \tag{2}$$

where, $y_{max} = max(f(X_i))$, $y_{min} = min(f(X_i))$, and $\hat{A}$ and M_e are two constants to control the explosion amplitude and the number of explosion sparks, respectively, and ε is the machine epsilon. In addition, to avoid the overwhelming effects of fireworks at good/bad locations, the max/min number of sparks are

Algorithm 1. General structure of conventional FWA

```
Initialize $N$ fireworks $X_i$
repeat
    Explosion operator
        $(i)$ Calculate explosion amplitude $A_i$ and explosion sparks number $s_i$
        $(ii)$ Generate the explosion sparks
    for each firework $X_i$, perform $s_i$ times do
        Initialize the location of the “explosion sparks”: $\hat{X}_i = X_i$
        Calculate offset displacement: $\triangle X = A_i \times rand(-1, 1)$
        $z = round(D * rand(0, 1))$
        Randomly select $z$ dimensions of $\hat{X}_i$
        for each select dimension of $\hat{X}_i^k$ do
            $\hat{X}_i^k = \hat{X}_i^k + \triangle X$
            if $\hat{X}_i^k$ out of bounds then
                $\hat{X}_i^k = X_{min}^k + |\hat{X}_i^k| \ \% \ (X_{max}^k - X_{min}^k)$
            end if
        end for
            $(iii)$ Evaluate fitness of newly created explosion sparks
    end for
    Gaussian mutation operator
        $(i)$ Generate the Gaussian sparks
    for perform $M_g$ times do
        Randomly initialize the location of the “Gaussian sparks”: $\tilde{X}_i = X_i$
        Calculate offset displacement: $e = Gaussian(1, 1)$
        Set $z^k = round(rand(0, 1))$, $k = 1, 2, ..., d$
        for each dimension of $\tilde{X}_i^k$, where $z^k == 1$ do
            $\tilde{X}_i^k = \tilde{X}_i^k \times e$
            if $\tilde{X}_i^k$ out of bounds then
                $\tilde{X}_i^k = X_{min}^k + |\tilde{X}_i^k| \ \% \ (X_{max}^k - X_{min}^k)$
            end if
        end for
    end for
        $(ii)$ Evaluate fitness of newly created Gaussian sparks
    Selection strategy
        $(i)$ Select fireworks for next iteration
until termination is met.
```

bounded by

$$s_i = \begin{cases} round(aM_e) & \text{if } s_i < aM_e, \\ round(bM_e) & \text{if } s_i > bM_e, \\ round(s_i) & otherwise. \end{cases} \tag{3}$$

where, a and b are constant parameters which confine minimal/maximal sparks number (the range of the sparks number). Then for each firework, the explosion sparks are generated according to Algorithm 1.

To increase the diversity, Gaussian mutation sparks are generated based on a Gaussian mutation operator (cf. Algorithm 1).

To retain the information to the next iteration, selection strategy is performed as most of the swarm intelligence algorithms and evolutionary algorithms. In the candidates set, the individual with minimal fitness is always selected while for the rest x_i in candidates set, the selection probability p_i is calculated as

$$p(x_i) = \frac{R(x_i)}{\sum_{j\in K} R(x_j)} \tag{4}$$

$$R(x_i) = \sum_{j\in K} d(x_i, x_j) = \sum_{j\in K} ||x_i - x_j|| \tag{5}$$

where K is the set of all current locations including original fireworks and both types of sparks.

3 The Selected Typical Improvement Work

3.1 Enhanced Fireworks Algorithm

Although FWA has shown its great performance when dealing with function optimization in [16], which outperforms SPSO [1] and CPSO [15] in the selected benchmark functions, in [18], Zheng et al presented a comprehensive study of operators in conventional FWA and proposed the enhanced FWA (EFWA). Some details of the EFWA are as following.

Amplitude of Explosion: In FWA, the explosion amplitude of the best firework is usually very close to 0. In EFWA, a lower bound A_{min} is introduced to avoid this problem:

$$A_i = max(A_i, A_{min}), \tag{6}$$

and A_{min} is non-linearly decreased with the evaluation times going on:

$$A^k_{\min}(t) = A_{init} - \frac{A_{init} - A_{final}}{evals_{max}} \sqrt{(2 * evals_{max} - t)t}, \tag{7}$$

where A_{init} and A_{final} are the initial and final minimum explosion amplitude, $evals_{max}$ is the maximum evaluation times and t is the current evaluation times.

Generating Sparks: In FWA, the number of to-be-mutated dimensions is calculated first and the displacement is calculated just once then used for all the selected dimensions. While in EFWA, for each dimension, an independent displacement $\Delta X^k_i = A_i \times U(-1, 1)$ is calculated with the selection probability $U(0, 1)$, ($U(a, b)$ denotes the generated random number is under the mean distribution between a and b).

$$\hat{X^k_i} = X^k_i + \Delta X^k_i. \tag{8}$$

Moreover, the way to generate Gaussian sparks makes use of the currently best location X_B to avoid the concentrated search on origin region.

$$\tilde{X}_i^k = X_i^k + e * (X_B^k - X_i^k), \tag{9}$$

where $e \sim N(0,1)$.

A new-generated spark will be mapped into a random place in the variable space with uniform distribution if the generated location exceeds the bounds.

$$\hat{X}_i^k = U(X_k^{min}, X_k^{max}). \tag{10}$$

Selection Strategy: To decrease the computational cost of selection strategy in FWA (Eq. 4), in EFWA, the best of the set will be selected first while the rest are randomly selected.

3.2 The dynFWA and AFWA

In FWA and EFWA, the explosion amplitude for fireworks is one of the most key features relevant to the performance. For each firework, its explosion amplitude is calculated by Eq. 1. In fact, the fitness of firework's position is only one kind of information to characterize the local information around X_i, good position needs further local search. However, for an optimization problem, the optimization process is dynamic, the previous static explosion amplitude calculation strategy only suggests that positions of fireworks are good or bad, not the local region within the positions of fireworks. So the explosion amplitude calculation method will lead to a bad local search ability and the experimental results in [18] are consistent with this idea.

For simplicity, the firework with minimal fitness in the fireworks swarm is defined as the core firework (CF, X_{CF}), which has the property that (*i*) its fitness is best among the fireworks, (*ii*) it is always selected to the next iteration. To overcome the limitations presented above, the dynamic search in FWA (dynFWA) and adaptive fireworks algorithm (AFWA) in [14] [11] are proposed respectively.

The dynFWA – Dynamic Explosion Amplitude Strategy: Assume that $f(\hat{X}_{best})$ is the minimal fitness among the explosion sparks and $f(X_{CF})$ denotes the fitness of CF. Here, in dynFWA, it concerns whether the generated explosion sparks can get better fitness than the CF, i.e. the $\Delta_f = f(\hat{X}_{best}) - f(X_{CF})$.

1) $\Delta_f < 0$

It means at least one of the newly generated sparks has smaller fitness than CF's fitness. If so, the $\hat{X}_{best}$ is probably created by the CF or the rest of fireworks other than CF. If $\hat{X}_{best}$ is created by CF, in order to speedup the search for the global optimum, the explosion amplitude of CF will become a bigger one compared with the current value. If $\hat{X}_{best}$ is created by one firework (X_i) other than CF, it has a high chance that the X_i is close to X_{CF}. If X_i is close to X_{CF}, the same explosion amplitude strategy for the CF in the next iteration is taken. If X_i is not close to X_{CF}, then the current explosion amplitude is in fact not effective for the newly generated CF for search any more. However, as it is hard to define the closeness and it is believed that the dynamic explosion amplitude strategy has its ability to adjust the explosion amplitude itself in the following iterations, so dynFWA just sets the explosion amplitude of newly selected CF with a increasing amplitude.

2) $\Delta_f \geq 0$

It means that none of the explosion sparks has found a position with better fitness compared to the CF. The reason for this situation is that the explosion amplitude of firework is too big for CF to search a better position. The CF needs to narrow down the search range. That is to reduce the explosion amplitude thus increasing the probability that the fireworks swarm can find a better position.

In fact, if the CF is far away from the global optimal position, increasing the explosion amplitude is one of the most effective methods to speedup the convergence. The reduction of the explosion amplitude makes it move towards the global optimal position, i.e. the CF finding a better solution.

In FWA and EFWA, to increase the diversity of the fireworks swarm, Gaussian mutation sparks are introduced. However, due to the selection method, the Gaussian mutation sparks do not work effectively as they are designed to, thus in dynFWA [14], they are eliminated.

The AFWA – Adaptive Fireworks Algorithm: The motivation of AFWA is to guarantee the progress made in current iteration is bigger than in the previous iteration [11].

AFWA tries to find the spark whose fitness is the minimal among the candidates whose fitness is worse than CF, and whose Infinite Norm is closest to the best candidate (i.e., Core firework or best spark), then the Infinite Norm between the found candidate with the firework will be taken as the explosion amplitude for the next iteration.

Under this explosion amplitude updating strategy, there are two cases. The first one is that the CF does not generate any good sparks whose fitness is smaller than the firework, then the fitness of all the sparks are larger than the firework, and the explosion amplitude will take the Infinite norm between the firework and the selected candidate. The explosion amplitude in the next iteration will be reduced. For the second case, it has two situations, and the explosion amplitude will be amplified according to the simulation with high chance. Moreover, as the Infinite Norm between the calculated candidate and the firework may change radically, in AFWA, it introduces the smoothing strategy.

4 Experiments

4.1 Experimental Setup

For the implementation of FWA, EFWA, dynFWA and AFWA in this paper, all the parameters are taken from [14] without any modifications. The experimental

Table 1. Run time results on f_9

	T1(s)	T2(s)	(T2-T1)/T1
dynFWA	28.0029	28.8269	0.0294
AFWA	28.0029	28.4186	0.0148

platform used in the experiments is MATLAB 2011b (Windows 7; Intel Core i7-2600 CPU @ 3.7 GHZ; 8 GB RAM) while ICSI-2014 competition problems are used as benchmark functions to compare the performance.

The description of the ICSI-2014 competition benchmark functions is as follows: It contains 30 functions, and for each function, the feasible range is set to $[-100, 100]$. Moreover, to make a comprehensive comparison, in the competition, three groups of experiments with dimension set as $D = 2, 10, 30$, and maximum evaluation times $D * 10000$ are designed. For each function, the max, min, mean, median value and standard deviation of 51 times results are recorded.

4.2 Experimental Results

The experimental results can be found in Table 2, Table 3 and Table 4. For the run time consuming, the experimental runs on f_9 of dynFWA and AFWA are given in Table 1 according to [17].

Table 2. Results for 2D functions

FWA	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Max	1.63E+03	1.89E+03	1.67E+00	1.13E+03	2.09E+00	1.68E-01	2.12E-03	2.58E+00	-3.74E+00	-4.14E-01	9.37E-03	6.62E+01	7.45E-02	8.05E-02	5.83E-02
	Min	8.03E-02	6.43E-01	1.67E+00	5.86E+00	1.39E-03	9.00E-05	2.00E-05	2.00E-05	-4.00E+00	-9.99E-01	1.90E-05	4.88E+01	1.90E-05	1.37E-03	3.40E-05
	Mean	1.97E+02	4.23E+02	1.67E+00	1.38E+02	3.39E-01	4.40E-02	6.91E-04	1.01E-01	-3.92E+00	-8.77E-01	1.04E-03	5.54E+01	2.10E-02	9.88E-03	1.03E-02
	Median	8.13E+01	2.10E+02	1.67E+00	7.99E+01	2.28E-01	2.37E-02	5.77E-04	4.16E-02	-3.93E+00	-9.10E-01	6.47E-04	5.48E+01	1.94E-02	4.46E-03	4.66E-03
	Std	2.96E+02	4.65E+02	2.26E-04	1.83E+02	3.94E-01	4.37E-02	4.89E-04	3.59E-01	6.26E-02	1.17E-01	1.44E-03	4.73E+00	1.14E-02	1.51E-02	1.30E-02
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	4.79E-02	5.94E-01	-6.01E+02	8.60E-03	0.00E+00	2.01E-01	1.62E-01	1.27E+01	-4.01E+00	-1.37E+00	1.03E+00	-3.62E+07	-4.82E+00	2.00E+01	1.01E+00
	Min	1.13E-03	2.32E-04	-8.38E+02	9.00E-06	0.00E+00	2.60E-05	2.00E-06	8.97E+00	-4.52E+00	-1.71E+00	5.74E-01	-3.74E+07	-5.86E+00	2.00E+01	2.68E-01
	Mean	1.54E-02	1.27E-01	-7.89E+02	2.71E-03	0.00E+00	4.35E-02	1.95E-02	9.36E+00	-4.31E+00	-1.50E+00	7.15E-01	-3.72E+07	-5.45E+00	2.00E+01	4.18E-01
	Median	1.34E-02	7.02E-02	-8.38E+02	2.81E-03	0.00E+00	7.93E-03	7.86E-03	8.98E+00	-4.31E+00	-1.39E+00	6.85E-01	-3.73E+07	-5.45E+00	2.00E+01	2.96E-01
	Std	1.10E-02	1.38E-01	9.06E+01	2.19E-03	0.00E+00	5.57E-02	3.19E-02	8.58E-01	1.16E-01	1.41E-01	9.67E-02	2.22E+05	2.54E-01	6.07E-03	2.17E-01
EFWA	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Max	7.31E+02	4.06E+03	1.67E+00	1.27E+03	7.12E-04	7.14E-01	2.28E-02	2.58E+00	-3.99E+00	-6.65E-01	3.13E-06	8.33E+01	3.56E-01	3.16E-01	4.78E-01
	Min	1.16E-04	1.27E-02	1.67E+00	7.69E-01	2.32E-06	2.47E-09	7.02E-07	3.44E-04	-4.00E+00	-1.00E+00	4.02E-10	4.86E+01	6.98E-08	2.30E-02	2.11E-03
	Mean	1.32E+02	1.01E+03	1.67E+00	3.43E+02	8.86E-05	1.40E-02	5.13E-03	5.37E-02	-3.99E+00	-9.45E-01	5.37E-07	5.14E+01	5.16E-02	1.43E-01	4.00E-02
	Median	2.73E+01	6.52E+02	1.67E+00	2.26E+02	5.91E-05	1.23E-06	3.69E-04	2.19E-03	-3.99E+00	-9.70E-01	2.50E-07	4.89E+01	1.94E-02	1.45E-01	3.02E-02
	Std	1.92E+02	1.13E+03	2.93E-09	3.33E+02	1.14E-04	1.00E-01	9.23E-03	3.61E-01	2.63E-03	7.60E-02	6.92E-07	8.41E+00	6.61E-02	7.72E-02	6.60E-02
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	1.88E-04	1.59E-05	-6.01E+02	4.86E-03	6.14E-32	1.08E-07	1.04E-05	1.07E+01	-2.11E+00	7.38E-01	1.31E+00	-1.46E+07	-4.51E+00	2.00E+01	1.12E+00
	Min	1.21E-05	2.68E-08	-8.38E+02	4.91E-10	1.59E-58	3.07E-11	4.61E-09	8.97E+00	-4.52E+00	-1.71E+00	5.72E-01	-3.74E+07	-5.74E+00	1.99E+01	2.67E-01
	Mean	8.94E-05	3.09E-06	-6.11E+02	5.68E-04	1.92E-33	1.46E-08	1.54E-06	9.01E+00	-4.09E+00	-1.32E+00	7.65E-01	-3.01E+07	-5.08E+00	2.00E+01	6.71E-01
	Median	8.51E-05	1.95E-06	-6.01E+02	1.16E-06	6.74E-39	4.82E-09	5.32E-07	8.97E+00	-4.34E+00	-1.39E+00	7.30E-01	-2.36E+07	-5.07E+00	2.00E+01	1.01E+00
	Std	4.64E-05	3.62E-06	4.64E+01	1.57E-03	9.39E-33	2.15E-08	2.04E-06	2.41E-01	6.85E-01	4.69E-01	1.69E-01	7.19E+06	2.68E-01	7.74E-03	3.76E-01
dynFWA	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Max	1.07E+03	5.40E+03	1.67E+00	1.45E+03	3.29E-02	1.81E-02	2.36E-02	1.60E-01	-3.87E+00	-9.13E-01	1.47E-03	5.33E+01	1.94E-02	2.85E-01	6.37E-02
	Min	3.37E-04	6.11E-02	1.67E+00	1.61E-02	5.16E-05	2.33E-07	2.39E-05	2.45E-04	-4.00E+00	-1.00E+00	1.42E-08	4.86E+01	1.90E-06	2.84E-03	7.42E-05
	Mean	1.56E+02	4.90E+02	1.67E+00	1.79E+02	3.85E-03	1.21E-03	9.70E-04	2.45E-02	-3.98E+00	-9.76E-01	7.02E-05	4.93E+01	1.55E-02	6.47E-02	1.74E-02
	Median	6.55E+01	8.70E+01	1.67E+00	3.54E+01	1.47E-03	1.21E-04	3.17E-04	1.05E-02	-3.99E+00	-9.80E-01	1.38E-05	4.91E+01	1.94E-02	4.13E-02	1.30E-02
	Std	2.44E+02	9.25E+02	1.16E-06	3.18E+02	6.32E-03	2.85E-03	3.30E-03	3.95E-02	2.45E-02	2.08E-02	2.09E-04	7.73E-01	7.75E-03	6.18E-02	1.55E-02
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	1.04E-02	7.81E-02	-8.38E+02	5.10E-03	0.00E+00	9.99E-02	1.59E-02	9.13E+00	-4.22E+00	-1.39E+00	9.18E-01	-3.73E+07	-4.73E+00	2.00E+01	1.01E+00
	Min	1.19E-04	7.38E-07	-8.38E+02	4.42E-07	0.00E+00	1.97E-08	1.07E-06	8.97E+00	-4.53E+00	-1.71E+00	5.76E-01	-3.74E+07	-5.89E+00	2.00E+01	2.67E-01
	Mean	2.31E-03	3.63E-03	-8.38E+02	1.14E-03	0.00E+00	1.98E-03	2.27E-03	8.99E+00	-4.42E+00	-1.57E+00	6.68E-01	-3.74E+07	-5.24E+00	2.00E+01	3.39E-01
	Median	1.83E-03	1.54E-04	-8.38E+02	4.24E-04	0.00E+00	4.57E-06	5.31E-04	8.98E+00	-4.46E+00	-1.71E+00	6.45E-01	-3.74E+07	-5.23E+00	2.00E+01	2.72E-01
	Std	2.16E-03	1.24E-02	3.38E-04	1.63E-03	0.00E+00	1.40E-02	4.05E-03	3.21E-02	9.62E-02	1.59E-01	7.80E-02	9.24E+02	2.50E-01	5.86E-03	1.45E-01
AFWA	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Max	1.97E+03	4.52E+03	1.67E+00	1.31E+03	3.04E-03	1.27E+01	2.38E-03	4.44E-15	-3.86E+00	-8.82E-01	2.74E-04	5.82E+01	1.94E-02	2.84E-01	7.07E-02
	Min	5.52E-02	5.52E-02	1.67E+00	5.15E-04	5.37E-07	1.42E-10	0.00E+00	8.88E-16	-4.00E+00	-1.00E+00	1.07E-14	4.86E+01	0.00E+00	5.11E-04	1.89E-05
	Mean	2.90E+02	1.03E+03	1.67E+00	3.00E+02	4.86E-04	5.61E-01	1.35E-04	1.10E-15	-3.99E+00	-9.71E-01	1.98E-05	5.04E+01	1.83E-02	5.75E-02	2.21E-02
	Median	1.32E+02	6.46E+02	1.67E+00	1.66E+02	3.05E-04	1.93E-05	9.65E-18	8.88E-16	-4.00E+00	-9.80E-01	8.55E-07	4.89E+01	1.94E-02	3.36E-02	1.51E-02
	Std	4.12E+02	1.23E+03	4.50E-09	3.53E+02	5.87E-04	2.07E+00	4.24E-04	8.44E-16	2.08E-02	2.56E-02	4.87E-05	2.76E+00	4.62E-03	7.09E-02	2.05E-02
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	0.00E+00	0.00E+00	-6.01E+02	4.85E-03	0.00E+00	0.00E+00	0.00E+00	1.07E+01	-3.12E+00	-6.50E-01	1.31E+00	-3.74E+07	-5.04E+00	2.00E+01	5.57E-01
	Min	0.00E+00	0.00E+00	-8.38E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.97E+00	-4.53E+00	-1.71E+00	5.71E-01	-3.74E+07	-5.93E+00	2.00E+01	2.67E-01
	Mean	0.00E+00	0.00E+00	-8.29E+02	1.22E-03	0.00E+00	0.00E+00	0.00E+00	9.01E+00	-4.35E+00	-1.49E+00	7.05E-01	-3.74E+07	-5.53E+00	2.00E+01	3.11E-01
	Median	0.00E+00	0.00E+00	-8.38E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.97E+00	-4.43E+00	-1.39E+00	6.67E-01	-3.74E+07	-5.54E+00	2.00E+01	2.70E-01
	Std	0.00E+00	0.00E+00	4.45E+01	2.03E-03	0.00E+00	0.00E+00	0.00E+00	2.41E-01	2.61E-01	1.94E-01	1.36E-01	8.99E+01	2.46E-01	8.51E-03	7.74E-02

Table 3. Results for 10D functions

	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FWA	Max	7.42E+06	1.35E+05	1.82E+02	3.52E+02	4.59E+00	1.63E+02	6.03E-02	7.97E+00	-1.51E+01	-3.20E+00	2.57E+01	6.15E-01	1.76E+00	2.06E-01	2.01E-01
	Min	2.25E+05	2.49E+03	1.71E+02	2.28E+01	9.06E-02	5.90E+00	9.08E-03	2.86E+00	-1.88E+01	-8.05E+00	2.36E-01	4.35E-01	2.19E-01	1.28E-02	1.12E-02
	Mean	2.78E+06	4.15E+04	1.74E+02	1.19E+02	1.95E+00	3.83E+01	3.39E-02	5.01E+00	-1.70E+01	-5.67E+00	8.25E+00	5.35E-01	8.91E-01	9.38E-02	7.14E-02
	Median	2.32E+06	3.35E+04	1.74E+02	1.01E+02	1.57E+00	2.84E+01	3.36E-02	4.83E+00	-1.72E+01	-5.77E+00	7.47E+00	5.36E-01	8.52E-01	8.77E-02	5.92E-02
	Std	1.81E+06	2.95E+04	2.57E+00	6.01E+01	1.23E+00	3.35E+01	1.47E-02	1.30E+00	9.77E-01	1.04E+00	6.28E+00	3.99E-02	3.02E-01	4.60E-02	4.34E-02
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	1.05E+00	2.54E+01	-1.66E+03	9.41E-01	4.65E+00	2.20E+00	1.71E+01	3.22E+01	-2.10E+01	4.54E+01	1.53E-02	-1.47E+07	-5.16E+00	2.00E+01	1.08E+00
	Min	9.41E-02	1.65E+00	-4.12E+03	3.36E-02	1.92E-05	9.99E-02	2.61E-01	2.09E+01	-3.27E+01	4.31E+01	8.55E-03	-9.88E+07	-5.93E+00	2.00E+01	1.01E+00
	Mean	4.22E-01	1.14E+01	-2.92E+03	2.65E-01	5.86E-01	7.65E-01	4.39E+00	2.63E+01	-2.78E+01	4.36E+01	1.12E-02	-5.30E+07	-5.53E+00	2.00E+01	1.03E+00
	Median	3.76E-01	1.16E+01	-2.95E+03	1.95E-01	5.78E-02	7.00E-01	3.30E+00	2.70E+01	-2.81E+01	4.35E+01	1.12E-02	-4.67E+07	-5.53E+00	2.00E+01	1.04E+00
	Std	2.07E-01	6.18E+00	1.04E+03	2.09E-01	1.02E+00	4.65E-01	3.29E+00	2.58E+00	2.66E+00	4.20E-01	1.77E-03	2.44E+07	1.97E-01	1.74E-03	1.50E-02
EFWA	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Max	1.37E+05	2.00E+04	1.70E+02	6.60E+01	2.53E-01	1.00E+01	3.07E+00	2.01E+01	-1.22E+01	8.07E+00	4.25E+01	5.84E-01	3.26E+00	2.02E+00	1.67E+00
	Min	1.73E+04	7.47E+02	1.70E+02	3.62E+00	1.07E-02	2.94E+00	1.98E-03	2.99E+00	-1.57E+01	-7.25E+00	3.65E+00	3.90E-01	6.40E-01	8.65E-02	8.09E-02
	Mean	5.83E+04	7.52E+03	1.70E+02	1.89E+01	1.12E-01	7.81E+00	8.51E-01	1.66E+01	-1.48E+01	-2.16E+00	2.04E+01	4.96E-01	2.11E+00	7.24E-01	3.02E-01
	Median	5.47E+04	6.49E+03	1.70E+02	1.71E+01	1.24E-01	8.46E+00	7.29E-01	2.00E+01	-1.51E+01	-2.53E+00	1.79E+01	5.03E-01	2.12E+00	3.40E-01	3.45E-01
	Std	2.81E+04	4.46E+03	4.11E-02	1.37E+01	6.73E-02	1.74E+00	9.63E-01	5.97E+00	8.49E-01	3.63E+00	1.05E+01	4.51E-02	5.64E-01	7.60E-01	2.40E-01
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	1.46E+00	5.94E-02	-3.00E+03	4.82E+00	4.84E+02	6.00E-01	2.28E-01	5.51E+01	-8.86E+00	5.29E+01	1.37E-02	-3.68E+07	-5.13E+00	2.00E+01	1.08E+00
	Min	3.79E-02	7.01E-03	-3.00E+03	1.51E-02	0.00E+00	9.99E-02	2.64E-03	2.36E+01	-3.26E+01	4.29E+01	7.05E-03	-1.07E+08	-5.66E+00	2.00E+01	1.01E+00
	Mean	2.45E-01	2.44E-02	-3.00E+03	8.23E-01	9.57E+00	2.29E-01	9.15E-02	3.88E+01	-2.18E+01	4.37E+01	9.84E-03	-9.75E+07	-5.37E+00	2.00E+01	1.04E+00
	Median	2.11E-01	2.13E-02	-3.00E+03	4.68E-01	3.90E-03	2.00E-01	9.34E-02	3.95E+01	-2.21E+01	4.29E+01	9.81E-03	-1.06E+08	-5.36E+00	2.00E+01	1.04E+00
	Std	2.31E-01	1.26E-02	3.33E-02	9.42E-01	6.77E+01	1.25E-01	6.71E-02	7.04E+00	5.33E+00	2.62E+00	1.57E-03	1.87E+07	1.29E-01	4.34E-04	1.68E-02
dynFWA	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Max	1.32E+05	5.20E+04	1.70E+02	1.86E+02	2.00E-01	9.71E+00	3.32E-01	4.56E+00	-1.69E+01	-5.32E+00	1.52E+01	5.67E-01	1.66E+00	2.03E-01	1.89E-01
	Min	1.80E+03	6.97E+02	1.70E+02	4.78E+00	4.23E-03	8.88E-01	5.74E-04	2.19E-02	-1.94E+01	-8.73E+00	7.35E-04	3.89E-01	2.08E-01	2.83E-02	7.27E-03
	Mean	3.18E+04	9.49E+03	1.70E+02	6.48E+01	6.31E-02	6.56E+00	1.64E-02	2.03E+00	-1.84E+01	-7.24E+00	6.54E+00	4.70E-01	9.80E-01	8.83E-02	1.03E-01
	Median	2.49E+04	6.61E+03	1.70E+02	5.74E+01	5.46E-02	7.04E+00	4.19E-03	2.04E+00	-1.85E+01	-7.34E+00	5.86E+00	4.63E-01	9.96E-01	8.04E-02	1.04E-01
	Std	2.90E+04	9.87E+03	7.52E-02	4.78E+01	5.23E-02	2.51E+00	4.85E-02	1.23E+00	6.30E-01	7.17E-01	4.17E+00	4.42E-02	3.85E-01	4.11E-02	4.01E-02
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	8.67E-01	1.68E-02	-1.68E+03	1.66E-01	8.76E+00	8.00E-01	2.33E-01	4.20E+01	-2.70E+01	5.33E+01	9.77E-03	-1.91E+07	-5.23E+00	2.00E+01	1.04E+00
	Min	3.89E-02	7.06E-05	-4.14E+03	1.32E-03	6.79E-12	9.99E-02	5.62E-05	1.96E+01	-3.51E+01	4.29E+01	6.68E-03	-1.09E+08	-5.94E+00	2.00E+01	1.01E+00
	Mean	2.67E-01	2.57E-03	-2.31E+03	2.42E-02	3.95E-01	2.88E-01	4.80E-02	2.90E+01	-3.29E+01	4.36E+01	7.92E-03	-7.83E+07	-5.77E+00	2.00E+01	1.02E+00
	Median	2.21E-01	2.00E-03	-2.02E+03	1.50E-02	1.75E-03	3.00E-01	1.78E-02	2.87E+01	-3.33E+01	4.29E+01	7.90E-03	-6.71E+07	-5.81E+00	2.00E+01	1.04E+00
	Std	1.89E-01	2.70E-03	7.54E+02	3.26E-02	1.44E+00	1.48E-01	5.98E-02	5.18E+00	1.70E+00	2.41E+00	6.37E-04	2.89E+07	1.37E-01	1.28E-04	1.39E-02
AFWA	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Max	1.06E+06	4.49E+04	1.71E+02	3.50E+02	1.20E+00	8.13E+01	9.26E-01	7.06E+00	-1.57E+01	-3.80E+00	2.51E+01	5.96E-01	2.03E+00	2.11E-01	2.32E-01
	Min	2.07E+03	8.97E+02	1.70E+02	1.04E+01	1.12E-02	1.93E+00	1.69E-03	1.18E+00	-1.95E+01	-8.54E+00	1.85E-03	3.89E-01	3.24E-01	1.16E-02	1.48E-02
	Mean	3.05E+05	1.42E+04	1.70E+02	9.24E+01	2.10E-01	1.11E+01	6.83E-02	3.60E+00	-1.77E+01	-7.00E+00	1.17E+01	5.10E-01	1.27E+00	1.01E-01	1.21E-01
	Median	2.26E+05	1.03E+04	1.70E+02	6.25E+01	8.45E-02	8.53E+00	1.52E-02	3.22E+00	-1.77E+01	-7.17E+00	1.20E+01	5.11E-01	1.29E+00	9.84E-02	1.15E-01
	Std	2.84E+05	1.16E+04	3.96E-01	7.98E+01	2.59E-01	1.24E+01	1.87E-01	1.47E+00	8.05E-01	9.47E-01	6.82E+00	4.17E-02	3.85E-01	4.52E-02	5.26E-02
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	1.23E+00	1.66E+00	-1.23E+03	2.82E-01	2.09E+01	1.90E+00	1.27E+00	4.19E+01	-2.52E+01	5.34E+01	1.15E-02	-1.73E+07	-4.86E+00	2.00E+01	1.04E+00
	Min	5.55E-02	7.31E-02	-4.14E+03	1.69E-03	0.00E+00	9.99E-02	0.00E+00	2.00E+01	-3.50E+01	4.29E+01	6.74E-03	-1.06E+08	-5.92E+00	2.00E+01	1.01E+00
	Mean	3.41E-01	4.24E-01	-2.22E+03	5.90E-02	7.22E-01	5.19E-01	2.78E-01	3.16E+01	-3.10E+01	4.39E+01	8.69E-03	-5.37E+07	-5.70E+00	2.00E+01	1.03E+00
	Median	2.95E-01	3.67E-01	-2.02E+03	3.04E-02	5.00E-03	5.00E-01	1.66E-01	3.17E+01	-3.13E+01	4.29E+01	8.52E-03	-4.54E+07	-5.78E+00	2.00E+01	1.04E+00
	Std	2.47E-01	3.28E-01	7.49E+02	6.69E-02	3.20E+00	3.38E-01	3.17E-01	5.21E+00	2.02E+00	2.99E+00	9.22E-04	2.86E+07	2.15E-01	2.38E-04	1.33E-02

Table 4. Results for 30D functions

	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FWA	Max	1.14E+07	5.51E+05	5.68E+03	2.25E+02	4.87E+00	2.06E+02	1.28E-01	7.67E+00	-4.62E+01	-1.26E+01	6.78E+00	1.54E-02	6.48E+00	7.64E-02	1.20E-01
	Min	2.79E+06	6.02E+04	4.79E+03	7.04E+01	1.07E+00	7.51E+01	3.25E-02	4.36E+00	-5.23E+01	-2.21E+01	1.59E+00	1.39E-02	1.48E+00	4.96E-03	8.57E-03
	Mean	6.27E+06	2.38E+05	5.16E+03	1.57E+02	2.61E+00	1.36E+02	6.65E-02	6.12E+00	-4.85E+01	-1.80E+01	4.38E+00	1.48E-02	3.68E+00	2.98E-02	3.30E-02
	Median	5.37E+06	1.98E+05	5.13E+03	1.55E+02	2.64E+00	1.31E+02	6.47E-02	6.19E+00	-4.86E+01	-1.81E+01	4.42E+00	1.48E-02	3.71E+00	2.85E-02	2.68E-02
	Std	2.36E+06	1.11E+05	2.24E+02	3.50E+01	8.88E-01	3.41E+01	2.08E-02	7.97E-01	1.37E+00	2.51E+00	1.37E+00	3.57E-04	1.19E+00	1.73E-02	2.32E-02
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	2.59E+00	1.07E+02	-8.94E+03	1.34E+00	3.87E+02	3.60E+00	1.91E+01	7.52E+01	-8.74E+01	1.40E+03	9.67E-04	-3.48E+08	-5.29E+00	2.00E+01	1.18E+00
	Min	8.97E-01	1.84E+01	-1.25E+04	3.11E-01	1.29E+00	1.10E+00	4.05E+00	3.67E+01	-1.02E+02	1.18E+03	4.77E-04	-7.54E+08	-5.94E+00	2.00E+01	1.08E+00
	Mean	1.68E+00	5.81E+01	-9.31E+03	7.27E-01	4.80E+01	2.08E+00	1.09E+01	5.72E+01	-9.50E+01	1.28E+03	6.19E-04	-5.17E+08	-5.61E+00	2.00E+01	1.10E+00
	Median	1.71E+00	5.63E+01	-8.97E+03	7.16E-01	3.36E+01	2.10E+00	1.09E+01	5.59E+01	-9.48E+01	1.28E+03	6.05E-04	-4.37E+08	-5.61E+00	2.00E+01	1.08E+00
	Std	3.89E-01	1.73E+01	1.04E+03	2.59E-01	6.31E+01	5.53E-01	3.37E+00	9.05E+00	3.35E+00	4.77E+01	8.01E-05	1.38E+08	1.63E-01	2.22E-04	2.66E-02
EFWA	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Max	3.62E+05	2.71E+04	4.53E+03	2.26E+01	1.62E-01	9.03E+01	3.79E+00	2.02E+01	-4.87E+00	-3.50E-01	1.53E+02	1.51E-02	1.20E+01	2.01E+00	1.92E+00
	Min	4.11E+03	3.27E+03	4.51E+03	2.97E+00	2.26E-02	2.70E+01	9.18E-03	6.94E+00	-4.75E+01	-1.98E+01	8.66E+00	1.29E-02	1.86E+00	5.92E-02	4.26E-02
	Mean	6.63E+04	1.23E+04	4.52E+03	8.91E+00	7.12E-02	3.03E+01	2.12E+00	1.92E+01	-1.07E+01	-1.27E+01	7.82E+01	1.41E-02	7.32E+00	1.94E+00	8.91E-01
	Median	4.60E+04	1.16E+04	4.52E+03	8.13E+00	6.62E-02	2.91E+01	1.99E+00	2.01E+01	-6.34E+00	-1.40E+01	7.43E+01	1.41E-02	7.52E+00	1.98E+00	4.62E-01
	Std	7.26E+04	5.59E+03	5.27E+00	4.43E+00	2.96E-02	8.60E+00	8.40E-01	2.93E+00	1.23E+01	4.22E+00	4.07E+01	4.78E-04	2.70E+00	2.69E-01	6.66E-01
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	1.35E+00	3.12E+00	-9.01E+03	4.99E+00	4.80E-05	1.20E+00	2.84E-01	1.26E+02	-7.58E+01	1.13E+03	4.73E-04	-5.08E+08	-5.31E+00	2.00E+01	1.18E+00
	Min	1.47E-01	1.39E-01	-9.01E+03	1.45E-01	3.00E-06	3.00E-01	1.63E-02	4.95E+01	-9.42E+01	1.12E+03	4.03E-04	-8.61E+08	-5.86E+00	2.00E+01	1.04E+00
	Mean	4.92E-01	1.07E+00	-9.01E+03	1.27E+00	1.20E-05	6.86E-01	6.72E-02	9.27E+01	-8.55E+01	1.13E+03	4.31E-04	-6.11E+08	-5.52E+00	2.00E+01	1.09E+00
	Median	4.11E-01	8.72E-01	-9.01E+03	7.90E-01	8.00E-06	7.00E-01	4.27E-02	9.27E+01	-8.60E+01	1.13E+03	4.28E-04	-5.21E+08	-5.50E+00	2.00E+01	1.08E+00
	Std	2.65E-01	6.63E-01	1.80E-01	1.26E+00	9.00E-06	2.10E-01	6.24E-02	2.04E+01	4.91E+00	1.30E+00	1.70E-05	1.50E+08	1.24E-01	2.31E-04	3.09E-02
dynFWA	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Max	2.43E+05	3.07E+04	4.55E+03	5.21E+01	2.43E-01	9.79E+01	3.16E-02	5.32E+00	-4.86E+01	-1.62E+01	5.21E+01	1.51E-02	5.04E+00	7.61E-02	4.73E-01
	Min	2.61E+03	4.01E+03	4.51E+03	1.04E+01	8.11E-03	2.74E+01	4.82E-03	9.59E-01	-5.63E+01	-2.61E+01	6.59E-04	1.31E-02	1.52E+00	2.39E-02	3.06E-02
	Mean	5.48E+04	1.26E+04	4.52E+03	3.03E+01	7.79E-02	3.64E+01	1.20E-02	2.75E+00	-5.33E+01	-2.22E+01	1.07E+01	1.41E-02	2.98E+00	4.61E-02	2.41E-01
	Median	3.41E+04	1.11E+04	4.52E+03	2.85E+01	6.34E-02	2.94E+01	1.15E-02	2.67E+00	-5.35E+01	-2.25E+01	2.50E+00	1.42E-02	3.00E+00	4.70E-02	1.11E-01
	Std	5.21E+04	6.72E+03	6.35E+00	1.14E+01	5.14E-02	2.01E+01	4.62E-03	8.24E-01	1.76E+00	2.02E+00	1.45E+01	5.07E-04	7.50E-01	1.31E-02	1.88E-01
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	1.21E+00	5.49E+00	-3.68E+03	9.42E-01	4.44E+01	2.00E+00	3.70E-01	7.09E+01	-9.33E+01	1.14E+03	4.72E-04	-3.95E+07	-5.38E+00	2.00E+01	1.13E+00
	Min	1.39E-01	1.73E-01	-6.06E+03	8.69E-03	2.36E-03	4.00E-01	2.52E-03	3.35E+01	-1.11E+02	1.12E+03	4.01E-04	-5.18E+08	-5.96E+00	2.00E+01	1.04E+00
	Mean	6.31E-01	2.17E+00	-5.03E+03	1.95E-01	2.76E+00	8.87E-01	7.58E-02	4.98E+01	-1.04E+02	1.13E+03	4.23E-04	-2.64E+08	-5.80E+00	2.00E+01	1.06E+00
	Median	5.94E-01	1.98E+00	-6.05E+03	1.29E-01	5.70E-01	9.00E-01	3.94E-02	4.97E+01	-1.05E+02	1.13E+03	4.19E-04	-2.90E+08	-5.84E+00	2.00E+01	1.04E+00
	Std	2.65E-01	1.17E+00	1.18E+03	2.15E-01	7.05E+00	3.07E-01	8.57E-02	9.16E+00	3.57E+00	3.47E+00	1.66E-05	1.02E+08	1.40E-01	2.14E-04	2.46E-02
AFWA	F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Max	2.85E+05	2.81E+04	4.55E+03	7.17E+01	2.00E-01	9.34E+01	4.94E-02	5.32E+00	-4.81E+01	-1.47E+01	4.20E+01	1.50E-02	5.82E+00	6.27E-02	4.80E-01
	Min	7.49E+02	2.83E+03	4.51E+03	1.43E+01	1.42E-02	2.69E+01	6.80E-03	2.02E+00	-5.62E+01	-2.59E+01	6.37E-03	1.31E-02	1.34E+00	1.80E-02	1.95E-02
	Mean	5.14E+04	1.19E+04	4.52E+03	3.01E+01	7.07E-02	3.26E+01	1.50E-02	3.19E+00	-5.26E+01	-2.18E+01	5.94E+00	1.42E-02	3.11E+00	4.00E-02	3.07E-01
	Median	1.97E+04	1.02E+04	4.52E+03	2.69E+01	5.79E-02	2.92E+01	1.40E-02	3.03E+00	-5.28E+01	-2.20E+01	1.35E-01	1.42E-02	2.97E+00	3.95E-02	4.22E-01
	Std	6.60E+04	5.84E+03	7.28E+00	1.18E+01	4.48E-02	1.48E+01	6.38E-03	7.93E-01	1.87E+00	2.03E+00	8.29E+00	3.87E-04	9.45E-01	1.20E-02	1.87E-01
	F	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	Max	1.44E+00	6.34E+00	-3.69E+03	6.64E-01	5.80E+01	1.80E+00	1.87E-01	7.59E+01	-9.54E+01	1.14E+03	4.51E-04	-7.18E+07	-5.37E+00	2.00E+01	1.13E+00
	Min	2.31E-01	5.86E-01	-9.01E+03	1.06E-02	7.86E-04	3.00E-01	3.93E-03	3.32E+01	-1.10E+02	1.12E+03	4.03E-04	-8.56E+08	-5.98E+00	2.00E+01	1.04E+00
	Mean	7.35E-01	2.40E+00	-5.39E+03	1.33E-01	3.70E+00	9.78E-01	4.91E-02	5.11E+01	-1.04E+02	1.13E+03	4.23E-04	-3.07E+08	-5.81E+00	2.00E+01	1.07E+00
	Median	7.15E-01	2.21E+00	-6.05E+03	8.82E-02	3.90E-01	1.00E+00	3.38E-02	5.02E+01	-1.04E+02	1.13E+03	4.21E-04	-2.91E+08	-5.85E+00	2.00E+01	1.08E+00
	Std	2.83E-01	1.34E+00	1.45E+03	1.55E-01	9.30E+00	3.48E-01	4.71E-02	8.84E+00	3.47E+00	3.17E+00	1.20E-05	1.53E+08	1.36E-01	2.07E-04	2.76E-02

From the results in different dimension, it can be seen that with the increasing of the dimension, the results optimized by all the algorithms get worsen, which is usually called "*dimension of curse*". From the run time results in Table 1, it can be seen that AFWA achieve smaller $(T2 - T1)/T1$ than dynFWA. Here we also need to point out that the implementation of the code is one of the core factors to influence the run time.

From the results of 2D functions in Table 2, it can be seen that AFWA achieves better results than FWA, EFWA and dynFWA. Especially on f_{16}, f_{17}, f_{20}, f_{21}, f_{22}, AFWA gets the optimum of these functions. Table 3 gives the results of 10D functions. The dynFWA and AFWA still outperform EFWA and FWA. For the comparison between dynFWA and AFWA, dynFWA achieves smaller mean fitness. Table 4 shows the results on 30D functions. None of the algorithms works well, since all the maximum and minimum are different for each function. The dynFWA and AFWA still outperform EFWA and FWA due to their great local search ability, while the performances of dynFWA and AFWA do not differ much.

5 Conclusion

In this paper, the FWA and its variants are used to take the ICSI2014 competition for solving competition problems which contains 30 functions, and the three groups of experimental results with the dimensions set to $2, 10, 30$ are recorded. In the competition, the error smaller than $2^{-52} \approx 2.22e^{-16}$ is set to 0. It can be seen that for some functions, the most recent work dynFWA and AFWA still can not get the optimum, thus further research needs to be taken and it is believed that there is a long way to go for fireworks algorithm in the future.

Acknowledgements. This work was supported by National Natural Science Foundation of China (NSFC), Grant No. 61375119, No. 61170057 and No. 60875080.

References

1. Bratton, D., Kennedy, J.: Defining a standard for particle swarm optimization. In: Swarm Intelligence Symposium, SIS 2007, pp. 120–127. IEEE (2007)
2. Yu, C., Kelley, L., Zheng, S.: Fireworks algorithm with differential mutation for solving the cec 2014 competition problems. In: 2014 IEEE Congress on Evolutionary Computation (CEC). IEEE (2014)
3. Ding, K., Zheng, S., Tan, Y.: A gpu-based parallel fireworks algorithm for optimization. In: Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference, GECCO 2013, pp. 9–16. ACM, New York (2013), http://doi.acm.org/10.1145/2463372.2463377
4. Gao, H., Diao, M.: Cultural firework algorithm and its application for digital filters design. International Journal of Modelling, Identification and Control 14(4), 324–331 (2011)

5. He, W., Mi, G., Tan, Y.: Parameter optimization of local-concentration model for spam detection by using fireworks algorithm. In: Tan, Y., Shi, Y., Mo, H. (eds.) ICSI 2013, Part I. LNCS, vol. 7928, pp. 439–450. Springer, Heidelberg (2013)
6. Imran, A.M., Kowsalya, M.: A new power system reconfiguration scheme for power loss minimization and voltage profile enhancement using fireworks algorithm. International Journal of Electrical Power & Energy Systems 62, 312–322 (2014)
7. Imran, A.M., Kowsalya, M., Kothari, D.: A novel integration technique for optimal network reconfiguration and distributed generation placement in power distribution networks. International Journal of Electrical Power & Energy Systems 63, 461–472 (2014)
8. Janecek, A., Tan, Y.: Iterative improvement of the multiplicative update nmf algorithm using nature-inspired optimization. In: 2011 Seventh International Conference on, Natural Computation (ICNC), vol. 3, pp. 1668–1672. IEEE (2011)
9. Janecek, A., Tan, Y.: Swarm intelligence for non-negative matrix factorization. International Journal of Swarm Intelligence Research (IJSIR) 2(4), 12–34 (2011)
10. Janecek, A., Tan, Y.: Using population based algorithms for initializing nonnegative matrix factorization. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) ICSI 2011, Part II. LNCS, vol. 6729, pp. 307–316. Springer, Heidelberg (2011)
11. Junzhi Li, S.Z., Tan, Y.: Adaptive fireworks algorithm. In: 2014 IEEE Congress on Evolutionary Computation (CEC). IEEE (2014)
12. Liu, J., Zheng, S., Tan, Y.: The improvement on controlling exploration and exploitation of firework algorithm. In: Tan, Y., Shi, Y., Mo, H. (eds.) ICSI 2013, Part I. LNCS, vol. 7928, pp. 11–23. Springer, Heidelberg (2013)
13. Pei, Y., Zheng, S., Tan, Y., Hideyuki, T.: An empirical study on influence of approximation approaches on enhancing fireworks algorithm. In: Proceedings of the 2012 IEEE Congress on System, Man and Cybernetics, pp. 1322–1327. IEEE (2012)
14. Zheng, S., Andreas, J., Li, J., Tan, Y.: Dynamic search in fireworks algorithm. In: 2014 IEEE Congress on Evolutionary Computation (CEC). IEEE (2014)
15. Tan, Y., Xiao, Z.: Clonal particle swarm optimization and its applications. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp. 2303–2309. IEEE (2007)
16. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010, Part I. LNCS, vol. 6145, pp. 355–364. Springer, Heidelberg (2010)
17. Tan, Y., Li, J., Zheng, Z.: Icsi 2014 competition on single objective optimization (2014)
18. Zheng, S., Andreas, J., Tan, Y.: Enhanced fireworks algorithm. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 2069–2077. IEEE (2013)
19. Zheng, S., Tan, Y.: A unified distance measure scheme for orientation coding in identification. In: 2013 IEEE Congress on Information Science and Technology, pp. 979–985. IEEE (2013)
20. Zheng, Y., Xu, X., Ling, H.: A hybrid fireworks optimization method with differential evolution. Neurocomputing (2012)
21. Zheng, Y.J., Song, Q., Chen, S.Y.: Multiobjective fireworks optimization for variable-rate fertilization in oil crop production. Applied Soft Computing 13(11), 4253–4263 (2013)

Performance of Migrating Birds Optimization Algorithm on Continuous Functions

Ali Fuat Alkaya[1], Ramazan Algin[1], Yusuf Sahin[2],
Mustafa Agaoglu[1], and Vural Aksakalli[3]

[1] Marmara University, Department of Computer Engineering, Istanbul, Turkey
[2] Marmara University, Department of Electrical and Electronics Engineering, Istanbul, Turkey
[3] Istanbul Sehir University, Department of Industrial Engineering, Istanbul, Turkey
{falkaya,ysahin,agaoglu}@marmara.edu.tr,
algin.ramazan@gmail.com, aksakalli@sehir.edu.tr

Abstract. In this study, we evaluate the performance of a recently proposed metaheuristic on several well-known functions. The objective of this evaluation is to participate in a competition where several metaheuristics are compared. The metaheuristic we exploit is the recently proposed migrating birds optimization (MBO) algorithm. Our contribution in this study is to develop a novel neighbor generating function for MBO to be used in multidimensional continuous spaces. After a set of preliminary tests presenting the best performing values of the parameters, the results of computational experiments are given in 2, 10 and 30 dimensions.

Keywords: migrating birds optimization, continuous functions, single objective optimization.

1 Introduction

The MBO algorithm is a newly proposed, population-based neighborhood search technique inspired from the V formation flight of the migrating birds which is proven to be an effective formation in energy minimization. In the analogy, initial solutions correspond to a flock of birds. Likewise the leader bird in the flock, a leader solution is chosen and the rest of the solutions is divided into two parts. Each solution generates a number of neighbor solutions. This number is a determiner value on exploration and it corresponds to the speed of the flock. The higher this value, the more detailed the flock explores its surroundings.

The algorithm starts with a number of initial solutions corresponding to birds in a V formation. Starting with the first solution (corresponding to the leader bird) and progressing on the lines towards the tales, each solution is tried to be improved by its neighbor solutions. If any of the neighbor solutions is better, the current solution is replaced by that one. There is also a benefit mechanism for the solutions (birds) from the solutions in front of them. Here we define the

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 452–459, 2014.
© Springer International Publishing Switzerland 2014

benefit mechanism as sharing the best unused neighbors with the solutions that follow. In other words, a solution evaluates a number of its own neighbors and a number of best neighbors of the previous solution and is replaced by the best of them. Once all solutions are improved (or tried to be improved) by neighbor solutions, this procedure is repeated a number of times (tours) after which the first solution becomes the last, and one of the second solutions becomes the first and another loop starts. The algorithm is terminated after a predetermined number of neighbors are generated. Pseudocode of our MBO is given in Figure 1.

```
Generate $n$ initial solutions in a random manner and place them on an hypothetical
$V$ formation arbitrarily
while termination condition is not satisfied
    for $m$ times
        Try to improve the leading solution by generating and evaluating $k$
        neighbors of it
        for each solution $s_i$ in the flock (except leader)
            Try to improve $s_i$ by evaluating ($k$-$x$) neighbors of it and $x$ unused
            best neighbors from the solution in the front
        endfor
    endfor
    Move the leader solution to the end and forward one of the solutions
    following it to the leader position
endwhile return the best solution in the flock
```

Fig. 1. Pseudocode of the MBO

MBO algorithm has four parameters: number of solutions (n), number of tours (m), number of neighbor solutions to be generated from a solution (k) and number of solutions to be shared with the following solution (x). However, due to the inherent design of the algorithm n value has to be equal to or greater than $2 * x + 1$.

This new metaheuristic was proposed by Duman et al. [1]. They applied it to solve quadratic assignment problem instances arising from printed circuit board assembly workshops. Its performance was compared with those of metaheuristics implemented and compared in two previous studies. These metaheuristics are simulated annealing, tabu search, genetic algorithm, scatter search, particle swarm optimization, differential evolution and guided evolutionary simulated annealing. In this comparison, the MBO outperformed the best performed metaheuristic (simulated annealing) in the previous studies by approximately three percent on the average. In addition, MBO was tested with some benchmark problem instances obtained from QAPLIB and in most of the instances it obtained the best known solutions. As a result of these tests, it is concluded that the MBO is a promising metaheuristic and it is a candidate to become one of the highly competitive metaheuristics. Duman and Elikucuk [2] applied MBO to solve fraud detection problem. They also proposed a new version of MBO where a different benefit mechanism is used. They tested the original MBO algorithm and its new version on real data and compared their performance with that of genetic algorithm hybridized with scatter search (GASS). Test results showed that the MBO algorithm and its new version performed significantly better than the GASS algorithm.

In this study, we exploit MBO to solve problems in continuous environments. The set of functions used are given in [3] which are tried to be minimized on 2, 10 and 30 dimensional spaces. The search space is $[-100, 100]^D$ where D is the dimension. We believe that defining an effective neighboring function is much more important than any other modifications on the MBO. In line with this observation, our contribution in this study is to develop a novel neighbor generating function for MBO to be used in multidimensional continuous spaces.

In the next section we present an effective neighbor generating function designed for MBO. Section three presents experimental setup which includes parameter analysis of the MBO algorithm. Section four gives the results where MBO is run on 30 different functions and various dimensions. Section five gives the conclusive remarks together with some future work.

2 A Novel Neighbor Generating Function

In order to design a well performing MBO algorithm, an effective neighbor generating function is essential. To have a more effective exploration plan in the D dimensional solution space, we used D dimensional spheres (D-spheres for short throughout this paper). A neighbor of a solution can be obtained only within the D-sphere around it. A neighbor of a solution can be at most r units away from the original solution where r is the radius of the D-sphere that surrounds it. To find the radius of a D-sphere, we firstly calculate the volume allocated to it using the following formula.

$$V_D = TV/n \tag{1}$$

where V_D is the volume of a D-sphere and TV is the total volume of the solution space. In order to calculate the radii for the D-spheres in a D dimensional space, the volume of the solution space is divided by n. In this way, we try to make an effective exploration and fair distribution of volume for all birds (solutions) to fly around. When the volume of a D-sphere is calculated, we need to find the radius of the D-sphere. The following inductive formulas give the volumes of D-spheres.

$$V_1 = 2 * r \tag{2}$$

$$V_2 = \pi * r^2 \tag{3}$$

$$V_D = V_{D-2} * 2 * \pi / r^D forD > 2 \tag{4}$$

Once the volume for each sphere is calculated, the radii of each sphere can be easily calculated using Equations(2-4). After calculating the radius of D-sphere, we can develop a method to find a neighbor solution (point) within the sphere using some trigonometry. The distance that how far will the new solution be away from the original solution will be a random number in $[0, r]$ where r is the radius of the sphere.

Additionally, we also need to determine the location (coordinate in each axis) of the point in the D dimensional space. For this, we used the following set of trigonometric formula.
$x_D = l * cos(\theta_{D-1})$
$x_{D-1} = l * sin(\theta_{D-1}) * cos(\theta_{D-2})$
$x_{D-2} = l * sin(\theta_{D-1}) * sin(\theta_{D-2}) * cos(\theta_{D-3})$
...
$x_2 = l * sin(\theta_{D-1}) * sin(\theta_{D-2}) * ... * sin(\theta_2) * cos(\theta_1)$
$x_1 = l * sin(\theta_{D-1}) * sin(\theta_{D-2}) * ... * sin(\theta_2) * sin(\theta_1)$

where l is the distance that how far will the new solution be away from the original solution, x_i is the coordinate of the point in the i^{th} axis and θ_i is the angle between i^{th} and $(i+1)^{th}$ axis. Before using this set of formula θ_i's must be obtained randomly such that $\theta_1 \in [0, 2\pi]$ and $\theta_i \in [0, \pi]$ for $i = 2, \ldots, D-1$. An example for the formulas given above is presented in Figure 2 for $D = 3$.

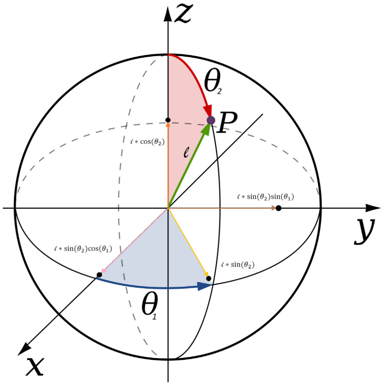


Fig. 2. Representation of a point (solution) and the vectors constituting it in three dimensions

From this setting, one can easily observe that if the number of birds (solutions) is small, then the volume that they are going to explore will be large whereas if the number of birds is large, the volume that they are going to explore will be small. Since we are limited by the number of function evaluation (neighbor generations) due to the competition rules, with a large number of solutions we will be able to explore small number of neighbors in smaller regions whereas with small number of solutions we will be able to explore large number of neighbors in larger regions. Hence, an efficient value for the n parameter must be found for the best performance of the algorithm.

3 Experimental Setup

The experiments are run on an HP Z820 workstation with Intel Xeon E5 processor at 3.0 GHz with 128 GB RAM running Windows 7. The MBO algorithm is implemented in Java language. The stopping criterion for the MBO algorithm is a given number of function evaluations which correspond to number of neighbors generated. Specifically the allowed number of function evaluations is 10000*D.

Table 1. Statistics of 51 runs on 30 different functions when D=2

Function ID	*min*	*max*	*avg*	*med*	*std*
1	40.82	19721.55	1951.86	1113.82	2964.70
2	2.48	4532.26	408.61	98.97	751.68
3	1.67	1.67	1.67	1.67	0.00
4	15.22	4166.54	488.89	227.25	704.77
5	0.17	1.20	0.57	0.59	0.24
6	0.03	3.58	0.95	0.82	0.80
7	0.00	0.01	0.00	0.00	0.00
8	0.91	6.44	3.99	4.17	1.17
9	0.07	0.43	0.26	0.26	0.10
10	0.02	0.39	0.20	0.21	0.10
11	0.01	1.10	0.35	0.23	0.36
12	0.08	1.28	0.55	0.55	0.30
13	0.02	0.08	0.02	0.02	0.01
14	0.23	1.21	0.73	0.75	0.21
15	0.28	2.62	0.98	0.81	0.54
16	0.01	0.15	0.07	0.06	0.04
17	0.01	0.82	0.16	0.12	0.15
18	-861.55	0.00	-840.06	-837.83	5.82
19	0.01	0.28	0.09	0.09	0.06
20	0.39	7.27	2.85	2.97	1.45
21	0.02	1.38	0.51	0.47	0.32
22	0.03	1.44	0.34	0.21	0.34
23	0.32	10.86	7.77	8.34	2.01
24	17.73	7069.62	551.74	261.12	1056.58
25	2.13	2390.53	355.57	138.30	535.51
26	1.33	12.67	2.75	2.03	1.92
27	-41721.46	0.00	-38311.70	-38587.87	2274.27
28	-1.57	0.00	-0.88	-0.87	0.26
29	8.47	17.97	12.75	12.35	2.77
30	0.05	0.84	0.33	0.33	0.16

In order to reveal the best performing parameter values of the MBO on the continuous functions, we run a set of extensive computational experiments. These preliminary tests are conducted on six functions selected out of 30 given in [3]. Best performing values for the parameters are as follows: $n = 5001, k = 3, m = 1, x = 1$.

4 Results

In this section we provide the results of the MBO algorithm on the aforementioned continuous benchmark functions. One of the results to be delivered as a rule of the competition is the $T1$ and $T2$ values. $T1$ is the average run time of five runs of the following piece of MATLAB code in our environment.

Table 2. Statistics of 51 runs on 30 different functions when D=10

Function ID	*min*	*max*	*avg*	*med*	*std*
1	20323203.08	109661239.94	61390076.47	63430416.62	18862528.62
2	240.72	1998.77	801.74	755.39	358.08
3	192.16	304.70	247.67	248.40	21.37
4	354.30	2204.80	1001.13	934.47	468.90
5	1.09	1.64	1.35	1.36	0.12
6	477.57	5023.73	1975.59	1786.44	1048.17
7	0.17	1.53	0.84	0.83	0.25
8	7.04	11.55	9.73	9.74	1.01
9	3.22	5.43	4.75	4.91	0.51
10	1.25	2.12	1.72	1.75	0.22
11	11.01	35.33	27.20	28.07	5.30
12	0.66	2.42	1.54	1.60	0.37
13	1.54	2.58	2.13	2.13	0.25
14	5.30	13.38	8.59	8.56	2.05
15	33.46	214.90	120.04	122.06	42.16
16	1.69	3.46	2.71	2.75	0.43
17	578.73	4028.84	1831.87	1830.40	681.23
18	-3470.64	0.00	-2717.47	-2617.14	376.96
19	1.97	8.05	4.56	4.36	1.30
20	5.90	13.83	9.96	9.91	1.73
21	5.10	19.54	13.41	13.80	3.44
22	38.72	213.23	122.00	126.23	38.49
23	29.19	43.87	38.97	39.58	2.96
24	395.74	1997.54	980.50	935.94	390.40
25	217.89	3921.61	1169.83	1048.91	596.34
26	838.83	6476.15	2421.21	2310.05	1154.72
27	-19655.46	0.00	-13727.18	-13385.63	2203.31
28	14.54	15.72	15.14	15.17	0.29
29	21.37	21.38	21.37	21.37	0.00
30	1.08	1.35	1.22	1.23	0.07

Table 3. Statistics of 51 runs on 30 different functions when D=30

Function ID	*min*	*max*	*avg*	*med*	*std*
1	131323505.00	371642428.18	272954269.37	278704299.47	49176717.62
2	370.11	1232.52	730.82	729.31	191.71
3	19702.97	36266.85	28845.16	29096.35	4050.11
4	397.31	2204.13	920.60	891.43	351.70
5	1.04	1.23	1.10	1.10	0.04
6	1607.06	12377.82	7458.20	7420.84	2176.52
7	1.80	3.42	2.80	2.80	0.39
8	10.20	11.89	11.05	11.11	0.46
9	13.70	17.27	15.64	15.67	0.88
10	2.97	4.82	3.96	4.06	0.42
11	81.94	140.23	114.50	117.65	13.35
12	2.77	4.47	3.74	3.74	0.39
13	8.00	10.33	9.18	9.27	0.58
14	6.30	11.09	8.99	9.18	1.14
15	194.15	371.56	298.56	311.88	42.88
16	5.71	8.75	7.47	7.60	0.72
17	35345.91	77582.40	54386.92	55674.38	10061.42
18	-5215.18	0.00	-4565.75	-4527.92	252.22
19	8.43	17.05	11.84	11.86	1.89
20	5.71	9.16	7.86	8.05	0.70
21	20.15	38.43	30.42	30.40	4.19
22	189.36	418.69	316.94	322.05	59.04
23	100.05	116.60	108.54	108.75	3.99
24	603.02	1419.04	951.32	938.95	199.66
25	11240.74	29044.32	19293.43	19268.09	4740.83
26	1980.60	11722.33	7516.65	8055.23	2296.92
27	-6756.60	0.00	-5080.52	-5009.40	753.36
28	54.70	55.72	55.18	55.18	0.28
29	21.61	21.62	21.61	21.61	0.00
30	1.38	1.47	1.43	1.44	0.02

```
for i=1:300000
    evaluate(9,rand(30,1)*200-100);
end
```

$T2$ is the average run time of five runs of the function 9 on D=30 in our environment.

According to our experimental work $T1$, $T2$ and $(T2\text{-}T1)/T1$ are as follows:

- $T1$=29.965
- $T2$=73.369
- $(T2\text{-}T1)/T1$=1.448

Table 1, 2 and 3 present the statistics when D=2, 10 and 30, respectively. The major observation among the tables is that in higher dimensions the performance of the MBO algorithm gets worse. This is an expected result because the search space grows much faster than the allowed number of function evaluations.

5 Conclusion

In this study we applied migrating birds optimization algorithm to 30 different functions on continuous domain. Our contribution in this study is to develop an effective novel neighbor generation function for MBO. The tests are conducted on 2, 10 and 30 dimensions. Results present that even though MBO is a recently proposed algorithm it is also promising for problems in continuous domain.

References

1. Duman, E., Uysal, M., Alkaya, A.F.: Migrating Birds Optimization: A New Metaheuristic Approach and Its Performance on Quadratic Assignment Problem. Information Sciences 217, 65–77 (2012)
2. Duman, E., Elikucuk, I.: Solving Credit Card Fraud Detection Problem by the New Metaheuristics Migrating Birds Optimization. In: Rojas, I., Joya, G., Cabestany, J. (eds.) IWANN 2013, Part II. LNCS, vol. 7903, pp. 62–71. Springer, Heidelberg (2013)
3. Website of Fifth International Conference on Swarm Intelligence, `http://www.ic-si.org/competition`

Erratum: IIS: Implicit Image Steganography

K. Jithesh[1] and P. Babu Anto[2]

[1] Department of Computer Science, M.G College, Iritty, Kannur Uniersity, Kerala, India
[2] Department of Information Technology, Kannur University, Kerala, India
jithukotheri@gmail.com

Y. Tan et al. (Eds.): ICSI 2014, Part II, LNCS 8795, pp. 275–283, 2014.
© Springer International Publishing Switzerland 2014

DOI 10.1007/978-3-319-11897-0_52

The name of the third author of the paper starting on page 275 of this publication was unfortunately omitted. The header should have been as follows:

K. Jithesh[1], P. Babu Anto[2], and V. Thavavel[3]

[1] Department of Computer Science, M.G College, Iritty, Kannur Uniersity, Kerala, India
[2] Department of Information Technology, Kannur University, Kerala, India
jithukotheri@gmail.com
[3] Prince Sultan University,
Saudi Arabia
thavavelmurugesanv@gmail.com

The original online version for this chapter can be found at
http://dx.doi.org/10.1007/978-3-319-11897-0_33

Author Index