



Computer Science and Information Systems

Published by ComSIS Consortium

**Special Issue on Recent Advances
in Systems and Informatics**

Volume 9, Number 4
December 2012

ComSIS is an international journal published by the ComSIS Consortium

ComSIS Consortium:

University of Belgrade:

Faculty of Organizational Science, Belgrade, Serbia

Faculty of Mathematics, Belgrade, Serbia

School of Electrical Engineering, Belgrade, Serbia

Serbian Academy of Science and Art:

Mathematical Institute, Belgrade, Serbia

Union University:

School of Computing, Belgrade, Serbia

University of Novi Sad:

Faculty of Sciences, Novi Sad, Serbia

Faculty of Technical Sciences, Novi Sad, Serbia

Faculty of Economics, Subotica, Serbia

University of Montenegro:

Faculty of Economics, Podgorica, Montenegro

EDITORIAL BOARD:

Editor-in-Chief: Mirjana Ivanović, University of Novi Sad

Vice Editor-in-Chief: Ivan Luković, University of Novi Sad

Managing Editors:

Gordana Rakić, University of Novi Sad

Miloš Radovanović, University of Novi Sad

Zoran Putnik, University of Novi Sad

Editorial Assistants:

Vladimir Kurbalija, University of Novi Sad

Jovana Vidaković, University of Novi Sad

Ivan Pribela, University of Novi Sad

Slavica Aleksić, University of Novi Sad

Srdan Škrbić, University of Novi Sad

Editorial Board:

S. Ambroszkiewicz, *Polish Academy of Science, Poland*

P. Andrae, *Victoria University, New Zealand*

Z. Arsovski, *University of Kragujevac, Serbia*

D. Banković, *University of Kragujevac, Serbia*

T. Bell, *University of Canterbury, New Zealand*

D. Bojić, *University of Belgrade, Serbia*

Z. Bosnić, *University of Ljubljana, Slovenia*

B. Delibašić, *University of Belgrade, Serbia*

I. Berković, *University of Novi Sad, Serbia*

L. Böszörményi, *University of Clagenfurt, Austria*

K. Bothe, *Humboldt University of Berlin, Germany*

S. Bošnjak, *University of Novi Sad, Serbia*

D. Letić, *University of Novi Sad, Serbia*

Z. Budimac, *University of Novi Sad, Serbia*

H.D. Burkhard, *Humboldt University of Berlin, Germany*

B. Chandrasekaran, *Ohio State University, USA*

G. Devedžić, *University of Kragujevac, Serbia*

V. Devedžić, *University of Belgrade, Serbia*

V. Čirić, *University of Belgrade, Serbia*

D. Domazet, *FIT, Belgrade, Serbia*

J. Đurković, *University of Novi Sad, Serbia*

G. Eleftherakis, *CITY College, International Faculty of the University of Sheffield, Greece*

M. Gušev, *FINKI, Skopje, FYR Macedonia*

S. Guttormsen Schar, *ETH Zentrum, Switzerland*

P. Hansen, *University of Montreal, Canada*

M. Ivković, *University of Novi Sad, Serbia*

L.C. Jain, *University of South Australia, Australia*

D. Janković, *University of Niš, Serbia*

V. Jovanović, *Georgia Southern University, USA*

Z. Jovanović, *University of Belgrade, Serbia*

L. Kalinichenko, *Russian Academy of Science, Russia*

Lj. Kaščelan, *University of Montenegro, Montenegro*

Z. Konjović, *University of Novi Sad, Serbia*

I. Koskosas, *University of Western Macedonia, Greece*

W. Lamersdorf, *University of Hamburg, Germany*

T.C. Lethbridge, *University of Ottawa, Canada*

A. Lojpur, *University of Montenegro, Montenegro*

M. Maleković, *University of Zagreb, Croatia*

Y. Manolopoulos, *Aristotle University, Greece*

A. Mishra, *Atilim University, Turkey*

S. Misra, *Atilim University, Turkey*

N. Mitić, *University of Belgrade, Serbia*

A. Mitrović, *University of Canterbury, New Zealand*

N. Mladenović, *Serbian Academy of Science, Serbia*

S. Mrdalj, *Eastern Michigan University, USA*

G. Nenadić, *University of Manchester, UK*

Z. Ognjanović, *Serbian Academy of Science, Serbia*

A. Pakstas, *London Metropolitan University, UK*

P. Pardalos, *University of Florida, USA*

J. Protić, *University of Belgrade, Serbia*

M. Racković, *University of Novi Sad, Serbia*

B. Radulović, *University of Novi Sad, Serbia*

D. Simpson, *University of Brighton, UK*

M. Stanković, *University of Niš, Serbia*

D. Starčević, *University of Belgrade, Serbia*

D. Surla, *University of Novi Sad, Serbia*

D. Tošić, *University of Belgrade, Serbia*

J. Trninić, *University of Novi Sad, Serbia*

M. Tuba, *University of Belgrade, Serbia*

P. Tumbas, *University of Novi Sad, Serbia*

J. Woodcock, *University of York, UK*

P. Zarate, *IRIT-INPT, Toulouse, France*

K. Zdravkova, *FINKI, Skopje, FYR Macedonia*

ComSIS Editorial Office:

University of Novi Sad, Faculty of Sciences,

Department of Mathematics and Informatics

Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia

Phone: +381 21 458 888; **Fax:** +381 21 6350 458

www.comsis.org; Email: comsis@uns.ac.rs

Volume 9, Number 4, 2012
Novi Sad

Computer Science and Information Systems

Special Issue on Recent Advances in Systems and Informatics

ISSN: 1820-0214

ComSIS Journal is sponsored by:

Ministry of Education, Science and Technological Development of Republic of Serbia -
<http://www.mpn.gov.rs/>



Com
SIS

Computer Science and Information Systems

AIMS AND SCOPE

Computer Science and Information Systems (ComSIS) is an international refereed journal, published in Serbia. The objective of ComSIS is to communicate important research and development results in the areas of computer science, software engineering, and information systems.

We publish original papers of lasting value covering both theoretical foundations of computer science and commercial, industrial, or educational aspects that provide new insights into design and implementation of software and information systems. ComSIS also welcomes survey papers that contribute to the understanding of emerging and important fields of computer science. Regular columns of the journal cover reviews of newly published books, presentations of selected PhD and master theses, as well as information on forthcoming professional meetings. In addition to wide-scope regular issues, ComSIS also includes special issues covering specific topics in all areas of computer science and information systems.

ComSIS publishes invited and regular papers in English. Papers that pass a strict reviewing procedure are accepted for publishing. ComSIS is published semiannually.

Indexing Information

ComSIS is covered or selected for coverage in the following:

- Science Citation Index (also known as SciSearch) and Journal Citation Reports / Science Edition by Thomson Reuters, with 2011 two-year impact factor 0.625,
- Computer Science Bibliography, University of Trier (DBLP),
- EMBASE (Elsevier),
- Scopus (Elsevier),
- Summon (Serials Solutions),
- EBSCO bibliographic databases,
- IET bibliographic database Inspec,
- FIZ Karlsruhe bibliographic database io-port,
- Index of Information Systems Journals (Deakin University, Australia),
- Directory of Open Access Journals (DOAJ),
- Google Scholar,
- Journal Bibliometric Report of the Center for Evaluation in Education and Science (CEON/CEES) in cooperation with the National Library of Serbia, for the Serbian Ministry of Education and Science,
- Serbian Citation Index (SCIndeks),
- doiSerbia.

Information for Contributors

The Editors will be pleased to receive contributions from all parts of the world. An electronic version (MS Word or LaTeX), or three hard-copies of the manuscript written in English, intended for publication and prepared as described in "Manuscript Requirements" (which may be downloaded from <http://www.comsis.org>), along with a cover letter containing the corresponding author's details should be sent to official Journal e-mail.

Criteria for Acceptance

Criteria for acceptance will be appropriateness to the field of Journal, as described in the Aims and Scope, taking into account the merit of the content and presentation. The number of pages of submitted articles is limited to 25 (using the appropriate Word or LaTeX template).

Manuscripts will be refereed in the manner customary with scientific journals before being accepted for publication.

Copyright and Use Agreement

All authors are requested to sign the "Transfer of Copyright" agreement before the paper may be published. The copyright transfer covers the exclusive rights to reproduce and distribute the paper, including reprints, photographic reproductions, microform, electronic form, or any other reproductions of similar nature and translations. Authors are responsible for obtaining from the copyright holder permission to reproduce the paper or any part of it, for which copyright exists.

Computer Science and Information Systems

Volume 9, Number 4, Special Issue, December 2012

CONTENTS

Editorial

Papers

- 1361 SMP-SIM: an SMP-based Discrete-event Execution-driven Performance Simulator**
Yufei Lin, Xinhai Xu, Yuhua Tang, Xin Zhang, Xiaowei Guo
- 1385 Orthogonal Sequences Based Multi-CFO Estimation and Semi-Blind ICA Based Equalization for Multiuser CoMP Systems**
Yufei Jiang, Xu Zhu, Enggee Lim, Yi Huang
- 1407 Speech Unit Category based Short Utterance Speaker Recognition**
Nakhat Fatima, Xiaojun Wu, Thomas Fang Zheng
- 1431 Formal Verification of Signature-monitoring Mechanisms by Model Checking**
Lanfeng Tan, Qingping Tan, Jianjun Xu, Huiping Zhou
- 1453 An Uncertain Optimal Control Model with n Jumps and Application**
Liubao Deng, Yuanguo Zhu
- 1469 A Formal Approach to Testing Programs in Practice**
Shaoying Liu, Wuwei Shen, Shin Nakajima
- 1493 Image Denoising Using Anisotropic Second and Fourth Order Diffusions Based on Gradient Vector Convolution**
Huaibin Wang, Yuanquan Wang, Wenqi Ren
- 1513 Active Semi-supervised Framework with Data Editing**
Wangxin Xiao, Xue Zhang
- 1533 An Ant System based on Moderate Search for TSP**
Ping Guo, and Zhujin Liu
- 1553 Optimal Node Placement of Industrial Wireless Sensor Networks Based on Adaptive Mutation Probability Binary Particle Swarm Optimization Algorithm**
Ling Wang, Wei Ye, Haikuan Wang, Xiping Fu, Minrui Fei, Muhammad Ilyas Menhas
- 1577 Automatic T-S fuzzy Model with Application to Designing Predictive Controller**
Zhi-gang Su, Pei-hong Wang, Yu-fei Zhang

- 1603 Superior Performance of Using Hyperbolic Sine Activation Functions in ZNN Illustrated via Time-Varying Matrix Square Roots Finding**
Yunong Zhang, Long Jin, Zhende Ke
- 1627 Clustering based Two-Stage Text Classification Requiring Minimal Training Data**
Xue Zhang, Wangxin Xiao
- 1645 A Novel Content Based Image Retrieval System using K-means/KNN with Feature Extraction**
Ray-I Chang, Shu-Yu Lin, Jan-Ming Ho, Chi-Wen Fann, Yu-Chun Wang
- 1663 Study on Prediction Models for Integrated Scheduling in Semiconductor Manufacturing Lines**
Lu Guo, Jinghua Hao, Min Liu
- 1679 Application of Grid-based K-means Clustering Algorithm for Optimal Image Processing**
Tingna Shi, Jeenshing Wang, Penglong Wang, Shihong Yue
- 1697 Agent Negotiation on Resources with Nonlinear Utility Functions**
Xiangrong Tong, Wei Zhang
- 1721 University Campus Social Network System for Knowledge Sharing**
Zhao Du, Xiaolong Fu, Can Zhao, Ting Liu, Qifeng Liu

EDITORIAL

As computer-based systems and informatics-related technologies proliferate, more and more research and development efforts have been made in recent years. The 1st International Conference on Systems and Informatics (ICSAI, May 2012, Yantai, China) aims to bring together researchers and engineers around the world to showcase their work in systems and informatics. We were delighted to receive well over one thousand submissions from 33 countries, out of which 587 papers were selected for inclusion in the conference program. The 18 papers in this special issue are extended from some of the best submissions to the conference and represent a cross-section of these exciting fields.

Yufei Lin et al. devise a discrete-event execution-driven performance simulation method based on symmetric multi-processors for designing and implementing a large-scale parallel system. Yufei Jiang et al. propose a low-complexity carrier frequency offset estimation approach and an independent component analysis-based semi-blind equalization structure for the multiuser coordinated multi-point (CoMP) communication system. Nakhath Fatima et al. investigate speech unit category-based short utterance speaker recognition. Lanfang Tan et al. study formal verification of signature-monitoring mechanisms by model checking. Liubao Deng and Yuanguo Zhu discuss an uncertain optimal control model with n jumps and its applications. A formal approach to testing programs in practice is presented by Shaoying Liu et al. Huaibin Wang et al. discuss image denoising using anisotropic second and fourth order diffusions based on gradient vector convolution.

Xue Zhang and Wangxin Xiao present an active semi-supervised framework with data editing in order to address the insufficient training data problem. Ping Guo and Zhujin Liu propose an ant system based on moderate search and demonstrate the technique in the traveling salesman problem. Ling Wang et al. solve the optimal node placement problem in industrial wireless sensor networks using an adaptive mutation probability binary particle swarm optimization algorithm. Zhi-gang Su et al. propose an automatic Takagi-Sugeno fuzzy model with applications to designing predictive controllers. Yunong Zhang et al. discuss superior performance of using hyperbolic sine activation functions in ZNN and illustrate it via time-varying matrix square roots finding. Clustering-based two-stage text classification requiring minimal training data is studied by Xue Zhang and Wang-xin Xiao. Ray-I Chang et al. present a content-based image retrieval system using K-means/K nearest neighbors (KNN) with feature extraction. Lu Guo et al. investigate prediction models for integrated scheduling in semiconductor manufacturing lines. Tingna Shi et al. apply grid-based K-means clustering for optimal image

processing. Xiangrong Tong et al. discuss agent negotiation on resources with nonlinear utility functions. Zhao Du et al. study university campus social networks for knowledge sharing.

We thank Professor Mirjana Ivanović, Editor-in-Chief of the Computer Science and Information Systems journal, for her support and comments regarding this special issue. We are grateful to the authors of this special issue for contributing their excellent work and to the reviewers for their timely evaluations.

Xu Zhu,
Wuwei Shen,
Lipo Wang
Editors of the special issue

SMP-SIM: an SMP-based Discrete-event Execution-driven Performance Simulator

Yufei Lin, Xinhai Xu, Yuhua Tang, Xin Zhang, and Xiaowei Guo

State Key Laboratory of High Performance Computing
National University of Defense Technology, Changsha, China
{linyufei,xuxinhai,yhtang}@nudt.edu.cn,zx2010@jfjb.com.cn, xiaow_g@126.com

Abstract. Designing and implementing a large-scale parallel system can be time-consuming and costly. It is therefore desirable to enable system developers to predict the performance of a parallel system at its design phase so that they can evaluate design alternatives to better meet performance requirements. Before the target machine is completely built, the developers can always build an symmetric multi-processor (SMP) for evaluation purposes. In this paper, we introduce an SMP-based discrete-event execution-driven performance simulation method for message passing interface (MPI) programs and describe the design and implementation of a simulator called SMP-SIM. As the processes share the same memory space in an SMP, SMP-SIM manages the events globally at the granularity of central processing units (CPUs). Furthermore, by re-implementing core MPI point-to-point communication primitives, SMP-SIM handles the communication virtually and sequential computation actually. Our experimental results show that SMP-SIM is highly accurate and scalable, resulting in errors of less than 7.60% for both SMP and SMP-Cluster target machines.

Keywords: simulator, SMP, MPI, performance prediction.

1. Introduction

Nowadays, large-scale parallel computers that comprise thousands of processors cost millions of dollars and take years to design and build. For system developers, it is greatly desired that the performance of a parallel system can be predicted efficiently and accurately at its design phase. This can help them evaluate different design alternatives to better meet performance requirements [25].

There are two popular performance prediction methods: model-based and discrete event simulation. The former [3, 11] builds a parameterized model based on the signatures extracted from the given system, such as the number of cores, the number of instructions executed, and memory access patterns. Then the model is used to estimate the system performance. However, a model built for an application cannot be applied to another one. Moreover, as the system complexity increases, the factors that affect the system performance become too complex to be extracted thoroughly. So it can be difficult to obtain accurate predictions for large-scale parallel computers with the model-based method.

Discrete event simulation is capable of simulating large-scale parallel systems. The simulation procedure jumps from one state to another upon the occurrence of an event [5]. A simulator updates the simulation time and state information by processing the events. Events are generated either by the pre-extracted trace (trace-driven) or by the execution of the program (execution-driven). For trace-driven techniques, such as Dimemas [12], PERC [19] and SIM-MPI [18], the event trace is obtained with instrumentation tools by running the program before the simulator runs. For execution-driven techniques, such as WWT [17], WWT II [14], MPI-SIM [15] and BigSim [26], the events are generated when the program is running.

In discrete event simulation, the large-scale parallel machine to be conducted is called the **target machine**, the machine on which the simulator executes is called the **host machine**, and the program whose performance is to be predicted is called the **target program**. Through analysis, we find that before the target machine is completely constructed, the system developers can always build a symmetric multi-processor (SMP), whose central processing units (CPUs) are the same as those in the target machine. If the target machine is an SMP, such as NEC SX-9 [20], CRAY CX1000-S [4] or IBM SP-SMP [8], the developers can use some of the target machine's CPUs to build a smaller SMP. If the target machine is an SMP-Cluster, a cluster whose nodes are SMPs, such as Roadrunner [2], Tianhe-1A [24] or K Computer [1], the developers can have at least one of the SMP nodes at the design phase. Therefore, we have opted to use the smaller SMP (or the SMP node) as the host machine to predict the performance of the target program on the target machine.

SMPs are well suited to discrete event simulation since the processes share the memory and are controlled by the same operating system. However, there has not been a simulation method utilizing these characteristics. Thus, in this paper, for message passing interface (MPI) [22], the de facto standard for parallel computing, we propose and design SMP-SIM, an SMP-based discrete-event execution-driven performance simulator. Our main contributions are as follows:

- We introduce a discrete-event execution-driven simulation method based on SMPs.
- We describe the design of SMP-SIM, a discrete-event execution-driven performance simulator, which consists of three modules, primitive decomposer, communication model, and event management.
- By re-implementing MPI core point-to-point communication primitives, we have implemented SMP-SIM based on MPICH2 [13].
- We show that SMP-SIM is highly accurate and scalable, with errors of less than 7.60% for both SMP and SMP-Cluster target machines.

The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 describes the basic ideas of SMP-SIM. Section 4 introduces the framework of SMP-SIM. Section 5 discusses the implementation of SMP-SIM based on MPICH2. Our experiments are presented in Section 6. Finally, Section 7 summarizes the paper.

2. Related Work

Discrete event simulation has been researched extensively in academia and industries. A number of simulators have been developed, including WWT [17] and WWT II [14] of University of Wisconsin, Dimemas [12] of CEPBA (European Center for Parallelism in Barcelona), MPI-SIM [15] of University of California, PERC [19] of San Diego Supercomputer Center, and BigSim [26] of University of Illinois at Urbana-Champaign. Among these simulators, MPI-SIM and BigSim are the most similar with SMP-SIM. All the three predict the performance while executing the target MPI applications.

In MPI-SIM, each process of the target program is simulated by a thread. MPI-SIM uses a portable library MPI-LITE to translate the target program into a multithreaded program, and measures the execution time of sequential computation codes when the multithreaded program is executing. However, threads differ from processes in terms of such program behaviours such as cache replacement policy and execution pattern. As a result, the sequential computation time cannot be measured accurately. Moreover, because MPI-SIM is not based on the SMP host machine, messages must carry the simulation timestamps that are used to calculate communication overheads.

BigSim is a simulator developed for BlueGene/C. The simulator contains two parts, a parallel-function emulator and a parallel-network simulator BigNet-Sim [27]. BigSim defines a set of application interfaces, such as `addMessage` and `sendPacket`, which are used to implement the MPI interfaces. All the application interfaces are executed by the simulator, and the other codes are directly executed on the host machine. In BigSim, the sequential computation time is calculated by heuristic approaches, which cannot lead to high accuracy. Several parallel programming languages are implemented on BigSim, including MPI, CHARM++ [10] and Adaptive MPI [7].

Zhai et al. [25] proposed a new method, called Phantom, to estimate the sequential computation time, which is used in a trace-driven simulator SIM-MPI [18]. Phantom needs to execute the target program twice. During the first execution, the communication traces of parallel applications are generated by intercepting all communication operations for each process and the computation between communication operations is marked as sequential computation unit. During the second execution, the real sequential computation time is measured on a target processing node for each process one by one. However, executing the target program twice is time-consuming. And it cannot deal with the uncertain programs because an inaccurate trace is usually generated.

The above simulation approaches are not based on SMP. To the best of our knowledge, there has not been an SMP-based performance simulator for MPI programs.

3. Basic Ideas of SMP-SIM

To design a discrete event simulator, three key questions need to be answered: what events exist in the simulator; how are the events generated; and how the

events are processed for performance prediction. In this section, we introduce the basic idea behind SMP-SIM by addressing these questions. Without loss of generality, we assume that the core count of the target machine is equal to the process count of the target program. It should be noticed that as mentioned in Section 1, the CPUs in the host and target machines are the same.

3.1. Event Definition

For an MPI program, at any point during execution, a process is in either the *sequential computation state* or the *communication state*. So there are four types of events during the simulation of an MPI program: **simulation start events**, **simulation end events**, **communication start events** and **sequential computation start events**. A simulation start event means that the simulation starts, i.e. a process enters the sequential computation state. A simulation end event means that the simulation ends, i.e. a process ends. A communication start event means that the process enters the communication state. A sequential computation start event means that the process enters the sequential computation state.

It is not difficult to find out that the simulation start event and the simulation end event for an MPI process are the first event and the last event, respectively, of the process. Communication start events and sequential computation start events are generated alternately, as shown in Fig. 1.

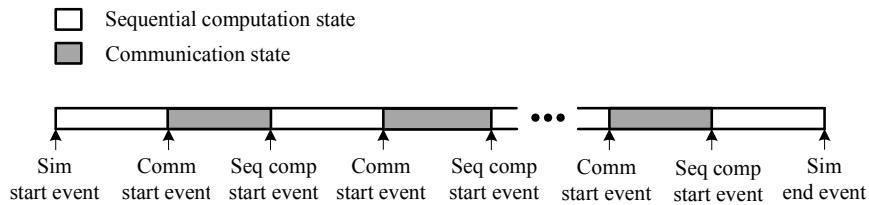


Fig. 1. The relationship between events and states.

3.2. Event Generation

Events are generated either by the pre-extracted trace (trace-driven) or by the execution of the program (execution-driven). For a trace-driven method, the event trace is extracted with instrumentation tools while running the program. Then the simulator uses the trace to drive the events and predict the execution time of the program. However, when predicting the performance of a large-scale parallel computing system, the time to extract the trace and the space to store it become intolerable. Moreover, due to some uncertain factors (e.g. branches, dynamic instruction generations and non-deterministic communications in the

program), trace-driven simulation may be inaccurate with the incorrect trace acquired.

For an execution-driven method, events are generated during program execution. Thus, program behaviours such as branch prediction and dynamic instruction generation can all be simulated. Moreover, an execution-driven method can make use of available system resources to directly execute portions of the application code and simulate the features that are of specific interest or unavailable [16]. We prefer an execution-driven method over a trace-driven method, because the former is closer to the program execution reality and has higher efficiency. Therefore, in SMP-SIM, the events are generated when the target program is running on the host machine:

- The first executable statement in an MPI process is MPI_Init and the last one is MPI_Finalize. So the execution time of a process under consideration is the time period between MPI_Init and MPI_Finalize. Therefore, when the MPI process encounters MPI_Init and MPI_Finalize, the simulation start event and simulation end event are generated respectively.
- When encountering an MPI communication primitive, such as MPI_Isend or MPI_Irecv, a process enters the communication state and a communication start event is generated.
- When an MPI communication primitive finishes (i.e. a process returns from an MPI communication primitive), the process enters the sequential computation state and a sequential computation start event is generated.

3.3. Processing the Events

The aim of processing the events is to estimate the execution time of the target program running on the target machine. For this purpose, SMP-SIM maintains the simulation time for each process of the target program, denoted as t^s . When an event is generated when the target program is executing on the host machine, the simulation time of this process will be updated to the generation time of the event. The generation time of an event is the time when the event is generated if the target program executes on the target machine. Therefore, when a simulation end event is being processed, the simulation time of its corresponding process (i.e. the generation time of the simulation end event) represents the execution time of the process when it runs on the target machine.

As a parallel simulator that can be run on a small-scale SMP, SMP-SIM utilizes the characteristics of SMP to deal with the events efficiently and accurately:

- Manage the events globally at the granularity of CPUs. Because the processes allocated to a CPU on the host machine may outnumber the cores in the CPU and all the processes share the memory in SMP, SMP-SIM globally manages the events that are generated by the processes allocated to the same CPU by using a shared-segment. The event with the smallest generation time will always be selected to be processed first.

- Use a virtual-actual combined method to process the selected event. If it is a communication start event, it will be processed virtually, i.e. the communication overhead is estimated by the communication model; if it is a sequential computation start event, it will be processed actually, i.e. the sequential computation time is measured by direct execution.

It should be mentioned that for the MPI programs with non-deterministic communications (e.g. a receive request contains `MPI_ANY_SOURCE` as the source), the simulator needs a synchronization mechanism to make sure that the right messages (i.e. the messages that are received when the program executes on the target machine) are accepted during the simulation on the host machine. The synchronization mechanism used in SMP-SIM is optimistic mechanism [9] [21], which allows to process the earliest available event with no regard to safety. When an older message arrives, a rollback mechanism is needed to undo an earlier out of order execution and re-execute the events to guarantee the correct sequence of event processing. However, synchronization mechanism is not the focus of this work. The interested readers please refer to [9] for a detailed description.

4. Framework of SMP-SIM

Due to its good configuration, high performance and portability, MPICH [6] has become one of the most popular MPI libraries. SMP-SIM is designed based on MPICH. We have modified the MPICH library and integrated all the functionalities of SMP-SIM into the modified MPICH library.

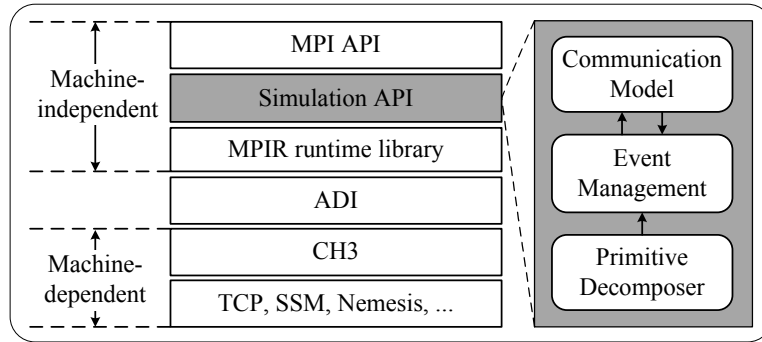


Fig. 2. The framework of SMP-SIM.

MPICH consists of two layers, a machine-independent layer and a machine-dependent layer, separated by an abstract device interface (ADI), as shown in Fig. 2. The machine-independent layer consists of an MPI application programmer interface (API) layer and an MPIR runtime library layer. The MPI API pro-

vides the user with programming interfaces and handles the MPI structures irrelevant to environments. The MPIR runtime library translates the complex MPI primitives in MPI API into point-to-point communication operations. The main functionalities of SMP-SIM are implemented by adding a **Simulation API Layer** between the MPI API and MPIR runtime library layers. As shown in Fig. 2, the simulation API layer comprises three modules: **primitive decomposer**, **communication model** and **event management**.

In the primitive decomposer module, all the MPI communication primitives are reconstructed by using core point-to-point communication primitives. This is the base of the other modules. When a process encounters a communication primitive as the target program runs on the host machine, the primitive decomposer module will decompose it into several core point-to-point communication primitives and then these core point-to-point communication primitives will be invoked one by one. The invocation and the return of a core point-to-point communication primitive will generate the corresponding events. When an event is generated, the event management module globally schedules the events, processes the selected event and updates the simulation time of the event's corresponding process. When processing a communication start event, the event management module will interact with the communication model module to calculate the time overheads of the corresponding core point-to-point primitive. Next, we will introduce these three modules in detail.

4.1. Primitive Decomposer

MPI provides users with a lot of communication primitives, including collective communication primitives and point-to-point communication primitives. In MPICH, all collective communication operations are implemented in terms of point-to-point communication operations. So, we choose four core point-to-point communication primitives from the MPI library: MPI_IbSend (non-blocking buffered send), MPI_Issend (non-blocking synchronous send), MPI_Irecv (non-blocking receive) and MPI_Wait [15]. Using these four core point-to-point communication primitives, we can reconstruct the other point-to-point and collective communication primitives.

Table 1. Point-to-point communication primitives reconstruction in SMP-SIM

Primitive	Functionality	Re-constructed by
MPI_Bsend	Blocking buffered send	Both MPI_IbSend and MPI_Wait
MPI_Ssend	Blocking synchronous send	Both MPI_Issend and MPI_Wait
MPI_Rsend	Blocking ready send	MPI_Ssend
MPI_Send	Blocking standard send	Either MPI_Bsend or MPI_Ssend
MPI_Recv	Blocking standard receive	Both MPI_Irecv and MPI_Wait

Table 1 lists the ways to reconstruct the other five point-to-point communication primitives in MPI. The collective communication primitives can be reconstructed by four core point-to-point communication primitives and the five primitives listed in the first column of Table 1. When a process of the target program encounters a communication primitive on the host machine, the primitive decomposer module will decompose it into several core point-to-point communication primitives and then these core point-to-point communication primitive will be invoked one by one. Consequently, the corresponding events will be generated. Therefore, the communication start events mentioned in Section 3.1 can be further divided into four types, *non-blocking buffered send start events*, *non-blocking synchronous send start events*, *non-blocking receive start events* and *wait start events*, which are generated due to the invocation of MPI_IbSend, MPI_Issend, MPI_Irecv and MPI_Wait, respectively. All the events that will appear in SMP-SIM are listed in Table 2, where the four items in the third row are all communication start events.

Table 2. All the events in SMP-SIM

Event	Symbol	Generated by
Simulation start event	E_{Start}	Invocation of MPI_Init
Non-blocking buffered send start event	E_{IbSend}	Invocation of MPI_IbSend
Non-blocking synchronous send start event	E_{Issend}	Invocation of MPI_Issend
Non-blocking receive start event	E_{Irecv}	Invocation of MPI_Irecv
Wait start event	E_{Wait}	Invocation of MPI_Wait
Sequential computation start event	E_{Seq}	Return of MPI_Init, MPI_IbSend, MPI_Issend, MPI_Irecv, MPI_Wait
Simulation end event	E_{End}	Invocation of MPI_Finalize

4.2. Communication Model

This module is responsible for calculating the time overheads of the point-to-point communication primitives. First, we list all the symbols to be used in Table 3. Then we discuss how to calculate the time overheads of core point-to-point communication primitives. The way to calculate the time overhead of an core point-to-point communication primitive is related to the two communicating processes' physical distributions in the target machine.

MPI_IbSend. MPI_IbSend starts a non-blocking buffered send. A buffer space for the data needs to be reserved at the sender side, and the message is always sent no matter whether there is a matching receive at the receiver side.

Table 3. Symbols to be used in Section 4.2

Symbol	Meaning
$t_{generate}^{send}$	Invocation time of MPI_Ibsend or MPI_Issend*
t_{return}^{send}	Return time of MPI_Ibsend or MPI_Issend
t_{safe}^{send}	The time when MPI_Ibsend or MPI_Issend is safe**
$t_{generate}^{recv}$	Invocation time of MPI_Irecv
t_{return}^{recv}	Return time of MPI_Irecv
t_{safe}^{recv}	The time when MPI_Irecv is safe
$t_{generate}^{wait}$	Invocation time of MPI_Wait
t_{return}^{wait}	Return time of MPI_Wait
t_{arrive}	The time when the message arrives at the receiver
L_{mem}	Memory latency
B_{mem}	Memory bandwidth
L_{net}	Network latency
B_{net}	Network bandwidth
m	Message (data) size
δt	The time overhead of invocation a primitive
O_a	The time overhead of reserving a buffer space

* The invocation time of an primitive is the generation time of the event generated by the primitive.

** The time when a communication primitive is safe is the time when its corresponding data are safe.

As shown in Fig. 3(a), if two communicating processes are physically distributed, MPI_Ibsend returns as soon as the buffer at the sender side is available. The data can be sent to the network while they are being copied to the buffer (generally speaking, $B_{mem} > B_{net}$), according to the configurations of the current machines. The data are safe when they have been completely copied to the buffer. We can calculate t_{return}^{send} , t_{safe}^{send} and t_{arrive} as in Equations 1 – 3:

$$t_{return}^{send} = t_{generate}^{send} + \delta t + O_a \quad (1)$$

$$t_{safe}^{send} = t_{generate}^{send} + \delta t + O_a + m/B_{mem} \quad (2)$$

$$t_{arrive} = t_{generate}^{send} + \delta t + O_a + L_{net} + m/B_{net} \quad (3)$$

As shown in Fig. 3(b), if two communicating processes are physically centralized, the buffer will not be reserved and MPI_Ibsend returns as soon as it is invoked. Then, we have:

$$t_{return}^{send} = t_{generate}^{send} + \delta t \quad (4)$$

$$t_{safe}^{send} = t_{generate}^{send} + \delta t + m/B_{mem} \quad (5)$$

$$t_{arrive} = t_{generate}^{send} + \delta t + L_{mem} + m/B_{mem} \quad (6)$$

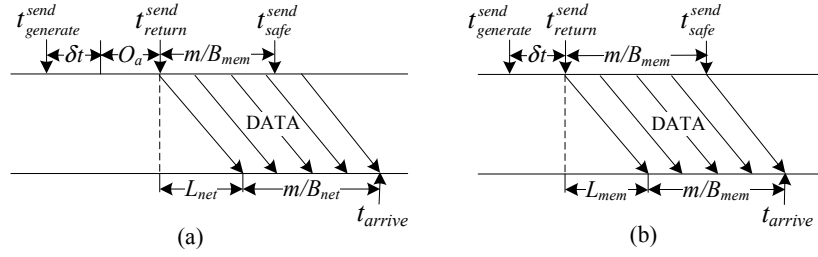


Fig. 3. Non-blocking buffered send. (a) Physically distributed. (b) Physically centralized.

MPI_Issend. MPI_Issend starts a non-blocking synchronous send. A handshake, which is used to make sure the matching receive has started, happens between the sender and the receiver via REQ and ACK messages before the data message is sent to the receiver. MPI_Issend will return when ACK from the receiver is received. The return time of MPI_Issend is affected by the relationship between $t_{generate}^{recv}$ and the REQ's arrival time (denoted as t_{REQ}).

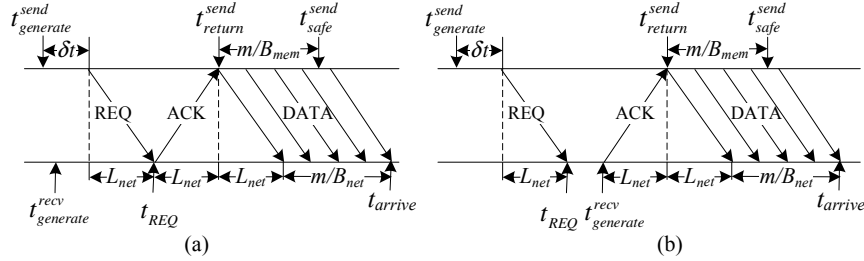


Fig. 4. Non-blocking buffered send (the sender and receiver are physically distributed) (a) $t_{generate}^{recv} \leq t_{REQ}$. (b) $t_{generate}^{recv} > t_{REQ}$.

Fig. 4(a)(b) shows the case where two communicating processes are physically distributed in target machine. The data can be sent to network while they are being copied to the buffer, and the data are safe after they have been copied to the buffer. t_{return}^{send} , t_{safe}^{send} and t_{arrive} can be calculated as in Equations 8 – 10:

$$t_{REQ} = t_{generate}^{send} + \delta t + L_{net} \quad (7)$$

$$t_{return}^{send} = \begin{cases} t_{generate}^{send} + \delta t + L_{net}, & t_{generate}^{recv} \leq t_{REQ} \\ t_{generate}^{send} + L_{net}, & t_{generate}^{recv} > t_{REQ} \end{cases} \quad (8)$$

$$t_{safe}^{send} = \begin{cases} t_{generate}^{send} + \delta t + 2L_{net} + m/B_{mem}, & t_{generate}^{recv} \leq t_{REQ} \\ t_{generate}^{recv} + L_{net} + m/B_{mem}, & t_{generate}^{recv} > t_{REQ} \end{cases} \quad (9)$$

$$t_{arrive} = \begin{cases} t_{generate}^{send} + \delta t + 3L_{net} + m/B_{net}, & t_{generate}^{recv} \leq t_{REQ} \\ t_{generate}^{recv} + 2L_{net} + m/B_{net}, & t_{generate}^{recv} > t_{REQ} \end{cases} \quad (10)$$

Due to the space limitation, we will not analyze the case where the communicating processes are physically centralized. We can calculate t_{return}^{send} , t_{safe}^{send} and t_{arrive} of this case by replacing the L_{net} and B_{net} in Equations 8 – 10 with L_{mem} and B_{mem} , respectively.

MPI.Irecv. MPI_Irecv returns as soon as it is invoked. The return of MPI_Irecv means that the message can be received, rather than having been received. Then, we have:

$$t_{return}^{recv} = t_{generate}^{recv} + \delta t \quad (11)$$

MPI.Wait. MPI_Wait is used to guarantee the safety of the data. It will block the process until the data sent or received by its related non-blocking communication are safe. The return time of MPI_Wait t_{return}^{wait} is the maximum value of the time when the data are safe (i.e. t_{safe}^{send} or t_{safe}^{recv}) and the invocation time of MPI_Wait $t_{generate}^{wait}$. So, we have:

$$t_{return}^{wait} = \begin{cases} \max\{t_{safe}^{send}, t_{generate}^{wait}\}, & \text{if the waiting object is a non-blocking send} \\ \max\{t_{safe}^{recv}, t_{generate}^{wait}\}, & \text{if the waiting object is a non-blocking receive} \end{cases} \quad (12)$$

t_{safe}^{send} in Equation 12 can be calculated as shown in Equations 2, 5 and 9. Then, we analyze the way to calculate t_{safe}^{recv} .

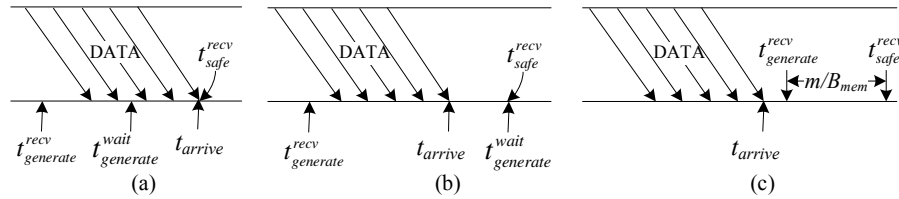


Fig. 5. Non-blocking receive. (a) $t_{generate}^{recv} < t_{arrive}$, $t_{generate}^{wait} \leq t_{arrive}$. (b) $t_{generate}^{recv} < t_{arrive}$, $t_{generate}^{wait} > t_{arrive}$. (c) $t_{generate}^{recv} \geq t_{arrive}$.

t_{safe}^{recv} is affected by the relationships among $t_{generate}^{recv}$, $t_{generate}^{wait}$ and t_{arrive} , as shown in Fig. 5. It is worth mentioning that the case shown in Fig. 5(c) only exists when the send primitive is MPI_lbsend. From Fig. 5, we can derive that

$$t_{safe}^{recv} = \begin{cases} \max\{t_{arrive}, t_{generate}^{wait}\}, & t_{generate}^{recv} < t_{arrive} \\ t_{generate}^{recv} + m/B_{mem}, & t_{generate}^{recv} \geq t_{arrive} \end{cases} \quad (13)$$

where t_{arrive} can be calculated as shown in Equations 3, 6 and 10.

4.3. Event Management Module

SMP-SIM is a parallel simulator, and the processes are fixed to their own CPUs while the target programs are running on the host machine. The target machine

has more CPUs than the host machine and the core count of the target machine is equal to the process count of the target program. So the processes allocated to a host machine's CPU outnumber the cores in it. The event management module maintains a *waiting event queue* for each CPU, i.e. the processes allocated to the same CPU share a waiting event queue. At the beginning of running a target program on the host machine, this module inserts an E_{Start} event for each process into its corresponding waiting event queue and sets the generation time of this event as -1. During the execution of the target program, the module manages the events at the granularity of CPUs using the workflow:

Step 1. The event management module selects the *un-processed executable event* e that has the smallest generation time from the CPU's waiting event queue. It should be noticed that E_{Start} , E_{End} , E_{Seq} , E_{Ibsend} and E_{Irecv} events are always executable; E_{Issend} is executable only when its corresponding receive primitive has been invoked; and E_{Wait} is executable only when the data of its related send or receive primitive are safe. For convenience, for a given event e , we use $e.t_{generate}$ to represent the generation time of e , $e.doing$ to determine whether e is being processed and $e.executable$ to determine whether e is executable at present.

Step 2. The event management module switches the process that is related to e onto an idle core and then processes e . As shown in Algorithm 1, the detailed dealing methods are different among different types of events. The *completing event queue* is used to store the E_{Ibsend} , E_{Issend} and E_{Irecv} events, which have already been processed, and the information of these events is necessary when processing the corresponding E_{Wait} events. For convenience, for a given communication event e , we use $e.t_{return}$, $e.t_{arrive}$ and $e.t_{safe}$ to refer to the return time, arrival time and the data's safe time of e 's corresponding communication primitive respectively.

Step 3. After accomplishing the processing of the selected event e , the event management module deletes e from the waiting event queue and may insert a new event e_{new} into the waiting event queue as follows:

- $e.type$ is E_{Start} , E_{Ibsend} , E_{Issend} , E_{Irecv} or E_{Wait} : after executing the communication operations corresponding to event e , the process will deal with sequential computation codes, so $e_{new}.type = E_{Seq}$ and $e_{new}.executable = True$.
- $e.type$ is E_{Seq} : denote the MPI primitive behind the sequential computation codes as S .
 - S is MPI.Ibsend: $e_{new}.type = E_{Ibsend}$ and $e_{new}.executable = True$.
 - S is MPI.Issend: $e_{new}.type = E_{Issend}$; if the E_{Irecv} event that matches e_{new} is in the waiting/completing event queue, $e_{new}.executable = True$, otherwise $e_{new}.executable = False$.
 - S is MPI.Irecv: $e_{new}.type = E_{Irecv}$ and $e_{new}.executable = True$.

Algorithm 1 The Algorithm of EventHandler**Input:** Event e **Output:** Void

```

1.  $e.doing \leftarrow True$ 
2. if  $e.type = E_{Start}$  then
3.   execute the body codes of MPI_Init;
4.    $t^s \leftarrow 0$ ;
5. else if  $e.type = E_{End}$  then
6.   print  $t^s$ ;
7.   execute the body codes of MPI_Finalize;
8. else if  $e.type = E_{Seq}$  then
9.   get the current system time  $t_1$ ;
10.  execute the sequential computation codes;
11.  get the current system time  $t_2$ ;
12.   $t^s \leftarrow t^s + (t_2 - t_1)$ ;
13. else if  $e.type = E_{Ibsend}$  then
14.  execute the body codes of MPI_Ibsend;
15.  calculate  $e.t_{return}$ ,  $e.t_{safe}$  and  $e.t_{arrive}$  based on Equations 1 – 3 and 4 – 6;
16.  if there is the  $E_{Irecv}$  event  $e'$  matched with  $e$  in waiting event queue then
17.     $e'.t_{arrive} \leftarrow e.t_{arrive}$ 
18.  else if there is the  $E_{Irecv}$  event  $e'$  matched with  $e$  in completing event queue then
19.     $e'.t_{arrive} \leftarrow e.t_{arrive}$ 
20.    if there is the  $E_{Wait}$  event  $e''$  matched with  $e'$  in waiting event queue then
21.       $e''.executable \leftarrow True$ ;
22.    end if
23.  end if
24.   $t^s \leftarrow e.t_{return}$ ;
25.  insert  $e$  into its completing event queue;
26. else if  $e.type = E_{Issend}$  then
27.  execute the body codes of MPI_Issend;
28.  get  $E_{Irecv}$  event  $e'$  which matches with  $e$  from the completing event queue;
29.  calculate  $e.t_{return}$ ,  $e.t_{safe}$  and  $e'.t_{arrive}$  based on Equations 8 – 10;
30.  if there is the  $E_{Wait}$  event  $e''$  matched with  $e'$  in waiting event queue then
31.     $e''.executable \leftarrow True$ ;
32.  end if
33.   $t^s \leftarrow e.t_{return}$ ;
34.  insert  $e$  into its completing event queue;
35. else if  $e.type = E_{Irecv}$  then
36.  execute the body codes of MPI_Irecv;
37.   $e.t_{arrive} \leftarrow -1$ ;
38.  if there is the  $E_{Ibsend}$  event  $e'$  matched with  $e$  in completing event queue then
39.     $e.t_{arrive} \leftarrow e'.t_{arrive}$ ;
40.  else if there is the  $E_{Issend}$  event  $e'$  matched with  $e$  in waiting event queue then
41.     $e'.executable \leftarrow True$ ;
42.  end if
43.   $t^s \leftarrow t^s + \delta t$ ;
44.  insert  $e$  into its completing event queue;
45. else if  $e.type = E_{Wait}$  then
46.  execute the body codes of MPI_Wait;
47.  get event  $e'$  which matches with  $e$  from the completing event queue;
48.  calculate  $e.t_{return}$  based on Equations 12 and 13;
49.  delete  $e'$  from the completing event queue;
50. end if

```

- S is MPI_Wait: $e_{new}.type = E_{Wait}$; if $e'.t_{arrive} > 0$ (e' is the E_{Irecv} event that matches e_{new}), $e_{new}.executable = True$, otherwise $e_{new}.executable = False$.
 - S is MPI_Finalize: $e_{new}.type = E_{End}$ and $e_{new}.executable = True$.
- $e.type$ is E_{End} : the process finishes after executing MPI_Finalize, so event management module will not insert any new event into the waiting event queue, i.e. no event e_{new} will be created.

Whatever the type of e_{new} is, $e_{new}.doing = False$ always holds after inserting e_{new} into the waiting event queue.

After the work of step 2 and step 3, if there is any un-processed event in the waiting event queue, the event management module will start the work of step 1 again and deal with the new selected event as described in step 2 and step 3. It is not difficult to find out that for each process, there is only one related event in the waiting event queue at any time during the simulation. Because only the processes whose events are selected to be processed can run, the unselected processes must be suspended. Processes allocated to the same CPU may be switched when a new event is selected to be processed, and the detailed implementation will be introduced in Section 5.

5. MPICH2-based Implementation of SMP-SIM

This section describes the MPICH2-based implementation of SMP-SIM. By recognizing the fact that the processes share the memory in SMPs, SMP-SIM creates event queues in the shared segment for all the processes at the granularity of CPUs. SMP-SIM re-implements the core MPI primitives and integrates the functionalities of event management into them.

Fig. 6 shows the directory of the source codes of MPICH2. The major modification of the implementation of SMP-SIM is in the sub-directory /src/mpi. The implementations of MPI_Init and MPI_Finalize are in /src/mpi/init; the implementations of all point-to-point MPI primitives are in /src/mpi/pt2pt; and the implementations of all collective MPI primitives are in /src/mpi/coll.

5.1. Data Structure

Event is the core of SMP-SIM, and we implement the data structure of an event as shown in Fig. 7.

- Attributes $type$, $processID$, $doing$, $executable$ and $t_{generate}$ are used for all types of events. $type$, $processID$ and $t_{generate}$ stand for the type, process ID and generation time of the event respectively; $doing$ and $executable$ judge whether the event is being and can be processed respectively.
- Attributes t_{return} , t_{arrive} , t_{safe} , $pairID$ and tag are used for three types of events: E_{IbSEND} , E_{IssEND} and E_{Irecv} . These attributes stand for the return time, message arrival time, safe time, matched process ID and tag of the corresponding communication primitive respectively. Given an E_{IbSEND} ,

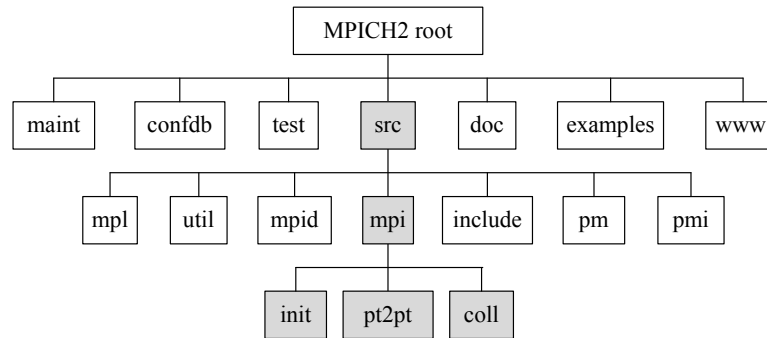


Fig. 6. The directory of the source codes of MPICH2

```

struct Event{
    int type;
    int processID;
    bool doing;
    bool executable;
    double t_generate;
    double t_return;
    double t_arrive;
    double t_safe;
    int pairID;
    int tag;
    MPI_Request *request;
}
  
```

Fig. 7. The data structure of event

E_{Issend} or E_{Irecv} event, we can find the matched E_{Irecv} , E_{Ibsend} or E_{Issend} event with $pairID$ and tag .

- Attribute $request$ is used for four types of events: E_{Ibsend} , E_{Issend} , E_{Irecv} and E_{Wait} . This attribute stands for the object of the corresponding non-blocking communication. Given an E_{Wait} event, we can find the matched E_{Ibsend} , E_{Issend} or E_{Irecv} event with $request$.

Based on the definition of the data structure of an event, we create two event queues, $Wait_Event_Queue$ and $Complete_Event_Queue$, at the granularity of CPUs, and store them in the shared segment so that all the processes can access them conveniently. It should be noticed that in order to guarantee correctness, all the operations on the event queues must be protected by the lock mechanism, i.e. only one process is permitted to access a given event queue at one time. In order to briefly describe our implementation, all the operations on the event queues described in the rest of this section are encapsulated by $lock()$ and $unlock()$ implicitly.

5.2. Re-implementations of Core Primitives

This subsection introduces the re-implementations of six core MPI primitives: MPI_Init, MPI_Finalize, MPI_Ibsend, MPI_Issend, MPI_Irecv and MPI_Wait. These re-implementations realize the functionalities of event management described in Section 4.3.

```

MPI_Init(){
    Codes of original body;

    // codes added for SMP-SIM
    new a event e;
    e.type=ESeq; e.processID=myRank; e.doing=False;
    e.t_executable=True; e.t_generate=0; t1=0;
    Insert e into its Wait_Event_Queue;
    MPI_Barrier();
    lock();
    if(Count<CORE_NUM){
        Count++;
        set the priority of this process HIGH;
        if(Count == CORE_NUM)
            suspend all the processes allocated to this CPU without HIGH priority;
    }
    unlock();
    e.doing=True;
    t1=gettime();

    return;
}
    
```

Fig. 8. The re-implementation of MPI_Init

MPI_Init As shown in Fig. 8, besides the codes of the original MPI_Init body, the re-implementation of MPI_Init does the work as follows: firstly, each process inserts an E_{Seq} event into its corresponding *Wait_Event_Queue*; secondly, make sure that only one process is running on a given processing core and the other processes are suspended; thirdly, the process that has not been suspended starts to deal with the E_{Seq} event.

In the re-implementation of MPI_Init, *Count* is a global variable shared by all the processes allocated to the same CPU, and it is used for logging the count of the processes whose E_{Seq} events have been started to be processed. *lock()* and *unlock()* are two functions used to protect the critical section between them. While the target program is running, the process switching mechanism is the priority scheduling mechanism instead of the time-sliced round-robin mechanism, so the line *set the priority of this process HIGH* in Fig. 8 guarantees that the current running process can not be switched.

```

MPI_Ibsend(buf, count, datatype, dest, tag, comm, request){
  // codes added for SMP-SIM: remove  $E_{Seq}$  type event
   $t_2$ =gettime();
  remove the event whose processID is myRank from the Wait_Event_Queue;

  // codes added for SMP-SIM: generate  $E_{Ibsend}$  type event
  new a event e;
  e.type= $E_{Ibsend}$ ; e.processID=myRank; e.doing=False;
  e.t_executable=True; e.t_generate= $t + (t_2 - t_1)$ ;  $t^s = e.t\_generate$ ;
  e.pairID=dest; e.tag=tag; e.request=request;
  Insert e into its Wait_Event_Queue;
  pID = findNextEvent();
  if (pID != myRank)
    suspend the current process and switch to the process whose ID is pID;

  Codes of original body;

  // codes added for SMP-SIM: deal with event e
  execute the codes whose function is same as line 15-25 in Algorithm 1;

  // codes added for SMP-SIM: generate  $E_{Seq}$  type event
  new a event e1;
  e1.type= $E_{Seq}$ ; e1.processID=myRank; e1.doing=False;
  e1.t_executable=True; e1.t_generate= $t^s$ ;
  Insert e1 into its Wait_Event_Queue;
  pID = findNextEvent();
  if (pID != myRank)
    suspend the current process and switch to the process whose ID is pID;
   $t_1$ =gettime();

  return;
}

```

Fig. 9. The re-implementation of `MPI_Ibsend`

MPI_Ibsend, MPI_Issend, MPI_Irecv and MPI_Wait The re-implementations of `MPI_Ibsend`, `MPI_Issend`, `MPI_Irecv` and `MPI_Wait` are similar, and the major work is divided into four steps as follows:

- Firstly, remove the E_{Seq} event that has just been processed from the *Wait_Event_Queue*.
- Secondly, insert a new communication event according to this MPI communication primitive into the *Wait_Event_Queue*; select the next event to process from the *Wait_Event_Queue*; switch to the process corresponding to the selected event.
- Thirdly, after this process is switched on again, execute the codes of the original body and process the event related to this primitive.
- Fourthly, insert a new E_{Seq} event into the *Wait_Event_Queue*; select the next event to process from the *Wait_Event_Queue*; switch to the process corresponding to the selected event.

The re-implementation of `MPI_Ibsend` is shown in Fig. 9, where the function `findNextEvent()` is used for selecting the un-processed executable event e with the minimum generation time, and the return value of `findNextEvent()` is $e.processID$. The major difference between `MPI_Issend`/`MPI_Irecv`/`MPI_Wait` and `MPI_Ibsend` is the work of processing the event related to the primitive, and the detailed processing methods can be found in Algorithm 1.

```
MPI_Finalize(){
  // codes added for SMP-SIM
  t2=gettime();
  remove the event whose processID is myRank from the Wait_Event_Queue;
  ts = ts + (t2-t1);
  print ts;

  Codes of original body;
  return;
}
```

Fig. 10. The re-implementation of `MPI_Finalize`

MPI_Finalize As shown in Fig. 10, in the re-implementation of `MPI_Finalize`, before executing the codes of the original `MPI_Finalize` body, we first remove the E_{Seq} event that has just been processed from the *Wait_Event_Queue*, and then update and print the current simulation time of this process.

6. Experiments

We demonstrate the validation and accuracy of SMP-SIM in this section. Section 6.1 introduces the benchmarks and experimental platform used. Section 6.2 describes the methodology used for evaluating our work. Section 6.3 presents and analyzes the experimental results.

6.1. Benchmarks and Platform

We select three parallel kernels EP, CG, FT from NPB3.3-MPI benchmarks and Sweep3D-2.2d to evaluate SMP-SIM. The problem sizes of EP, CG and FT are all Class C, and the problem size of Sweep3D is $100 \times 100 \times 100$. All the benchmarks are executed in Red Hat Enterprise Linux Server Release 5.5, and MPICH2-1.3.1 is used.

Our platform is a cluster with eight nodes, and each node is an SMP equipped with two 2.93G Intel Xeon X5670 CPUs and 24GB RAM. The interconnection is described in [23]. Notice that although there are 6 cores in Xeon X5670 CPU, we only use it as a 4-core CPU because most of the benchmarks must be executed by 2^n processes.

6.2. Evaluation Methodology

In order to validate the accuracy of SMP-SIM for different target machines, our experiments include two parts:

- SMP target machine: for a single SMP node in our platform, firstly, we use the two CPU as the target machine and only one CPU as the host machine, and test the accuracy of SMP-SIM; secondly, we use only one core as the host machine, and test the scalability of SMP-SIM with varying the scale of the target machine.
- SMP-Cluster target machine: for the whole platform, we use all the eight nodes as the target machine and only one node as the host machine, and test the accuracy of SMP-SIM.

6.3. Results and Analysis

SMP Target Machine As shown in Fig. 11, when using two CPUs as the target machine and only one CPU as the host machine, for all the four benchmarks, the errors of SMP-SIM are between 2.56% and 6.14%. The accuracy of SMP-SIM for EP benchmark is the highest, because EP has the fewest communications. The sequential computation time is measured by direct execution, so the simulation of sequential computation is more accurate than that of communication. Consequently, the more communications a benchmark contains, the lower accuracy of SMP-SIM is.

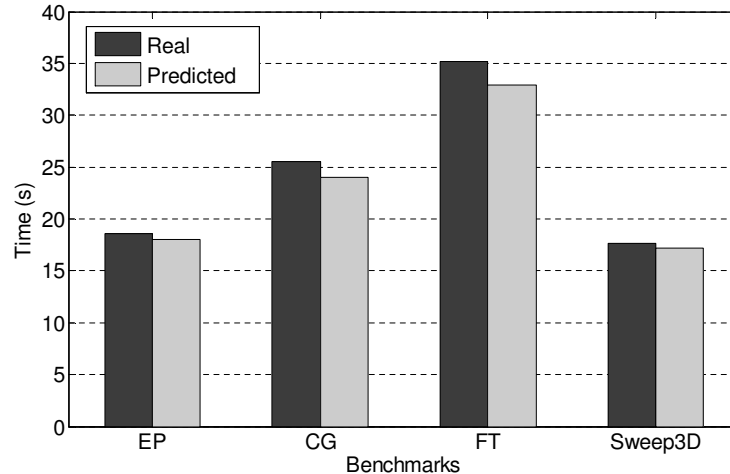


Fig. 11. The accuracy of SMP-SIM for SMP target machine

In order to test the scalability of SMP-SIM, we fix the host machine as a single processing core, and vary the scale of the target machine as 2, 4, and

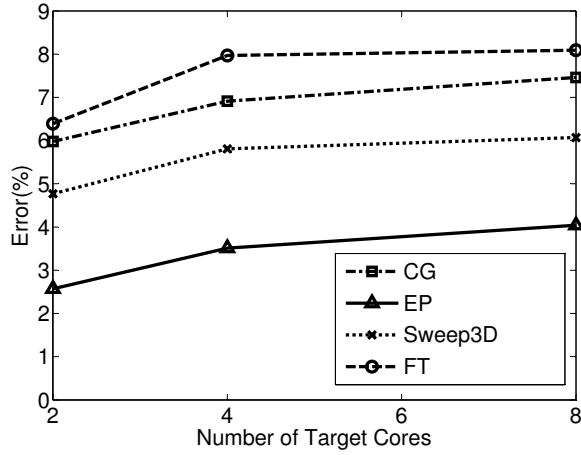


Fig. 12. The scalability of SMP-SIM

8 processing cores. As shown in Fig. 12, the results show that SMP-SIM has good scalability.

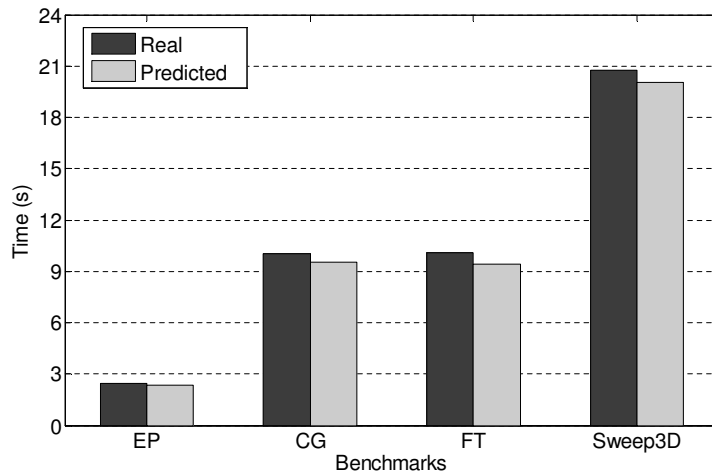


Fig. 13. The accuracy of SMP-SIM for SMP-Cluster target machine

SMP-Cluster Target Machine As shown in Fig. 13, when using all the eight nodes as the target machine and only one node as the host machine, for all the four benchmarks, the accuracies of SMP-SIM are high and the errors are between 2.67% and 7.60%.

7. Conclusion

For the system developers, it has been long desired that the performance of a parallel system can be predicted at the design phase. Before the target machine is completely constructed, the developers can always build an SMP machine used as a host machine. In this paper, we introduce SMP-SIM, an SMP-based discrete-event execution-driven performance simulator that exploits the characteristics of SMP to achieve accurate and scalable performance prediction.

In SMP-SIM, we have integrated three modules, primitive decomposer, communication model and event management, by adding a simulation API layer on top of the MPICH library. By executing the target program on the host machine with the modified MPICH library, SMP-SIM manages the events globally, handles the communication start events virtually and sequential computation start events actually. In our experimental evaluation, SMP-SIM shows high accuracy and good scalability with prediction errors of less than 7.60%.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (NSFC) No.60921062.

References

1. Ajima, Y., Takagi, Y., Inoue, T., Hiramoto, S., Shimizu, T.: The tofu interconnect. In: High Performance Interconnects (HOTI), 2011 IEEE 19th Annual Symposium on. pp. 87–94 (aug 2011)
2. Barker, K.J., Davis, K., Hoisie, A., Kerbyson, D.J., Lang, M., Pakin, S., Sancho, J.C.: Entering the petaflop era: the architecture and performance of roadrunner. In: Proceedings of the 2008 ACM/IEEE conference on Supercomputing. pp. 1:1–1:11. SC '08, IEEE Press, Piscataway, NJ, USA (2008)
3. Barker, K., Pakin, S., Kerbyson, D.: A performance model of the krak hydrodynamics application. In: Parallel Processing, 2006. ICPP 2006. International Conference on. pp. 245–254 (aug 2006)
4. Website, <http://www.cray.com/Products/CX1000/Chassis/CX1000S.aspx>
5. Fujimoto, R.M.: Parallel discrete event simulation. *Commun. ACM* 33, 30–53 (October 1990)
6. Gropp, W., Lusk, E., Doss, N., Skjellum, A.: A high-performance, portable implementation of the mpi message passing interface standard. *Parallel Comput.* 22, 789–828 (September 1996)
7. Huang, C., Lawlor, O., Kal, L.: Adaptive mpi. In: Languages and Compilers for Parallel Computing. Lecture Notes in Computer Science, vol. 2958, pp. 306–322. Springer Berlin / Heidelberg (2004)
8. Website, <http://static.msi.umn.edu/tutorial/scicomp/sp/intro-Power3SP/index.html>
9. Jefferson, D., Beckman, B., Wieland, F., L., B., Diloroto, M.: Time warp operating system. *SIGOPS Oper. Syst. Rev.* 21, 77–93 (November 1987)
10. Kale, L.V., Krishnan, S.: Charm++: a portable concurrent object oriented system based on c++. *SIGPLAN Not.* 28(10), 91–108 (Oct 1993)
11. Kerbyson, D.J., Alme, H.J., Hoisie, A., Petrini, F., Wasserman, H.J., Gittings, M.: Predictive performance and scalability modeling of a large-scale application. In: Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM). pp. 37–37. Supercomputing '01, ACM, New York, NY, USA (2001)

12. Labarta, J., Girona, S., Pillet, V., Cortes, T., Gregoris, L.: Dip: A parallel program development environment. *Lecture Notes in Computer Science* 1124, 665 – 665 (1996)
13. Website, <http://www.mcs.anl.gov/research/projects/mpich2/>
14. Mukherjee, S.S., Reinhardt, S.K., Falsafi, B., Litzkow, M., Hill, M.D., Wood, D.A., Huss-Lederman, S., Larus, J.R.: Wisconsin wind tunnel ii: A fast, portable parallel architecture simulator. *IEEE Concurrency* 8, 12–20 (October 2000)
15. Prakash, S., Bagrodia, R.L.: Mpi-sim: using parallel simulation to evaluate mpi programs. In: *Proceedings of the 30th conference on Winter simulation*. pp. 467–474. WSC '98, IEEE Computer Society Press, Los Alamitos, CA, USA (1998)
16. Prakash, S., Deelman, E., Bagrodia, R.: Asynchronous parallel simulation of parallel programs. *IEEE Trans. Softw. Eng.* 26, 385–400 (May 2000)
17. Reinhardt, S.K., Hill, M.D., Larus, J.R., Lebeck, A.R., Lewis, J.C., Wood, D.A.: The wisconsin wind tunnel: virtual prototyping of parallel computers. In: *Proceedings of the 1993 ACM SIGMETRICS conference on Measurement and modeling of computer systems*. pp. 48–60. SIGMETRICS '93, ACM, New York, NY, USA (1993)
18. Website, <http://sourceforge.net/projects/sim-mpi/>
19. Snaveley, A., Carrington, L., Wolter, N., Labarta, J., Badia, R., Purkayastha, A.: A framework for performance modeling and prediction. In: *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*. pp. 1–17. Supercomputing '02, IEEE Computer Society Press, Los Alamitos, CA, USA (2002)
20. Soga, T., Musa, A., Shimomura, Y., Egawa, R., Itakura, K., Takizawa, H., Okabe, K., Kobayashi, H.: Performance evaluation of nec sx-9 using real science and engineering applications. In: *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*. pp. 28:1–28:12. SC '09, ACM, New York, NY, USA (2009)
21. Steinman, J.S.: Breathing time warp. *SIGSIM Simul. Dig.* 23, 109–118 (July 1993)
22. Website, <http://www-unix.mcs.anl.gov/mpi>
23. Xie, M., Lu, Y., Liu, L., Cao, H., Yang, X.: Implementation and evaluation of network interface and message passing services for tianhe-1a supercomputer. In: *Proceedings of the 2011 IEEE 19th Annual Symposium on High Performance Interconnects*. pp. 78–86. HOTI '11, IEEE Computer Society, Washington, DC, USA (2011)
24. Yang, X.J., Liao, X.K., Lu, K., Hu, Q.F., Song, J.Q., Su, J.S.: The tianhe-1a supercomputer: Its hardware and software. *Journal of Computer Science and Technology* 26(3), 344–351 (2011)
25. Zhai, J., Chen, W., Zheng, W.: Phantom: Predicting performance of parallel applications on large-scale parallel machines using a single node. In: *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPOPP*. pp. 305 – 314. Bangalore, India (2010)
26. Zheng, G., Kakulapati, G., Kale, L.: Bigsim: a parallel simulator for performance prediction of extremely large parallel machines. In: *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*. p. 78 (april 2004)
27. Zheng, G., Wilmarth, T., Lawlor, O., Kale, L., Adve, S., Padua, D., Guebelle, P.: Performance modeling and programming environments for petaflops computers and the blue gene machine. In: *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*. p. 197 (april 2004)

Yufei Lin was born in 1985. She received her B.S. degree in automation from the University of Science and Technology of China (USTC) in 2006 and M.S. degree in computer science from the National University of Defense Technology (NUDT) in 2008. She is now a Ph.D. candidate in Computer Science from State Key Laboratory of High Performance Computing at NUDT. Her research interest lies in high performance computing and performance evaluation. Her email address is linyufei@nudt.edu.cn.

Xinhai Xu was born in 1984. He received his B.S., M.S. and Ph.D. degrees in computer science from NUDT in 2006, 2008 and 2012. He is now an assistant professor from School of Computer Science at NUDT. His research interest lies in high performance computing and fault tolerance. His email address is xuxinhai@nudt.edu.cn.

Yuhua Tang was born in 1962. She received her B.S. and M.S. degrees in computer science from NUDT in 1983 and 1985. She is now a professor from School of Computer Science at NUDT. Her research interest lies in high performance computing and its router design. Her email address is yhtang@nudt.edu.cn.

Xin Zhang was born in 1986. He received his B.S. degree in computer science from NUDT in 2009. He is now an M.S. candidate in Computer Science from School of Computer Science at NUDT. His research interest lies in performance evaluation. His email address is zx2010@jfjb.com.cn.

Xiaowei Guo was born in 1986. He received his B.S. and M.S. degrees in computer science from NUDT in 2009 and 2011. He is now a Ph.D. candidate in Computer Science from State Key Laboratory of High Performance Computing at NUDT. His research interest lies in high performance computing and its applications. His email address is xiaow_g@126.com.

Received: January 18, 2012; Accepted: December 4, 2012.

Orthogonal Sequences Based Multi-CFO Estimation and Semi-Blind ICA Based Equalization for Multiuser CoMP Systems

Yufei.Jiang^{1,2}, Xu Zhu¹, Enggee Lim², and Yi Huang¹

¹ Dept. of Electrical Engineering and Electronics, the University of Liverpool, L69 3GJ
Brownlow Hill, Liverpool, U.K.

{Yufei.Jiang, Xuzhu, Yihuang}@liverpool.ac.uk

² Dept. of Electrical and Electronic Engineering, Xi'an Jiao Tong University,
215000, Suzhou, China
{enggee.lim@xjtu.edu.cn}

Abstract. We propose a low-complexity carrier frequency offset (CFO) estimation approach and an independent component analysis (ICA) based semi-blind equalization structure for the orthogonal frequency division multiplexing (OFDM) based multiuser coordinated multi-point (CoMP) systems. A group of orthogonal sequences are employed to enable separation and simultaneous estimation of multiple CFOs at base stations, dividing a complex multi-dimensional search into series of low-complexity mono-dimensional searches. Then, a small number of pilot symbols attached to the source data, are used to resolve the ambiguity problem in the ICA equalized signals. It has a low complexity, as the permutation and quadrature ambiguity can be eliminated simultaneously, rather than sequentially by the previous precoding aided method. Simulation results show that the proposed semi-blind equalization structure has a bit error rate (BER) performance close to the ideal cases of the zero forcing (ZF) and minimum mean square error (MMSE) based equalizations with perfect channel state information (CSI) and no CFO, and also significantly outperforms the constant amplitude zero autocorrelation (CAZAC) sequences based CFO estimation method.

Keywords: carrier frequency offset (CFO), coordinated multi-point (CoMP), orthogonal frequency division multiplexing (OFDM), independent component analysis (ICA).

1. Introduction

Inter-cell interference is a major bottleneck for achieving a very high data rate in wireless communications [1]-[2]. Many methods have been considered to manage interference in wireless networks. Previously, adjacent cells are operating on different frequencies to effectively reduce the inter-cell interference. However, as the demand for the high mobile data rates is exponentially growing, higher spectral efficiency is driven [3]-[4]. One of the best ways to manage interference and to improve the spectral efficiency in wireless networks, is to

allow base stations to connect to each other through a backhaul link, where they could exchange the messages and jointly process the received multiple users' signals on the same frequency band. This technology is called the uplink co-ordinated multi-point reception (CoMP) [2], which has received much attention recently, and has been adopted by the 3GPP LTE-Advanced standard [2] as a key technology to reduce inter-cell interference and to improve the service coverage. CoMP can explore the interferences between the cells, which is different from other existing methods by treating them as noise [3]-[6]. Therefore, CoMP enables vast spectral efficiency in current wireless communication systems structure.

1.1. Related Work

Channel estimation and multiuser detection play an important role in multiuser CoMP systems. Two types of approaches, namely training based methods [7]-[9] and blind methods [10]-[15], are commonly used in wireless communications to estimate the channel state information (CSI) and to recover the transmit signals. On the one hand, transmitting training signals or pilots reduces the spectral efficiency, especially in wireless communication systems with very scarce bandwidth resource. Most of previous works related to the training sequence bring considerably extra bandwidth. [7] and [8] propose an adaptive MIMO single carrier frequency-domain equalization. However, their training overhead are up to 13% and 10%, respectively. In [9], the training sequences are used to initialize the equalizer coefficients to combat the Doppler shift. Although the training overhead is as low as 0.05%, the actual training sequences are up to 160 for one transmit antenna at each time of initialization. On the other hand, the pure blind methods increase the computational complexity, particularly at the receiver. If little prior knowledge or information is available at the receiver, the semi-blind approaches [16]-[18] can be employed as a good trade-off relationship between the number of training symbols and the computational complexity.

Blind source separation method can obtain the channel state information, without introducing extra training sequences. The equalizer coefficient can be directly obtained from the statistics of the received data, with the no need to use extra bandwidth and spectral. In fact, blind source separation method is divided into second order statistics (SOS) [12]-[15] and higher order statistics (HOS) [9], [16]-[18]. The Higher Order Statistic is more against the Gaussian noise, as the fourth or higher cumulants of Gaussian noise are equal to zero. In [12]-[13], a non-redundant linear precoding is applied to the source data, and the blind channel estimation is obtained, by exploring the signal covariance matrix via the precoding, based on the second order statistics of the received signals at the receiver. This work is extended to the MIMO case in [14] and [15]. However, all the papers above are based on the SOS only, which is sensitive to the Gaussian noise.

Independent component analysis [10]-[11], is one of the higher order statistics (HOS) based blind source separation approaches, which can estimate the

data directly from the statistics of the received signals, by maximizing the non-Gaussianity of the received signals, without knowing the CSI, as long as the received signal corresponds to a linear mixture. However, as a common drawback of blind approaches, phase, quadrature and permutation ambiguity still exist in the ICA based approaches. In [9], ICA was applied on all subcarriers, and the statistical correlation between subcarriers is used to resolve the frequency dependent permutation. However, this approaches propagate bit errors and ambiguity errors across subcarriers. In [16] and [17], a large number of training sequences are used to eliminate the ambiguity, which, however, reduces the spectral efficiency of the system. In [18], the precoding is employed at the transmitter, by superimposing the reference data to the source data. At the receiver, the permutation and phase ambiguity is eliminated, by using the correlation property between the reference symbols, without consuming extra spectral. However, it is very sensitive to the precoding constant and the frame length, and has a good bit error rate (BER) performance only when data frame size is large enough.

Orthogonal frequency division multiplexing (OFDM) [19]-[28] is an effective technology to combat frequency selective fading, and has been adopted in the fourth generation (4G) wireless communication systems. One of the main drawbacks of OFDM systems is their sensitivity to the carrier frequency offset (CFO), due to the imperfect local oscillators between transmitter and receiver, destroying the orthogonality among the subcarriers, and inducing the additional inter-carrier interference (ICI). In the OFDM based multiuser-CoMP system, multiple CFOs exist between mobile terminals and base stations, introducing multiple access interference (MAI). The optimum maximum likelihood (ML) based CFO estimation method is proposed in [19]. However, it can only be used in single CFO systems, and its complexity is prohibitive in real systems with multiple CFOs, due to the exhaustive multi-dimensional search. In [20], a sub-optimal CFO estimation, using constant amplitude zero autocorrelation (CAZAC) sequences, is proposed. However, there is a mutual interference due to multiple CFOs at each base station, resulting in a biased CFO estimation. Space time block codes transmission with multiple CFOs are proposed in cooperative wireless communication in [21]. However, it does not consider the multi-CFO estimation and the channel estimation. An optimum ML based estimator is proposed for downlink multiuser CoMP systems in [22]. However, its complexity is very high. An approach of approximation for ICI matrix based on the Hadamard sequence is proposed in [23], However it is a biased CFO estimation method. Moreover, in [24]-[28], they require transmitting a large number of training sequences for accurate CFO estimation.

1.2. Main Contributions of this Work

In this paper, we propose a low-complexity multi-CFO estimation method and an ICA based semi-blind equalization structure for multiuser-CoMP OFDM systems. Our work is different in the following aspects:

- First, a group of orthogonal sequences are employed for separation of multiple CFOs during the preamble, based on their orthogonal cross-correlation. The simultaneous multi-CFO estimation is performed at each base station, by using the property of the CFO-induced inter-carrier interference (ICI). This is different from other existing works, as a multi-dimensional search is divided into a series of low-complexity mono-dimensional searches.
- Second, an ICA based semi-blind equalization structure in CoMP OFDM systems is proposed, where a small number of pilot symbols are used to eliminate the permutation and quadrature ambiguity induced by ICA, based on exploring the cross correlation between the original and equalized pilot symbols. It also requires a lower complexity than the method in [18], as the permutation ambiguity and the quadrature ambiguity can be solved simultaneously, rather than sequentially as in the precoding aided approach.
- Simulation results show that the proposed orthogonal sequences based multi-CFO estimation and ICA based semi-blind equalization scheme provides a BER performance close to the ideal case with perfect CSI and no CFO, and has a much better mean square error (MSE) performance than the CAZAC sequence based CFO estimation method [20].

The system model is presented in Section 2. The orthogonal sequences based multiple CFOs estimation is described in Section 3. The ICA based semi-blind structure is proposed in Section 4. Section 5 is the complexity analysis. Simulation results are shown in Section 6. Section 7 draws the conclusion.

Throughout the paper, We use bold symbols to represent vectors/matrices, and use superscripts $*$, T and H to denote the complex conjugate, transpose, and complex conjugate transpose of a matrix, respectively. \mathbf{I}_N and $\mathbf{0}_N$ are an $(N \times N)$ identity matrix and an $(N \times N)$ all-zero matrix, respectively. $\text{diag}\{\mathbf{x}\}$ denotes a diagonal matrix whose diagonal elements are entries of vector \mathbf{x} . $[\mathbf{X}]_{u,v}$ is the entry of matrix \mathbf{X} in the u -th row and v -th column. The real part of a complex is denoted by $\Re\{\cdot\}$. $|\cdot|$ denotes the absolute value of a complex number. The angle of a complex is expressed as $\angle\{\cdot\}$. $\|\cdot\|$ is the Frobenius norm.

2. SYSTEM MODEL

We consider an uplink multiuser-CoMP OFDM system. A total of K users transmit different signals simultaneously, using the same set of N subcarriers to the M separated base stations. A single antenna is assumed for each base station and each user. Each receiver experiences multiple CFOs from K users. The source signal of each user is transmitted in frames, containing N_g OFDM blocks, while each OFDM block is prepended with a cyclic prefix (CP) of length L_{cp} before the transmission, and removed at the receiver to avoid inter-block interference.

We define the k -th user's signal vector as $\mathbf{s}^k(i) = [s^k(0, i), s^k(1, i), \dots, s^k(N-1, i)]^T$, where $s^k(n, i)$ denotes the symbol on the n -th subcarrier ($n = 0, 1, \dots, N-1$).

1) in the i -th ($i = 0, 1, \dots, N_s - 1$) block transmitted by the k -th user. \mathbf{F} is the N -point unitary DFT matrix with the (u, v) entry $[\mathbf{F}]_{u,v} = \frac{1}{\sqrt{N}} W^{uv}$, where $W = e^{-j2\pi/N}$ ($u, v = 0, 1, \dots, N - 1$). The channel is assumed to be quasi-static block fading [1], and the CSI remains constant for the duration of a frame. $\mathbf{h}^{m,k}$ is the channel impulse response matrix between the k -th user and the m -th base station with L paths. $\phi^{m,k} = \Delta f^{m,k} / f_0$ is the normalized CFO, where $\Delta f^{m,k}$ is the CFO between the k -th user and the m -th base station, and f_0 is the subcarrier spacing. At the m -th receiver, after the removal of CP, the received signals vector $\mathbf{y}^m(i) = [y^m(0, i), y^m(1, i), \dots, y^m(N - 1, i)]^T$ in the i -th block combined with all K users' signals and multiple CFOs is expressed as [19]-[28]

$$\mathbf{y}^m(i) = \sum_{k=0}^{K-1} \mathbf{E}(\phi^{m,k}) \mathbf{h}^{m,k} \mathbf{F}^H \mathbf{s}^k(i) + \mathbf{z}^m(i) \quad (1)$$

where $\mathbf{E}(\phi^{m,k}) = \text{diag}\{[1, Q, \dots, Q^{(N-1)}]\}$ is the normalized CFO matrix with $Q = e^{\frac{j2\pi\phi^{m,k}}{N}}$, $\mathbf{z}^m(i)$ is the additive white Gaussian noise (AWGN) vector, whose entries are independent and identically distributed (i.i.d.) complex Gaussian random variables with zero mean and variance of σ^2 . Equation (1) is transformed to the frequency domain by taking the N -point DFT of $\mathbf{y}^m(i)$:

$$\begin{aligned} \mathbf{Y}^m(i) = \mathbf{F} \mathbf{y}^m(i) &= \sum_{k=0}^{K-1} \mathbf{F} \mathbf{E}(\phi^{m,k}) \cdot \mathbf{h}^{m,k} \cdot \mathbf{F}^H \mathbf{s}^k(i) + \mathbf{Z}^m(i) \\ &= \sum_{k=0}^{K-1} \mathbf{C}^{m,k} \mathbf{H}^{m,k} \mathbf{s}^k(i) + \mathbf{Z}^m(i) \end{aligned} \quad (2)$$

where $\mathbf{Y}^m(i) = [Y^m(0, i), Y^m(1, i), \dots, Y^m(N - 1, i)]^T$ is the frequency-domain received signals vector at the m -th base station. $\mathbf{C}^{m,k} = \mathbf{F} \mathbf{E}(\phi^{m,k}) \mathbf{F}^H$ is the ICI matrix caused by the CFO between the k -th user and the m -th base station. $\mathbf{H}^{m,k} = \mathbf{F} \mathbf{h}^{m,k} \mathbf{F}^H = \text{diag}\{[H^{m,k}(0), H^{m,k}(1), \dots, H^{m,k}(N - 1)]\}$ is the relevant channel frequency response matrix between the m -th base station and the k -th user, with $H^{m,k}(n)$ denoting the channel frequency response element on the n -th subcarrier. $\mathbf{Z}^m(i) = \mathbf{F} \mathbf{z}^m(i)$ is the frequency-domain noise matrix.

It can be shown that the ICI matrix $\mathbf{C}^{m,k}$ have the following properties.

Property I: $\mathbf{C}^{m,k}$ is a circulant matrix transformed from diagonal matrix $\mathbf{E}(\phi^{m,k})$ [11].

Property II: The sum of the elements in any row or any column is equal to one. That is, $\sum_{u=0}^{N-1} [\mathbf{C}^{m,k}]_{u,v} = 1$, $\sum_{v=0}^{N-1} [\mathbf{C}^{m,k}]_{u,v} = 1$ and $\sum_{u=0}^{N-1} \sum_{v=0}^{N-1} [\mathbf{C}^{m,k}]_{u,v} = N$.

3. Orthogonal Sequences Based Multi-CFO Estimation

In this section, we describe the orthogonal sequences based multi-CFO estimation method. Let \mathbf{S}_{oth}^k denote the K -th user's orthogonal training sequence

matrix of size $(N_p \times P)$, where N_p is the number of subcarriers and P is the training sequence length during the preamble. The k -th user's ICI matrix along with the channel frequency response matrix, *i.e.*, $\mathbf{C}^{m,k} \mathbf{H}^{m,k}$, at the m -th base station can be separated from other users' signals, accomplished by the application of a group of orthogonal sequences on N_p subcarriers and K users during the preamble. The properties listed in Section 2 are used to estimate the channel frequency response matrix $\hat{\mathbf{H}}^{m,k}$, and then the k -th user's ICI matrix $\hat{\mathbf{C}}^{m,k}$ at the m -th base station is decoupled. Finally, multiple CFOs are searched for from a range of trail CFO values in the frequency domain at M base stations simultaneously.

With the known k -th user's orthogonal training sequence matrix \mathbf{S}_{orth}^k , the combination of the ICI matrix and the channel frequency response matrix $\mathbf{U}^{m,k}$ between the k -th user and the m -th receiver is separated from other users's signals at the m -th receiver as

$$\mathbf{U}^{m,k} = \frac{1}{P} \mathbf{Y}_{orth}^m [\mathbf{s}_{orth}^k]^H \quad (3)$$

where \mathbf{Y}_{orth}^m is the received training sequences matrix of size $N_p \times P$ during the preamble at the m -th receiver. In order to allow multi-CFO separation at each base station, the orthogonal training matrices \mathbf{S}_{orth}^k and $\mathbf{S}_{orth}^{k'}$ of the k -th user and the k' -th user must satisfy:

$$\mathbf{S}_{orth}^k [\mathbf{S}_{orth}^{k'}]^H = \begin{cases} P \mathbf{I}_N & k = k' \\ \mathbf{0}_N & k \neq k' \end{cases} \quad (4)$$

We found out that these sequences can be obtained from a group of well studied Hadamard matrices [18], [30]-[31], where any two different rows are orthogonal to each other. We select a unique row in a Hadamard matrix, and apply it onto each subcarrier of different users during the preamble. Note that, the training sequences are obtained from a Hadamard matrix which can be created only for the case where P is a multiple of 4 [30]-[31].

Shown from (4), although any two different users' training sequences matrices are orthogonal to each other on all subcarriers, and thus each user's CFO-induced ICI matrix can be separated from other users' signals, the separated ICI matrix of each user is still coupled with the channel frequency response matrix. However, the ICI matrix is a circulant matrix, and the channel frequency response matrix is a diagonal matrix. Based on the properties of the ICI matrix and the channel frequency response matrix, $\hat{\mathbf{H}}^{m,k}$, the estimate of the channel frequency response matrix between the m -th receiver and the k -th user, is written as

$$\hat{\mathbf{H}}^{m,k} = \text{diag} \left\{ \sum_{u=0}^{N-1} [\mathbf{U}^{m,k}]_u \right\} \quad (5)$$

where $[\mathbf{U}^{m,k}]_u$ is the u -th row of matrix $\mathbf{U}^{m,k}$, and $\sum_{u=0}^{N-1} [\mathbf{U}^{m,k}]_u$ is an $(N \times 1)$ vector.

Then, the CFO-induced ICI matrix $\hat{\mathbf{C}}^{m,k}$ between the k -th user and the m -th base station is obtained with very low complexity as:

$$\hat{\mathbf{C}}^{m,k} = \mathbf{U}^{m,k} [\hat{\mathbf{H}}^{m,k}]^{-1} \quad (6)$$

In reality, the value of CFO $\Delta f^{m,k}$, is less than half of the subcarrier spacing [9],[11], so the normalized CFO is in the range of -0.5 to 0.5 . Since the ICI matrix is estimated, the CFO $\hat{\phi}^{m,k}$ between the k -th user and the m -th base station is estimated by searching for the minimum distance between $\hat{\mathbf{C}}^{m,k}$ and $\mathbf{C} = \mathbf{F}\mathbf{E}(\tilde{\phi})\mathbf{F}^H$ with the trial CFO value $\tilde{\phi}$ as

$$\begin{aligned} \hat{\phi}^{m,k} &= \arg \min_{\tilde{\phi} \in (-0.5, 0.5)} \|\hat{\mathbf{C}}^{m,k} - \mathbf{C}\|_2^2 \\ &= \arg \min_{\tilde{\phi} \in (-0.5, 0.5)} \|\hat{\mathbf{C}}^{m,k} - \mathbf{F}\mathbf{E}(\tilde{\phi})\mathbf{F}^H\|_2^2 \end{aligned} \quad (7)$$

The multiple CFOs can be estimated simultaneously, dividing the complex multi-dimensional search into a series of low-complexity mono-dimension searches. However, it is hard to compensate for multiple CFOs if not impossible [21], because each base station has received multiple CFOs from users, resulting in that the number of different CFOs may be more than the number of users' signals. The traditional CFO compensation by pre-multiplying the CFO value is impossible at the receiver. To solve this problem, a CFO correction technique is used under the feedback model in [31]-[32], by feeding the CFOs between different users from base stations back to users, aiming at reducing the differences of transmission frequencies among users. Then, the CFO compensation is performed by pre-multiplying the inverse of the ICI matrix $\hat{\mathbf{C}}_m^H$ with adjusted CFO value to $\mathbf{y}(i) = [\mathbf{y}_0(i), \mathbf{y}_1(i), \dots, \mathbf{y}_{M-1}(i)]^T$ at each base station as $\hat{\mathbf{y}}(i) = \hat{\mathbf{C}}_m^H \mathbf{y}(i)$ in the i -th block. The compensation error matrix between base station m and user k is denoted as $\mathbf{Q}_{m,k} = \hat{\mathbf{C}}_m^H \mathbf{C}_{m,k}$. The compensation error matrix may become a non-identity matrix, meaning that a subcarrier frequency component is affected by other subcarrier frequency components.

4. ICA Based Semi-Blind Equalization

With accurate CFO estimation and compensation, the received signals become a linear mixture of the transmitted signals. The separated base stations can jointly detect and equalize the multiple user's signals by employing ICA [10]-[11] on each subcarrier. However, as the instinct drawback of ICA models, the permutation ambiguity and phase ambiguity exist in the ICA separated signals. In order to eliminate the ambiguity, the de-rotation process is first used to correct some possible phase shifts in the ICA separated signals, enabling the signal to have a same phase rotation in a data frame on a separated substream. Second, the pilots of length N_{pil} , known at the receiver, are extracted to resolve the

permutation and quadrature ambiguity, by using the cross-correlation between the equalized and original pilot symbols. The received signals vector $\mathbf{Y}(n, i) = [\dot{Y}^0(n, i), \dot{Y}^1(n, i), \dots, \dot{Y}^{M-1}(n, i)]^T$ on the n -th subcarrier is re-expressed as

$$\dot{\mathbf{Y}}(n, i) = \mathbf{Q}(n)\mathbf{H}(n)\mathbf{s}(n, i) + \mathbf{Z}(n, i) \quad (8)$$

where $\mathbf{s}(n, i) = [s^0(n, i), s^1(n, i), \dots, s^{K-1}(n, i)]^T$ is the K users' signals vector on the n -th subcarrier, $\mathbf{Q}(n)$ is the CFO estimation error matrix on the n -th subcarrier, $\mathbf{H}(n)$ and $\mathbf{Z}(n, i)$ are the $M \times K$ channel frequency response matrix and $M \times 1$ noise vector on the n -th subcarrier, respectively.

4.1. ICA Based Frequency-Domain Equalization

ICA aims to extract the original source data from the linear mixture based on the statistics of received signals. To obtain the spatially uncorrelated signals, principal component analysis (PCA) [10]-[11] is first used to whiten the received signals, as a standard ICA preprocessing step. The whitening matrix $\mathbf{W}(n)$ is obtained from the eigenvalue decomposition of the autocorrelation matrix of received signals as

$$\mathbf{R}_{yy}(n) = E\{\dot{\mathbf{Y}}(n, i)\dot{\mathbf{Y}}^H(n, i)\} = \mathbf{U}(n)\mathbf{\Lambda}(n)\mathbf{U}^H(n) \quad (9)$$

The whitening matrix is given as

$$\mathbf{W}(n) = \mathbf{\Lambda}^{-1/2}(n)\mathbf{U}^H(n) \quad (10)$$

such that

$$\mathbf{W}(n)E\{\dot{\mathbf{Y}}(n, i)\dot{\mathbf{Y}}^H(n, i)\}\mathbf{W}^H(n) = \mathbf{I}_K \quad (11)$$

JADE [10]-[11], one of the well-established ICA algorithms, based on the joint diagonalization of cumulant matrices of the received signal, is applied on each frame of length N_s to get the unitary matrix $\mathbf{V}(n)$ on the n -th subcarrier. The ICA equalized signals vector $\tilde{\mathbf{s}}(n, i) = [\tilde{s}^0(n, i), \tilde{s}^1(n, i), \dots, \tilde{s}^{K-1}(n, i)]^T$ on the n -th subcarrier and in the i -th OFDM block is given by:

$$\tilde{\mathbf{s}}(n, i) = \mathbf{V}(n)\mathbf{W}(n)\dot{\mathbf{Y}}(n, i) + \tilde{\mathbf{Z}}(n, i) \quad (12)$$

where $\tilde{\mathbf{Z}}(n, i) = \mathbf{V}(n)\mathbf{W}(n)\mathbf{Z}(n, i)$ is the output noise matrix from the ICA based equalization. However, there exists an ambiguity matrix $\mathbf{G}(n)$ in $\tilde{\mathbf{s}}(n, i)$, *i.e.*

$$\tilde{\mathbf{s}}(n, i) = \mathbf{G}(n)\mathbf{s}(n, i) \quad (13)$$

It is assumed that the N_s blocks in a data frame on the n -th subcarrier has the same ambiguity matrix $\mathbf{G}(n)$ composed of two indeterminacies as

$$\mathbf{G}(n) = \mathbf{P}(n)\mathbf{D}(n) \quad (14)$$

where $\mathbf{P}(n)$ denotes the permutation ambiguity matrix and $\mathbf{D}(n)$ denotes the quadrant ambiguity matrix. The ambiguity which remains in the ICA equalized signals can be resolved by the proposed pilot aided method described below.

4.2. Phase Deviation

A de-rotation process can correct possible phase deviations in the ICA equalized signals, to ensure that all of the symbols, including the separated pilots and source symbols on the n -th subcarrier in a frame, have a same phase rotation on each separated substream. The equalized signals vector $\tilde{\mathbf{s}}(n, i) = [\check{s}^0(n, i), \check{s}^1(n, i), \dots, \check{s}^{K-1}(n, i)]^T$ after phase correction is denoted as

$$\check{\mathbf{s}}(n, i) = \tilde{\mathbf{s}}(n, i)[\alpha_k(n)/|\alpha_k(n)|] \quad (15)$$

where

$$\alpha_k(n) = \left(\frac{1}{N_s - 1} \sum_{i=0}^{N_s} [\check{s}^k(n, i)]^4 \right)^{-\frac{1}{4}} e^{j\frac{\pi}{4}} \quad (16)$$

is the de-rotation factor obtained from the statistical characteristics of $\tilde{\mathbf{s}}(n, i)$ with QPSK modulation. The de-rotation process introduces a phase rotation of θ ($\theta \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$) in $\check{\mathbf{s}}(n, i)$, which might vary from frame to frame, and from user to user.

4.3. Permutation and Quadrature Ambiguity Resolution

After the de-rotation process, the data in a frame for each separated user on a subcarrier has the same quadrature and permutation ambiguity, which can be resolved by utilizing the cross correlation property between the original and equalized pilot symbols. The cross correlation $\rho_{\pi_k(n), k}(n)$ between the ICA equalized pilot symbols $\check{s}_{pil, \pi_k(n)}(n, i)$ and the original pilot symbols $s_{pil, k}(n, i)$ on the n -th subcarrier for the k -th user is defined as

$$\rho_{\pi_k(n), k}(n) = \frac{1}{N_{pil}} \sum_{i=0}^{N_{pil}-1} \{ [\check{s}_{pil, \pi_k(n)}(n, i) e^{j\theta_{\pi_k(n)}}] s_{pil, k}^*(n, i) \} \quad (17)$$

where $\theta_{\pi_k(n)} = \frac{\pi}{2}l$ $l \in \{0, 1, 2, 3\}$ is the possible phase rotation for the $\pi_k(n)$ -th separated user on the n -th subcarrier. In the noiseless case, *i.e.* $\sigma^2 = 0$, we have

$$\rho_{\pi_k(n), k}(n) = \begin{cases} 1 & \pi_k(n) = k \\ 0 & \pi_k(n) \neq k \end{cases} \quad (18)$$

The phase rotation of $\theta_{\pi_k(n)}$ on the $\pi_k(n)$ -th user can be found by

$$e^{j\theta_{\pi_k(n)}} = \frac{s_{pil, k}(n, i)}{\check{s}_{pil, \pi_k(n)}(n, i)} \quad (19)$$

In the presence of noise, we can search for the largest real part of cross correlation to find the correct order $\pi_k^{cor}(n)$ and the phase rotation $\theta_{\pi_k^{cor}(n)}^{cor}$ si-

multaneously for the separated π_k -th user:

$$\begin{aligned}
 [\theta_{\pi_k^{cor}(n)}, \pi_k^{cor}(n)] &= \arg \max_{\theta_{\pi_k(n)}, \pi_k(n)} \Re\{\rho_{\pi_k(n),k}\} \\
 &= \arg \max_{\theta_{\pi_k(n)}, \pi_k(n)} \Re\left\{ \frac{1}{N_{pil}} \sum_{i=1}^{N_{pil}} \{[\check{s}_{pil, \pi_k(n)}(n, i) e^{j\theta_{\pi_k(n)}}] \cdot s_{pil,k}^*(n, i)\} \right\}
 \end{aligned} \tag{20}$$

$\pi_k^{cor}(n)$ is considered as the correct order which can be arranged into the matrix of

$$\mathbf{P}^{-1}(n) = \text{diag}\{[\pi_0^{cor}(n), \pi_1^{cor}(n), \dots, \pi_{K-1}^{cor}(n)]\} \tag{21}$$

and $\theta_{\pi_k^{cor}(n)}$ is the correct phase rotation matrix as

$$\mathbf{D}^{-1}(n) = \text{diag}\{[e^{j\theta_{\pi_0^{cor}(n)}}, e^{j\theta_{\pi_1^{cor}(n)}}, \dots, e^{j\theta_{\pi_{K-1}^{cor}(n)}}]\} \tag{22}$$

The $\pi_k^{cor}(n)$ and $\theta_{\pi_k^{cor}(n)}$ are considered as the highest possibility of being the correct order and the correct phase rotation, when the real part is the largest one.

5. Complexity Analysis

In this section, the complexities of the proposed CFO estimation approach and the pilot aided ICA based equalization are presented in terms of the number of complex multiplications. The complexity of the ML based CFO estimation method [19] and the precoding aided and ICA based equalization structure [8] are also analyzed for comparison. $q = 1/\Delta$ is the homogenization of step size Δ between -0.5 and 0.5 for the CFO search.

As shown in Table 1, the proposed CFO estimation method has a much lower complexity than the ML based CFO estimation approach which requires an exhaustive search with an prohibitive complexity for real systems and is impractical for real systems. The multiple CFOs are separated with the complexity order of N^2MKP , while the matrix inversion dominates the computational complexity in the ML based CFO estimator, and has $N^2K^3P^3M$ times higher complexity than the proposed CFO estimation method. For the CFO search, the complexity of the ML based CFO estimator increases exponentially with the number of base stations and users, *i.e.*, q^{MK} multiplications are required, while the hadamard sequences based multi-CFO estimation has the computational complexity of qK , which increases linearly with the increase of the number of users. The proposed CFO estimation method benefits from dividing a complex multi-dimensional into a number of low-complexity one-dimensional searches. Also, the complexity of the simultaneous multi-CFO estimation is independent of the number of base stations.

Both of the proposed pilot aided method and the precoding aided method [18] use JADE [10]-[11] for equalization. However, they use different indeterminacy resolutions for permutation and quadrature ambiguity.

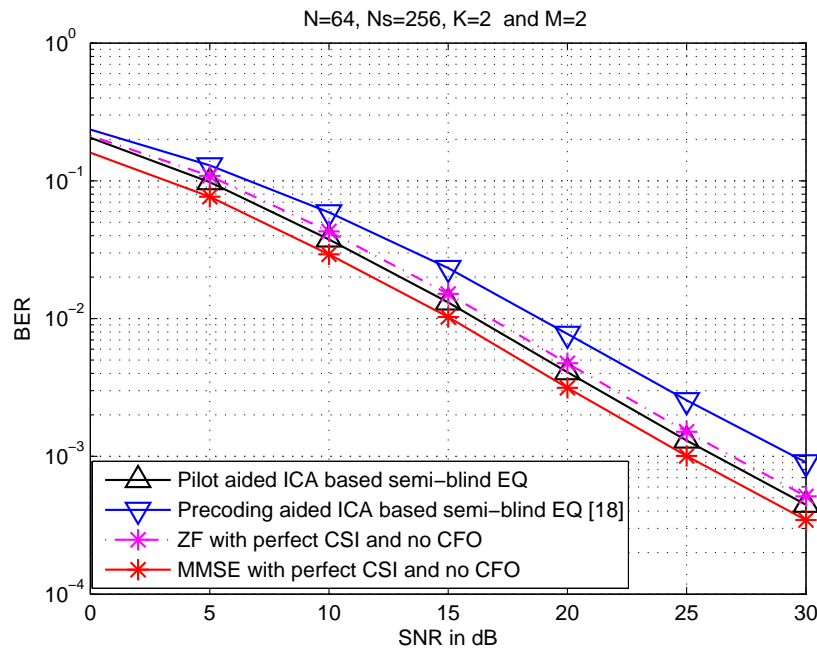


Fig. 1. BER performances of semi-blind equalization structures with CFO, in comparison to ZF and MMSE based equalization in the perfect case (EQ: equalization)

The proposed pilot aided semi-blind equalization requires a much lower complexity than the precoding aided method. The proposed semi-blind equalization uses a small number of pilot symbols to eliminate the ambiguity, with only a complexity of $4NN_{pil}K$. While the precoding aided semi-blind structure requires a pair of precoding-decoding process at both transmitter and receiver side, and has a good BER performance only when the frame size is big, because a large number of received blocks are required to eliminate the ambiguity. Also, the permutation and quadrature ambiguity can be resolved simultaneously by the proposed method, rather than sequentially as in [18]. Therefore, the proposed pilot aided and ICA based equalization has lower computational complexity.

6. Simulation Results

We use simulation results to demonstrate the performance of the proposed orthogonal sequence based CFO estimation and pilot aided ICA based signal equalization for multiuser-CoMP OFDM systems with $K=2$ users and $M=2$ base stations. A CP of length $L_{cp}=16$ is used and the number of subcarriers is $N=64$ for the OFDM systems. Each data frame with QPSK modulation consists of

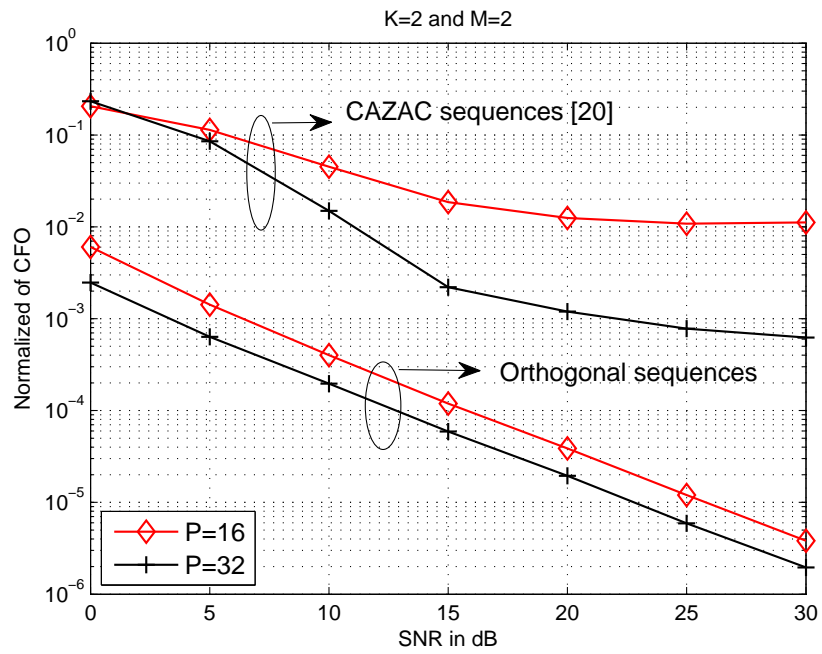
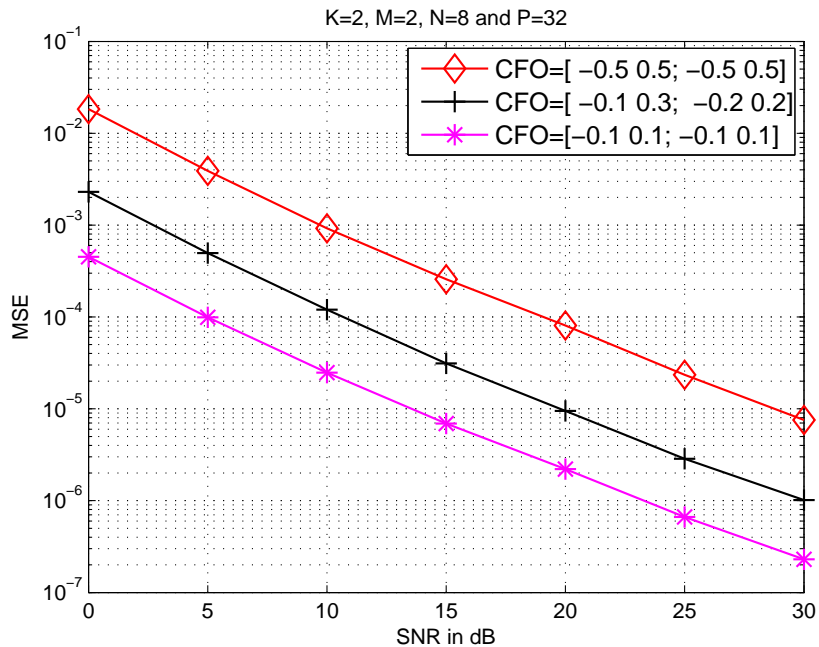


Fig. 2. MSE performance of the proposed Hadamard pilot based multiple CFOs estimation for multiuser OFDM CoMP systems

Table 1. Computational Complexity

CFO Estimation			
Orthogonal sequences		Maximum likelihood [19]	
Item	Complexity order	Item	Complexity order
CFO separation	$N^2 MKP$	Problem formulation for CFO	$(NKP)^4 M^2$
CFO search	qK	CFO search	q^{MK}
Ambiguity Elimination			
Pilot aided		Precoding aided [18]	
Item	Complexity order	Item	Complexity order
Ambiguity elimination	$4NN_{pil}K$	Reordering	$NN_s(K!)$
		Quadrature elimination	NK

**Fig. 3.** Impact of different CFOs on the MSE of CFO estimation

$N_s=256$ symbol blocks, where the first 6 blocks ($N_{pil}=6$) of each frame are used as pilots, resulting in a low training overhead of 2.3%. The Clarke's block fading channel model [1] with an exponential power delay profile is employed, where the CSI remains constant during a frame. The root mean square (RMS) delay spread of the channel is 86 ns. The resulting channel impulse response length is $L=7$. The orthogonal sequences of length $P=32$ are assigned across $N_p = 16$ subcarriers to separate and estimate the multiple CFOs at each base

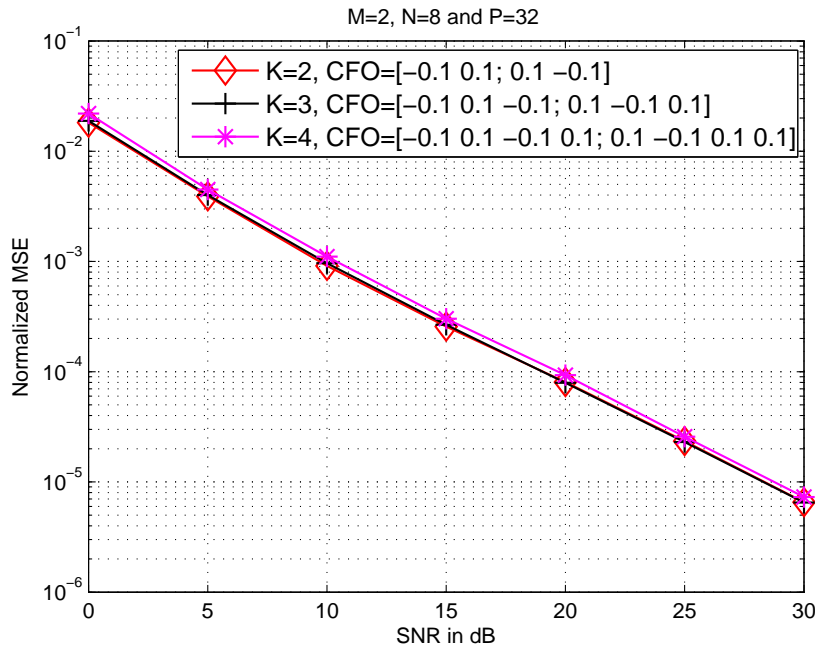


Fig. 4. Impact of the number of users on the normalized MSE performance of the CFO estimation

station during the preamble. A step size $\Delta=0.001$ results in a search of $q = 1000$ possible CFO values within the range of -0.5 to 0.5 for the CFO search.

Fig. 1 demonstrates the BER performance of the proposed orthogonal sequences based multi-CFO estimation and ICA based equalization scheme, compared with the zero forcing (ZF) based and the minimum mean square error (MMSE) based equalization methods. The precoding aided semi-blind equalization method [18] is also used for comparison. The CFOs are set to $[-0.3 \ 0.3; 0.3 \ -0.3]$ and the precoding constant is set to 0.25 [18]. The proposed multi-CFO estimation and ICA based structure provides a BER performance close to the ZF based and the MMSE based equalization methods with perfect CSI and no CFO. The pilot aided and ICA based structure outperforms the precoding aided method by around $1 - 2$ dB, showing the effectiveness of our proposed method in ambiguity elimination.

Fig. 2 demonstrates the MSE performance of the proposed orthogonal training sequence based multi-CFO estimation for multiuser OFDM CoMP systems, with training sequences length $P = 16$ and $P = 32$. The CFO estimation MSE is defined as

$$\text{MSE} = \sum_{m=0}^{M-1} \sum_{k=0}^{K-1} [(\phi^{m,k} - \hat{\phi}^{m,k})]^2 \quad (23)$$

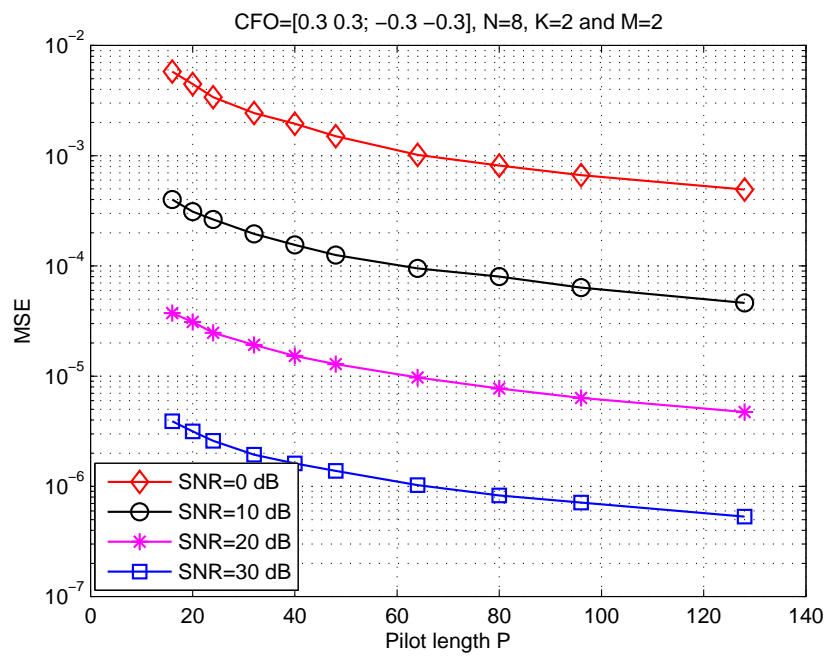


Fig. 5. Impact of pilot length on the MSE performances of the proposed multi-CFO estimation

The CFOs are set to $[-0.3 \ 0.3; 0.3 \ -0.3]$. The proposed CFO estimation method significantly outperforms the CAZAC sequences based method in [20] in terms of the MSE performance, especially at high signal-to-noise ratio (SNR). This is because our proposed method can successfully separate multiple user's ICI matrices, and achieve multi-CFO estimation at each base station, while the CAZAC sequence based CFO estimation can not completely eliminate the MAI caused by different CFOs at base station.

Fig. 3 shows the impact on the MSE performance of multi-CFO estimation with different sets of CFOs over a range of SNRs. The MSE performances improve as the SNR increases and have no error floor at high SNR region. The CFOs as $[-0.1 \ 0.1; 0.1 \ -0.1]$ has the best MSE performance, while the CFOs as $[-0.5 \ 0.5; 0.5 \ -0.5]$ has the worse performance. Their performance gap implies that the performance becomes better when the CFO value is closer to zero.

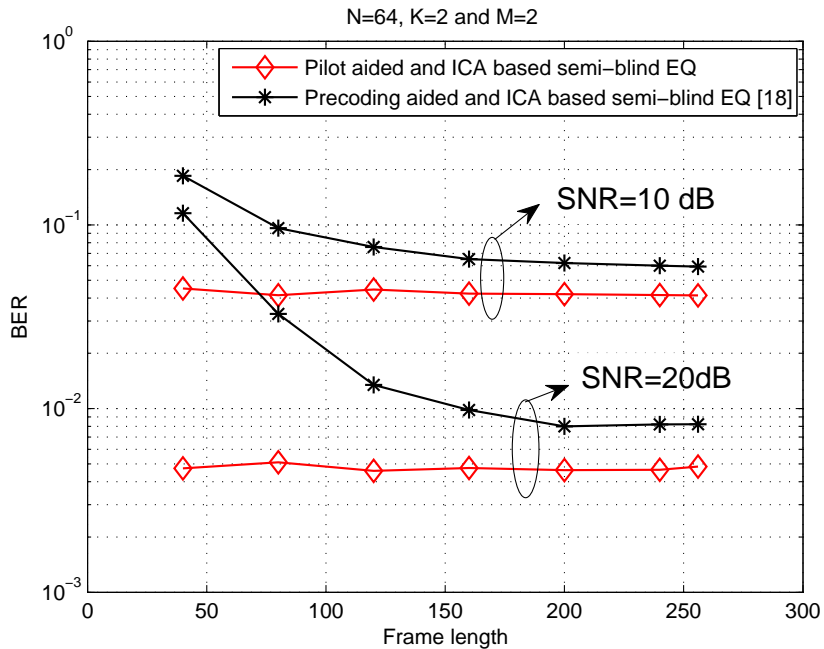


Fig. 6. Impact of frame size on the BER performances of pilot aided and precoding aided semi-blind equalization structures (EQ: equalization)

The impact of the number of users on the proposed multi-CFO estimation is showed in Fig. 4, where the number of users varies from 2 to 4. For fair comparison, CFO values are set to 0.1 or -0.1 only, and the MSE for the proposed

CFO estimation is normalized to

$$\text{NMSE} = \frac{\sum_{m=0}^{M-1} \sum_{k=0}^{K-1} [(\phi^{m,k} - \hat{\phi}^{m,k})]^2}{\sum_{m=0}^{M-1} \sum_{k=0}^{K-1} [\phi^{m,k}]^2} \quad (24)$$

The proposed CFO estimation presents a robust performance against the increase of the number of users.

Fig. 5 shows the impact of the pilot length on performance of the proposed CFO estimation method with different SNRs. The MSE performance for CFO estimation improves with the increase of pilot length, until the pilot length reaching a around $P = 32$ for different SNRs. It then becomes steady.

Fig. 6 demonstrates the impact of frame size on the BER performance of the pilot aided and precoding aided semi-blind equalization structures, at SNR = 10 dB and 20 dB. The performance of the pilot aided method remains constant over a wide range of frame sizes, while the precoding aided method is very sensitive to the change of frame length from small to medium. This is because the precoding aided approach requires a large frame of data to achieve a good performance. Thus, the pilot aided ICA based equalization is more robust against channel variations.

7. Conclusion

We have proposed a low-complexity orthogonal sequences based multi-CFO estimation approach, and a pilot aided and ICA based semi-blind equalization structure for multiuser-CoMP OFDM systems. Based on the properties of the orthogonal sequences extracted from the Hadamard matrix, the multiple CFOs are separated and estimated simultaneously. The multi-dimensional multi-CFO search is divided into a serials of one-dimensional searches with low complexity and high accuracy. By utilizing the cross-correlation between the original and equalized pilot symbols, the permutation and phase ambiguity due to the ICA model are resolved simultaneously, without increasing the computational complexity. The proposed pilot-aided semi-blind equalization structure does not provide a BER performance closed to the ideal case with perfect CSI and no CFO, but also has a much better MSE performance than the CAZAC sequence based CFO estimation scheme [10]. The proposed CFO estimation has a robust performance against the increase of the number of users, and its MSE performance becomes better when the CFO value is closer to zero. The proposed pilot aided ICA based equalization is more robust against channel variations, than the precoding aided semi-blind structure.

Appendix: Proof of Property II of ICI Circulant Matrix

The CFO matrix is right multiplied by a unitary IDFT matrix, with the element in the a -th row and the v -th column ($a = 0, 1, \dots, N - 1; v = 0, 1, \dots, N - 1$)

written as

$$[\mathbf{E}(\phi)\mathbf{F}^H]_{a,v} = \frac{1}{\sqrt{N}}[Q^a W^{-av}]_{a,v} \quad (25)$$

By left multiplying the unitary DFT matrix to (25), the ICI matrix is formed, and its element $[\mathbf{C}]_{u,v}$ in the u -th row and the v -th column ($u = 0, 1, \dots, N-1; v = 0, 1, \dots, N-1$) is written as

$$\begin{aligned} [\mathbf{C}]_{u,v} &= [\mathbf{F}\mathbf{E}(\phi)\mathbf{F}^H]_{u,v} \\ &= \left[\frac{1}{N} \sum_{a=0}^{N-1} W^{ua} (Q^a W^{-av}) \right]_{u,v} \\ &= \frac{1}{N} \sum_{a=0}^{N-1} \left[Q^a W^{a(u-v)} \right]_{u,v} \end{aligned} \quad (26)$$

Substituting $Q = e^{\frac{j2\pi\phi}{N}}$ and $W = e^{-j2\pi/N}$ into (26) yields

$$\begin{aligned} \sum_{v=0}^{N-1} [\mathbf{C}]_{u,v} &= \frac{1}{N} \sum_{u=0}^{N-1} \sum_{a=0}^{N-1} \left[e^{j2\pi a(\phi+v-u)} \right]_{u,v} \\ &= \frac{1}{N} \sum_{a=0}^{N-1} \left[e^{\frac{j2\pi a(\phi+v)}{N}} + e^{\frac{j2\pi a[(\phi+v)-1]}{N}} + \dots + e^{\frac{j2\pi a[(\phi+v)-(N-1)]}{N}} \right]_v \\ &= \frac{1}{N} \left[N + \sum_{a=1}^{N-1} \left[e^{\frac{j2\pi a(\phi+v)}{N}} + e^{\frac{j2\pi a[(\phi+v)-1]}{N}} + \dots + e^{\frac{j2\pi a[(\phi+v)-(N-1)]}{N}} \right] \right]_v \\ &= \frac{1}{N} \left[N + \sum_{a=1}^{N-1} e^{\frac{j2\pi a(\phi+v)}{N}} \left[1 + e^{\frac{-j2\pi a}{N}} + e^{\frac{-j4\pi a}{N}} + \dots + e^{\frac{-j2\pi(N-1)a}{N}} \right] \right]_v \end{aligned} \quad (27)$$

since $(1 + e^{\frac{-j2\pi a}{N}} + \dots + e^{\frac{-j2\pi(N-1)a}{N}})$ in (27) is the geometric sequence with common ratio of $e^{\frac{-j2\pi a}{N}}$ and initial term of 1, (27) is rewritten as

$$\begin{aligned} \sum_{u=0}^{N-1} [\mathbf{C}]_{u,v} &= \frac{1}{N} \left[N + \sum_{a=1}^{N-1} e^{\frac{j2\pi a(\phi+v)}{N}} \left(\frac{e^{-j2\pi a} - 1}{e^{\frac{-j2\pi(N-1)a}{N}} - 1} \right) \right]_v \\ &= 1 \end{aligned} \quad (28)$$

Similarly, $\sum_{v=0}^{N-1} [\mathbf{C}]_{u,v} = 1$ and $\sum_{v=0}^{N-1} \sum_{u=0}^{N-1} [\mathbf{C}]_{u,v} = N$ can be proved.

References

1. Goldsmith, A.: Wireless communications. Cambridge University Press, London, U.K. (2005)

2. 3GPP Technical Report 36.913 Version 9.0.0: Requirements for further advancements for Evolved Universal Terrestrial Radio Access (E-UTRA)(LTE-Advanced) (Release 9) (Dec. 2009)
3. Sklavos, A., Weber, T., Costa, E. Haas, H., Schulz, E.: Joint detection in multi-antenna and multi-user OFDM systems. *Multi-Carrier Spread Spectrum and Related Topics*. pp. 191-198 (May 2002)
4. Shamai, S., Somekh, O., Zaidel, B.: Multi-cell communications: a new look at interference. *IEEE Journal on Selected Areas in Communications*. vol. 28, no. 9, pp. 1380-1408 (Dec. 2010)
5. Andrews, J.: Interference cancellation for cellular systems: a contemporary overview. *IEEE Transaction on Wireless Communication*. vol. 12, no. 2, pp. 19-29 (Apr. 2005)
6. Marsch, P., Khattak, S., Fettweis, G.: A framework for determining realistic capacity bounds for distributed antenna systems. In: *Proceedings of the IEEE Information Theory Workshop*. pp. 571-575, Chengdu, China (Oct. 2006)
7. Coon, J., Armour, S., Beach, M., McGeehan, J.: Adaptive frequency domain equalization for single-carrier MIMO systems. In: *Proceedings of the IEEE International Conference on Communications*. pp. 2487-2491, Paris, France (Jun. 2004)
8. Wu, Y., Zhu, X. Nandi, A. K.: Adaptive layered space-frequency equalization for single-carrier MIMO systems. In: *Proceedings of the 13th European Signal Processing Conference*. Antalya, Turkey (Sep. 2005)
9. Sarperi, L., Zhu, X. Nandi, A. K.: Semi-blind layered space-frequency equalization for single-carrier MIMO system with block transmission. *IEEE Transactions on Wireless Communications*. vol. 7, no. 4, pp. 1203-1207 (Apr. 2008)
10. Cardoso, J.F.: High-order contrasts for independent component analysis. *Neural Computation*. vol. 11, pp. 157-192 (Jan. 1999)
11. Hyvarinen, A., Karhunen, J., Oja, E.: *Independent component analysis*. John Wiley & Sons, New York, U.S.A. (May 2002)
12. Petropulu, A., Zhang, R., Lin, R.: Blind OFDM channel estimation through simple linear precoding. *IEEE Transactions on Wireless Communication*. vol. 3, no. 2, pp. 647-655 (Mar. 2004)
13. Lin, R., Petropulu, A.: Linear precoding assisted blind channel estimation for OFDM systems. *IEEE Transactions on Vehicular Technology*. vol. 56, no. 3, pp. 1155-1164 (May 2007)
14. Gao, F., Nallanathan, A.: Blind channel estimation for OFDM systems via a generalized precoding. *IEEE Transactions on Vehicular Technology*. vol. 56, no. 3, pp. 1155-1164 (May 2007)
15. Gao, F., Nallanathan, A.: Blind channel estimation for MIMO OFDM systems via nonredundant linear precoding. *IEEE Transactions on Signal Processing*. vol. 55, no. 2, pp. 784-789 (Jan. 2007)
16. Obradovic, D. Madhu, N., Szabo, A., Wong, C. S.: Independent component analysis for semiblind signal separation in MIMO mobile frequency selective communication channels. In: *Proceedings of the IEEE international Conference on Neural Networks*. pp. 53-58. Budapest, Hungary (Jul. 2004)
17. He, L., Ma, S., Wu, Y., Ng, T.: Semiblind iterative data detection for OFDM systems with CFO and doubly selective channels. *IEEE Transaction on Communications*. vol. 58, no. 12, pp. 3491-3499 (Dec. 2010)
18. Gao, J., Zhu, X., Nandi, A. K.: Non-redundant precoding and PAPR reduction in MIMO OFDM systems with ICA based blind equalization. *IEEE Transactions on Wireless Communications*. vol. 8, no. 6, pp. 3038-3049 (Jun. 2009)

19. Morelli, M., Mengali, U.: Carrier-frequency estimation for transmissions over selective channels. *IEEE Transactions on Communications*. vol. 48, no. 9, pp. 1580-1589 (Aug. 2000)
20. Wu, Y., Bergmans, J. W. M., Attallah, S.: Carrier frequency offset estimation for multiuser MIMO OFDM uplink using CAZAC sequences: performance and sequence optimization. In: *Proceedings of the IEEE Wireless Communications and Networking Conference*. pp. 1-5, Budapest, Hungary (Apr. 2009)
21. Wang, H., Xia, X., Yin, Q.: Distributed space-frequency codes for cooperative communications systems with multiple carrier frequency offsets. *IEEE Transactions on Wireless Communications*. vol. 8, no. 2, pp. 1045-1055 (Feb. 2009)
22. Zirikoff, B. W., Cavers, J. K.: Coordinated multi-cell systems: carrier frequency offset estimation and correction. *IEEE Transactions on Selected Areas in Communications*. vol. 28, no. 9, pp. 1490-1501 (Dec. 2010)
23. Wu, M., Wu, Y.: A new ICI matrices estimation scheme using Hadamard sequences for OFDM systems. *IEEE Transactions on Broadcasting*, vol. 51, no. 3, pp. 305-314 (Sep. 2005)
24. Xia, X., Ching, P., Chen, Y., Ma, W.: Signal Detection in a Space-Frequency Coded Cooperative Communication System With Multiple Carrier Frequency Offsets by Exploiting Specific Properties of the Code Structure. *IEEE Transactions on Vehicular Technology*. vol. 58, no. 7, pp. 3396-3409 (Sep. 2009)
25. Weng, L., Au, E. K. S., Chen, P. W. C., Murch, R. D., Cheng, R. S., Mow, W. H., Lau, V. K. N.: Effect of carrier frequency offset on channel estimation for SISO/MIMO-OFDM systems. *IEEE Transaction on Wireless Communications*. vol. 6, no. 5, pp. 1854-1863 (May 2007)
26. Sezginer, S., Bianchi, P., Hachem, W.: Asymptotic Cramer Rao bounds and training design for uplink MIMO-OFDMA systems with frequency offsets. *IEEE Transactions on Signal Processing*. vol. 55, no. 7, pp. 3606-3622 (Jul. 2007)
27. Tsai, Y., Huang, H., Chen, Y., Yang, K.: Simultaneous carrier frequency offset estimation for multi-point transmission in OFDM systems. In: *Proceedings of the IEEE Global Telecommunication Conference*. pp. 1-5, Houston, USA (Dec. 2011)
28. Mehrpouyan, H., Blöstein, S.: Synchronization in cooperative networks: Estimation of multiple carrier frequency offsets. In: *Proceedings of IEEE International Conference on Communications*. pp. 1-6, Cape Town, South Africa (May 2010)
29. Iossifides, A. C.: Complex orthogonal coded binary transmission with amicable Hadamard matrices over Rayleigh fading channels. In: *Proceedings of the IEEE Symposium on Computers and Communications*. pp. 335-340, Kerkyra, Greece (Jun. 2011)
30. Trinh, Q. K., Fan, P. Z.: Construction of multilevel Hadamard matrices with small alphabet. *Electronics Letters*. vol. 44, no. 21, pp. 1250-1252 (Oct. 2008)
31. Papadogiannis, A., Bang, H.J., Gesbert, D., Hardouin, E.: Efficient selective feedback design for multicell cooperative networks. *IEEE Transactions on Vehicular Technology*. vol. 60, no. 1, pp. 196-205 (Jan. 2011)
32. Zirikoff B. W., Cavers, J. K.: Coordinated multi-cell systems: carrier frequency offset estimation and correction. *IEEE Journal on Selected Areas in Communications*. vol. 28, no. 9, pp. 1490-1501 (Dec. 2010)

Yufei Jiang received the M.Sc degree in Telecommunications and Computer Networks from the London South Bank University, London, U.K., in 2008. He is currently working towards a Ph.D. degree on a joint programme between the University of Liverpool, Liverpool, U.K., and the Xi'an JiaoTong Liverpool University, Suzhou, China. His research interests include MIMO, space-time process, OFDM, cooperative communication, synchronization and blind source separation.

Xu Zhu received the B.Eng. degree (with first class honors) from the Department of Electronics and Information Engineering, the Huazhong University of Science and Technology, Wuhan, China, in 1999, and the Ph.D. degree from the Department of Electrical and Electronic Engineering, the Hong Kong University of Science and Technology, Hong Kong, in 2003. Since May 2003, she has been with the Department of Electrical Engineering and Electronics, the University of Liverpool, Liverpool, U.K., where she is currently a senior lecturer. Dr. Zhu was the vice chair of the 2006 and 2008 ICA Research Network International Workshops. She has acted as session chair and technical program committee member for more than 20 international conferences, such as IEEE GLOBECOM 2009 and IEEE ICC 2010. Her research interests include cooperative communications, cognitive networks, cross-layer design, MIMO and adaptive equalization, etc.

Eng Gee Lim received the BEng(Hons) and PhD degrees in Electrical and Electronic Engineering from the University of Northumbria, UK in 1998 and 2002 respectively. Dr Lim worked for Andrew Ltd, a leading communications systems company in the United Kingdom from 2002 to 2007. His responsibilities were to conduct the research and development for a cost-effective solution for all terrestrial microwave systems operating at frequencies between 400 MHz and 90 GHz. Dr Lim joined Xi'an Jiaotong-Liverpool University in 2007, where he is currently an associate professor. His research interests are Antennas, RF/microwave engineering, EM measurements/simulations, and Wireless Communication Network.

Yi Huang received BSc in Physics (Wuhan, China), MSc (Eng) in Microwave Engineering (Nanjing, China), and DPhil in Communications from the University of Oxford, UK in 1994. He has been conducting research in the areas of wireless communications, applied electromagnetics, radar and antennas for the past 25 years. His experience includes 3 years spent with NRIET (China) as a Radar Engineer and various periods with the Universities of Birmingham, Oxford, and Essex at the UK as a member of research staff. He worked as a Research Fellow at British Telecom Labs in 1994, and then joined the Department of Electrical Engineering & Electronics, the University of Liverpool, UK as a Faculty in 1995, where he is now a full Professor in Wireless Engineering, the Head of High Frequency Engineering Research Group and MSc Programme Director.

Prof. Huang has published over 200 refereed papers in leading international journals and conference proceedings, and is the principal author of the popular

Yufei Jiang, Xu Zhu, Enggee Lim and Yi Huang

book *Antennas: from Theory to Practice* (John Wiley, 2008). He has received many research grants from research councils, government agencies, charity, EU and industry, acted as a consultant to various companies, and served on a number of national and international technical committees and been an Editor, Associate Editor or Guest Editor of four of international journals. He has been a keynote/invited speaker and organiser of many conferences and workshops (e.g. IEEE iWAT 2010, WiCom 2006, 2010 and LAPC2012). He is at present the Editor-in-Chief of *Wireless Engineering and Technology*, a UK National Rep of European COST-IC1102, Executive Committee Member of the IET Electromagnetics PN, a Senior Member of IEEE, and a Fellow of IET.

Received: March 3, 2012; Accepted: December 5, 2012.

Speech Unit Category based Short Utterance Speaker Recognition

Nakhat Fatima¹, Xiaojun Wu¹ and Thomas Fang Zheng^{1*}

¹ Center for Speech and Language Technologies, Division of Technical Innovation and Development, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, 100084, Beijing, China

fatima@csl.tit.tsinghua.edu.cn, xjwu@tsinghua.edu.cn

* Corresponding Author: fzheng@tsinghua.edu.cn

Abstract. Information of speech units like vowels, consonants and syllables can be a kind of knowledge used in text-independent Short Utterance Speaker Recognition (SUSR) in a similar way as in text-dependent speaker recognition. In such tasks, data for each speech unit, especially at the time of recognition, is often not enough. Hence, it is not practical to use the full set of speech units because some of the units might not be well trained. To solve this problem, a method of using speech unit *categories* rather than individual phones is proposed for SUSR, wherein similar speech units are put together, hence solving the problem of sparse data. We define Vowel, Consonant, and Syllable Categories (VC, CC and SC) with Standard Chinese (Putonghua) as a reference. A speech utterance is recognized into VC, CC and SC sequences which are used to train Universal Background Models (UBM) for each speech unit category in the training procedure, and to perform speech unit category dependent speaker recognition, respectively. Experimental results in Gaussian Mixture Model-Universal Background Model (GMM-UBM) based system give a relative equal error rate (EER) reduction of 54.50% and 40.95% from minimum EERs of VCs and SCs, respectively, for 2 seconds of test utterance compared with the existing SUSR systems.

Keywords: Short Utterance Speaker Recognition, Vowel Categories, Universal Background Vowel Category Model.

1. Introduction

In real-time speaker recognition there can be circumstances where obtaining appropriate speech data might become difficult. Conditions like noisy background, faulty devices or unwilling speakers, are a few of the many factors that might reduce the available amount of speech data. Hence, it becomes essential to make the most of the available data efficiently by employing short utterances of speech to identify a speaker. Short Utterance Speaker Recognition (SUSR) has recently emerged as an important area of

research. SUSR technology attempts to use small amount of data from short utterances of speech for recognition purpose. In conventional speaker recognition systems, a large amount of data is required for a successful speaker identification system. However, SUSR attempts to use smaller amount of data for the recognition task. Different research endeavors have described different lengths for short utterance from around a minute to less than 10 seconds. However, recently the meaning of short utterance has taken a turn to mean less than 3 seconds of speech.

When utterance lengths are short, there is not enough variation in speech for an efficient speaker recognition task. Therefore, the performance of speaker recognition systems deteriorates sharply when utterance lengths are shorter than 10 seconds.

It is because of the declining performance of speaker recognition with short utterances that we were motivated to investigate the characteristics of sounds at phoneme level, which is the smallest meaningful unit of speech. The pronunciation idiosyncrasies of a speaker begin at phoneme level. Therefore we study the phonemes in different languages. To begin our study we limited ourselves to the vowels. We created a set of vowel categories to use them as phoneme sequences and then perform speaker recognition on 1-3 seconds of the sequence lengths. We extend our research onto other speech units like consonants and later the combination of vowels and consonants i.e. syllables.

Ultimately this research is intended to have the following contributions:

1. To describe the importance of phonemes and their combination (speech units like vowels and syllables) in Short Utterance speaker recognition.
2. Making the SUSR system essentially language and text independent by designing speech unit categories.
3. Improving the performance of speaker recognition by employing speech unit category sequences for training speaker models and for performing recognition with utterance lengths of 1-3 seconds.
4. Reduction of comparative training length by using speech unit categories.

1.1. Related Work

Generally, speaker recognition comprises of three steps: feature extraction, statistical modeling and score calculation. Most of the conventional speaker recognition systems perform feature extraction using various frame sizes shifts or vocal source features called wavelet octave coefficients of residues (WOCOR) [1, 2, 3]. For statistical modeling, eigenvoice and factor analysis subspace estimation are applied [4]. CGMM-UBM (Clustered GMM-UBM) and sub GMM-UBM methods are also used in many system, which find their basis in clustering and subspace estimation [5]. Though conventional, the improvements built on these techniques are widely used in speaker recognition. These methods, however, require a large amount of speech data for training as well as recognition purposes. This is the reason why these

methods do not perform well when utterance length becomes shorter than a minute.

The innovative method of Phonetic or prosodic speaker recognition makes use of idiosyncrasies in a person's speech to identify a speaker e.g. pronunciation habits [6, 7]. This method, too, requires a large amount of speech processing and training data, again making it quite difficult to be performed when speech data is not enough.

In order to solve the problem of large data required to perform speaker recognition, research has lead the use of Factor Analysis, Joint Factor Analysis (JFA), Support Vector Machine (SVM) and I-vector based technologies [1, 6, 7]. In other works performing short utterance speaker recognition, Dimension decoupled GMM is applied [8]. Training and testing with 10 seconds of speech on variations of GMM and SVM have also shown improvements in results [7]. Since SUSR is a complicated task, it has also been used in combination with video aid for better results [9]. Mel-frequency cepstrum coefficients (MFCC) and Hidden Markov Models (HMM) based system has been employed in using phonemes for speaker recognition [10] which suggests that using larger amount of training data; phoneme level speaker recognition can be improved. A study of formant contours at consonant-vowel boundary has shown that a speaker can be identified using information at the consonant to vowel transition boundary, which makes use of the distinctive "speaker-style" [11]. The biannual National Institute of Standards and Technology (NIST) Speaker Recognition Evaluation (SRE) also presents the optional test of 10-second evaluation [12]. The length of utterance in most of these cases is around 10 seconds.

I-vector based technology has proven its vast advantage in recent studies [6]. The i-vector based SUSR achieves a minimum of 21.98% of EER for 2 seconds of test utterance. In performing SUSR, most of the research has been conducted using 10 seconds of utterances..

1.2. Performance of Existing SUSR Techniques

The term "Short Utterances" has taken various meanings and hence evolved in recent years, ranging from more than a minute to only a few seconds. Different studies have taken a different measure for short utterance. Previously most of the works defined short utterance to be around 10 seconds of speech [12]. However, as the need of short utterance speaker recognition grew, utterance lengths were defined to be shorter. Of late, research has taken a turn and utterances as short as 2-3 seconds are being used in order to recognize, validate and discern a speaker.

Working in the domain of short utterances, it has been observed that the overall performance decreases significantly when utterance length becomes smaller than 20 seconds. A comparison of different researches on SUSR [6, 7, 13, 14, 15, 16, 17] was made and it was observed that although performance of speaker recognition in terms of EER% remains close to 5-6% on average (minimum EER of 3.13% and maximum being 13.47% for more

than a minute of test utterances) when utterance lengths are more than a minute long down to 20 seconds, there is a considerable drop in performance when utterance lengths go below 20 seconds of speech (EER reaching around 12% with 20 seconds of speech). The decline in performance is more marked when utterances are shorter than 10 seconds (average EER at 10 seconds from different systems being 18.32%). The average EER% of SUSR systems using 4 seconds of speech decline to 24.15% (min: 17.38%, max: 31.3%) and when 2 seconds of speech is used, the performance of different methods give an average of 28.94% EER (min: 21.98%, max: 36.16%). Hence, there is an exponential drop in performance when utterance lengths are below 10 seconds for test speech.

Considering the dwindling efficiency below 10 seconds, it becomes intriguing as well as incumbent that shorter utterances than 10 seconds be investigated for better results. Our motivation to strive for a solution to the problem of short utterances in speaker recognition comes from Phonetic Speaker Recognition (PSR). We attempt to find the usefulness of phoneme similarities in speaker recognition when using short utterances. Hence, we propose a category based short segment speaker recognition that can use vowel, consonant and syllable categories in SUSR.

The rest of the paper is organized as follows. In Section 2 we describe the framework of the Speaker Recognition system used. Experimentation details on continuous speech divided in random segments are given in Section 3. Section 4 discusses Vowel Categories and their performance in SUSR. This discussion and performance analysis is continued in Section 5 with Syllable, Consonant and improved Vowel Categories. Finally, we draw our conclusions and present future research prospects in Section 6.

2. Speaker Recognition System Framework

2.1. Proposal and Framework Overview

In our previous work [18], we devised an innovative design of speaker recognition, which was based on phoneme-classes for speaker recognition, creating a Phoneme Category Based Short Utterance Speaker Recognition (PCBSUSR). For this purpose, a vowel-class set was defined, i.e. a language-independent set of vowel categories. The vowel-class set was defined using linguistic knowledge of vowels. Consonants were not used for the study because consonants are more complex to recognize and categorize. The vowel categories were defined to help cover maximum number of the vowels in most languages. In training phase a group of vowel-class models was built with respect to each speaker making a speaker vowel-class model. In test phase the test utterance was first recognized into a sequence of phonemes and then speaker recognition was performed on the utterance using the vowel categories from the recognized phonemes. The system performance was not

very efficient as it gave 46% EER. There were, however, many constraints during the experiments including quality of speech segmentation and recognition. In the subsequent study, initially the speech segmentation problem was addressed. We decided to use speech segments such that each segment was a complete unit of meaningful sound, vowels in this case. In this study, we devised five VCs. In [19], we presented Syllable categories (SCs), wherein utterances were recognized into syllables and then assigned to appropriate categories. The results of this study showed that the syllables are very significant speech with respect to speaker recognition and they contain a large amount of speaker information.

In the current research, we present how improvement in category classification can be beneficial to speaker recognition. Hence we propose new vowel categories and their improvement, and a new set of syllables categories (SC). In addition, we present consonant categories (CC) and compare their performance with vowel and syllable categories so that we can determine which type of speech units give better performance.

This research would help in understanding that though speech recognition plays an important role in text dependent speaker recognition, when the duration gets shorter than one second, exact phone recognition is not important. Instead, broader phoneme-classes/categories can provide good results as well. This makes the system reasonably text independent. The use of phoneme properties helps bring together similar phonemes under single category for speaker recognition. Also, this research will help in understanding that there are speaker related phonemic idiosyncrasies which can be put to use in SUSR by exploring the role of vowels and other speech units in speaker recognition. Furthermore, this study will strive to show that in speaker recognition with short utterances, it is important to use complete segments of phonemes rather than continuous speech cut in random segments. The random segments lose speaker information where a complete speech unit retains it.

2.2. Baseline System

Our baseline system has been organized as follows. Feature extraction is performed on a 20-millisecond frame every 10-milliseconds. The pre-emphasis coefficient is 0.97 and hamming windowing is applied to each pre-emphasized frame. Voice activity detection based on energy is performed with each frame, resulting in the labeling of either valid or invalid. 16-dimensional MFCC features are extracted from the utterances only for those valid frames with 30 triangular Mel filters used in the MFCC calculation. For each frame, the MFCC coefficients and their first delta coefficients form a 32-dimensional feature vector. Gender-independent UBM consists of 1,024 mixture components and are trained using EM algorithm [20]. For the MAP training [20], only mean vectors were adapted with a relevance factor of 16. The baseline system is a speaker verification system based on the conventional

GMM-UBM. Note, that previously [18], the EER of this system was 46% using phoneme categories.

2.3. Database

The experiments were performed using the “Annotated Speech Corpus of Chinese Discourse (ASCCD) from Institute of Linguistics, Chinese Academy of Social Sciences, Beijing, China” [21]. The dataset consists of sentences uttered by 10 Chinese speakers (5 males and 5 females). Each speaker spoke 47 sentences of average duration of 25 seconds, in normal speech. 30 utterances of the speakers were selected to train the UBM while the remaining (17) utterances were used for testing purpose.

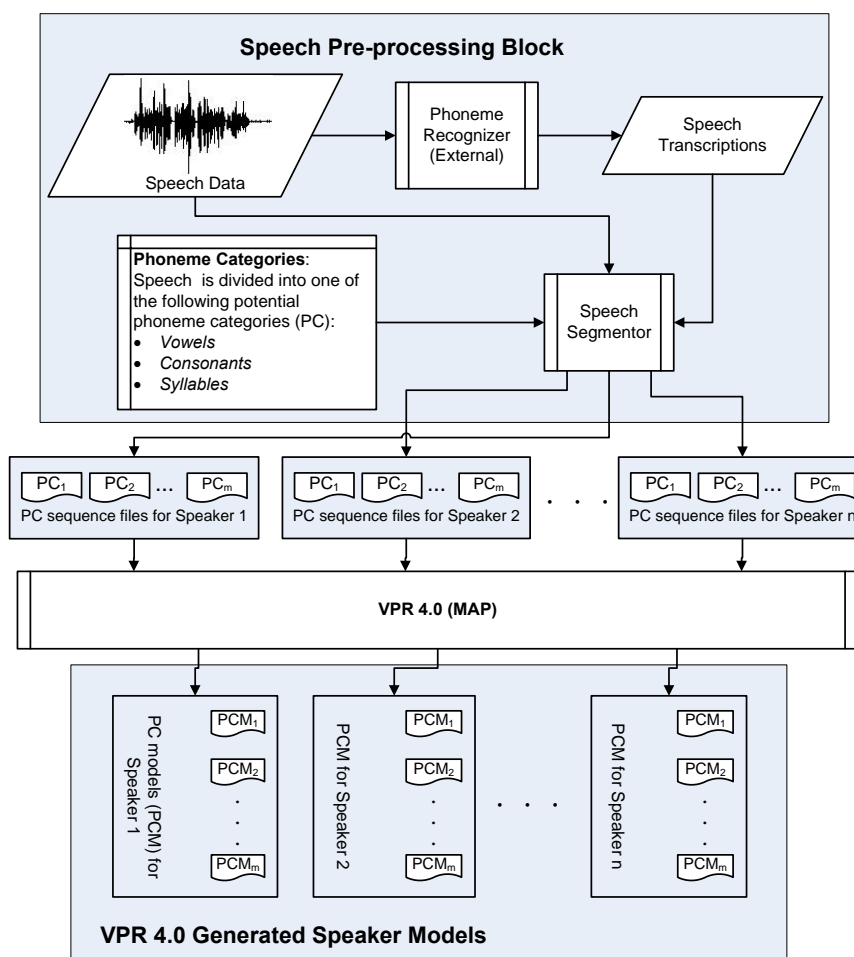


Fig. 1. Block diagram of Model Training in SUSR using Phoneme Categories (PC)

2.4. Universal Background Model Training and Testing

In our experiments, a Universal Background Model (UBM) [7] is built based on the EM algorithm and the GMM modeling using MAP algorithm. During speaker recognition, the test utterance is scored against its corresponding UBM, giving n scores. These n scores are fused to obtain the final score.

We propose a SUSR framework which is described as follows:

The speech would first be passed through a phoneme recognizer, which would recognize the speech into phoneme categories sequences. This would result into the phoneme category sequence files for each speaker. For example category 'a' for "speaker A" would contain 'a' type sounds from the training utterance of "speaker A". Similarly phoneme category sequence file of each phoneme category for every speaker would be created. These sequences would then be given as input to the GMM-UBM system wherein a Speaker Phoneme Category Model would be created for each Speaker Phoneme Category, resulting in speaker models against each phoneme category.

Later the same process of recognition of speech into phoneme categories would be applied during testing. The resulting phoneme category files of specified length would then be matched against the corresponding models and scored to give the recognition result.

Fig. 1 and Fig. 2 are block diagram representation of our proposed SUSR system with Phoneme Categories.

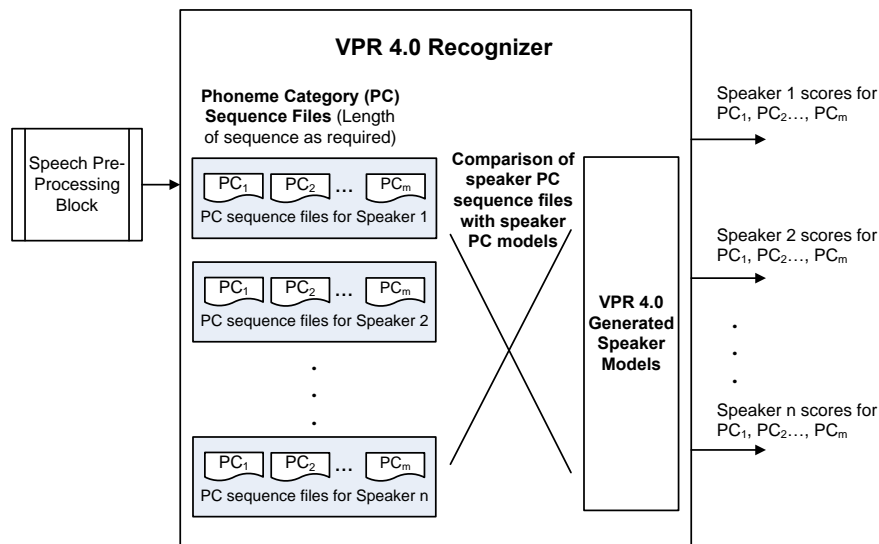


Fig. 2. Block Diagram of Recognition and Scoring in SUSR

2.5. Designing Phoneme Categories

Each speech utterance, most likely a sentence, a phrase or an exclamation, comprises of words. Each word is formed of syllables and syllables are made up of vowels, consonants or both. Therefore, it is important to take into account these meaningful units of sounds to design meaningful categories for SUSR.

We propose the following process flow for designing phoneme categories. This process flow has been shown in Fig. 3:

1. Studying the types of sound.
2. Learning the types of vowels and their properties.
3. Studying the vowel structure of most common languages spoken in the world.
4. Designing the vowel categories which would cover most languages.
5. Studying the consonant types.
6. Designing consonant categories in the most generic form.
7. Designing syllable categories based upon the vowel and consonant categories.

3. Performance of Speaker recognition with continuous speech divided randomly

3.1. Importance of Meaningful Segmentation of Speech

When we are developing a speaker recognition system to make use of short utterances, speech segmentation is an important factor. It is important that each segment of sound should be meaningful. According to Adami et al. [22], segmentation of speech in an appropriate manner is needed when prosodic/high level features are being used for speaker recognition. During normal speech, there can be many clicks, hiss and random noise sounds, like ingressive sounds at an expression of regret or laughter or other similar expressions which carry little speaker information. Also, random picking up of segments might give an unknown mix of sounds like a combination of a palatal stop and low vowel, none of which, if chosen at random, would give a consistent result. A model developed from such random sounds would vary dramatically when segmentation of speech would be changed. To elaborate this fact, we show in Section 3.2 that a system based on utterances segmented in random, does not perform well when utterance lengths are as short as 3, 2 and 1 second.

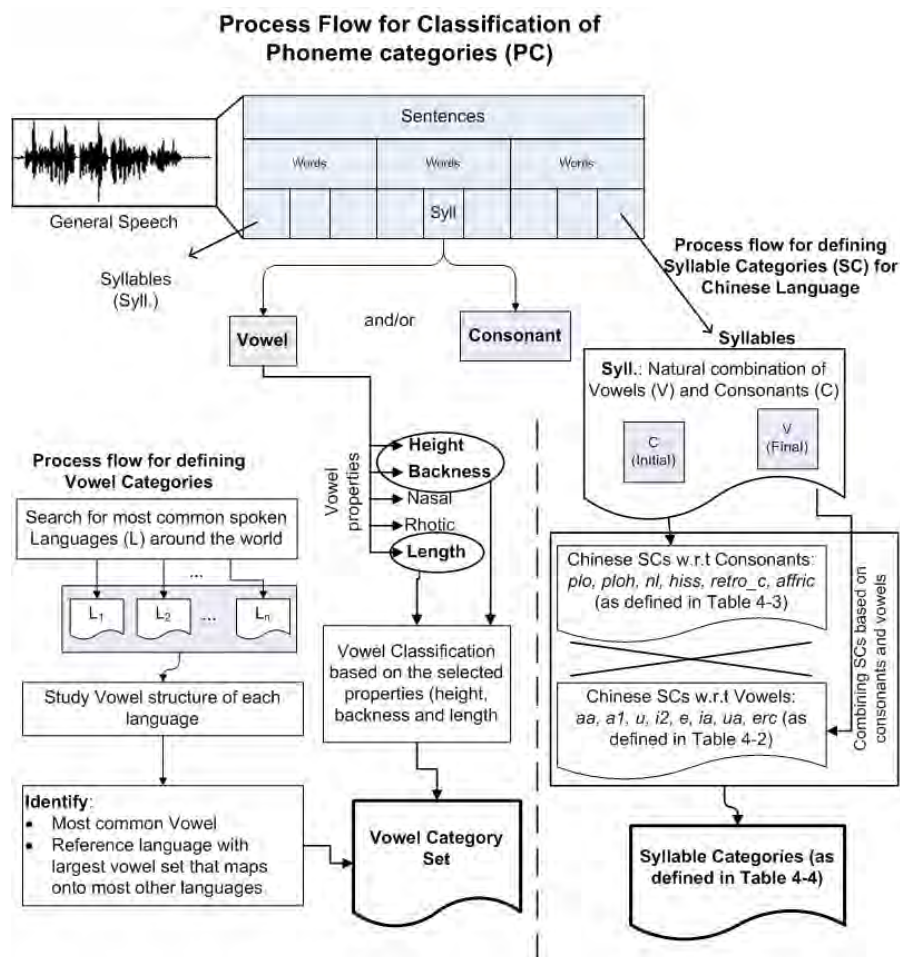


Fig. 3. Process flow of Designing Phoneme Categories

3.2. Experimentation and Results

In order to test the system performance, we decided to first conduct speaker recognition experiments using our system without dividing speech into phoneme categories, so that a comparison could be made between categorical and continuous speech based speaker recognition systems.

We conducted an experiment with 2.5 minutes of training data for each speaker, which was used to train the GMM-UBM models for each speaker. Recognition was performed on full length of test utterances (9 minutes on average), and then by dividing the utterances into segments of 3 seconds, 2 seconds and 1 second in testing phase. The training of speaker models and testing was done with the system and database described in Section 2.

The results of this experiment are shown in Table 1.

Table 1. System Performance of SUSR with continuous Speech with Training length of 2.5 minutes

Utterance Length (Training-Testing)	EER%
2.5min-full (8min)	9.09
2.5-3 sec	19.46
2.5-2 sec	22.53
2.5-1 sec	29.56

It can be seen from Table 1 that although the system performs considerably well with long test utterance, there is a sharp decline in performance in terms of EER% when test length is equal to or less than 3 seconds. The high values of EERs show that the system performance remains unreliable and it can be improved by investigating further into the short utterance properties.

Hence the SUSR using continuous speech poses a concern for speaker recognition systems. This is why we propose categorical division (i.e. segmentation of sounds) of speech units for speaker recognition system.

This also shows that randomly dividing speech into segments does not give good results. Hence, investigating further into speech units when the utterance lengths are short, becomes important. This is why we propose categorical division of speech units for speaker recognition system as we describe in Section 4.

3.3. Importance of Phoneme based Short Utterance SR

Due to the several difficulties presented by using continuous speech and low performance of current SUSR systems, we decided to look deeper into using phonetic cues for speaker recognition. There are the following potential disadvantages of using continuous speech for speaker recognition with short utterances:

1. Using continuous speech, the performance of speaker recognition falls exponentially below 10 seconds of speech.
2. The random segments generated from continuous speech can result in unexpected and unintelligible mix of sounds when segment lengths become smaller.
3. There is not much continuous speech available in emergency situations where there might be a lot of bursts and breaks in speech.

We believe phonemes are the correct, meaningful and practical unit which should be employed in speaker recognition using short utterances. We believe phonemes based Short utterance SR will have vast advantage over continuous speech for the following reasons:

1. When recognizing a speaker using short utterances of sound, it is important that the units of sounds should be meaningful. Phonemes are the smallest meaningful unit of sound.
2. For SUSR, the utterances of speech should be practical. The random segmentation of sounds do not allow intelligible and practical utterances.
3. It has been shown by study of phonetic speaker recognition that phonetic idiosyncrasies improve speaker recognition vastly [24].
4. Various studies have shown that high level characteristics in speech have vast advantage in speaker recognition [25].
5. High level characteristics can be mapped onto the low level features like formant frequencies [11]. Therefore, high level idiosyncrasies can be mapped onto the acoustic features. This is only possible with short utterances of speech.
6. Due to physiology of phoneme production, the phonemes contain distinct wave patterns which vary from speaker to speaker. This is because of the habit of speaking e.g. style of tongue rolling, or lisping etc.
7. Phoneme combinations (like syllables) can cover more speaker specific information because of speaker habits in uttering a specific combination of sounds.

4. Vowel Categories for PCBSUSR

As described in Section 3, continuous speech, if randomly divided into segments, does not perform well when utterance lengths are short. This is why we study smaller speech units by taking the whole phoneme/syllable into account, such that the speech segment is not random, but a meaningful unit of sound.

For an SUSR system that is language independent, we studied the phonology (systematic organization of sounds in languages) of phoneme production. The detailed study was made in order to figure out those phonemes which overlap in most commonly spoken languages of the world. The kind of sounds studied were mainly the egressive sounds (sounds produced by pushing the air out of the vocal tract), e.g. plosives, affricates, fricatives, continuants and vowels. Like in our previous study [18], we initially confine ourselves to the study of vowels and their properties to choose a universal vowel set for languages. Fig. 4 is the illustration of vowels according to International Phonetic Association [26].

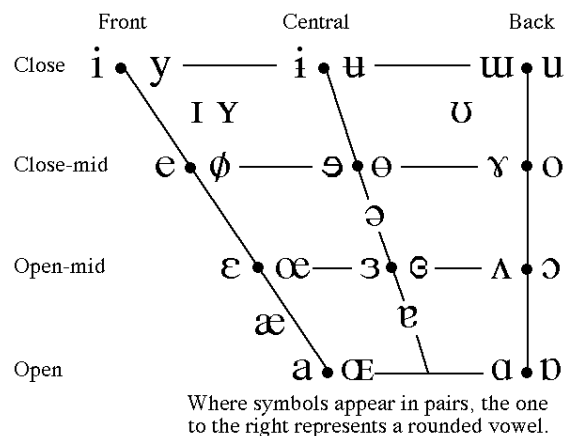


Fig. 4. Vowels in languages

4.1. Vowel Characteristics

Height, Backness, Roundedness, Nasalization and R-coloration are the features that define a vowel [27]. Considering height and backness to be the main features, vowels were searched for in main languages of the world. For the purpose of our experimentation, we selected two languages i.e. English and Chinese, keeping them as baselines for the future system. The reason for choosing English is because it has a fairly large vowel inventory, which can be mapped onto most other common languages.

Diphthongs or gliding vowels are those, which glide from one quality to another during s production, e.g. “cow”. They are different from unilateral vowels, such that boundary where change takes place is not distinctly recognizable. English and Chinese, both contain many diphthongs. It is hard to ignore them altogether, since a huge amount of speech consists of vowels and a large portion of them are diphthongs. Diphthongs, however, pose a problem since they are vowels which might contain two very different vowel types. Most of the times, one vowel is dominant. Hence we propose to solve this problem by categorizing vowels into categories in a way that diphthongs are categorized on the basis of the first dominant vowel that appears in the diphthong.

4.2. Proposed Vowel Categories

Previously in [18], we defined eight categories of vowels. However we now define the following five categories of vowels:

1. Category a: low centre vowels, and those vowels which have dominant lower central vowels.
2. Category e: mid centre.

3. Category i: front high followed by neutral vowel (e.g. /a/), front high i followed by front vowel or none (e.g. /i/, /e/) and front high i followed by back vowel (e.g. /o/, /u/).
4. Category o: mid back rounded and those vowels which have dominant mid back vowels.
5. Category u: high back and (yu): high back unrounded.

The new categories are defined because of the following reasons:

1. To make a considerably uniform distribution of vowels across categories.
2. Some previous categories did not give very good results; it is assumed that along with the quality of segmentation, there was less number of phones in those categories.

The categories are defined based on dominant vowels in case of diphthongs. This distribution can be subject to scrutiny and subsequent change.

4.3. Phoneme Extraction

The speech data from “Annotated Speech Corpus of Chinese Discourse (ASCCD) from Institute of Linguistics, Chinese Academy of Social Sciences, Beijing, China”, was used in this study. The speech data was segmented into phones based on the labels and placed into separate wave files. For the purpose of training and testing, the phones from each Vowel Category (VC) were concatenated into single wave file, representing one VC.

4.4. Universal Background Vowel Category Model (UBVCM) Training and Testing

Our study confines to vowel categories only, therefore, in our experiments, a Universal Background Vowel Category Model (UBVCM) [7] is built in a UBM fashion. A batch of vowels is obtained from the annotated speech data. The vowels are then used to estimate UBVCM based on the EM algorithm and the GMM modeling, giving a group of UBVCMS.

Vowel category (VC) models replace conventional speaker models. Vowel sequence and then VC sequences are obtained from training utterances of the speaker. Each VC model is estimated from UBVCM using the MAP algorithm with features of the vowels corresponding to the VC. During speaker recognition, the speech annotated test utterance is used to obtain VCs. Each VC is scored against its corresponding UBVCM, giving n scores. These n scores are fused to obtain the final score.

4.5. Results and Discussion

Table 2 represents the results of individual categories. Compared with the EER% of our previous work [18], which was of no less than 42%, this experiment has shown a huge improvement.

Table 2. Results Comparison Of UBVCM with [18]

EER %				
VCs	Complete length	3 sec	2 sec	1 sec
Category “a”	10.00	13.90	14.27	17.46
Category “e”	17.78	20.53	20.42	21.81
Category “i”	10.56	13.76	14.03	16.18
Category “o”	10.56	15.85	15.09	18.5
Category “u”	17.78	15.44	18.63	18.9
Average	13.336	15.90	16.49	18.57
Min	10.00	13.76	14.03	16.18
Max	17.78	20.53	20.42	21.81

The following measures have been different in this approach compared to the baseline system, yielding better results.

1. **Vowel Categories:** In the previous unilateral vowels [a, e, i, o, u, v] represented the main categories. Diphthongs were divided into the categories based on the first vowel. e.g. “iang” was put under category i1 (i followed by neutral vowel). However, in the present work, vowel categories have been merged into 5 broad level categories, such that the unilateral vowels come under the same name category except the unrounded high-back vowel “v”. “v” has been merged with “u”, the rounded high-back vowel category, hence giving the categories [a, e, i, o and u]. The diphthongs are categorized based on the dominant/open vowel in it, e.g. “iang” is placed in category “a” because /a/ is the dominant vowel in it. The categories of mid vowels e.g. “e” have not been changed in order to keep a fairly balanced distribution of phoneme across the vowel categories.
 - a. **Segmentation method:** Segmentation of speech has been a big difference between our current and previous work. In the previous work [18], speech was segmented randomly before phoneme recognition and categorization. In the current work, however, annotated speech data was used on which phoneme level recognition had been performed earlier. The speech was subsequently segmented using the labels and each category was separated. Later, training and testing was performed on those vowel segments combined into one wave (.wav) file.

Following observations are made from the results in Table 2:

1. Data segmentation plays an important role. It shows that when performing SUSR, it is necessary to have a meaningful unit of sound like a complete phoneme. If speech is segmented randomly, as

described in Section 3, the individual information from each phoneme is wasted. Using a complete vowel, on the other hand, allows a speaker recognition system to make use of idiosyncrasies of a speaker in pronouncing that vowel.

2. If an entire phoneme is taken, it does not remain necessary to have accurate speech recognition. Instead broad phoneme categories can provide good results.
3. The vowel segments may be shorter than 500 ms, but being complete segments, they contain a large amount of speaker related information, pronunciation habits in particular.
4. Category "a" has given the best performance as shown in Tables 2. It is important to note that "a" is the most open vowel category in the set of VCs. Therefore, in articulating open vowels as those that fall in VC "a", speaker specific voice attributes are represented in the speech unconstricted. This is the reason vowel /a/ and other similar vowels can be a preferred choice in speaker recognition based on phonemes.
 - a. Categories "e" and "u" have higher EER as shown in Table 2.
 - b. For category "e", the reason could be the presence of phoneme "er" in category "e". The r-colored "e" or 儿 (in Chinese) has frequent occurrence in speech. The presence of the retroflex "r" brings about a constriction in the airway and consonant-like properties to the vowel, which is a variation from pure vowel. This causes a variation in speech in which one type of speech is different from other. This is why mismatch might have occurred and affected the results. This further strengthens our hypothesis that phonemes that fall under the same category provide higher chances of better results.
 - c. Similarly, for category "u", the reason of low performance could be the relative closeness of /u/. The narrowed airway due to backness of tongue and roundedness of mouth allow less amount of vocal tract information to enter the speech signal.
5. When taking phoneme categories all by themselves, pure vowels contain a larger amount of speaker information compared to diphthongs because their open and unrestricted properties allow vocal tract information to enter the speech signals.
6. Lastly but most importantly, it is worthy to note that the average training length of speech data used in our method was about 1 minute on average for each vowel category compared to 2.5 minutes of training length with continuous speech as described in Section 3. This shows that vowel categories perform better than continuous speech with smaller training durations hence reducing the computation cost and time of training process significantly.

The results from our method using VC sequences has shown an improvement of 37.73% of relative EER reduction compared to our baseline system using continuous speech and a relative EER reduction of 36.17%

compared to the minimum value of EER from the existing works (Section 1.2) using 2 seconds of speech.

5. Speech Unit Category based SUSR

In order to extend our research of phonemes from vowels to other units of speech, we initially gave a Syllable Category bases SUSR system in [19]. In Section 4, we presented improvement in vowel categories, which yielded vastly better results compared with [18]. In this section we present VCs, SCs as well as CCs for short utterance speaker recognition. Universal Background VC, CC and SC Models in Universal Background Model (UBM) fashion [7] as we did in our previous works [18, 28]. We use phonemic knowledge of phonemes to bring together similar speech units under single category. The purpose of this research is to determine which type of speech units provide better speaker recognition information.

5.1. Vowel, Consonant and Syllable Categories

Tables 3 and 4 respectively describe vowel and consonant categories (Based on Standard Chinese) used in current experiments.

Table 3. Vowel categories

VC	Representation	Vowels
Long /a/ type	aa	a, an, ang, ai, ao
Short /a/ type	a1	e, en, eng, i (zi)
/u/ type	u	u, ou, un, ong, iong, iou
/i/ and /v/ type	i2	i, v, in, ing, vn
/e/ type	e	ei, ie, a, ian, ve, van, uei, ven
/i/ diphthongs	ia	ia, iao, iang, iu,
/u/ diphthongs	ua	ua, uai, uan, uang, o, uen, ui, uo, yuan
retroflex	er	i (zhi), er

We also devise the following Syllable Categories by combining vowel and consonant categories. Since there is limitation to the valid combination of initials (consonants) and finals (vowels) [29], all vowel categories do not appear under each consonant category. liq category represents liquids, e.g. no-initial syllables and syllable starting in //.

plo_aa, plo_a1, plo_u, plo_i2, plo_e, plo_ua, ploh_aa, ploh_a1, ploh_u, ploh_i2, ploh_e, ploh_ua, nl_aa, nl_a1, nl_u, nl_i2, nl_e, hiss_aa, hiss_a1, hiss_u, hiss_ua, retro_aa, retro_a1, retro_u, retro_ua, affric_aa, affric_a1,

affric_u, affric_i2, affric_e, affric_ia, affric_ua, liq_aa, liq_a1, liq_u, liq_i2, liq_e, liq_ia, liq_ua, and retro_er. The *retro_er* category incorporates all syllables ending in retroflex vowels.

Table 4. Consonant categories

CC	Representation	Consonants
Plosives	plo	b, d, g
Aspirated Plosives	ploh	p, t, k
Nasals and Laterals	nl	m, n, l
Hiss sounds	hiss	h, f
Retroflex	retro_c	zh, ch, sh, r
Affricates and Fricatives	affric	s, z, c, j, q, x

Knowledge based approach is used to devise the categories because rules of Chinese Language limit the total number of syllables and there are some syllables against which very scant data is obtained.

5.2. Experimental Results and Discussion

With the similar training and testing conditions as described in Section 2, we perform Speaker Recognition on each of the vowel, consonant and syllable categories. Each speech unit based UBM corresponds to its specific vowel, consonant or syllable category.

The results for syllable, vowel and consonant categories have been described in Tables 5 to Table 9.

Tables 5 and 6 present the results of syllable categories and Table 7 gives statistical summary of syllable category results.

Tables 8 and 9 respectively show results of vowel and consonant categories.

Tables 5, 6 and 7 show the syllable category results. Minimum values of EER% are 9.9%, 10% and 15.38% for 3 seconds, 2 seconds and 1 second respectively. In addition, 1.1% of EER has been obtained for Retro_er category for an average 18 seconds of test length.

The following observations have been made from these results:

1. It is observed from the results in Tables 5, 6 and 7 that longer test lengths are generally associated with better performance.
2. /a/ and /a/ based diphthongs, e.g. ia and ua based categories have shown better results proving that openness in vowels contributes to better performance.
3. The lower performance in case of /u/ based categories show that performance can deteriorate with closeness of vocal tract.
4. In some cases there is a sudden degradation of performance when sequence length becomes smaller (3 seconds or less). However, generally, length of test utterance, when smaller, does not deteriorate greatly.

5. In some instances, the EERs reduce when shorter utterances are used (e.g. SC retro_ua, SC affric_u). The reason could be that when smaller utterances are used, the effect of potentially unfavorable speech units is reduced.
6. Overall, long /a/ vowels (aa and ia), /e/ based categories, plosive based syllables and affricate/fricative categories in general have performed better.
7. Retroflex categories, especially the syllables ending in a retroflex (e.g. retro_er) show the promise of extremely good results. With training length ranging between 33 seconds to 44 seconds and test length of no longer than 22 seconds, the EER% was 1.11 with ID rate of 100%. This is because the articulation of retroflex sounds varies a lot from one person to another.

Table 5. Syllable category results in terms of EER% - Part 1

SC	Full Length		3 Sec	2 Sec	1 Sec
	Length (sec)	EER%			
plo_aa	0:09-0:12	10.00	17.37	16.56	17.82
plo_a1	0:08-0:11	18.33	25.00	24.17	26.85
plo_u	0:06-0:08	12.22	19.44	20.22	20.83
plo_i2	0:02-0:03	18.88	14.81	15.00	17.85
plo_e	0:03-0:08	21.11	20.55	26.08	29.29
plo_ua	0:10-0:13	10.00	15.78	17.27	19.39
ploh_aa	0:05-0:06	18.33	17.85	16.97	18.78
ploh_a1	0:03-0:04	21.67	26.11	29.39	32.80
ploh_u	0:03-0:03	21.11	34.25	30.00	38.23
ploh_i2	0:02-0:02	18.33	18.33	26.31	24.13
ploh_e	140ms-500ms	29.44	29.44	29.44	29.44
ploh_ua	0:01-0:02	20.00	20.00	27.77	25.92
nl_aa	0:03-0:04	22.22	34.31	25.00	29.62
nl_a1	0:03-0:03	28.33	30.40	30.00	32.43
nl_u	688ms-941ms	37.78	37.78	37.78	37.78
nl_i2	0:01-0:02	21.11	21.11	25.25	20.10
nl_e	0:07-0:09	9.44	11.80	12.76	16.40

The results show that speech unit categories perform very well in SUSR. Even with conventional GMM-UBM system, the results have shown that idiosyncrasies of speaker at phoneme level have its advantages in SUSR and they do not go wasted at short segments, rather each unit of speech; vowel, consonant or syllable, contains significant information about speaker's identity.

Table 6. Syllable category results in terms of EER% - Part 2

SC	Full Length (sec)	EER%	3 sec	2 sec	1 sec
	Length	EER%			
hiss_aa	0:06-0:08	12.22	23.33	25.00	25.00
hiss_a1	0:08-0:09	27.78	28.25	27.03	27.71
hiss_u	0:04-0:06	27.78	27.27	22.77	33.53
hiss_ua	0:05-0:07	10.00	17.85	21.05	22.22
retro_aa	0:08-0:10	10.00	15.78	19.66	19.56
retro_a1	0:22-0:28	10.00	15.56	17.87	23.75
retro_u	0:11-0:13	10.00	21.49	20.87	21.66
retro_ua	0:03-0:04	18.88	28.89	26.26	26.57
affric_aa	0:09-0:10	8.33	17.83	16.88	18.42
affric_a1	0:05-0:06	11.11	21.75	26.76	28.74
affric_u	0:08-0:11	40.00	30.55	32.33	38.97
affric_i2	0:31-0:39	2.22	10.74	13.00	16.90
affric_e	0:12-0:15	10.00	12.50	14.39	15.38
affric_ia	0:15-0:19	18.33	17.56	15.73	18.25
affric_ua	0:03-0:04	11.11	14.91	10.00	21.05
liq_aa	0:06-0:09	12.22	15.79	17.77	21.69
liq_a1	0:06-0:08	20.00	26.66	26.42	32.85
liq_u	0:06-0:09	18.88	19.44	20.80	21.58
liq_i2	0:16-0:23	10.00	14.86	17.03	18.00
liq_e	0:08-0:11	9.44	13.73	13.62	17.88
liq_ia	0:04-0:05	18.33	20.00	20.68	20.40
liq_ua	985ms-0:01	20.55	20.55	20.55	22.83
retro_er	0:16-0:22	1.11	9.93	11.00	15.95

Table 7. Summary of Syllable category results in terms of EER%

	Full Length	3 sec	2 sec	1 sec
Average	16.91	20.99	21.69	24.16
Minimum	1.11	9.93	10.00	15.38
Maximum	40.00	37.78	37.78	38.97

It can be seen from Tables 8 that Vowels present an extremely feasible option for SUSR.

Minimum values of EER% for 3 seconds, 2 seconds and 1 second of test length has been 13.72, 12.93 and 16.09 respectively. The Category ia has been seen to be consistent with the best results.

In conformity with results in Section 4.5, vowel categories have shown good results. This proves that with careful selection of phoneme categories, performance of speaker recognition improves significantly when phoneme category sequences are used for speaker recognition with short utterances, even when they are as little as 1 second.

Table 8. Vowel category results in terms of EER%

VC	Complete Length (sec)		3 sec	2 sec	1 sec	
	Utterance length		EER%			
aa	0:35	0:43	10.00	14.72	15.36	17.40
a1	0:34	0:47	18.33	21.00	21.36	22.58
U	0:31	0:42	10.55	17.16	18.47	20.29
i2	0:39	0:52	10.56	14.74	14.97	16.09
E	0:30	0:38	12.22	16.41	15.85	18.73
la	0:11	0:15	11.11	13.72	12.93	17.28
ua	0:20	0:28	12.22	18.84	18.98	21.39
er	0:09	0:14	18.88	20.51	19.54	21.30
Average			12.98	17.14	17.18	19.38
Min			10.00	13.72	12.93	16.09
Max			18.88	21.00	21.36	22.58

Table 9. Consonant category results in terms of EER%

CC	Complete Length (sec)		3 sec	2 sec	1 sec	
	Utterance length	EER%	EER%			
plo	0:04	0:09	40.55	45.83	42.10	45.58
ploh	0:05	0:08	30.00	30.65	31.70	33.33
nl	0:07	0:11	30.00	29.72	28.30	30.80
hiss	0:08	0:14	30.00	30.23	38.00	36.20
retro_c	0:22	0:31	30.00	35.41	34.79	36.87
affric	0:33	0:43	30.00	29.99	32.98	34.40
plo	0:04	0:09	40.55	45.83	42.10	45.58
ploh	0:05	0:08	30.00	30.65	31.70	33.33
Average			31.76	33.64	34.65	36.20
Min			30.00	29.72	28.30	30.80
Max			40.55	45.83	42.10	45.58

Table 9 shows the results of speaker recognition performance with consonant categories. With minimum EER% of 29.72, 28.3% and 30.8% for 3 seconds, 2 seconds and 1 second, consonant categories have not shown promising results. The minimum values have been obtained with the nasal-

lateral category. The reasons behind poor performance of SUSR with consonant categories are:

1. Consonants generally have very little duration. The duration does not allow speaker variations in speech signals to be captured to their full extents.
2. Many consonants are produced without vibration of vocal cords i.e. unvoiced consonants. Due to this factor, the properties of sound are reduced to white noise like hiss, bursts and silences. Such sounds do not have any significant speaker related information in them.
3. Consonants, when on their own, do not have active signals and idiosyncratic information in them.
4. The constriction in vocal tract, when uncoupled with vowels, contributes to the lower performance of SUSR.
5. The minimum values of EER% have been obtained from nasal-lateral consonant category. This is the consonant category with most voicing activity as well as vowel like properties of laterals allow some speaker variation in speech signals.

It is because of the above factors that we do not recommend consonant sequence based speaker recognition i.e. when consonants are used on their own without their combination with vowels. However when coexisting with vowels, they impact the properties of vowels (especially at the transition boundary). This fact can be proven by the good results shown by syllable categories in Table 5, 6 and 7. Therefore, although the importance of consonants in speech is undeniable and their impact on speaker recognition can be considerable, they are not a feasible choice when used alone.

Using Syllable Categories for our SUSR system there has been a relative EER reduction of 55.61% and 54.50% compared to our baseline system with continuous speech and minimum EER recorded in literature (Section 1.2) respectively for 2 seconds of test utterance length.

6. Conclusions and Future Work

Our research discusses how phone unit categories can be used in SUSR. We used conventional GMM-UBM models for different category types. We show from an experiment on randomly divided continuous speech that a complete speech unit is very important when we are performing speaker recognition on units as small as 1 second. Our results on the experiments based on speech units like vowels, and syllables show that text dependent speaker recognition based on speech units gives significantly better results in SUSR. Although consonants are extremely important in speech, when they are used alone, they do not provide good results. However in combination with vowels i.e. syllables, they provide quite good results. We conclude from our experiments that when short utterances are used, it is better to use phonemes rather than continuous speech. Phonemes can use knowledge of utterances, while using

categories allowing the system to use rudimentary speech recognition, not requiring high accuracy. Our results show that different speech units have different performances. So, in performing real time speaker recognition, priority can be given to specific speech units while poor performing speech units can be avoided to ensure recognition.

As part of future work, we propose the following potential research directions:

1. Applying data driven statistical method for determining the phone units as an alternative of current knowledge based approach.
2. Study of formant dynamics at speech units and their role in SUSR.
3. Exploring speaker recognition in complete absence of a specific speech unit category due to fewer amounts of training data.
4. Cross speech unit category experimentation in order to understand which categories perform well against each other.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grant No. 61271389 and the National Basic Research Program (973 Program) of China under Grant No. 2013CB329302.

References

1. Vogt, R., Baker, B., Sridharan, S.: Factor analysis subspace estimation for speaker verification with short utterances. In Proceedings of INTERSPEECH 2008. 853-856. (2008).
2. Chan, W. N., Zheng, N., Lee, T.: Discrimination Power of Vocal Source and Vocal Tract Related Features for Speaker Segmentation. IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING, Vol. 15, No. 6, 1884-1892. (2007).
3. Jayanna, H. S., Prasanna, S. R. M.: Multiple Frame Size and Rate Analysis for Speaker Recognition under Limited Data Condition. IET SIGNAL PROCESSING, Vol. 3, No. 3, 189-204. (2009).
4. Kenny, P., Boulianne, G., Dumouchel, P.: Eigenvoice Modeling with Sparse Training Data. IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, Vol. 13, No. 3, 345-354. (2005).
5. Wu-fu, Lu, X. Y., Bei-qian, D., Hui, L.: Speaker Recognition based on the phonal speech Short CGMM-UBM. Journal of Circuits and Systems, Vol. 12, No. 5, 131-136. (2007).
6. Kanagasundaram, A., Vogt, R., Dean, D. B., Sridharan, S., Mason, M. W.: i-vector based speaker recognition on short utterances. in Proceedings of the 12th Annual Conference of the International Speech Communication Association, (2011).
7. Fauve, B., Evans, N., Pearson, N., Bonastre, J.-F., Mason, J.: Influence of task duration in text-independent speaker verification. In Proceedings of Interspeech, 2007. 794-797. (2007).
8. Stadelmann, T., Freisleben, B.: Dimension-Decoupled Gaussian Mixture Model for Short Utterance Speaker Recognition. in 20th International Conference on Pattern Recognition, (2010).

9. Larcher, A., Bonastre, J.-F., Mason, J. S. D.: Short Utterance-based Video Aided Speaker Recognition. in International Workshop on Multimedia Signal Processing, Cairns, Australia, (2008).
10. Fattah, M. A.: Phoneme Based Speaker Modeling to Improve Speaker Recognition. *Information*, Vol. 9, No. 1, 135-147. (2006).
11. Fatima, N., Aftab, S., Sultan, R., Shah, S. A. H., Hashmi, B. M., Majid, A., *et al.*: Speaker Recognition Using Lower Formants. In Proceedings of Proceedings of IEEE INMIC 2004. Lahore, 125-130. (2004).
12. (NIST), N. I. o. S. a. t.: (2012). [Online]. Available: <http://www.nist.gov/index.html> (current February 2012).
13. Li-fu, W., Yan-lu, X., Bei-qian, D., Hui, L.: CGMM-UBM based speaker Verification using short telephone speech. *Journal of Circuits and Systems*, Vol. 12, No. 5, 131-136. (2007).
14. Lin, L., Shu-xun, W., Gang, G.: Novel Speaker Recognition Method Based on Little Speech Data. *Journal of System Simulation*, Vol. 19, No. 10, 2272-2275. (2007).
15. McLaren, M., Vogt, R., Baker, B., Sridhara, S.: Experiments in SVM-based Speaker Verification Using Short Utterances. In Proceedings of Odyssey: The Language and Speaker Workshop. (2010).
16. Vogt, R., Lustri, C., Sridharan, S.: Factor Analysis Modelling for Speaker Verification with Short Utterances. In Proceedings of In Proceedings Odyssey 2008: The Speaker and Language Recognition Workshop. South Africa. (2008).
17. Xiao-han, L., Nan-chen, H., Bei-qian, D., Zhi-qiang, Y.: Research on the HMM-UBM and Short Text Based Speaker Verification. *Information and Control*, Vol. 33, No. 6, 762-764. (2004).
18. Fatima, N., Wu, X., Thomas Fang Zheng, Chenhao Zhang, Wang, G.: A Universal Phoneme-Set Based Language Independent Short Utterance Speaker Recognition. in 11th National Conference on Man-Machine Speech Communication (NCMMSC '11), Xi'an, China, (2011).
19. Fatima, N., Zheng, T. F.: Syllable Category Based Short Utterance Speaker Recognition. in International Conference on Audio, Language and Image Processing, Shanghai, China, (2012).
20. Reynolds, D. A.: Channel robust speaker verification via feature mapping. In Proceedings of ICASSP. 53-56. . (2003)
21. Aijun, L., Maocan, L., Xiaxia, C., Yiqing, Z., Guohua, S., Wu, H., *et al.*: Speech corpus of Chinese discourse and the phonetic research. In Proceedings of International conference of speech and language processing (ICSLP). 13-18. (2000)
22. Adami, A. G., Hermansky, H.: Segmentation of speech for speaker and language recognition. In Proceedings of 8th European Conference on Speech and Communication and Technology (Eurospeech). (2003)
23. Hagiwara, R.: Monthly mystery spectrogram webzone (2009). [Online]. Available: <http://home.cc.umanitoba.ca/~robh/archives/arc0601.html> (current March 4 2012)
24. Kohler, M. A., Andrews, W. D., Campbell, J. P., Hernandez-Cordero, J.: Phonetic Refraction for Speaker Recognition. In Proceedings of Workshop on Multilingual Speech and Language Processing. Aalborg, Denmark. (2001)
25. Kinnunen, T., Li, H.: An Overview of Text-Independent Speaker Recognition: from Features to Supervectors. *Speech Communication*, Vol. 52, No. 1, 12-40. (2010).
26. IPA: Vowel Types in languages (2005). [Online]. Available: <http://www.langsci.ucl.ac.uk/ipa/vowels.html> (current March 4 2012)
27. Wikipedia: Vowel (2012). [Online]. Available: <http://en.wikipedia.org/wiki/Vowels> (current march 4 2012)

Nakhat Fatima, Xiaojun Wu and Thomas Fang Zheng

28. Fatima, N.,Zheng, T. F.: Vowel Category Based Short Utterance Speaker Recognition. in International Conference on Systems and Informatics, YanTai, China, (2012).
29. Jiyong, Z., Fang, Z., Mingxing, X.,Shuqing, L.: Intra-syllable dependent phonetic modeling for Chinese speech recognition. In Proceedings of International Symposium on Chinese Spoken Language Processing (ISCSLP'00). Beijing, China, 73-76. (2000).

Nakhat Fatima received the Bachelor's and Master's degree in Computer Science from National University of Computer and Emerging Sciences (NUCES-FAST), Lahore, Pakistan in 2004 and 2007 respectively. She worked as a Software Development Engineer in Centre for Research in Urdu Language Processing (CRULP). She passed her defense for Ph.D. in Computer Science in December 2012 from Tsinghua University, Beijing, China. Her interests include Speaker Recognition and Text to Speech processing.

Xiaojun Wu received the Ph.D. degree in computer science and technologies from Tsinghua University, Beijing, China, in 2004. He is currently an associate professor in the Research Institute of Information Technology, Tsinghua University. His research interests include speaker recognition, natural language understanding and dialogue management.

Thomas Fang Zheng received the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 1997. He is now a Research Professor and Director of the Center for Speech and Language Technologies, Tsinghua University. His research focuses on speech and language processing. He has published more than 210 papers. He plays active roles in a number of communities, including Vice President of the Asia-Pacific Signal and Information Processing Association (APSIPA), IEEE Senior Member, Chair of the Chinese Corpus Consortium, Chair of the Standing Committee of China's National Conference on Man-Machine Speech Communication, Council Member of Chinese Information Processing Society of China, and Council Member of the Acoustical Society of China. He is an Associate Editor of the IEEE Transactions on Audio, Speech and Language Processing and APSIPA Transaction on Signal and Information Processing, he is on the editorial boards of Speech Communication, Journal of Signal and Information Processing, and the Journal of Chinese Information Processing. (Please refer to <http://csit.riit.tsinghua.edu.cn/~fzheng> for details).

Received: February 08, 2012; Accepted: November 27, 2012.

Formal Verification of Signature-monitoring Mechanisms by Model Checking

Lanfang Tan, Qingping Tan, Jianjun Xu, and
Huiping Zhou

School Computer, National University of Defense Technology,
410073 Changsha, China
{tanlanfang1022, eric.tan.6508, jjun.xu, icent}@gmail.com

Abstract. In recent decades, reliability in the presence of transient faults has been a significant problem. To mitigate the effects of transient faults, fault-tolerant techniques are proposed. However, validating the effectiveness of fault-tolerant techniques is another problem. In this paper, we present an original approach to evaluate the effectiveness of signature-monitoring mechanisms. The approach is based on model-checking principles. First, the fault tolerant model is proposed using step-operational semantics. Second, the fault model is refined into a state transition system that is translated into the input program of the symbolic model checker NuSMV. Using NuSMV, two reprehensive signature-monitoring algorithms are verified. The approach avoids the state space explosion problem and the verification was completed with practical time. The verification results reveal some undetected errors, which have not been previously observed.

Keywords: software fault-tolerance, model checking, formal verification, fault tolerance, signature monitoring mechanisms.

1. Introduction

In recent years, with the growing demand for high availability and reliability of computer systems, validating the effectiveness of fault-tolerant techniques constitutes a significant problem [1]. Fault injection techniques have been instinctively proposed, in which faults are injected into target systems to see whether the fault tolerant procedures could properly behave [2]. However, fault injection techniques have a common drawback, which is their failure to cover all fault scenarios. These techniques also tend to be very time consuming. Therefore, the development of efficient verification techniques for fault-tolerant systems is urgent. Given that the formal verification technique can target all possible "corner cases," which may be missed by conventional fault injection methods, the verification technique for fault-tolerant systems becomes a promising candidate solution. A few methods for formal verification of fault-tolerant techniques have been proposed. Generally, these methods can be classified into two categories:

- Deductive verification: a technique that proves the correctness of fault-tolerant systems using axioms and proof rules [3]. The intrinsic advantage of the deductive technique is its ability to reason out all nondeterministic errors. However, it can only be used by experts who are educated in logical reasoning and have considerable experience. Deductive verification also cannot be automatically performed.
- Model checking: In earlier research [4][5][16-18], model checking was proposed to evaluate the reliability of fault-tolerant systems. It explores all possible system states to cover all fault scenarios. However, it may suffer from the state space explosion problem.

Though model-checking techniques are promising, many theoretical questions remain. In particular, a general approach based on model checking for the evaluation of fault-tolerant techniques has not been formulated. An approach is usually specialized for a specific system, and different techniques need different validation paradigms. For example, Nicolescu et al. verified a control-flow checking technique by constructing a hypothetical program that augmented the technique, and then checks the program for undetected errors [4]. For other detection mechanisms, construction of similar programs is difficult.

In this paper, we propose a general approach to evaluate the effectiveness of signature-monitoring techniques [9-11] using model checking. It aims at formally proving the capacity of the signature-monitoring techniques to detect errors over a general class of possible applications. In other words, can we verify that some control-flow errors (CFEs) can be detected by the signature-monitoring technique, while some CFEs cannot? The purpose is to catch all undetected CFEs that potentially escape from conventional detection. The main contributions of our work are highlighted as follows:

- An original approach for the evaluation of signature-monitoring techniques is proposed, which provides a universal validation paradigm for all signature-monitoring mechanisms.
- A fault tolerant model is proposed to describe the execution of the assembly program that is strengthened by the signature-monitoring mechanism. A state transition system is refined and a procedure on how to translate the state transition system into the input program of the model checker NuSMV is explained in detail. The translation procedure can be applied to all cases.
- Two reprehensive signature-monitoring algorithms are evaluated. Some undetected errors that have not been found before are revealed. Especially for the dynamic signature-monitoring (DSM) technique, which can detect all illegal interblock errors [4][9], four kinds of errors that escaped detection were found.

The remainder of this article is organized as follows. Section 2 lists the related work. Section 3 describes the basic principle of the signature-monitoring mechanism. Section 4 presents the fault tolerant model for the assembly programs that are strengthened by the signature-monitoring mechanism. Section 5 describes the state transition system refined by the fault tolerant model and the translation of the state transition system into the input language of the

model checker. In Section 6, two representative algorithms are verified and the verification results are analyzed. Section 7 lists the comparisons. Section 8 states the conclusions and future works.

2. Related Work

Some earlier formal verification techniques have been proposed to evaluate the system dependability. Nicolescu et al. [4] proposed an original approach to evaluate the system reliability with respect to transient errors. Based on a generic model over a general class of all possible applications, the proposed validation approach allows exploring all fault scenarios. The validation confirms that DSM technique satisfies the property that if an error corrupts the control flow execution, it will ultimately be detected. Our approach verifies the DSM technique and the comparisons are illustrated in Section 7.

Tomoyuki et al. [5] proposed a symbolic model checking method for verification of fault tolerance of systems. At first the fault tolerant system is specified in the form of a guarded-command program. A modeling language suited for describing guarded-command programs is defined, and then a translation method from the modeling language to the SMV language is proposed. By specifying the fault tolerance property as a CTL formula, the examples were verified to demonstrate the usefulness of the proposed method. The difference with our work is the target fault model. The faults in [5] mainly are crash faults and byzantine faults, while our study focuses on transient faults [19].

Arora and Gouda [16] first gave a formal definition of “fault tolerance” for a system, which consists of a safety requirement, closure and convergence. Then a formal framework for reasoning about fault-tolerant system was proposed, which enables reasoning independent of technology, architecture and application considerations. Similar to the work in [5], this method deals with the stuck-at, crash, failstop, omission and byzantine faults.

John [17] presented a methodology that can ease the formal specification and assurance of critical fault-tolerant system. The functional program is first transformed into an untimed synchronous system and then into its time-triggered implementation. The first step is specific to the algorithm concerned, but the second is generic and its correctness was proved. This proof was formalized and mechanically checked with the PVS verification system. This work is applicable to the critical real-time applications.

Daniel and Ruben [18] proposed a method of first using aspect oriented programming (AOP) to add fault tolerance to software, and then formally verifying the fault tolerance property of a program. Meng et al. [20] analyzed the impact on formal verification effort and testing effort due to adding different fault tolerance mechanisms to baseline systems. By comparing the experimental results of different designs, they concluded that re-execution (time redundancy) was the most efficient mechanism, followed by parity code, dual modular redundancy (DMR), and triple modular redundancy (TMR). Moreover, the ratio of verification effort to testing effort to assist designers in their trade-

off analysis when deciding how to allocate their budget between formal verification and testing was defined, which could be used in practical industrial production.

3. Signature-monitoring Mechanisms

In this paper, the target errors to be tolerated are transient errors [19], which emerge from external events such as energetic particles striking the chip due to the shrinking chip size and increasing transistor density of the modern processors. Transient faults do not cause permanent damage, but may result in incorrect program executions by altering signal transfers or stored values. In terms of effects, transient errors can be classified into data errors and CFEs. Data errors appear when the contents of variables in the memory or in the microprocessor registers are altered. CFEs refer to deviations from the program's normal instruction execution flow, such as upsets to the target addresses of branch instructions or the program counter (PC). Studies have shown that CFEs account for 33% to 77% of all transient errors [7]. Therefore, computer systems must be equipped with CFE detection mechanisms.

A multitude of CFE detection techniques have been proposed [8–11]. Signature-monitoring techniques are apparently the most universal and effective techniques. Several signature-monitoring techniques have been developed [9–11]. Control-flow checking by Software Signatures (CFCSS) [11] and DSM [9] are two representative signature-monitoring techniques, which will be verified in Section 6. As a general characteristic, signature-monitoring techniques are designed to detect illegal interblock transitions that are incorrect jumps to other blocks corrupting the program control flow. Signature-monitoring techniques are based on the partition of the program code into basic blocks (BB) [12]. A basic block is a maximal set of sequential instructions that start from the first instruction and terminate at the last instruction. No branch instruction exists in a basic block except for the last instruction.

After dividing the program into BBs, a unique static signature is associated to each BB during the pre-compilation phase. Some assistant signature variables are also introduced. Based on the signatures, checking instructions are added. When a program is executed, a dynamic signature variable is computed and compared with the static signature of the current executing block. If the dynamic signature equals to the static signature, no error occurs. Otherwise, the checking instructions see the mismatch and detect the error.

Checking instructions can be classified into three kinds in terms of their effects. Generation instructions produce the dynamic signature for the current block. Compare instructions identify whether the control has correctly reached this block. Preparing instructions update signatures for the transfer of control to the next block. For different signature-monitoring mechanisms, the locations and numbers of checking instructions are different. Fig. 1a shows an abstract description of how checking instructions are added. For some algorithms, preparing instructions may follow the original instructions. Whether

preparing instructions emerge before or after the original instructions, the branch instruction should come last.

Fig. 1b illustrates how checking instructions are added in CFCSS. G and S represent the dynamic signature and static signature, respectively. D and d represent the assistant signatures. Two instructions “xor G,G,d_i ” and “xor G,G,D ” [11] update the dynamic signature G . The compare instruction “bne $G, S_i, error$ ” compares the dynamic signature with the static signature. If they mismatch, program control flow transfers to the error handler “error”. The instruction “xor D,S_h,S_i ” generates the assistant signature D to prepare for the control flow transfer.

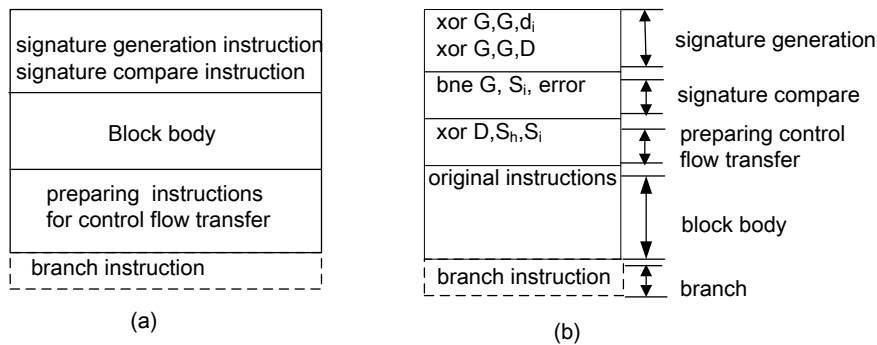


Fig. 1. (a) An abstract description of adding checking instruction in a basic block; (b) Example of checking instructions in CFCSS algorithm

4. The Fault Tolerant Model

A control flow machine is introduced to describe the control flow transfer of the fault tolerant program strengthened by signature-monitoring mechanism. Suppose the hardware affected by transient faults is based on a simple RISC architecture, the execution of the fault tolerant program strengthened by signature-monitoring mechanism can be specified using a step-operational semantics, which maps a machine state to other machine states.

4.1. The Syntax of the Machine

Considering that signature-monitoring mechanisms only focus on CFE, operations on the data segment (general-purpose registers and memory) can be abstracted from the machine. For clarity, we adopt a minimal assembly instruction set that is composed of control flow instructions jump (jmp) and conditional branch (brz), as well as signature-generation (generate), signature-compare (compare), and signature-preparing (prepare) instructions. Fig. 2 details the syntax of the machine states.

For clarity and elegance, assume that the values of signature and address are integers. Meta-variable n generally ranges over integers. When we emphasize that an integer is used as an address, the meta-variable l is used. Similarly, a meta-variable s is introduced to express the values of signature variables. rz denotes the branch condition of the conditional branch instructions, with the value “true” and “false”. rt denotes the target address of the control flow instructions.

Control flow instructions consist of jump instruction ($\text{jmp } rt$) and conditional branch instructions ($\text{brz } rz, rt$). Checking instructions consist of signature-generation (generate), signature-compare (compare), and signature-preparing (prepare) instructions. For different signature-monitoring techniques, the operations of checking instructions are different.

Instructions are grouped together in basic blocks BB . These blocks always begin with signature-generation and signature-compare instructions, are then filled with the block body and terminate by signature-preparing instructions. If a block has control flow transfer, control flow instruction ($\text{jmp } rt$) or ($\text{brz } rz, rt$) ends with the block. Since the original instructions of the block do not alter the program control flow, they can be abstracted as an entirety, in other words, the block body.

The machine states can be modeled as a tuple (C, H, BB, S, PC) that are composed of code memory C , history H , basic block being executed BB , signature variables S , and program counter PC . These are defined as follows:

- Code memory $C[l \rightarrow BB]$ is a partial map from addresses to valid basic blocks. Block addresses are all ordered. We use the notation $l+1$ to refer to the address of the block, which follows the block at l . If a block at l ends with a conditional branch, we assume that $l+1$ inhabit the domain of C . In other words, conditional branch always have a block to fall through to.
- History H is a sequence of labels that record basic blocks being executed during the current execution. When a program is completed executing, H denotes the execution path.
- Block BB denotes the basic block being executed.
- Signature variables S have two components, namely, the dynamic signature variable G and the assistant signature file A . $A[a \rightarrow s]$ is a mapping from assistant signature variables to the values they contain. Different signature-monitoring mechanisms introduce different assistant signature variables.
- Program Counter PC refers to the next instruction to be executed.

To describe the fault-tolerant program behavior after an error occurs, we introduce three special “final states”. The *detected* state represents a state in which an error has been detected, whereas the *invalid* state represents a state when an error causes transition to an invalid address. The *exit* state represents a state that program exits normally, though an error occurs.

Address values	l	::= n
Program Counter	PC	::= l
Branch condition	rz	::= true false
Branch intention register	rt	::= l
Signature values	s	::= n
Dynamic signature	G	::= s
Assistant signature variables	a	::= $a_1 \dots a_n$
Assistant signature file	A	::= . $A, a \rightarrow s$
Signature variables	S	::= (G, A)
History	H	::= l_1, \dots, l_n
Control flow instruction	i	::= jmp rt brz rz, rt
Checking instruction	c	::= generate G, a, s compare G, s prepare a, a, s
Blocks Body	b	::= original instruction list
Basic blocks	BB	::= generate G, a, s ; compare G, s ; b ; prepare a, a, s ; i .
Code memory	C	::= . $C, l \rightarrow BB$
State	Σ	::= (C, H, BB, S, PC)
Final state	Γ	::= detected exit invalid

Fig. 2. Syntax of instructions and machine states

4.2. The Semantics of the Machine

This section formalizes the operational semantics of the control flow machine. Generally speaking, the operational semantics are defined as the rule (1), which expresses that state Σ_1 transfers to state Σ_2 by a step execution under certain conditions.

$$\frac{\text{Conditions}}{\Sigma_1 \xrightarrow{\text{step}} \Sigma_2} \quad (1)$$

Rule (2) illustrates the fault model. As an established standard, the current study adhered to the Single Event Upset (SEU) model [7], which states that only one fault occurs per execution. With signature-monitoring mechanisms only focusing on CFE, SEU can be modeled as illegal transitions to other instructions that are different from the expected target. Under the semantics of the control flow machine, error occurrence can be described by the rule error, which expresses that when CFE occurs, the program control flow transfers to a random instruction rather than to the expected instruction (denoted as ex-

$pect(PC)$). Sometimes, a CFE may cause a program to jump to an invalid address. CFE may occur any time as the rule can apply any time.

$$\frac{PC' \neq (expect(PC)) \ \&\& \ BB' \neq BB}{(C, H, BB, S, PC) \xrightarrow{error} (C, (H; BB), BB', S, PC') / invalid(H)} \quad (2)$$

Several notations are introduced to explain the following rules. Notation $S_{val}(a)$ denotes the value of a . Similarly, $R_{val}(rt)$ refers to the value of rt . Notation $S[a \mapsto s]$ corresponds to updated assistant signature variables file with variable a mapped to s . $PC++$ denotes incrementing PC by 1, when the execution of an instruction is complete.

Rule (3) expresses the execution of a jump instruction. The control flow is transferred to a new block when rt contains the address of a valid block.

$$\frac{R_{val}(rt) \in Dom(C) \ \&\& \ BB' = C(R_{val}(rt))}{(C, H, BB, S, PC) \xrightarrow{jmp \ rt} (C, (H; BB), BB', S, R_{val}(rt))} \quad (3)$$

The rule (4), which is for the conditional branch instruction, similarly follows jump instructions when the branch condition is satisfied.

$$\frac{R_{val}(rt) \in Dom(C) \ \&\& \ BB' = C(R_{val}(rt)) \ \&\& \ R_{val}(rz) = false}{(C, H, BB, S, PC) \xrightarrow{brz \ rz, rt} (C, (H; BB), BB', S, R_{val}(rt))} \quad (4)$$

Accordingly, the rule (5) describes that the program control flow proceeds to the next block when the condition is not satisfied.

$$\frac{R_{val}(rt) \in Dom(C) \ \&\& \ R_{val}(rz) = true}{(C, H, BB, S, PC) \xrightarrow{brz \ rz, rt} (C, (H; BB), C(BB) + 1, S, PC++)} \quad (5)$$

The following rules describe state transitions by updating signature variables. The rule (6) indicates the generation of the dynamic signature of the current block, where the function $f = generate(s_1, s_2)$ defines the operations for signature generation.

$$\frac{G' = generate(S_{val}(a), s) \ \&\& \ S' = (G', A)}{(C, H, BB, S, PC) \xrightarrow{generate \ G, a, s} (C, H, BB, S', PC++)} \quad (6)$$

The rule (7) describes the mismatch that the dynamic signature is not equal to the static signature. Hence, an error is detected and the program transfers to a final $detected(H)$ state, where H represents the sequence of the blocks during execution.

$$\frac{G \neq s}{(C, H, BB, S, PC) \xrightarrow{compare \ G, s} detected(H)} \quad (7)$$

By contrast, the rule (8) describes when the dynamic signature equals the static signature of the current block, no error is detected and the program continues with its execution.

$$\frac{G = s}{(C, H, BB, S, PC) \xrightarrow{\text{compare } G, s} (C, H, BB, S, PC++)} \quad (8)$$

The rule (9) indicates that assistant signature variables are prepared for transfer control to the next block, where the function $f = \text{prepare}(s_1, s_2)$ defines the operations for updating the assistant signatures.

$$\frac{A' = \text{prepare}(a \mapsto S_{val}(a), s) \ \& \ S' = (G, A')}{(C, H, BB, S, PC) \xrightarrow{\text{prepare } a, a, s} (C, H, BB, S', PC++)} \quad (9)$$

5. Implementation

In this section, we first refine the execution of the control flow machine into a state transition system, and then describe its translation to the input language of the model checker for automatic verification.

5.1. The State Transition System

As aforementioned, a state of the control flow machine is formally represented by a tuple $\Sigma = (C, H, BB, S, PC)$. Transitions between these states formalize the effect of the execution of an instruction. The fault-tolerant program execution then can be refined into a state transition system with one assembly instruction per state.

To explain the component C in the transition system, all basic blocks are numbered and each one is uniquely identified by an address represented by a special symbol. The addresses of the instructions in the basic blocks are also represented by symbols. For instance, the address of the first basic block (BB_1) is “# BB_1 ”. The addresses of the instructions in BB_1 are identified as # $BB_1(\text{gen})$, # $BB_1(\text{comp})$, # $BB_1(\text{body})$, # $BB_1(\text{pre})$, and # $BB_1(\text{bran})$.

In the machine state, H represents the sequence of blocks involved during the current execution. When the execution is complete, H becomes the execution path. In an assembly program, if the branch conditions are identified, their execution paths are also identified. To express all execution paths, a tuple $Con = (c_1, c_2, \dots, c_N)$ is introduced, where c_i is a Boolean variable that denotes the branch condition of BB_i and N is the total number of basic blocks in the program. If BB_i is not a branch block that exits with a branch instruction, c_i is reserved as “false” and not modified by the model-checking tool during the verification.

The states of control flow machine can be simplified by a tuple $\Omega = (PC, S, Con, error)$, where Ω is the set of states of the state transition system. The state variables are defined as follows:

- *PC*: a Program Counter variable *PC* explicitly controls the sequencing of instructions. As aforementioned, the addresses of all basic blocks are represented by symbols. Therefore, the domain of *PC* can be represented by a set $\{\#BB_1(\text{gen}), \#BB_1(\text{comp}), \#BB_1(\text{body}), \#BB_1(\text{pre}), \#BB_1(\text{bran}), \dots, \#BB_N(\text{gen}), \#BB_N(\text{comp}), \#BB_N(\text{body}), \#BB_N(\text{pre}), \#BB_N(\text{bran}), \text{Normal_exit}, \text{Fault_detected}$ and $\text{Invalid_address}\}$. To describe the final states, three special symbols, namely, “Normal_exit”, “Fault_detected” and “Invalid_address” are introduced. In the absence of error, program normally exits and *PC* eventually transforms to “Normal_exit”. Conversely, if an error occurs and is caught by the signature-monitoring mechanism, *PC* transforms to “Fault_detected”. Moreover, if an error causes transition to an invalid address, *PC* transforms to “Invalid_address”.
- *S* is a set of signature variables $\{G, a_1, a_2, \dots, a_M\}$ that are introduced to represent the signature information, where *G* denotes the dynamic signature, a_i denotes the assistant signature variable and *M* denotes the number of introduced assistant signature variables. The values of the signature variables are updated upon executions of signature generation and preparing instructions. In the CFCSS algorithm, for example, when the generation instruction “xor *G*, *G*, d_i ” is executed, the dynamic signature *G* is updated as “ $G \oplus d_i$ ”. Similarly, when the preparing instruction “xor *D*, S_h, S_k ” is executed, the assistant signature *D* is updated as “ $S_h \oplus S_k$ ”. The symbol “ \oplus ” denotes the bitwise XOR operation. Different signature-monitoring mechanisms have different signature computations.
- *Con*: To express all execution paths, a tuple $Con=(c_1, c_2, \dots, c_N)$ is introduced. During verification, the values of (c_1, c_2, \dots, c_N) are comprehensively modified by the model-checking tool, thus allowing the exploration of all execution paths.
- *error* is introduced to specify when a CFE occurs. According to the fault model, an error can occur any time during the execution. To describe the occurrence of the error, a set of Boolean expressions $\{\text{error} == \#BB_1(\text{gen}), \text{error} == \#BB_1(\text{comp}), \text{error} == \#BB_1(\text{body}), \dots, \text{error} == \#BB_N(\text{body}), \text{error} == \#BB_N(\text{pre})$ and $\text{error} == \#BB_N(\text{bran})\}$ are introduced. Only one expression can be true in an execution and the identification of the true expression is random. For instance, if the value of “ $\text{error} == \#BB_i(\text{body})$ ” is true, it denotes that an illegal transition is transferred from the original instructions of BB_i . And the destination of the illegal transition is comprehensively modified by the model-checking tool, thus covering all fault scenarios.

The transition system can then be modeled as $\langle \Omega, A, \rightarrow, \theta \rangle$, where Ω is the set of states, *A* denotes the set of actions, \rightarrow denotes the transition relation, and θ is the set of initial states. A state is initial if and only if *PC* is equal to the first instruction of the entry block, and then an initial value is declared for every other state variable. For signature variables, their initial values are determined by the signature-monitoring mechanism. For variables in *Con* and *error*, their initial value is randomly selected in their pre-defined domain. By

modeling each instruction execution as an action, the transition is defined as $\Omega_1 \rightarrow \Omega_2$, where Ω_1 and Ω_2 are the current and next states.

Fig. 3 demonstrates an example how states transfer with the execution of the program. Fig. 3a illustrates the execution of a basic block, where the solid lines show the correct control flows and the dashed line represents the illegal transition caused by a CFE. The corresponding state transitions of Fig. 3a are shown in Fig. 3b. The circles denote the states, which are represented by a tuple that contains the assignment of current values to the state variables. By modeling each instruction execution as an action, the current state is transferred to the next state with some variables updating by the execution. If an error occurs, program control flow transfers to a random instruction that is different from the expected one. Correspondingly, the current state transfers to the next state that is not the expected one, such as the transition from Ω_3 to Ω_r that illustrates the occurrence of a CFE.

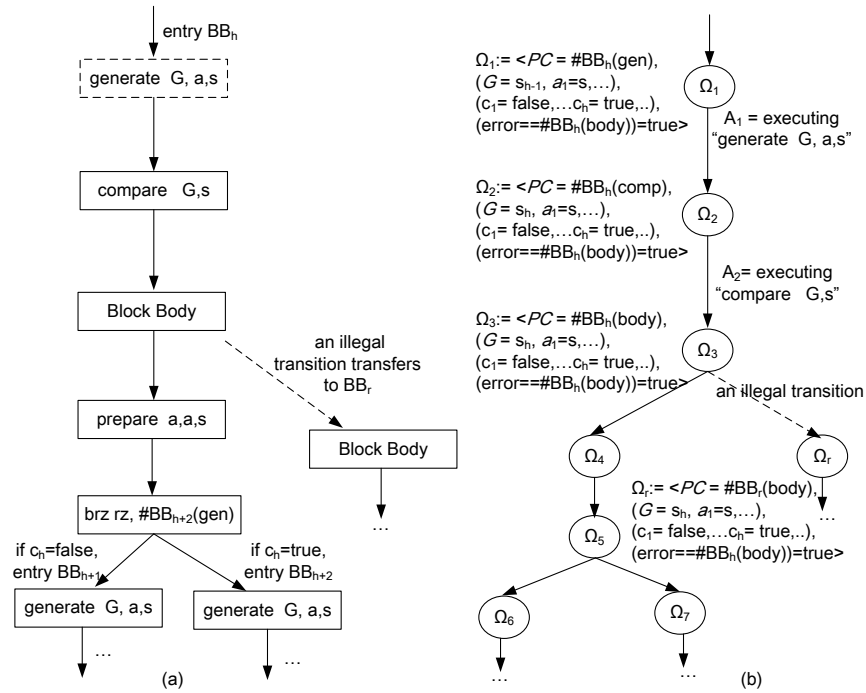


Fig. 3. Examples of state transitions

5.2. Translation to NuSMV

In the current study, the model-checking technique is adopted to verify the effectiveness of signature-monitoring mechanisms. The model checker NuSMV is selected for two reasons. First, the input language of NuSMV is syntactically and semantically similar to the general description of a finite state

transition system. Second, NuSMV has a powerful symbolic representation and an expressive Computation Tree Logic (CTL) that specify properties to be checked. To verify a finite system in NuSMV, it has to be described in the SMV input program. The translation of the state transition system into an SMV program is then described.

An SMV program is composed of variable declaration, state initialization, state transitions and a list of properties written in CTL formula. The complete syntax of the SMV language is described in the SMV documentation [6]. The steps for the translation are as follows:

Step1: Defining an SMV module

The state transition system can be described using a model. The SMV program must have a main MODULE, so the system can be represented by the main MODULE.

Step2: Defining each state variable

Each state variable in the transition system is directly mapped into an SMV variable. An SMV variable for each variable in the transition system is then declared and its domain is pre-defined. The keyword VAR is used to declare variables.

Step3: Defining the initial state

Each SMV variable is assigned to its initial value, using the INIT statement.

Step4: Defining transition relations

First, for each transition in the state transition system, the current state is defined with its enabling conditions in the DEFINE statement. For a transition from Ω_1 to Ω_2 , as shown in Fig. 4, its corresponding DEFINE statement is “ $t_i := (PC = \#BB_h(\text{gen})) \wedge (G = S_{h-1} \wedge a = s \wedge \dots) \wedge (c_i = \text{false} \wedge \dots \wedge c_h = \text{true} \dots) \wedge (\text{error} == \#BB_h(\text{body})) = \text{true} \dots$ ”.

Second, for each SMV variable v , a next statement is declared, which is expressed as follows:

```
next (v) := case
    t1 : v';
    t2 : v'';
    . . . . .
    TRUE : v;
esac;
```

Thus, if condition t_1 is true, v is updated as v' . Similarly, v is updated as v'' when t_2 is satisfied. Otherwise, no condition is true, v keeps the original value.

Step5: Defining the constraints for the error model

According to the SEU model, only one fault occurs during an execution. Hence, a constraint should be declared under the keyword INVAR. As aforementioned, a set of Boolean expressions $\{\text{error} == \#BB_1(\text{gen}), \dots, \text{error} == \#BB_N(\text{pre}), \text{error} == \#BB_N(\text{bran})\}$ is introduced to describe the error occurrence. By defining each expression as a DEFINE statement, such as “ $h_i = (\text{error} == \#BB_j(\text{body}))$ ”, the fault model, which indicates that only one error occurs per execution, can be expressed as the following INVAR statement:

$$h_1 + h_2 + h_3 + \dots + h_w = 1$$

where W represents the number of Boolean expressions.

Step 6: Defining the property to be verified

The basic principle of model checking is that the model under validation should satisfy a desired property. The property, which characterizes signature-monitoring mechanisms, states that if a CFE disrupts the control flow execution, the error is ultimately detected. This property can be described in a CTL formula.

$$AG (h_1 + h_2 + h_3 + \dots + h_W = 1 \rightarrow AF(PC=fault_detected))$$

The CTL formula shows that in all state sequences from the initial state, if a CFE occurs, the program control flow is eventually transferred to the final state "fault detected".

6. Case Studies

In this section, we apply the verification method to two comprehensive error detection algorithms: CFCSS and DSM.

6.1. Model Checking of CFCSS algorithm

CFCSS is a pure software method proposed by researchers in the Stanford University [11]. CFCSS has been widely used in safety-critical systems. In the CFCSS algorithm, each block is numbered and assigned a unique static signature. Two basic checking instructions, namely, "xor G, G, d" and "bne G, S, error" are inserted at the beginning of each basic block, where G denotes the dynamic signature and S denotes the static signature. For each block that has more than one predecessor, a checking instruction "xor G,G,D" is inserted after the instruction "xor G, G, d", where D denotes the assistant signature. Finally, for each block that has more than one successor, a checking instruction "xor D,S_n,S_i" is inserted after the instruction "bne G, S, error". Given the limited space, the details of the CFCSS algorithm are presented in [11].

According to the characteristic of checking instructions in CFCSS algorithm, basic blocks can be classified into the following kinds (seeing Fig. 4):

- one to one: a block has only one predecessor and each of its successors has only one predecessor.
- one to many: a block has only one predecessor and at least one of its successors has more than one predecessor.
- many to one: a block has more than one predecessor and each of its successors has only one predecessor.
- many to many: a block has more than one predecessor and at least one of its successors has more than one predecessor.

Let S_i be the static signature of basic block BB_i , and d_i be the signature difference which is calculated as $d_i = S_i \oplus S_j$ (BB_j is the successor of BB_i).

The added checking instructions in Fig. 4 are PowerPC instructions [15].

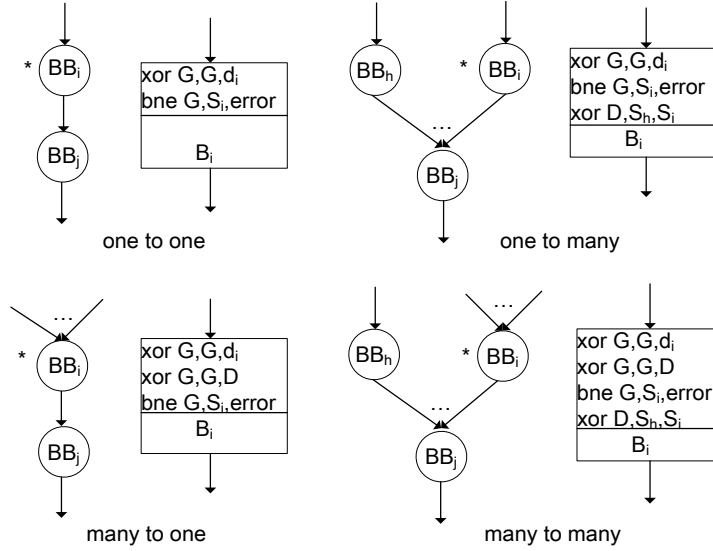


Fig. 4. The added checking instructions for different kinds of blocks

According to the verification, the results are shown in Table 1 and Table 2. They are obtained by statistically analyzing the counterexamples NuSMV reports. Before identifying the results, we introduce several notations. We use the notation $Suc(BB_i)$ to denote the set of all successors of BB_i . Correspondingly, we use the notation $Pred(BB_i)$ to denote the set of all predecessors of BB_i . We use the symbol “ IG_{i1} ” to denote the checking instruction “ $xor\ G, G, d_i$ ” of BB_i . Similarly, the symbols “ IG_{i2} ”, “ IC_i ”, “ IP_i ” are used to denote the checking instructions “ $xor\ G, G, D$ ”, “ $bne\ G, S_i, error$ ” and “ $xor\ D, S_h, S_i$ ”, respectively. And the symbol “ IO_i ” is introduced to represent the original instructions of BB_i .

A CFE can be modeling as a transition from the source instruction to the sink instruction. Table 1 lists the verification results for the CFEs between two blocks without any direct legal control flow transfers. In other words, BB_m is not in $Suc(BB_i)$ and BB_i is not in $Pred(BB_m)$. Table 2 lists the verification results for the CFEs between two blocks BB_i and BB_j , where $BB_j \in Suc(BB_i)$.

Table 1. Part I: the verification results of checking capacity for CFCSS .

source sink	one to one				one to many				many to one				many to many				
	IG_{i1}	IC_i	IO_i		IG_{i1}	IC_i	IP_i	IO_i	IG_{i1}	IG_{i2}	IC_i	IO_i	IG_{i1}	IG_{i2}	IC_i	IP_i	IO_i
IG_{m1}	◇	√	√		◇	◇	√	√	√	◇	◇	√	√	◇	◇	√	√
IG_{m2}	√	√	√		√	√	#	#	▽	▽	▽	▽	▽	▽	▽	#	#
IC_m	√	√	√		√	√	√	√	▽	√	√	√	▽	√	√	√	√
IP_m	√	√	√		√	√	√	√	▽	√	√	√	▽	√	√	√	√
IO_m	◇	√	√		◇	◇	#	#	√	◇	◇	√	*	◇	◇	#	#

Table 2. Part II: the verification results of checking capacity for CFCSS .

source sink	one to one			one to many				many to one				many to many				
	IG _{i1}	IC _i	IO _i	IG _{i1}	IC _i	IP _i	IO _i	IG _{i1}	IG _{i2}	IC _i	IO _i	IG _{i1}	IG _{i2}	IC _i	IP _i	IO _i
IG _{j1}	x	x	x	√	√	x	x	√	x	x	x	*	√	√	x	x
IG _{j2}	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
IC _j	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
IP _j	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
IO _j	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

The symbol “√” indicates that the illegal transitions from the source instruction to the sink instruction can be checked by CFCSS algorithm, while symbol “x” indicates that the illegal transitions cannot be detected. Other symbols describe illegal transitions that cannot be detected under certain conditions. The undetected errors and proofs are obtained by analyzing the counter-examples reported by the model checker NuSMV. They are as follows:

- “x”: When the dynamic signature G and the assistant signature D are updated at BB_i, control flow skips IO_i of BB_i and transfers to IG_{j1} of BB_j, where BB_j ∈ Suc(BB_i). These illegal transitions cannot be detected.
- “#”: Undetected errors are grouped into two cases: 1) when new G and new D are generated at BB_i, control flow transfers to IO_m of BB_m, where BB_m ∈ Pred(BB_j) and BB_j ∈ Suc(BB_i) and 2) when new G and new D are generated at BB_i, control flow transfers to IG_{m2} of BB_m, where BB_m ∈ Pred(BB_j) and BB_j ∈ Suc(BB_i).
Proof. Suppose an illegal branch br_{im} takes from IO_i of BB_i to IO_m of BB_m, where BB_m ∈ Pred(BB_j) and BB_j ∈ Suc(BB_i). At BB_i, G is equal to S_i and D is equal to S_i ⊕ S_m (picked arbitrarily). After br_{im} is taken, the original instructions of BB_m are executed, and then control flow transfers to its successor BB_j. The checking instructions of BB_j update G = G ⊕ d_j ⊕ D = S_i ⊕ (S_i ⊕ S_m) ⊕ (S_i ⊕ S_m) = S_j. With G equal to S_j, the checking instruction “bne G, S_j, error” does not detect the mismatch. Therefore, the error cannot be detected. Other cases labeled by “#” can be proved in the same way.
- “*”: An illegal transition taken from IG_{i1} of BB_i to IG_{j1} of its successor BB_j cannot be detected if and only if BB_i is a “many to many” block. Similarly, illegal transitions taken to IO_m of BB_m cannot be detected, where BB_m ∈ Pred(BB_j).
Proof. Suppose br_{ij} is an illegal branch, where BB_j ∈ Suc(BB_i), BB_h ∈ Pred(BB_i), and BB_k ∈ Pred(BB_j). At BB_i, G is equal to (S_h ⊕ d_i) and D is equal to (S_h ⊕ S_k). After br_{ij} is taken, the checking instructions of BB_j update G, where G = (S_h ⊕ d_i) ⊕ d_j ⊕ D = S_h ⊕ (S_i ⊕ S_k) ⊕ (S_i ⊕ S_j) ⊕ (S_h ⊕ S_k) = S_j. Therefore, this illegal branch is not detected. Illegal transitions taken from IG_{i1} to IO_m of BB_m can be proved similarly.
- “∇”: Undetected errors of this kind are caused by the design of signatures. Suppose that BB_h ∈ Pred(BB_i) and BB_k ∈ Pred(BB_j), BB_m is a random block. If equation (10) is satisfied, illegal transitions taken from IG_{i1}

of BB_i to IC_m of BB_m or from IG_{i2} of BB_i to IG_{m2} of BB_m cannot be detected, where BB_m has more than one predecessor.

$$S_m = S_h \oplus S_k \oplus S_i \quad (10)$$

Proof. With br_{im} as an illegal branch, as aforementioned, and the signature design satisfies equation(10), after the instruction “xor G,G,d_i” of BB_i is executed, br_{im} is taken, where G is equal to $(S_h \oplus d_i) = S_h \oplus (S_k \oplus S_i)$. With S_m equal to $S_h \oplus S_k \oplus S_i$, the instruction “bne G, S_m, error” of BB_m cannot detect the error. The detection process of illegal transition from “xor G,G,D” of BB_i to “xor G,G,D” of BB_m is similar.

- Undetected errors of this kind are also caused by the design of assigned signatures. If equation (11) is satisfied, illegal branches taken from IC_i , where the new G at BB_i is generated, to IG_{m1} of BB_m are not detected, where BB_m has more than one predecessor. Similarly, illegal transitions to the original instructions of the predecessor BB_m also cannot be detected. D is determined by the block that is executed prior to BB_i .

$$S_m = S_i \oplus d_m \oplus D \quad (11)$$

Proof. With br_{im} as an illegal branch, as aforementioned, the signature design satisfies equation (11). After the new G of BB_i is generated, br_{im} is taken, where G is equal to S_i . The checking instructions of BB_m update G, where $G=S_i \oplus d_m \oplus D=S_m$. Therefore, br_{im} cannot be detected.

Oh et al. [11] evaluated the effectiveness of CFCSS by fault injection experiment and their experimental results show that the CFCSS algorithm increases the error detection capability by an order of magnitude. However, the results do not exactly specify which faults do not attempt detection. As of this writing, the present study is a pioneer evaluation of the effectiveness of the algorithm and the identification of undetected errors.

6.2. Model Checking of DSM algorithm

DSM is a software-based signature analysis technique presented by TMA laboratory [9]. In DSM, each basic block is associated to an identification number (IDB) and each interblock transition is assigned as:

$$\text{signature} = \text{IDB}_{\text{source}} | \text{IDB}_{\text{destination}} \quad (12)$$

where operator “|” represents the concatenation function.

In addition to the transition signatures, some local cumulative signatures N_{i1}, N_{i2}, N_{i3} are introduced. These signatures are integer numbers that are unique for each basic block. The condition expressed by (13) is satisfied by the design.

$$N = N_{i1} + N_{i2} + N_{i3} = \dots = N_{N1} + N_{N2} + N_{N3} = 0 \quad (13)$$

The local cumulative signatures ensure that: (1) a source block transfers control to the first instruction of the destination block and (2) the signature-

checking instructions are correctly executed. Fig. 5 demonstrates how control flow is checked [9], where B denotes the dynamic transition signatures and R denotes the static transition signatures. The details of DSM algorithm refer to [9].

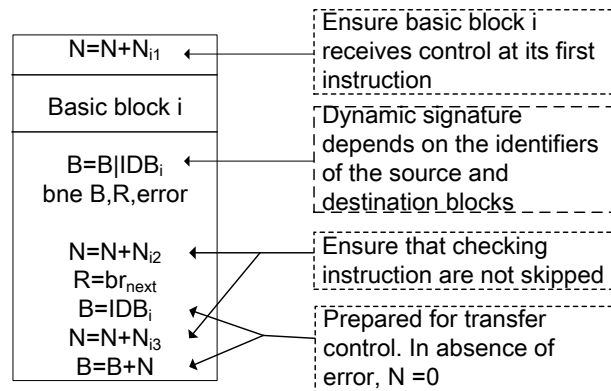


Fig. 5. The implementation of DSM algorithm

By applying the proposed method to validate DSM algorithm, the verification results are listed in Table 3. Notations “ IL_{i1} ”, “ IL_{i2} ” and “ IL_{i3} ” are introduced to represent the instructions “ $N=N+N_{i1}$ ”, “ $N=N+N_{i2}$ ” and “ $N=N+N_{i3}$ ”, respectively. “ IO_i ” is used to denote the original instructions of BB_i . “ IG_i ” denotes the signature generation instruction “ $B=B|IDB_i$ ”. “ IC_i ” denotes the compare instruction “ $bne B,R,error$ ”. “ IP_{i1} ”, “ IP_{i2} ” and “ IP_{i3} ” are used to denote the prepare instruction “ $R=br_{next}$ ”, “ $B=IDB_i$ ” and “ $B=B+N$ ”, respectively. In Table 3, there are no legal control flow transitions between BB_m and BB_i .

Table 3. the verification results of checking capacity for DSM.

source sink	IL_{i1}	IO_i	IG_i	IC_i	IL_{i2}	IP_{i1}	IP_{i2}	IL_{i3}	IP_{i3}
IL_{m1}	√	√	√	√	√	√	#	#	√
IO_m	√	√	√	√	√	√	√	√	√
IG_m	√	√	√	√	√	√	√	√	√
IC_m	√	√	*	*	*	√	√	√	√
IL_{m2}	√	√	√	√	◇	◇	◇	√	√
IP_{m1}	▽	▽	▽	▽	√	√	√	√	√
IP_{m2}	√	√	√	√	√	√	√	√	√
IL_{m3}	√	√	√	√	√	√	√	√	√
IP_{m3}	√	√	√	√	√	√	√	√	√

In [4] and [9], DSM algorithm is validated to have the ability to detect all the illegal interblock transitions. However, we found four kinds of errors that cannot be detected under certain conditions. These undetected errors are as follows [14]:

- “#”: Illegal branches that are transferred from one of the instructions between “ $B=IDB_i$ ” and “ $N=N+N_{i3}$ ” of BB_i , to the instruction “ $N=N+N_{n1}$ ” of BB_n cannot be detected, where $BB_i \in Pred(BB_n)$ and $Suc(BB_n) \in \emptyset$. These faults do not cause wrong outputs and can thus be negligible.
- “*”: Illegal branches taken from one of the instructions “ $B=B|IDB_i$; bne B, R, error” of BB_i to the instruction “bne B, R, error” of BB_n cannot be detected, where $Suc(BB_n) \in \emptyset$.
Proof. Assuming that br_{in} is an illegal branch and the branch is transferred from “ $B=B|IDB_i$ ” of BB_i and landed at “bne B, R, error” of BB_n , where $Suc(BB_n) \in \emptyset$, when br_{in} is taken, R is still equal to B because the new R is not generated at BB_i . Therefore, they match and br_{in} is not detected.
- “ ∇ ”: Undetected errors of this kind are caused by the design of local cumulative signatures. If equation (14) is satisfied, illegal branches are transferred from one of the original instructions of BB_i to the checking instruction “ $R=br_{next}$ ” of BB_m , which are cannot be detected. BB_i and BB_m are arbitrary basic blocks of the program.

$$N_{i1} + N_{m3} = 0 \quad (14)$$

Proof. If br_{im} is an illegal branch and it is taken from one of the original instructions of BB_i , skipped the rest of the checking instructions, and then landed at the instruction “ $R=br_{next}$ ” of BB_m . At BB_i , N is equal to N_{i1} . After br_{im} is taken, the new R is generated to the transition signature, depending on the identifiers of BB_m and its successor. B is updated to the identifier of BB_m and N is added by N_{m3} . If equation (14) is satisfied, N equals to zero. The control flow then transfers to the successor of BB_m . Considering that R and B are updated at BB_m , the execution of the program continues without detecting br_{im} .

- “ \diamond ”: Undetected errors of this kind are also caused by the design of local cumulative signatures. If equation (15) is satisfied, illegal branches that are transferred from the instructions “ $N=N+N_{i2}$ ” of BB_i to the checking instruction “ $N=N+N_{m2}$ ” of BB_m are not detected. BB_i and BB_m are arbitrary blocks of program.

$$N_{i1} + N_{i2} + N_{m2} + N_{m3} = 0 \quad (15)$$

Proof. If br_{im} is an illegal branch, as aforementioned, and the signature design satisfies equation (15). After the instruction “ $N=N+N_{i2}$ ” of BB_i is executed, br_{im} is taken, where N is equal to $(N_{i1} + N_{i2})$. At BB_m , the checking instructions generate R and B. N is updated as $N = N + N_{m2} + N_{m3} = (N_{i1} + N_{i2}) + N_{m2} + N_{m3} = 0$. Given that equation (15) is satisfied, no error is detected when the control flow is transferred to the successor of BB_m .

7. Comparisons

As illustrated in Section 2, some earlier works have proposed the use of model checking in evaluating system dependability. The work with more similarity to ours is described in [4], but ours is much more general and practical. The comparisons are as follows:

- We provide a general approach to evaluate all signature monitoring techniques, while the technique in [4] is just aimed at the DSM technique.
- In [4], the model checking computation time increases at a polynomial rate of the addition of instructions of the model, and the amount of memory needed is a potential bottleneck for large models. It is known as the state-space explosion problem. In this study, we offer a solution to solve this problem. The state variables consist of $(PC, S, Con, error)$, so the state space increases at a polynomial rate of the addition of the basic blocks for a given program. Therefore, the memory cost and computation time can be seen as a biquadrate function of N , where N denotes the number of basic blocks. Apparently, the number of basic blocks is much less than the number of instructions for a given program. So our approach is more efficient in optimizing the state space exploration. From the viewpoint of model checking, our approach avoids the state-space explosion problem.
- In contrast with the verification results in [4], our verification results can be used to design signatures as the signature variables are involved in the model. For instance, to dissatisfy the condition expressed by " $N_{i1} + N_{i2} + N_{m1} + N_{m2} = 0$ ", illegal transitions symbolized by " \diamond " in Table 3 can be detected by DSM algorithm.

8. Conclusions

In this paper, we presented a novel approach for evaluating the effectiveness of signature-monitoring techniques. We initially modeled the program that was strengthened by signature-monitoring algorithms as a control flow machine. The execution of the assembly program modeled by the control flow machine is specified using a step-operational semantics, which maps a machine state to other machine states. We then refined the control flow machine into a state transition system and proposed a translation procedure to translate the state transition system into the input program of the model checker NuSMV for automatic verification. Finally, we applied the approach to two reprehensive error detection algorithms, namely, CFCSS and DSM. The undetected errors were analyzed based on the counter-example reported by NuSMV. As of this writing, this is a pioneer evaluation of signature-monitoring techniques, where undetected errors are also uncovered for the first time.

In our future research, we shall improve the algorithm design according to the verification results. We shall also design a lexical analyzer which can translate the state transition system into SMV programs automatically.

References

1. Gnesi S., Lenzini G., Latella D., Abbaneo C., Amendola A., and Marmo P.: An Automatic SPIN Validation of a Safety Critical Railway Control System. In Proceedings of the International Conference on Dependable Systems and Networks, New York, NY, USA, 119-124. (2000)
2. Ramachandran P., Kudva P., Kellington N.: Statistical Fault Injection. In Proceedings of the 38th International Conference on Dependable Systems and Networks. Conference Publishing Services, Anchorage, USA, 122-127. (2008)
3. Kljaich J., Smith B. T., Wojcik A. S. : Formal Verification of Fault Tolerance Using Theorem-Proving Techniques. IEEE Transaction on Computers, Vol. 38, No. 3, 366-376. (1989)
4. Nicolescu B., Gorse N., Savaria Y.: On the Use of Model Checking for the Verification of a Dynamic Signature Monitoring Approach. IEEE Transactions on Nuclear Science, Vol. 52, No. 5, 1555-1561.(2005)
5. Tomoyuki Y., Tatsuhiro T., Tohru K.: Automatic verification of Fault Tolerance Using Model Checking. In Proceedings of the 8th Pacific Rim International Symposium on Dependable computing. Conference Publishing Services, Seoul, Korea 95-102. (2001)
6. Roberto C., Alessandro C., Charles A. J. , Gavin K., Emanuele O., Pistore M. , Roveri M., Andrei T.: NuSMV 2.4 User Manual. [Online]. Available: <http://nusmv.fbk.eu/NuSMV/userman/v24/nusmv.pdf> (current 2005)
7. Vemu, R., Gurusurthy, S., Abraham J. A.: ACCE: Automatic correction of control-flow errors. In Proceedings of 2007 International Conference on TEST. Conference Publishing Services, Santa Clara, USA, 1-10. (2007)
8. Goloubeva O., Reaudengo M., Sonza Reorda M., Violante M.: Soft-Error Detection Using Control Flow Assertions. In Proceedings of the 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems. Boston, USA, 581-588. (2003)
9. Nicolescu B., Savaria Y., Velazco R. : Software detection mechanisms providing full coverage against single bit-flip faults. IEEE Transaction on Nuclear Science, Vol. 51, No. 6, 3510-3518. (2004)
10. Borin E., Wang C., Wu Y.F., Araujo G. : Software-Based Transparent and Comprehensive Control-Flow Error Detection. In Proceedings of the 4th ACM/IEEE International Symposium on Code Generation and Optimization. New York, USA. (2006)
11. Oh N. , Shirvani P.P., McCluskey E.J.: Control-Flow Checking by Software Signatures. IEEE Transactions on Reliability, Vol. 51, No. 2, 111-122. (2002)
12. Aho A.,Sethi R., Ullman J.: Compilers: Principles, Techniques and Tools(2nd Edition). Addison-Wesley, Boston, USA.(2007)
13. Reinhardt S. K., Mukherjee S. S.: Transient fault detection via simultaneous multi-threading. In Proceedings of the 27th Annual International Symposium on Computer Architecture. ACM Press, Vancouver, Canada.(2000)
14. Tan L., Tan Q., Xu J., Li J.: A note on "On the Use of Model Checking for the Verification of a Dynamic Signature Monitoring Approach. IEEE Transaction on Nuclear Science, Vol. 58, No. 1, 359-359. (2011)
15. Programming Environments Manual for 32-Bit Implementations of the PowerPC™ Architecture. [Online]. Available: www.freescale.com (current 2005)
16. Arora A. and Gouda M.: Closure and Convergence: A Foundation of Fault-Tolerant Computing. IEEE Transactions on Software Engineering, Vol.19, No.11, 1015–1027. (1993)

17. John R.: Systematic Formal Verification for Fault-Tolerant Time-Triggered Algorithms. IEEE Transactions on Software Engineering, Vol.25, No.5, 651–660. (1999)
18. Daniel L. and Ruben A.: Formal Verification of Fault Tolerance Aspects. [Online]. Available: <http://www.software3.net/ff/formal-verification-of-fault-tolerance-aspects-w290> (current 2005)
19. Sexton F. W.: Destructive single-event effects in semiconductor devices and ICs. IEEE Transactions on Nuclear Science, Vol. 50, No. 3, 603–621. (2003)
20. Meng Z., Anita L. and Daniel J. S.: Analyzing Formal Verification and Testing Efforts of Different Fault Tolerance Mechanisms. In Proceedings of the 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Chicago, IL, USA, 277-285.(2009)

Lanfang Tan received the B. S. degree from the Department of Computer Science, National University of Defense Technology in 2008. She is currently pursuing the Ph.D. degree. Her research interests include software fault-tolerance, formal verification, model checking and dependable computing.

Qingping Tan received his M.S. and Ph.D. degrees in computer science from the National University of Defense Technology in 1988 and 1992, respectively. He is currently a professor and Ph. D. supervisor at National University of Defense Technology. He is also a visiting scholar at the University of Pisa in 2009. His research interests include software engineering, programming languages, compilers, software fault-tolerance and dependable computing.

Jianjun Xu received his M.S. and Ph.D. degrees in computer science from the National University of Defense Technology in 2006 and 2010, respectively. He is a full-time Lecturer in the Computer Science School at the National University of Defense Technology. His research is focused on program analysis, compiler for fault tolerance.

Huiping Zhou is an Associate Professor in the Computer Science School at the National University of Defense Technology. His research is focused on programming languages, compilers and software fault-tolerance.

Received: February 18, 2012; Accepted: November 26, 2012.

An Uncertain Optimal Control Model with n Jumps and Application

Liubao Deng and Yuanguo Zhu

Department of Applied Mathematics, Nanjing University of Science and Technology
Nanjing 210094, Jiangsu, China
dengliubao@163.com
ygzhu@njust.edu.cn

Abstract. Optimal control theory is an important branch of modern control theory which has been widely applied in various sciences. Uncertain optimal control is a theory dealing with optimal control problems which are based a new uncertainty theory and differs from the stochastic optimal control based on probability theory and fuzzy optimal control based on fuzzy set theory or credibility theory. As the further work of the uncertain optimal control with jump in the one-dimensional case and multidimensional linear-quadratic (LQ) uncertain optimal control problem with jump which has a quadratic objective function for a linear uncertain control system with jump, a general uncertain optimal control problem with n jumps in the multi-dimensional cases is considered in this paper. The principle of optimality is presented and the equation of optimality is obtained about multidimensional uncertain optimal control with n jumps. Finally, as an application, an optimal control problem in R&D (Research and Development) fiscal subsidy policy is discussed and the optimal control decisions are obtained.

Keywords: optimal control, uncertainty, jump, multidimensional, R&D fiscal subsidy policy.

1. Introduction

Optimal control theory is an important branch of modern control theory. It is to seek the optimal control decision among the admissible control strategies for maximizing or minimizing some objective functions, which relate to a dynamic process driven by a differential equation. With the more use of methods and results on mathematics and computer science, optimal control theory has made considerable advances, and has been used widely in real-life problems such as production engineering, national defence, programming, finance, and economic management.

There has been an enormously rich theory on deterministic optimal control problem. The Pontryagin's maximum principle, Bellman's dynamic programming and Kalman's optimal linear feedback regulator design theory are the main tools studying deterministic optimal control problems. When a control system is an uncertain dynamic system, we are facing an uncertain optimal control problem.

Randomness and fuzziness are two traditional uncertainties. By applying probability theory and the Zadeh's fuzzy set theory or the credibility theory [16], the stochastic optimal control problem and the fuzzy optimal control problem have been developed by many researchers. However, the complexity of the world makes the events we face uncertain in various forms. A lot of surveys show that some imprecise quantities, such as information and knowledge represented by human language, behave neither like randomness nor like fuzziness. In order to model these imprecise quantities, an uncertainty theory was founded by Liu [17] in 2007 and refined in 2011 [18]. Nowadays, uncertainty theory has become a new branch of axiomatic mathematics and is well developed on both theory and applications.

Based on uncertain canonical process in uncertainty theory, Zhu [33] first introduced and dealt with an uncertain optimal control problem without jump by using dynamic programming in 2010. It is reasonable model for uncertain systems with the continuous uncertain canonical process without jump. Nevertheless, in real world, some external extreme events or noises have a great influence on uncertain dynamic systems. In [5–7], Deng and Zhu studied uncertain optimal control problems with jump, including a general uncertain optimal control problem with one jump in one-dimensional case, a linear quadratic (LQ) uncertain optimal control problem with jump which has a quadratic objective function for a linear uncertain control system with one jump, and a multidimensional linear quadratic (LQ) uncertain optimal control model with n jumps. In this paper, we will extend the special multidimensional uncertain LQ model to a general one.

The subsequent sections of the paper are organized as follows. In the next section related work will be reviewed. In Section 3, some preliminaries will be given. In Section 4, a multidimensional uncertain control model with n jumps will be put forward. Then, the equation of optimality will be obtained for the proposed model. As an application of the equation, an optimal control problem in R&D (Research and Development) fiscal subsidy policy will be discussed. In Section 5, some concluding remarks will be made.

2. Related Work

Merton [21] studied stochastic optimal control for finance in the late of 1960s. In recent decades, the study of stochastic optimal control greatly attracted the attention of many researchers, and has been made considerable advances, especially for finance. For example, Fleming and Rishel [8], Harrison [10] and Karatzas [13] studied optimal control problems of Brownian motion or stochastic differential equations and applications in finance and engineering. Dixit and Pindyck [4] discussed the use of dynamic programming in optimization over Ito's process. Zhou and Li [29] introduced the stochastic LQ control as a general framework to analyze the continuous-time mean-variance portfolio optimization and hedging problem. Jensen [11] studied stochastic optimal stopping problem in risk theory. Cairns [3] and Boulier [1] established the optimal control

models of stochastic pension fund. Ou-Yang [22] and Sung [25] designed optimal contracts of dynamic delegated portfolio management by applying stochastic optimal control approach under Merton's continuous-time stochastic finance framework.

Komolov et al [14] presented the concept of optimal fuzzy control in 1979 based on the fuzzy set theory introduced by scientist on cybernetics Zadeh in 1965. Following that, the optimal fuzzy control problem has been studied in some work, see e.g., [9, 12, 23, 26] and references therein. In recent years, based on the credibility theory found by Liu [16] in 2004, which gives a self-dual measure for fuzzy events, a fuzzy optimal control problem with fuzzy Liu process was studied in some literature [20, 24, 27, 30–32].

Zhu [33] put forward an uncertain optimal control problem without jump based on the uncertainty theory in 2010. The equation of optimality was obtained and applied to solve a portfolio selection problem. The results showed the effectiveness of the proposed method. Then Kang and Zhu [15] and Xu and Zhu [28] discussed uncertain bang-bang optimal control problems for multi-stage and continuous time uncertain systems without jump in 2012, respectively. A general uncertain optimal control problem with one jump in one dimensional case was presented and dealt with by Deng and Zhu [5] in 2012. The principle of optimality was presented and the equation of optimality was obtained. Furthermore, in one-dimensional case, a special model, i.e., linear quadratic (LQ) uncertain optimal control model with one jump, was discussed by Deng and Zhu [6]. A necessary and sufficient condition for the existence of optimal control was derived. Deng [7] studied a linear quadratic (LQ) uncertain optimal control model with n jumps in multidimensional case. However, for many multidimensional uncertain optimal control problems, uncertain dynamic system with jumps may be nonlinear and the objective function may be not quadratic. Therefore, it is a meaningful work to extend the special multidimensional uncertain LQ model to general one. In this paper, we will focus on a general multidimensional uncertain optimal control problem with n jumps.

3. Preliminary

The concepts about uncertain measure, uncertain variable, uncertain process, *et al* may be referred to Liu [17], [18] and references therein.

Definition 1. (Liu [19]) An uncertain process C_t is said to be a canonical process if (i) $C_0 = 0$ and almost all sample paths are Lipschitz continuous; (ii) C_t has stationary and independent increments; (iii) every increment $C_{s+t} - C_s$ is a normally distributed uncertain variable with expected value 0 and variance t^2 , whose uncertainty distribution is

$$\Phi(x) = \left(1 + \exp \left(\frac{-\pi x}{\sqrt{3t}} \right) \right)^{-1}, \quad x \in \mathfrak{R}. \quad (1)$$

Theorem 1. (Zhu [33]) Let ξ be a normally distributed uncertain variable with expected value 0 and variance σ^2 , whose uncertainty distribution is

$$\Phi(x) = \left(1 + \exp\left(\frac{-\pi x}{\sqrt{3}\sigma}\right)\right)^{-1}, \quad x \in \mathfrak{R}. \quad (2)$$

Then for any real number a , we have $\frac{\sigma^2}{2} \leq E[a\xi + \xi^2] \leq \sigma^2$.

Theorem 2. (Liu [18]) Let f be a convex function on $[a, b]$, and ξ an uncertain variable that takes values in $[a, b]$ and has expected value e . Then

$$E[f(\xi)] \leq \frac{b-e}{b-a}f(a) + \frac{e-a}{b-a}f(b). \quad (3)$$

In order to model the discontinuous jump part of an uncertain system, an uncertain V -jump process, which is associated with an uncertain Z -jump variable $Z(r_1, r_2, t)$ defined by a jump uncertainty distribution was introduced by Deng and Zhu [5]. For the sake of simplicity, only one jump point is considered there. Now we generalize them to the following case of n jump points.

Definition 2. An uncertain variable $Z(r_{i1}, r_{i2}, n, t)$ is said to be an uncertain Z - n jumps variable with parameters r_{i1} and r_{i2} ($0 < r_{i1} < r_{i2} < r_{(i+1)1} < r_{(i+1)2} < r_{(n+1)1} = 1, i = 1, 2, \dots, n - 1$) for $t > 0$ if it has a jump uncertainty distribution

$$\Phi(x) = \begin{cases} 0, & \text{if } x < 0, \\ \frac{(n+1)r_{11}}{t}x, & \text{if } 0 \leq x < \frac{t}{n+1}, \\ r_{i2} + \frac{(n+1)(r_{(i+1)1} - r_{i2})}{t} \left(x - \frac{it}{n+1}\right), & \text{if } \frac{it}{n+1} \leq x < \frac{(i+1)t}{n+1}, i = 1, \dots, n, \\ 1, & \text{if } x \geq t. \end{cases} \quad (4)$$

Theorem 3. Assume ξ_1 and ξ_2 are independent uncertain Z - n jumps variables $Z(r_{i1}, r_{i2}, n, t_1)$ and $Z(r_{i1}, r_{i2}, n, t_2)$, respectively. Then the sum $\xi_1 + \xi_2$ is also an uncertain Z - n jumps variable $Z(r_{i1}, r_{i2}, n, t_1 + t_2)$, i.e.,

$$Z(r_{i1}, r_{i2}, n, t_1) + Z(r_{i1}, r_{i2}, n, t_2) = Z(r_{i1}, r_{i2}, n, t_1 + t_2). \quad (5)$$

The product of an uncertain Z - n jumps variable $Z(r_{i1}, r_{i2}, n, t)$ and a scalar number $\bar{k} > 0$ is also an uncertain Z - n jumps variable $Z(r_{i1}, r_{i2}, n, \bar{k}t)$, i.e.,

$$\bar{k} \cdot Z(r_{i1}, r_{i2}, n, t) = Z(r_{i1}, r_{i2}, n, \bar{k}t) \quad (6)$$

Definition 3. An uncertain process V_t is said to be an uncertain V - n jumps process with parameters r_{i1} and r_{i2} ($0 < r_{i1} < r_{i2} < r_{(i+1)1} < r_{(i+1)2} < r_{(n+1)1} = 1, i = 1, 2, \dots, n - 1$) for $t > 0$ if (i) $V_0 = 0$; (ii) V_t has stationary and independent increments; (iii) every increment $V_{s+t} - V_s$ is an uncertain Z - n jumps variable $Z(r_{i1}, r_{i2}, n, t)$.

Let V_t be an uncertain V - n jumps process, and $\Delta V_t = V_{t+\Delta t} - V_t$. Then by using definition of expected value of uncertain variable, we get

$$\begin{aligned} \tilde{k} &= E[\Delta V_t] = \int_0^{+\infty} (1 - \Phi(x)) dx = \int_0^{\frac{\Delta t}{n+1}} \left(1 - \frac{(n+1)r_{11}}{\Delta t} x \right) dx \\ &\quad + \sum_{i=1}^n \int_{\frac{i\Delta t}{n+1}}^{\frac{(i+1)\Delta t}{n+1}} \left(1 - r_{i2} - \frac{(n+1)(r_{(i+1)1} - r_{i2})}{\Delta t} \left(x - \frac{i\Delta t}{n+1} \right) \right) dx \\ &= \left(1 - \frac{1}{2(n+1)} \left(\sum_{i=1}^n (r_{i1} + r_{i2}) + 1 \right) \right) \Delta t. \end{aligned}$$

Theorem 4. (Existence Theorem) *There is an uncertain V - n jumps process.*

The proof of the theorem 3 and 4 are parallel to the corresponding results in Deng and Zhu [5].

Theorem 5. *Let V_t be an uncertain V - n jumps process and $\zeta = \Delta V_t$. Then for any real number a ,*

$$E[a\zeta + \zeta^2] = a\tilde{k}\Delta t + o(\Delta t).$$

Proof: To begin with we have

$$E[a\zeta + \zeta^2] \geq E[a\zeta] = aE[\zeta] = a\tilde{k}\Delta t. \tag{7}$$

On the other hand, we can get

$$E[a\zeta + \zeta^2] \leq \frac{E[\zeta]}{\Delta t} (a\Delta t + (\Delta t)^2) = a\tilde{k}\Delta t + o(\Delta t) \tag{8}$$

by Theorem 2. Combining inequality (7) and inequality (8), we can obtain

$$E[a\zeta + \zeta^2] = a\tilde{k}\Delta t + o(\Delta t).$$

Definition 4. (Deng and Zhu [5]) *Suppose that C_t is an uncertain canonical process, V_t is an uncertain V - n -jumps process, and g_1, g_2 and g_3 are some given functions. Then*

$$dX_t = g_1(X_t, t) dt + g_2(X_t, t) dC_t + g_3(X_t, t) dV_t \tag{9}$$

is called an uncertain differential equation with jump. A solution is an uncertain process X_t that satisfies (9) identically in t .

4. Multidimensional Uncertain Optimal Control Model with n Jumps and Application

4.1. Multidimensional Uncertain Optimal Control Model with n Jumps

An uncertain optimal control model with jump under one-dimensional case was proposed and the equation of optimality for solving the model was obtained by

Deng and Zhu [5]. In this subsection, we present the following general multidimensional uncertain optimal control model with n jumps:

$$\begin{cases} J(t, \mathbf{x}) \equiv \sup_{\mathbf{u}_s \in \mathbf{U}} E \left[\int_t^T L(s, \mathbf{X}_s, \mathbf{u}_s) ds + F(T, \mathbf{X}_T) \right], \\ \text{subject to} \\ d\mathbf{X}_s = \mathbf{A}(s, \mathbf{X}_s, \mathbf{u}_s) ds + \mathbf{B}(s, \mathbf{X}_s, \mathbf{u}_s) d\mathbf{C}_s + \mathbf{H}(s, \mathbf{X}_s, \mathbf{u}_s) d\mathbf{V}_s, \\ \mathbf{X}_t = \mathbf{x}, \end{cases}$$

where $\mathbf{X}_s = (X_{s1}, X_{s2}, \dots, X_{sn})^\tau$ is a n -dimensional state vector with the initial condition $\mathbf{X}_t = \mathbf{x} = (x_1, x_2, \dots, x_n)^\tau$ at time t , \mathbf{u}_s the decision vector of dimension r (represents the function $\mathbf{u}(s, \mathbf{X}_s)$ of time s and state \mathbf{X}_s) in a domain \mathbf{U} , $L : [0, T] \times \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}$ the objective function, and $F : [0, T] \times \mathbb{R}^n \rightarrow \mathbb{R}$ the terminal reward function. In addition, $\mathbf{A} : [0, T] \times \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^n$ is a column-vector function, $\mathbf{B} : [0, T] \times \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^n \times \mathbb{R}^k$ and $\mathbf{H} : [0, T] \times \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^n \times \mathbb{R}^k$ are two matrix functions, and $\mathbf{C}_s = (C_{s1}, C_{s2}, \dots, C_{sk})^\tau$, $\mathbf{V}_s = (V_{s1}, V_{s2}, \dots, V_{sk})^\tau$, where $C_{s1}, C_{s2}, \dots, C_{sk}$ are independent uncertain canonical processes, $V_{s1}, V_{s2}, \dots, V_{sk}$ are independent uncertain V - n jumps processes, and C_{si} and V_{sj} for any $i, j = 1, 2, \dots, k$ ($i \neq j$) are independent. Note that \mathbf{y}^τ represents the transpose vector of the vector \mathbf{y} , and the final time $T > 0$ is fixed or free.

Theorem 6 (Principle of optimality). For any $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}^n$, and $\Delta t > 0$ with $t + \Delta t < T$, we have

$$J(t, \mathbf{x}) = \sup_{\mathbf{u}_t \in \mathbf{U}} E [L(t, \mathbf{x}, \mathbf{u}_t) \Delta t + J(t + \Delta t, \mathbf{x} + \Delta \mathbf{X}_t) + o(\Delta t)], \quad (10)$$

where $\mathbf{x} + \Delta \mathbf{X}_t = \mathbf{X}_{t+\Delta t}$.

The proof of the theorem is parallel to the corresponding result in Deng and Zhu [5].

Theorem 7 (Equation of optimality). Let $J(t, \mathbf{x})$ be twice differentiable on $[0, T] \times \mathbb{R}^n$. Then we have

$$\begin{aligned} -J_t(t, \mathbf{x}) = \sup_{\mathbf{u}_t \in \mathbf{U}} \left\{ L(t, \mathbf{x}, \mathbf{u}_t) + \nabla_{\mathbf{x}} J(t, \mathbf{x})^\tau \mathbf{A}(t, \mathbf{x}, \mathbf{u}_t) \right. \\ \left. + \tilde{k} \nabla_{\mathbf{x}} J(t, \mathbf{x})^\tau \mathbf{H}(t, \mathbf{x}, \mathbf{u}_t) \mathbf{1} \right\} \end{aligned} \quad (11)$$

where $\tilde{k} = 1 - \frac{1}{2(n+1)} (\sum_{i=1}^n (r_{i1} + r_{i2}) + 1)$, $J_t(t, \mathbf{x})$ is the partial derivative of the function $J(t, \mathbf{x})$ in t , $\nabla_{\mathbf{x}} J(t, \mathbf{x})$ is the gradient of $J(t, \mathbf{x})$ in \mathbf{x} and $\mathbf{1}$ is a k -dimension column-vector with the all terms being 1.

The proof of theorem 7 is showed in the Appendix.

4.2. Optimal Control Problem in R&D Fiscal Subsidy Policy

The technology improvement is increasingly playing an important role in modern economic growth and economic development. As the source of the technology improvement, R&D (Research and Development) exert a direct or indirect effect on economic growth. The R&D investment related to new technology or new knowledge is very important and necessary. However, only market power in such field makes not enough incentive because for the risk contained in the process of R&D. So the government's R&D fiscal subsidy policy is one of the effective means to guide the market corresponding input.

The government departments usually cannot directly take part in R&D activities with fiscal subsidy policy. The relationship between government and R&D departments is a principle-agent relationship in which the officer is the principle and the other (representative consumer) is the agent. The problem of moral hazard may arise because the principle is unable to directly monitor R&D subsidy program implementation, and agent can change the use of the grant of R&D investment from principle in real world.

In this subsection, we employ the result obtained in the above subsection to solve the principle-agent problem in R&D fiscal subsidy policy with more than one uncertain factor. We present the dynamic optimization model of the principle and the agents with considering the possible moral hazard under the framework of uncertain environment, and seek their optimal paths.

We consider the principle-agent problem with a principle (government) and n agents (different industries' representative consumers).

Agents' Optimization Problem Let \mathbf{K} denote the n -dimensional agents' capital stock column-vector at time t with the initial condition $\mathbf{K}_0 = \mathbf{k}_0$ (state vector), \mathbf{c} and \mathbf{w} denote the n -dimensional consumption level vector selected by n agents at time t and the fraction vector of the capital stock allocated by n agents in R&D activities (control vector), respectively. \mathbf{C}_t and \mathbf{V}_t denote, respectively, independent canonical process vector and uncertain V - n jumps process vector. \mathbf{r}_y , \mathbf{r}_c , Θ_1 , Θ_2 , \mathbf{s} and \mathbf{a} are all diagonal matrixes with the diagonal terms being n representative consumers' capital tax rate, the consumption tax rate collected by the government, the constant diffusion coefficient of volatility, the constant jump coefficient of volatility, the input/output ratio in daily production activities and the average rate of return in R&D activities, respectively. \mathbf{I} is the unit matrix. In addition, \mathbf{W} and $\tilde{\mathbf{K}}$ are also the diagonal matrixes with the diagonal terms vector being \mathbf{w} and \mathbf{K} , respectively.

Thus n representative consumers' dynamic budget constraint equation may be described by a multidimensional uncertain differential equation with n jumps as follows

$$d\mathbf{K} = [(\mathbf{I} - \mathbf{r}_y)(\mathbf{I} - \mathbf{W})\mathbf{s}\mathbf{K} + (\mathbf{I} - \mathbf{r}_y)\mathbf{W}\mathbf{a}\mathbf{K} - (\mathbf{I} + \mathbf{r}_c)\mathbf{c}] dt + (\mathbf{I} - \mathbf{r}_y)\mathbf{W}\Theta_1\tilde{\mathbf{K}}d\mathbf{C}_t + (\mathbf{I} - \mathbf{r}_y)\mathbf{W}\Theta_2\tilde{\mathbf{K}}d\mathbf{V}_t. \quad (12)$$

Assume that the agents are interested in maximizing their discount value of expected total utilities over an infinite time horizon. Then n representative consumer's optimization problem may be described as follows

$$\max_{\mathbf{c}, \mathbf{w}} E \left[\int_0^\infty e^{-\rho t} U(\mathbf{c}, \mathbf{G}) dt \right], \quad (13)$$

where ρ and $U(\cdot)$ denote the consumer's subjective discount rate and instantaneous utility function at time t , respectively. \mathbf{G} denotes the government's unproductive public input vector.

Further assume that agents' instantaneous utility function is the specific form below [2]

$$U(\mathbf{c}, \mathbf{G}) = \frac{1}{b} [(\mathbf{W}\mathbf{c})^b]^\tau \mathbf{G}^{1-b}, \quad (14)$$

where coefficient b satisfies $0 < b < 1$.

Unless in particular stated otherwise, we assume $\mathbf{A}^s = (a_{ij}^s)_{m \times n}$ for any $\mathbf{A} = (a_{ij})_{m \times n}$. $(\mathbf{A}\mathbf{B})^s = \mathbf{A}^s \mathbf{B}^s$. Then the agents' optimal control model is provided by

$$\begin{cases} J(0, \mathbf{k}_0) \equiv \max_{\mathbf{c}, \mathbf{w}} E \left[\int_0^\infty \frac{1}{b} e^{-\rho t} [(\mathbf{W}\mathbf{c})^b]^\tau \mathbf{G}^{1-b} \right], \\ \text{subject to} \\ d\mathbf{K} = [(\mathbf{I} - \mathbf{r}_y)(\mathbf{I} - \mathbf{W})\mathbf{s}\mathbf{K} + (\mathbf{I} - \mathbf{r}_y)\mathbf{W}\mathbf{a}\mathbf{K} - (\mathbf{I} + \mathbf{r}_c)\mathbf{c}] dt \\ \quad + (\mathbf{I} - \mathbf{r}_y)\mathbf{W}\Theta_1 \tilde{\mathbf{K}} d\mathbf{C}_t + (\mathbf{I} - \mathbf{r}_y)\mathbf{W}\Theta_2 \tilde{\mathbf{K}} d\mathbf{V}_t, \quad \mathbf{K}_0 = \mathbf{k}_0. \end{cases} \quad (15)$$

By the equation of optimality (11), we have

$$\begin{aligned} -J_t(t, \mathbf{k}) = \max_{\mathbf{c}, \mathbf{w}} \left\{ \frac{1}{b} e^{-\rho t} [(\mathbf{W}\mathbf{c})^b]^\tau \mathbf{G}^{1-b} + \nabla_{\mathbf{k}} J(t, \mathbf{k})^\tau [(\mathbf{I} - \mathbf{r}_y)(\mathbf{I} - \mathbf{W})\mathbf{s}\mathbf{k} \right. \\ \left. + (\mathbf{I} - \mathbf{r}_y)\mathbf{W}\mathbf{a}\mathbf{k} - (\mathbf{I} + \mathbf{r}_c)\mathbf{c} + \tilde{k}(\mathbf{I} - \mathbf{r}_y)\mathbf{W}\Theta_2 \mathbf{k}] \right\} = \max_{\mathbf{c}, \mathbf{w}} L(\mathbf{c}, \mathbf{w}), \quad (16) \end{aligned}$$

where $L(\mathbf{c}, \mathbf{w})$ represents the term in the braces. The optimal (\mathbf{c}, \mathbf{w}) satisfies

$$\frac{\partial L(\mathbf{c}, \mathbf{w})}{\partial \mathbf{c}} = e^{-\rho t} [(\mathbf{W}\mathbf{c})^{b-1}]^\tau \mathbf{W}\mathbf{G}^{1-b} - \nabla_{\mathbf{k}} J(t, \mathbf{k})^\tau (\mathbf{I} + \mathbf{r}_c) = 0, \quad (17)$$

$$\begin{aligned} \frac{\partial L(\mathbf{c}, \mathbf{w})}{\partial \mathbf{w}} = e^{-\rho t} [(\mathbf{W}\mathbf{c})^{b-1}]^\tau \mathbf{C}\mathbf{G}^{1-b} \\ + \nabla_{\mathbf{k}} J(t, \mathbf{k})^\tau (\mathbf{I} - \mathbf{r}_y) (\mathbf{a} + \tilde{k}\Theta_2 - \mathbf{s}) \mathbf{k} = 0. \end{aligned} \quad (18)$$

Solving the equation (17) and (18), respectively, we get

$$(\mathbf{W}\mathbf{c})^b = e^{\rho t} \mathbf{C} \left(\tilde{\mathbf{G}}^{1-b} \right)^{-1} (\mathbf{I} + \mathbf{r}_c) \nabla_{\mathbf{k}} J(t, \mathbf{k}), \quad (19)$$

$$(\mathbf{W}\mathbf{c})^b = e^{\rho t} \mathbf{W} \left(\tilde{\mathbf{G}}^{1-b} \right)^{-1} \tilde{\mathbf{k}} (\mathbf{I} - \mathbf{r}_y) \left(\mathbf{s} - \mathbf{a} - \tilde{k} \Theta_2 \right) \nabla_{\mathbf{k}} J(t, \mathbf{k}), \quad (20)$$

where \mathbf{C} , $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{k}}$ are the diagonal matrixes with the diagonal terms vector being \mathbf{c} , \mathbf{G} and \mathbf{k} , respectively.

By the equation (19) and (20), we have

$$\mathbf{C} (\mathbf{I} + \mathbf{r}_c) = \mathbf{W} \tilde{\mathbf{k}} (\mathbf{I} - \mathbf{r}_y) \left(\mathbf{s} - \mathbf{a} - \tilde{k} \Theta_2 \right).$$

Hence

$$\mathbf{c}^* = \tilde{\mathbf{k}} (\mathbf{I} - \mathbf{r}_y) \left(\mathbf{s} - \mathbf{a} - \tilde{k} \Theta_2 \right) (\mathbf{I} + \mathbf{r}_c)^{-1} \mathbf{w}^*. \quad (21)$$

We conjecture that $J(t, \mathbf{k}) = e^{-\rho t} \mathbf{A} \tilde{\mathbf{k}}^b \mathbf{G}^{1-b}$, where \mathbf{A} is a row-vector. Then

$$J_t(t, \mathbf{k}) = -\rho e^{-\rho t} \mathbf{A} \tilde{\mathbf{k}}^b \mathbf{G}^{1-b}, \quad (22)$$

$$\nabla_{\mathbf{k}} J(t, \mathbf{k}) = b e^{-\rho t} \mathbf{A} \mathbf{G}^{1-b} \mathbf{k}^{b-1}. \quad (23)$$

Substituting Eq. (21), (22) and (23) into Eq. (16) yields

$$\begin{aligned} \rho \mathbf{A} \tilde{\mathbf{k}}^b \mathbf{G}^{1-b} &= (\mathbf{k}^b)^\tau \mathbf{A} (\mathbf{I} - \mathbf{r}_y) \left(\mathbf{s} - \mathbf{a} - \tilde{k} \Theta_2 \right) \mathbf{w}^* \mathbf{G}^{1-b} \\ &+ b (\mathbf{k}^{b-1})^\tau \mathbf{A} \mathbf{G}^{1-b} \left[(\mathbf{I} - \mathbf{r}_y) \mathbf{s} - 2 (\mathbf{I} - \mathbf{r}_y) \left(\mathbf{s} - \mathbf{a} - \tilde{k} \Theta_2 \right) \mathbf{W}^* \right] \mathbf{k}. \end{aligned} \quad (24)$$

Solving Eq. (24), we may get

$$\mathbf{w}^* = \frac{1}{1-2b} \left(\mathbf{s} - \mathbf{a} - \tilde{k} \Theta_2 \right)^{-1} \left[\rho (\mathbf{I} - \mathbf{r}_y)^{-1} - b \mathbf{s} \right] \mathbf{1}. \quad (25)$$

Substituting Eq. (25) into Eq. (21), we have

$$\mathbf{c}^* = \frac{1}{1-2b} (\mathbf{I} + \mathbf{r}_c)^{-1} [\rho \mathbf{I} - b \mathbf{s} (\mathbf{I} - \mathbf{r}_y)] \mathbf{k}. \quad (26)$$

Thus agents' optimal fraction of capital stock allocated in R&D and optimal consumption path are obtained.

Substituting Eq. (23) into Eq. (19) yields

$$(\mathbf{W}\mathbf{c})^b = b \mathbf{A} (\mathbf{I} + \mathbf{r}_c) \mathbf{c} \mathbf{k}^{b-1}. \quad (27)$$

Substituting Eq. (25) and (26) into Eq. (27), we can derive

$$\begin{aligned} \mathbf{A} &= \frac{1}{b(1-2b)^{2b-1}} \mathbf{1}^\tau \left(\mathbf{s} - \mathbf{a} - \tilde{k} \Theta_2 \right)^{-b} (\mathbf{I} - \mathbf{r}_y)^{b-1} \\ &\left[\rho (\mathbf{I} - \mathbf{r}_y)^{-1} - b \mathbf{s} \right]^{2b-1} (\mathbf{I} + \mathbf{r}_c)^{-b}. \end{aligned} \quad (28)$$

Then agent' optimal value may be obtained.

Principle's Optimization Problem Next, we consider the optimization problem of principle (government).

The principle's dynamic budget constraint equation may be described by a multidimensional uncertain differential equation with n jumps as follows

$$d\mathbf{K} = [(\mathbf{I} - \mathbf{W})\mathbf{s}\mathbf{K} + \mathbf{a}\mathbf{W}\mathbf{K} - \mathbf{c} - \mathbf{G}]dt + \Theta_1\mathbf{W}\tilde{\mathbf{K}}d\mathbf{C}_t + \Theta_2\mathbf{W}\tilde{\mathbf{K}}d\mathbf{V}_t. \quad (29)$$

Let instantaneous utility function of principle be also the form by (14). And assume that the principle is interested in maximizing the expected total utilities over an infinite time horizon. Then the principle's optimization problem may be described as follows

$$\begin{cases} J(0, \mathbf{K}_0) \equiv \max_{\mathbf{G}} E \left[\int_0^\infty \frac{1}{b} e^{-\rho t} [(\mathbf{W}\mathbf{c})^b]^\tau \mathbf{G}^{1-b} \right], \\ \text{subject to} \\ d\mathbf{K} = [(\mathbf{I} - \mathbf{W})\mathbf{s}\mathbf{K} + \mathbf{a}\mathbf{W}\mathbf{K} - \mathbf{c} - \mathbf{G}]dt \\ \quad + \Theta_1\mathbf{W}\tilde{\mathbf{K}}d\mathbf{C}_t + \Theta_2\mathbf{W}\tilde{\mathbf{K}}d\mathbf{V}_t, \quad \mathbf{K}_0 = \mathbf{k}_0. \end{cases} \quad (30)$$

By the equation of optimality (11), we have

$$\begin{aligned} -J_t(t, \mathbf{k}) = \max_{\mathbf{G}} \left\{ \frac{1}{b} e^{-\rho t} [(\mathbf{W}\mathbf{c})^b]^\tau \mathbf{G}^{1-b} + \nabla_{\mathbf{k}} J(t, \mathbf{k})^\tau [(\mathbf{I} - \mathbf{W})\mathbf{s}\mathbf{k} \right. \\ \left. + \mathbf{a}\mathbf{W}\mathbf{k} - \mathbf{c} - \mathbf{G} + \tilde{\mathbf{k}}\Theta_2\mathbf{W}\mathbf{k}] \right\} = \max_{\mathbf{G}} L(\mathbf{G}), \end{aligned} \quad (31)$$

where $L(\mathbf{G})$ represents the term in the braces. The optimal \mathbf{G} satisfies

$$\frac{\partial L(\mathbf{G})}{\partial \mathbf{G}} = \frac{1-b}{b} e^{-\rho t} [(\mathbf{W}\mathbf{c})^b]^\tau \mathbf{G}^{-b} - \nabla_{\mathbf{k}} J(t, \mathbf{k})^\tau \mathbf{1} = 0,$$

or

$$[(\mathbf{W}\mathbf{c})^b]^\tau \mathbf{G}^{-b} = \frac{be^{\rho t}}{1-b} \nabla_{\mathbf{k}} J(t, \mathbf{k})^\tau \mathbf{1}.$$

So

$$\mathbf{G}^{-b} = \frac{be^{\rho t}}{1-b} \mathbf{W}^{-b} \tilde{\mathbf{c}}^{-b} \nabla_{\mathbf{k}} J(t, \mathbf{k}). \quad (32)$$

We conjecture that $J(t, \mathbf{k}) = e^{-\rho t} \mathbf{B} \tilde{\mathbf{c}}^b \mathbf{k}^{1-b}$, where \mathbf{B} is a row-vector. Then

$$J_t(t, \mathbf{k}) = -\rho e^{-\rho t} \mathbf{B} \tilde{\mathbf{c}}^b \mathbf{k}^{1-b}, \quad (33)$$

$$\nabla_{\mathbf{k}} J(t, \mathbf{k}) = (1-b)e^{-\rho t} \mathbf{B} \tilde{\mathbf{c}}^b \mathbf{k}^{-b}. \quad (34)$$

Substituting Eq. (34) into Eq. (32), we have

$$\mathbf{G} = b^{-\frac{1}{b}} \mathbf{B}^{-\frac{1}{b}} \mathbf{w}\mathbf{k}. \quad (35)$$

Substituting Eq. (33), (34) and (35) into Eq. (31) yields

$$\begin{aligned} \rho \mathbf{B} \tilde{\mathbf{c}}^b \mathbf{k}^{1-b} &= b^{-\frac{1}{b}} \mathbf{B}^{1-\frac{1}{b}} \mathbf{c}^b (\mathbf{k}^{1-b})^\tau \mathbf{w} + (1-b) \mathbf{B} \mathbf{c}^b (\mathbf{k}^{-b})^\tau [(\mathbf{I} - \mathbf{W}) \mathbf{s} \mathbf{k} \\ &+ \mathbf{a} \mathbf{W} \mathbf{k} - \mathbf{c} - b^{-\frac{1}{b}} \mathbf{B}^{-\frac{1}{b}} \mathbf{w} \mathbf{k} + \tilde{k} \Theta_2 \mathbf{W} \mathbf{k}]. \end{aligned} \quad (36)$$

Solving Eq. (36), we derive

$$\mathbf{B}^{-\frac{1}{b}} = b^{\frac{1}{b}-1} \left\{ \rho \mathbf{1}^\tau \mathbf{W}^{-1} - (1-b) \mathbf{1}^\tau \left[(\mathbf{s} - \mathbf{C} \tilde{\mathbf{k}}^{-1}) \mathbf{W}^{-1} - (\mathbf{s} - \mathbf{a} - \tilde{k} \Theta_2) \right] \right\} \quad (37)$$

where \mathbf{W} and \mathbf{C} satisfies (25) and (26), respectively.

Substituting Eq. (37) and (25) into Eq. (35), we can get government' optimal R&D fiscal subsidy \mathbf{G}^* . At the same time, government' optimal value can be obtained.

5. Conclusion

Based on the one-dimensional uncertain optimal control with one jump, in this paper, a multidimensional uncertain optimal control problem with n jumps is studied. The equation of optimality is presented to solve the problem. Finally, an optimal control model in R&D fiscal subsidy policy is solved by the equation of optimality to show the usefulness of the equation. The difference between our work and the previous works is that in the previous work, only one-dimensional model with one jump or a special multidimensional LQ model with n jumps was considered, and in this paper, a general multidimensional uncertain optimal control model with n jumps is established. In our previous work and this work, we adapted expected value operator to measure the objective rewards in uncertain optimal control problems with jump. We may consider to do so by employing optimistic value or pessimistic value operator in future work.

Appendix: Proof of Theorem 7

Proof: By Taylor formula, we get

$$\begin{aligned} J(t + \Delta t, \mathbf{x} + \Delta \mathbf{X}_t) &= J(t, \mathbf{x}) + J_t(t, \mathbf{x}) \Delta t + \nabla_{\mathbf{x}} J(t, \mathbf{x})^\tau \Delta \mathbf{X}_t + \frac{1}{2} J_{tt}(t, \mathbf{x}) \Delta t^2 \\ &+ \frac{1}{2} \Delta \mathbf{X}_t^\tau \nabla_{\mathbf{xx}} J(t, \mathbf{x}) \Delta \mathbf{X}_t + \nabla_{\mathbf{x}} J_t(t, \mathbf{x})^\tau \Delta \mathbf{X}_t \Delta t + o(\Delta t) \end{aligned} \quad (38)$$

where $\nabla_{\mathbf{xx}} J(t, \mathbf{x})$ is the Hessian matrix of $J(t, \mathbf{x})$. Since

$$\Delta \mathbf{X}_t = \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t + \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{C}_t + \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{V}_t,$$

the expansion of (38) may be rewritten as

$$J(t + \Delta t, \mathbf{x} + \Delta \mathbf{X}_t)$$

$$\begin{aligned}
 &= J(t, \mathbf{x}) + J_t(t, \mathbf{x})\Delta t + \nabla_{\mathbf{x}}J(t, \mathbf{x})^T \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t \\
 &\quad + \nabla_{\mathbf{x}}J(t, \mathbf{x})^T \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{C}_t + \nabla_{\mathbf{x}}J(t, \mathbf{x})^T \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{V}_t \\
 &\quad + \frac{1}{2} \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t^2 \\
 &\quad + \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{C}_t \Delta t \\
 &\quad + \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{V}_t \Delta t \\
 &\quad + \frac{1}{2} (\mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{C}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{C}_t \\
 &\quad + (\mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{C}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{V}_t \\
 &\quad + \frac{1}{2} (\mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{V}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{V}_t \\
 &\quad + \nabla_{\mathbf{x}}J_t(t, \mathbf{x})^T \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t^2 + \nabla_{\mathbf{x}}J_t(t, \mathbf{x})^T \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{C}_t \Delta t \\
 &\quad + \nabla_{\mathbf{x}}J_t(t, \mathbf{x})^T \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{V}_t \Delta t + o(\Delta t) \\
 &= J(t, \mathbf{x}) + J_t(t, \mathbf{x})\Delta t + \nabla_{\mathbf{x}}J(t, \mathbf{x})^T \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t \\
 &\quad + \{ \nabla_{\mathbf{x}}J(t, \mathbf{x})^T \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) + \nabla_{\mathbf{x}}J_t(t, \mathbf{x})^T \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t \\
 &\quad + \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t \} \Delta \mathbf{C}_t \\
 &\quad + \{ \nabla_{\mathbf{x}}J(t, \mathbf{x})^T \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) + \nabla_{\mathbf{x}}J_t(t, \mathbf{x})^T \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t \\
 &\quad + \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t \} \Delta \mathbf{V}_t \\
 &\quad + \frac{1}{2} (\mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{C}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{C}_t \\
 &\quad + (\mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{C}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{V}_t \\
 &\quad + \frac{1}{2} (\mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{V}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta \mathbf{V}_t + o(\Delta t). \quad (39)
 \end{aligned}$$

Denote

$$\begin{aligned}
 \mathbf{m} &= \nabla_{\mathbf{x}}J(t, \mathbf{x})^T \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) + \nabla_{\mathbf{x}}J_t(t, \mathbf{x})^T \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t \\
 &\quad + \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t, \\
 \mathbf{n} &= \nabla_{\mathbf{x}}J(t, \mathbf{x})^T \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) + \nabla_{\mathbf{x}}J_t(t, \mathbf{x})^T \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t \\
 &\quad + \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t, \\
 \mathbf{P} &= \frac{1}{2} \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t), \\
 \mathbf{Q} &= \mathbf{B}(t, \mathbf{X}_t, \mathbf{u}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t), \\
 \mathbf{R} &= \frac{1}{2} \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t)^T \nabla_{\mathbf{xx}}J(t, \mathbf{x}) \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t).
 \end{aligned}$$

Therefore the equation (39) may be simply expressed as

$$\begin{aligned}
 &J(t + \Delta t, \mathbf{x} + \Delta \mathbf{X}_t) \\
 &= J(t, \mathbf{x}) + J_t(t, \mathbf{x})\Delta t + \nabla_{\mathbf{x}}J(t, \mathbf{x})^T \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t + \mathbf{m} \Delta \mathbf{C}_t \\
 &\quad + \mathbf{n} \Delta \mathbf{V}_t + \Delta \mathbf{C}_t^T \mathbf{P} \Delta \mathbf{C}_t + \Delta \mathbf{C}_t^T \mathbf{Q} \Delta \mathbf{V}_t + \Delta \mathbf{V}_t^T \mathbf{R} \Delta \mathbf{V}_t + o(\Delta t). \quad (40)
 \end{aligned}$$

Substituting equation (40) into equation (10) yields

$$0 = \sup_{\mathbf{u}_t \in \mathbf{U}} \{L(\mathbf{x}, \mathbf{u}_t, t)\Delta t + J_t(t, \mathbf{x})\Delta t + \nabla_{\mathbf{x}} J(t, \mathbf{x})^T \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t + E[\mathbf{m}\Delta C_t + \mathbf{n}\Delta V_t + \Delta C_t^T \mathbf{P}\Delta C_t + \Delta C_t^T \mathbf{Q}\Delta V_t + \Delta V_t^T \mathbf{R}\Delta V_t] + o(\Delta t)\}. \quad (41)$$

Let $\mathbf{m} = (m_i)_{1 \times k}$, $\mathbf{n} = (n_i)_{1 \times k}$, $\mathbf{P} = (p_{ij})_{k \times k}$, $\mathbf{Q} = (q_{ij})_{k \times k}$ and $\mathbf{R} = (r_{ij})_{k \times k}$. Then we have

$$\begin{aligned} & \mathbf{m}\Delta C_t + \mathbf{n}\Delta V_t + \Delta C_t^T \mathbf{P}\Delta C_t + \Delta C_t^T \mathbf{Q}\Delta V_t + \Delta V_t^T \mathbf{R}\Delta V_t \\ &= \sum_{i=1}^k (m_i \Delta C_{ti} + n_i \Delta V_{ti}) + \sum_{i=1}^k \sum_{j=1}^k (p_{ij} \Delta C_{ti} \Delta C_{tj} + q_{ij} \Delta C_{ti} \Delta V_{tj} + r_{ij} \Delta V_{ti} \Delta V_{tj}). \end{aligned}$$

Since $|p_{ij} \Delta C_{ti} \Delta C_{tj}| \leq \frac{1}{2} |p_{ij}| (\Delta C_{ti}^2 + \Delta C_{tj}^2)$, $|q_{ij} \Delta C_{ti} \Delta V_{tj}| \leq \frac{1}{2} |q_{ij}| (\Delta C_{ti}^2 + \Delta V_{tj}^2)$, $|r_{ij} \Delta V_{ti} \Delta V_{tj}| \leq \frac{1}{2} |r_{ij}| (\Delta V_{ti}^2 + \Delta V_{tj}^2)$, we have

$$\begin{aligned} & \sum_{i=1}^k \left[m_i \Delta C_{ti} + n_i \Delta V_{ti} - \sum_{j=1}^k \left\{ \left(|p_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta C_{ti}^2 + \left(|r_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta V_{ti}^2 \right\} \right] \\ & \leq \mathbf{m}\Delta C_t + \mathbf{n}\Delta V_t + \Delta C_t^T \mathbf{P}\Delta C_t + \Delta C_t^T \mathbf{Q}\Delta V_t + \Delta V_t^T \mathbf{R}\Delta V_t \\ & \leq \sum_{i=1}^k \left[m_i \Delta C_{ti} + n_i \Delta V_{ti} + \sum_{j=1}^k \left\{ \left(|p_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta C_{ti}^2 + \left(|r_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta V_{ti}^2 \right\} \right]. \end{aligned}$$

It follows from the independence of C_{ti} and V_{tj} ($i, j = 1, 2, \dots, k$) that

$$\begin{aligned} & \sum_{i=1}^k E \left[m_i \Delta C_{ti} + n_i \Delta V_{ti} - \sum_{j=1}^k \left\{ \left(|p_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta C_{ti}^2 + \left(|r_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta V_{ti}^2 \right\} \right] \\ & \leq E[\mathbf{m}\Delta C_t + \mathbf{n}\Delta V_t + \Delta C_t^T \mathbf{P}\Delta C_t + \Delta C_t^T \mathbf{Q}\Delta V_t + \Delta V_t^T \mathbf{R}\Delta V_t] \\ & \leq \sum_{i=1}^k E \left[m_i \Delta C_{ti} + n_i \Delta V_{ti} + \sum_{j=1}^k \left\{ \left(|p_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta C_{ti}^2 + \left(|r_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta V_{ti}^2 \right\} \right]. \end{aligned}$$

Theorem 1 implies that

$$E \left[m_i \Delta C_{ti} - \sum_{j=1}^k \left(|p_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta C_{ti}^2 \right] = o(\Delta t),$$

$$E \left[m_i \Delta C_{ti} + \sum_{j=1}^k \left(|p_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta C_{ti}^2 \right] = o(\Delta t).$$

Theorem 5 implies that

$$E \left[n_i \Delta V_{ti} - \sum_{j=1}^k \left(|r_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta V_{ti}^2 \right] = n_i \tilde{k} \Delta t + o(\Delta t),$$

$$E \left[n_i \Delta V_{ti} + \sum_{j=1}^k \left(|r_{ij}| + \frac{|q_{ij}|}{2} \right) \Delta V_{ti}^2 \right] = n_i \tilde{k} \Delta t + o(\Delta t).$$

Therefore

$$E [\mathbf{m} \Delta \mathbf{C}_t + \mathbf{n} \Delta \mathbf{V}_t + \Delta \mathbf{C}_t^\tau \mathbf{P} \Delta \mathbf{C}_t + \Delta \mathbf{C}_t^\tau \mathbf{Q} \Delta \mathbf{V}_t + \Delta \mathbf{V}_t^\tau \mathbf{R} \Delta \mathbf{V}_t]$$

$$= \sum_{i=1}^k n_i \tilde{k} \Delta t + o(\Delta t).$$

Obviously, we have

$$\sum_{i=1}^k n_i \tilde{k} \Delta t = \mathbf{n} \mathbf{1} \tilde{k} \Delta t = \nabla_{\mathbf{x}} J(t, \mathbf{x})^\tau \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \mathbf{1} \tilde{k} \Delta t + o(\Delta t).$$

Hence

$$E [\mathbf{m} \Delta \mathbf{C}_t + \mathbf{n} \Delta \mathbf{V}_t + \Delta \mathbf{C}_t^\tau \mathbf{P} \Delta \mathbf{C}_t + \Delta \mathbf{C}_t^\tau \mathbf{Q} \Delta \mathbf{V}_t + \Delta \mathbf{V}_t^\tau \mathbf{R} \Delta \mathbf{V}_t]$$

$$= \nabla_{\mathbf{x}} J(t, \mathbf{x})^\tau \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \mathbf{1} \tilde{k} \Delta t + o(\Delta t). \tag{42}$$

Substituting equation (42) into equation (41) yields

$$0 = \sup_{\mathbf{u}_t \in \mathbf{U}} \left\{ L(\mathbf{x}, \mathbf{u}_t, t) \Delta t + J_t(t, \mathbf{x}) \Delta t + \nabla_{\mathbf{x}} J(t, \mathbf{x})^\tau \mathbf{A}(t, \mathbf{X}_t, \mathbf{u}_t) \Delta t \right.$$

$$\left. + \nabla_{\mathbf{x}} J(t, \mathbf{x})^\tau \mathbf{H}(t, \mathbf{X}_t, \mathbf{u}_t) \mathbf{1} \tilde{k} \Delta t + o(\Delta t) \right\}. \tag{43}$$

Dividing the equation (43) by Δt , and letting $\Delta t \rightarrow 0$, we can obtain the result (11). The theorem is proved.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No.61273009) and the Natural Science Foundation of Anhui Province of China (1208085QG131).

References

1. Boulier, J-F., Trussant, E., Florens, D.: A dynamic model for pension funds management. Proceedings of the 5th AFIR International Colloquium, No.1, 361-384. (1995)

2. Bakshi, G. S., Chen, Z.: The Spirit of Capitalism and Stock-market Prices. *The American Economic Review*, Vol.3, 133-157. (1996)
3. Cairns, A.: Some Notes on the Dynamics and Optimal Control of Stochastic Pension Fund Models in Continuous Time. *ASTIN Bulletin*, Vol.30, No.1, 19-55. (2000)
4. Dixit, A. K., Pindyck, R. S.: *Investment under Uncertainty*. Princeton University Press, Princeton. (1994)
5. Deng, L., Zhu, Y.: Uncertain Optimal Control with Jump. *ICIC Express Letters, Part B: Applications*, Vol.3, No.2, 419-424. (2012)
6. Deng, L., Zhu, Y.: Uncertain Optimal Control of Linear Quadratic Models with Jump. *Mathematical and Computer Modelling*, doi:10.1016/j.mcm.2012.07.003 (2012)
7. Deng, L.: Multidimensional Uncertain Optimal Control of Linear Quadratic Models with Jump, *Journal of Computational Information Systems*, Vol.8, No.18, 7441-7448. (2012)
8. Fleming, W. H., Rishel, R. W.: *Deterministic and Stochastic Optimal Control*. Springer-Verlag, New York. (1986)
9. Filev, D., Angelov, P.: Fuzzy Optimal Control. *Fuzzy Sets and Systems*, Vol.47, No.2, 151-156. (1992)
10. Harrison, J. M.: *Brownian Motion and Stochastic Flow Systems*. John Wiley & Sons, New York. (1985)
11. Jensen, U.: An Optimal Stopping Problem in Risk Theory. *Insurance: Mathematics and Economics*, Vol.22, No.2, 177-178. (1998)
12. Jia, L. M., Zhang, X. D.: The Fuzzy Multiobjective Optimal Control Problem. *Engineering Applications of Artificial Intelligence*, Vol.6, No.2, 153-164. (1993)
13. Karatzas, I.: Optimization Problems in the Theory of Continuous Trading. *SIAM Journal on Control and Optimization*, Vol.27, No.6, 1221-1259. (1989)
14. Komolov, S. V., Makeev, S. P., Serov, G. P., Shaknov, I. F.: Optimal Control of a Finite Automaton with Fuzzy Constraints and a Fuzzy Target. *Cybernetics and Systems Analysis*, Vol.15, No.6, 805-810. (1979)
15. Kang, Y., Zhu, Y.: Bang-bang optimal control for multi-stage uncertain systems, *Information: An International Interdisciplinary Journal*, Vol.15, No.8, 3229-3238. (2012)
16. Liu, B.: *Uncertainty Theory: An Introduction to its Axiomatic Foundations*. Springer-Verlag, Berlin. (2004)
17. Liu, B.: *Uncertainty Theory*. 2nd ed., Springer-Verlag, Berlin. (2007)
18. Liu, B.: *Uncertainty Theory: A Branch of Mathematics for Modeling Human Uncertainty*. Springer-Verlag, Berlin. (2011)
19. Liu, B.: Some Research Problems in Uncertainty Theory. *Journal of Uncertain Systems*, Vol.3, No.1, 3-10. (2009)
20. Liu, J., Zhu, Y.: Multi-dimensional Multistage Fuzzy Optimal Control. 8th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). Shanghai, China, 189-192. (2011)
21. Merton, R. C.: Optimal Consumption and Portfolio Rules in a Continuous Time Model. *Journal of Economic Theory*, Vol.42, No.3, 373-413. (1971)
22. Ou-Yang, H.: Optimal contracts in a continuous-time delegated portfolio management problem. *The Review of Financial Studies*, Vol.16, No.1, 173C208. (2003)
23. Ostermark, R.: A Fuzzy Control Model for Dynamic Portfolio Management. *Fuzzy Sets and Systems*, Vol.78, No.3, 243-254. (1996)
24. Qin, Z., Bai, M., Ralescu, D.: A Fuzzy Control System with Application to Production Planning Problems. *Information Sciences*, Vol.181, No.5, 1018-1027. (2011)

25. Sung, J.: Optimal Contracts Under Adverse Selection and Moral Hazard: A Continuous-Time Approach. *The Review of Financial Studies*, Vol.18, No3, 1021-1073. (2005)
26. Willaelys, D.: Optimal Control of Fuzzy Systems. In *proc. Int. Congress on Applied Systems Research and Cybern*, Vol.16, No.6, 805-810. (1979)
27. Wang, B., Zhu, Y.: Multidimensional Fuzzy Optimal Control with Application to Optimal Stopping Time. *Proceedings of the 8th International Conference on Information and Management Sciences*. Kunming, China, 560-566. (2009)
28. Xu, X., Zhu, Y.: Uncertain bang-bang control for continuous time model, *Cybernetics and Systems: An International Journal*, Vol.43, No.6, 515-527. (2012)
29. Zhou, X., Li, D.: Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics and Optimization*, Vol.42, No.1, 19-33. (2000)
30. Zhu, Y.: A Fuzzy Optimal Control Model. *Journal of Uncertain Systems*, Vol.3, No.4, 270-279. (2009)
31. Zhao, Y., Zhu, Y.: Fuzzy Optimal Control of Linear Quadratic Models. *Computers and Mathematics with Applications*, Vol.60, No.1, 67-73. (2010)
32. Zhu, Y.: Fuzzy Optimal Control for Multi-stage Fuzzy Systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol.41, No.4, 964-975. (2011)
33. Zhu, Y.: Uncertain Optimal Control with Application to a Portfolio Selection Model. *Cybernetics and Systems: An International Journal*, Vol.41, No.7, 535-547. (2010)

Liubao Deng received the B.S degree in Mathematics from Anhui Normal University, China, in 1985 and M.S degree in Management from University of Electronic and Science and Technology of China, in 2004. He is pursuing the Ph.D. degree in Systems Engineering in Nanjing University of Science and Technology. He is an Associate Professor with School of Statistics and Applied Mathematics, Anhui University of Finance and Economics, China. His current research interests include optimal control, uncertainty and financial mathematics.

Yuanguo Zhu received the B.S and M.S degrees in Mathematics from Jiangxi Normal University, China, in 1984 and 1988, respectively, and the Ph.D. degree in Mathematics from Tsinghua University, China, in 2004. He is a Professor with the Department of Applied Mathematics, Nanjing University of Science and Technology, China. His major research interests include optimal control, uncertainty, optimization, and intelligent algorithm. His major projects include fuzzy optimal control, uncertain optimal control and application to portfolio selection.

Received: February 25, 2012; Accepted: November 25, 2012.

A Formal Approach to Testing Programs in Practice

Shaoying Liu¹, Wuwei Shen², and Shin Nakajima³

¹ Department of Computer Science, Hosei University, Japan
sliu@hosei.ac.jp

² Department of Computer Science, Western Michigan University, USA
wshen@wmich.edu

³ NII, Japan
nkjm@nii.ac.jp

Abstract. A program required to be tested in practice often has no available source code for some reason and how to adequately test such a program is still an open problem. In this paper, we describe a formal specification-based testing approach to tackle this challenge. The principal idea is first to formalize the informal requirements into formal operation specifications that take the interface scenarios of the program into account, and then utilize the specifications for test case generation and test result analysis. An example and case study of applying the approach to an IC card system is presented to illustrate its usage and analyze its performance.

Keywords: Specification-based testing, Formal specification, Black-box testing

1. Introduction

Programs required to be tested in practice often have no source code for some reasons (e.g., confidentiality) but testing such programs is necessary and important for software quality assurance in industry. A traditional approach to dealing with this problem is black-box testing (or functional testing) [1], but how to carry out an adequate black-box testing in this situation still remains a challenge.

The central issue in tackling this problem is how to make the tester thoroughly understand the desired functionality of the program under testing. One way is to discover it from the requirements that are usually written informally, but due to the ambiguities of the requirements, this may not be easily achieved. Another possibility is to use some test cases to run the program to detect the input-output relations. Each input-output relation can be reflected by a program interface trace, which we call an *interface scenario*. Each interface scenario indicates the implementation of a functional scenario (a sequence of operations), but it does not provide a complete definition of the functional scenario due to the fact that the test cases do not cover all necessary input values for the implementation of the functional scenario.

In this paper, we describe an approach to using formalization to help the tester understand the desired functionality of the program and build a firm foundation for testing the program. Specifically, our approach suggests that the requirements be formalized into a set of formal operation specifications whose signatures are constructed based on the program interface scenarios. The formalization will force the tester to draw an accurate, complete, and precise picture about the desired behaviors of the program and to prepare a solid foundation for testing. To make our approach work effectively, the formal specification must be ensured to be internally consistent and valid with respect to the informal requirements. This task can be fulfilled by means of formal specification inspection [2] and/or specification animation [3]. Since this issue is beyond the scope of this paper, we omit further discussions. The reader can refer to the related work for details of the inspection and animation techniques. In this paper, we focus on the descriptions of how the interface scenarios of a program are derived, how the formal operation specifications are constructed based upon the program interface scenarios, and how testing is carried out based upon the formal specifications.

The remainder of the paper is organized as follows. Section 2 describes the essential ideas of the testing approach and the process of testing. Section 3 discusses how to formalize the informal requirements based upon the program interface scenarios. Section 4 focuses on the test case generation and test result analysis. Section 5 presents an example to illustrate how our approach is applied to test an IC Card software. Section 6 introduces the related work and compares it with our approach. Finally, in Section 7 we conclude the paper and point out future research directions.

2. Essential Ideas

The essential ideas of our approach is first to formalize the requirements and then to carry out a testing. When formalizing the requirements, we abstract the program under testing into a set of operations, each defining an independent service, such as *Withdraw* or *Deposit* in an automated telling machine (ATM) software. The goal of the formalization is therefore to write a formal specification for each operation. To this end, we need to determine the signature of each operation, including the input variables, output variables, and the related external variables (or state variables). This signature is expected to be consistent with the interface of the program in order to ensure that test cases generated from the specification can be directly adopted for testing. Since there might be a gap between the informal requirements and the interface of the program, it is usually difficult to achieve the consistency without examining the program. However, since the source code of the program is assumed not to be available, it is impossible to examine the implementation details. The only information about the program we can grasp is its interface scenarios and its dynamic behaviors when it is executed. For this reason, our approach suggests that the formation of each operation's signature takes the *interface scenarios* of the program into

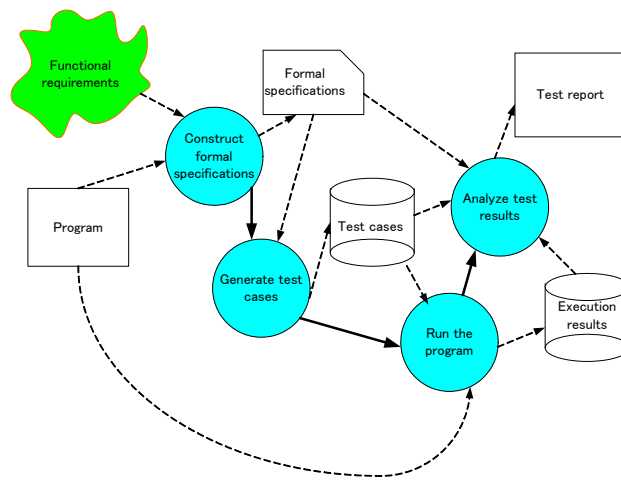


Fig. 1. The testing process using our approach

account. As far as defining the functionality of the operation is concerned, we use pre- and post-conditions, the most commonly applied notation in the literature [4], to specify it, based on the informal requirements, as shown in an example later. One may argue that writing such a formal specification may not be a routine exercise in practice and therefore our proposed approach depending on formal specifications may not be practical enough. While this might be true for the current industry where formal methods are rarely used and software crisis continues, the future software development would be significantly improved when some of the activities, such as testing, are undertaken, with effective tool support, by well trained practitioners or consultants in formal methods.

An interface scenario of the program is represented by a sequence of inputs and the corresponding output of the program. It can be identified by executing the program using sample inputs, but the precise relation between the sequence of inputs and the output is unknown. The goal of testing is to identify whether such a relation is implemented correctly in the program with respect to its definition in the specification. Assume that we have achieved formal specifications for the program under testing, what we need to do next is to generate test cases based upon the specifications to test the program. The whole process of testing using our approach is illustrated in Figure 1.

3. Construction of Formal Specifications

The construction of a formal specification for the program under testing takes the following three steps:

A software system for a simplified Automated Teller Machine (ATM) needs to provide two services for customers. ***The services include withdrawing money from the customer's bank account and inquiry for the account balance. To withdraw money, the customer's card and password as well as the requested amount are required. The bank policy does not allow overdraft when the service for withdrawal is used.*** For an inquiry about the customer's account balance, the system should provide the current balance of the customer's account.

Fig. 2. A description of informal requirements for the simplified ATM.

1. Find important functional scenarios described in the informal requirements, where each functional scenario describes an independent, desired function in terms of the input-output relation.
2. Form an operation to define all of the identified functional scenarios, but the signature of the operation must be formed based on both the requirements and the interface scenarios of the corresponding program.
3. Write the pre- and post-conditions for the operation using a formal specification language.

Next, we discuss these three issues, respectively.

3.1. Finding functional scenarios

The first step is to find the functional scenarios important to the end user from the requirements description. This can be done by reviewing and analyzing the informal requirements specification. We define each identified functional scenario as a single operation.

Let us consider a simplified ATM system as an example. Suppose the informal requirements for the system are given in Figure 2 and we want to find all of the important functional scenarios. We analyze the document to identify the relevant requirements and highlight the related text segments using the bold italic font, as indicated in the figure. Through the analysis, we realize that this functional scenario is concerned with withdrawing money from a bank account. Therefore, we form an operation and name it *Withdraw*. Applying the same principle, all of other functional scenarios can be identified and defined as appropriate operations.

3.2. Identifying interface scenarios

The next step is to determine the signature of the operation. As discussed previously, this signature needs to be kept consistent with that of the program in order to facilitate the direct adoption of test cases to be generated from the specification. To this end, we need to identify the corresponding interface scenarios when the program is executed. For this purpose, we can select test cases from

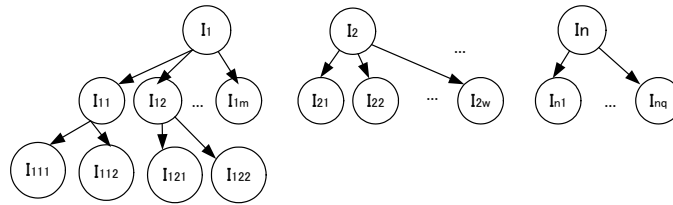


Fig. 3. Illustration of an interface tree

relevant types randomly. During the execution of the program, we continue to supply test cases if they are required by the program (e.g., via its GUI). Such an activity continues until the final execution result is provided. Since this technique is not aimed at really testing the program but at finding potential interface scenarios, the test cases needed should be as merely necessary as possible. This can be interpreted differently in practice when practical constraints are taken into account, but the following specific criterion can be considered as a principle for striking a balance between finding interface scenarios and generating test cases at this stage.

Let the first level (top level) interface of the program under testing, called *Interface*₁, show n exclusive input patterns: I_1, I_2, \dots, I_n , indicating n different choices for the input of the program. For each selected pattern I_i ($i = 1 \dots n$), another lower level interface showing m input patterns $I_{11}, I_{12}, \dots, I_{1m}$ will be displayed. Applying this principle to all interface patterns at all levels, an interface tree for the whole program could be built, as illustrated in Figure 3. Then, a set of test cases need to be generated to cover every path from one of the top nodes (denoting one possible input pattern on the top level interface) to one of the bottom nodes (denoting one possible input pattern on the related bottom level interface), such as $path_1 = I_1, I_{11}, I_{111}$ and I_1, I_{12}, I_{121} . Note that there might be many functional scenarios (behaviors) under each interface scenario (input). Creating test cases to cover all of such functional scenarios would increase the cost while missing identifying any interface scenario would jeopardize the functional coverage in formalizing the specification. In this sense, the criterion for test case generation described above provides a good balance.

Assume that we have obtained all interface scenarios, the next issue is how to represent them so that they will present a clear guideline for the formalization of the corresponding functional requirements. We adopt the following finite state machine (FSM) notation for this purpose.

Definition 1. A finite-state machine (FSM) is a quintuple $(\Sigma, S, s_0, \delta, F)$, where Σ is a non-empty set of input symbols, S is a finite, non-empty set of states, $s_0 \in S$ is an initial state, δ is the state-transition function, and $F \subseteq S$ is the set of final states.

In the FSM representing interface scenarios of a program, the five elements are interpreted as follows. Each state S denotes one page of the GUI of the

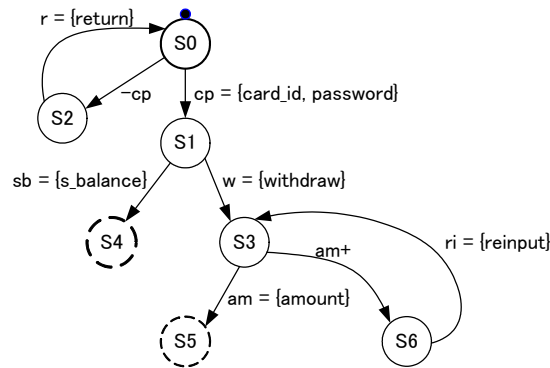


Fig. 4. An example of the finite state machine for a simplified ATM

program, where a page of GUI is an image with certain layout and contents shown on the display of computer. Each input symbol in Σ denotes a set of input items whose concrete values are necessary for executing the program. $s_0 \in S$ denotes the very top page of the GUI of the program. δ defines transitions of pages of the GUI based upon the input items. F represents the set of final pages showing the execution result of the program.

Figure 4 shows a simple example of FSM for a simplified Automated Teller Machine (ATM) program. The state s_0 , marked with a black dot at the top, is the initial state, representing the top page of the GUI of the ATM system, and the states s_4 and s_5 are both final states which are deliberately drawn in broken line circles. The FSM describes two interface scenarios. One is the scenario for withdrawing money from the bank account, and another is to show the account balance. The scenario for withdrawing money starts from the initial state s_0 . When correct user's *card_id* and *password* are provided, which is denoted by symbol cp , the state will transfer to s_1 ; otherwise, if any of them is incorrect, denoted by symbol $-cp$, the state will transfer to state s_2 and will return to state s_0 . At state s_1 , if the input symbol is w (denoting the *withdraw* command), the state will transfer to state s_3 ; otherwise, the state will transfer to the final state s_4 denoting the final page of GUI in which the requested account balance is shown. At state s_3 , if the requested amount smaller than the account balance is provided, denoted by am , the state will transfer to the final state s_5 ; otherwise, if the amount is greater than the account balance, the state will transfer to state s_6 , and then returns to state s_3 for re-input of appropriate amount.

In the context of the FSM, an interface scenario is in fact defined as a set of state transition sequences. Each sequence has the form: $[s_0, s_1, s_2, \dots, s_n]$, where s_0 must be the initial state of the FSM and s_n must be a final state. For example, the scenario for withdrawal discussed above is defined by the set of state transition sequences:

$$Scenario_1 = \{[s_0, s_1, s_3, s_5], [s_0, s_1, s_3, s_6, s_3, s_5]\},$$

$$[s_0, s_1, s_3, s_6, s_3, s_6, s_3, s_5], \dots,$$

$$[s_0, s_1, s_3, s_6, s_3, s_6, \dots, s_3, s_5]\}.$$

Obviously, this interface scenario is infinite, but the input values associated with the state transitions can be classified into a finite set based on which corresponding input variables can be declared in the signature of the corresponding operation. For example, three input variables for representing inputs *card_id*, *password*, and *withdrawal amount* can be declared for the signature of the *withdraw* operation. As a result, the operation has the following format:

```
process Withdraw(card_id : string,
                password : nat0,
                amount : nat0)
    cash : nat | error_message : string
end_process
```

where the output variables *cash* and *error_message* are created for the need of writing the formal specification.

The operation contains three input variables *card_id*, *password*, and *amount*. The *card_id* is a string, and both *password* and *amount* are a natural number (including zero). The output of the operation has two exclusive possibilities: either *cash* or *error_message*. The pre- and post-conditions of the operation are empty at the moment, but will be completed in the next step. The operation is written in the SOFL formal specification language [5] for the sake of our expertise. SOFL is an extension of VDM-SL [6] to have more capability for modelling of practical systems. Note that in spite of using SOFL for discussion, our approach presented in this paper is independent of specification languages; it can be applied to any formal notation using pre- and post-conditions.

3.3. Construction of Formal Specifications

Having understood all of the implemented interface scenarios, the next key issue is how to formally define the identified functional scenarios as operation specifications. As mentioned in Section 3.1, the potential behavior of each operation can be derived from the informal requirements, but the understanding of it may remain limited due to the ambiguities of the informal expressions. Writing a formal specification for the operation can force the tester to consider every possible aspect and to establish a firm foundation for testing. The key issue in writing the formal specification is how to provide appropriate pre- and post-conditions for the operation.

This can be done by analyzing the corresponding functional scenario. We first target on the post-condition because it focuses on the function of the operation by defining the relation between input and output. During this process, we may find that some state variables (or external variables as we call in both VDM and SOFL) are necessary for expressing the post-condition. We then try to figure out the pre-condition to ensure that the post-condition defines a valid final state when the pre-condition is satisfied by an initial state. Consider the operation *Withdraw* mentioned previously as an example. Applying this technique, we provide the following specification:

```

process Withdraw(card_id: string,
                  password: nat0,
                  amount: nat0)
    cash: nat | error_message: string
ext wr accounts: set of Account
pre exists![acc: accounts] |
    acc.card_id = card_id and
    acc.password = password
post let acc: accounts |
    acc.card_id = card_id and
    acc.password = password
    in
    amount ≤ acc.balance and
    cash = amount and
    accounts =
    union(
        diff(~accounts, {acc}),
        {modify(acc, balance →
                acc.balance - amount)}
    )
    or
    not (amount ≤ acc.balance) and
    bound(error_message) and
    accounts = ~accounts
end_process

```

where the logical operator “and” is used for \wedge and “or” for \vee .

In this specification, an external variable *accounts* is declared. This variable is used to hold a set of customers’ bank accounts, each being a member of the type *Account* that is declared as a composite type with three fields: *card_id*, *password*, and *balance*. The pre-condition requires the existence of a unique account in *accounts* such that its *card_id* and *password* are the same as the input *card_id* and *password*, respectively. The post-condition defines two cases: one is for a successful withdrawal and the other is a failed withdrawal. When the requested amount is not greater than the balance, the withdrawal will be successful and the requested amount will be delivered, which is denoted by the variable *cash*. The balance will also be updated in the account to reflect the reduction of the requested amount. On the other hand, when the withdrawal is failed (i.e., the amount is greater than the account balance), an error message denoted by the variable *error_message* is provided. For testing purpose, we are not interested in the specific content of the error message implemented by the program, therefore, we use the expression **bound**(*error_message*) in the post-condition to mean that *error_message* is made available but its content is yet to be determined by the programmer later.

4. Test Case Generation and Result Analysis

The goal of testing is to check whether each specification scenario defined in the specification is correctly implemented by the program. Theoretically, for any *specification scenario*, there should be an *implementation scenario* in the program to correctly implement it if the program refines the specification. A natural strategy for test case generation is therefore to let the generated test cases cover all the specification scenarios. To effectively apply this strategy, we have worked out a decompositional approach to test set generation from a pre-post style formal specification, which will be introduced later in this section. To this end, we first need to precisely define the fundamental concepts known as *functional scenario form (FSF)* and *functional scenario (FS)*.

4.1. Functional scenarios

For simplicity, let $S(S_{iv}, S_{ov})[S_{pre}, S_{post}]$ denote the specification of an operation S , where S_{iv} is the set of all input variables whose values are not changed by the operation, S_{ov} is the set of all output variables whose values are produced or updated by the operation, and S_{pre} and S_{post} are the pre- and post-conditions of S , respectively. In S_{post} , we use \tilde{x} and x to represent the initial value before the operation and the final value after the operation of external (or state) variable x , respectively. Thus, \tilde{x} serves as an actual input variable of the operation while x is treated as an output variable; that is, $\tilde{x} \in S_{iv}$ and $x \in S_{ov}$. Of course, S_{iv} and S_{ov} may contain other input parameter variables and output parameter variables, respectively.

Definition 2. Let $S_{post} \equiv (C_1 \wedge D_1) \vee (C_2 \wedge D_2) \vee \dots \vee (C_n \wedge D_n)$, where each C_i ($i \in \{1, \dots, n\}$) is a predicate called a “guard condition” that contains no output variable in S_{ov} and $\forall_{i,j \in \{1, \dots, n\}} \cdot i \neq j \Rightarrow C_i \wedge C_j = \text{false}$; D_i a “defining condition” that contains at least one output variable in S_{ov} but no guard condition. Then, a (functional) scenario f_s of S is a conjunction $S_{pre} \sim \wedge C_i \wedge D_i$, and the expression $(S_{pre} \sim \wedge C_1 \wedge D_1) \vee (S_{pre} \sim \wedge C_2 \wedge D_2) \vee \dots \vee (S_{pre} \sim \wedge C_n \wedge D_n)$ is called a functional scenario form (FSF) of S .

The decorated pre-condition $S_{pre} \sim = S_{pre}[\tilde{\sigma}/\sigma]$ denotes the predicate resulting from substituting the initial state $\tilde{\sigma}$ for the final state σ in pre-condition S_{pre} . We treat a conjunction $S_{pre} \sim \wedge C_i \wedge D_i$ as a scenario because it defines an independent behavior: when $S_{pre} \sim \wedge C_i$ is satisfied by the initial state (or intuitively by the input variables), the final state (or the output variables) is defined by the defining condition D_i . To facilitate our discussion throughout this paper, we call the conjunction $S_{pre} \sim \wedge C_i$ the *test condition* of the scenario $S_{pre} \sim \wedge C_i \wedge D_i$.

Note that simply treating a disjunctive clause in the disjunctive normal form of the post-condition as a functional scenario is not necessarily correct in supporting our method for checking the refinement relation between the specification and the program. For example, let $x > 0 \wedge (y = x \vee y = -x) \vee x \leq 0 \wedge y = x + 1$ be the post-condition of an operation whose pre-condition is assumed

to be *true*, where x is the input and y the output. It states that when $x > 0$, y is defined either as x or as $-x$ (the specifier does not care which definition will be implemented). In this case, the programmer may refine it into the post-condition $x > 0 \wedge y = x \vee x \leq 0 \wedge y = x+1$ and then implement it, which will of course satisfy the original specification. However, if we convert the original specification into the disjunctive normal form $x > 0 \wedge y = x \vee x > 0 \wedge y = -x \vee x \leq 0 \wedge y = x+1$, and treat each of the two disjunctive clauses $x > 0 \wedge y = x$ and $x > 0 \wedge y = -x$ as an individual functional scenario, respectively, and generate test cases to cover each of these two scenarios, we may not find a satisfactory answer in the program for the scenario $x > 0 \wedge y = -x$ because no program paths implementing this scenario exist in the program. This may lead to a confusion in testing and consequently increase the cost. Another concern is that a disjunctive clause in a DNF may not properly reflect the client's perception of a desired function. For instance, neither of the two expressions $x > 0 \wedge y = x$ and $x > 0 \wedge y = -x$ is appropriate to reflect the client's desire individually, but the functional scenario $x > 0 \wedge (y = x \vee y = -x)$ obtained by their combination does. For the above reasons, our strategy of generating test cases based on functional scenarios is adopted in this paper.

Any operation specification can be transformed to an equivalent FSF and a proved algorithm for such a transformation is made available in our previous publication [7]. Below we use the operation *Swap_Increase_Maintain* as an example to explain the concepts of FSF and functional scenario. The example may look a little artificial but it allows us to illustrate all the relevant aspects of the concepts.

$$\begin{aligned}
 & \textit{Swap_Increase_Maintain} \\
 & (\{\tilde{x}, \tilde{y}\}, \{x, y\}) \\
 & [x \geq 0, \\
 & \tilde{x} < \tilde{y} \wedge y = \tilde{x} \wedge x = \tilde{y} \wedge \tilde{x} < 100 \vee \\
 & \tilde{x} < \tilde{y} \wedge (y > \tilde{x} \wedge x > \tilde{y}) \wedge \tilde{x} \geq 100 \vee \\
 & \tilde{x} \geq \tilde{y} \wedge (y = \tilde{y} \wedge x = \tilde{x}) \vee \\
 & \tilde{x} \geq \tilde{y} \wedge (y + \tilde{x} = x + \tilde{y})],
 \end{aligned}$$

where $\textit{Swap_Increase_Maintain}_{iv} = \{\tilde{x}, \tilde{y}\}$; $\textit{Swap_Increase_Maintain}_{ov} = \{x, y\}$; $x \geq 0$ is the pre-condition; the last predicate expression is the post-condition; and all the variables involved are of the type *real*. The operation exchanges, increases or "sustains" the values of x and y , depending on certain conditions satisfied by the input. An FSF of the specification is derived as follows:

- (1) $\tilde{x} \geq 0 \wedge \tilde{x} < \tilde{y} \wedge \tilde{x} < 100 \wedge y = \tilde{x} \wedge x = \tilde{y} \vee$
- (2) $\tilde{x} \geq 0 \wedge \tilde{x} < \tilde{y} \wedge \tilde{x} \geq 100 \wedge (y > \tilde{x} \wedge x > \tilde{y}) \vee$
- (3) $\tilde{x} \geq 0 \wedge \tilde{x} \geq \tilde{y} \wedge (y = \tilde{y} \wedge x = \tilde{x} \vee y + \tilde{x} = x + \tilde{y})$

This FSF is derived from the DNF of the post-condition and includes the three functional scenarios marked by (1), (2), and (3) (after removing the last operator \vee in each case), respectively. In (1), $\tilde{x} \geq 0$ is the decorated pre-condition; $\tilde{x} < \tilde{y} \wedge \tilde{x} < 100$ is the guard condition; and $y = \tilde{x} \wedge x = \tilde{y}$ is the defining condition. The other two scenarios can be interpreted similarly.

Before describing the test strategy, we need to clarify the concepts of *test case* and *test set* precisely.

Definition 3. Let $S_{iv} = \{x_1, x_2, \dots, x_r\}$ be the set of all input variables of operation S and $Type(x_i)$ denotes the type of x_i ($i = 1, \dots, r$). Then, a test case for S , denoted by T_c , is a mapping from S_{iv} to the set *Values*:

$$T_c : S_{iv} \rightarrow Values$$

$$\text{where } Values = Type(x_1) \cup Type(x_2) \cup \dots \cup Type(x_r).$$

A test case is usually expressed as a set of pairs of input variables and their values, for example, $\{(x_1, 5), (x_2, 10), \dots, (x_r, 20)\}$. A test set for operation S is a set of test cases, and is usually expressed as a set of sets of pairs. With the above preparation, we can now define precisely the test strategy.

Test Strategy: Let operation S have an FSF $(S_{pre} \sim \wedge C_1 \wedge D_1) \vee (S_{pre} \sim \wedge C_2 \wedge D_2) \vee \dots \vee (S_{pre} \sim \wedge C_n \wedge D_n)$ where ($n \geq 1$). Let T be a test set for S . Then, T must satisfy the condition $(\forall_{i \in \{1, \dots, n\}} \exists_{t \in T} \cdot S_{pre} \sim(t) \wedge C_i(t))$.

We call this strategy *scenario-coverage strategy* (SCS). The strategy states that for each functional scenario there exist some test cases in test set T that satisfy its test condition. If T meets this condition, it will ensure that every functional scenario defined in the specification is tested (at least once), thus ensuring that no specified functional behavior is missed in the test.

To ensure that the test strategy is fully applied, a decompositional test set generation approach is proposed, which is described in detail next.

4.2. Decompositional Approach to Test Set Generation

Using the decompositional test set generation approach, the test set generated from an operation specification is realized by generating test sets from all of its functional scenarios. The production of test set from a functional scenario is realized by generating them from its test condition, which can be divided further into test set generations from every disjunctive clause of the test condition. To clearly describe the proposed method, we first need to define a function, called *test case generator* (TCG), that yields a test set for a given logical expression such that every test case in the set plays an independent role while it satisfies the expression (e.g., assuming the expression is a disjunction, each test case in the set makes each disjunctive clause true). We use \mathcal{G} to denote the TCG, which is a function from the universal set of logical expressions L_E to the universal set of test sets T_S , formally,

$$\mathcal{G} : L_E \rightarrow T_S.$$

By logical expressions here we mean the first order logic expressions written in SOFL. Let p be a logical expression in L_E . Then, $\mathcal{G}(p)$ represents the test set generated, which is a member of T_S and satisfies p (i.e., every test case in the test set satisfies p). The key issue now is how \mathcal{G} is defined. We will define \mathcal{G} to ensure that our test strategy discussed previously is supported.

An FSF of operation specification S is a disjunction of all its functional scenarios, defining a set of independent behaviors, the test set generated from the

FSF should therefore be the union of all the test sets generated from all of its functional scenarios, as formalized in **Criterion 1** below.

Criterion 1: $\mathcal{G}((S_{pre} \sim \wedge C_1 \wedge D_1) \vee (S_{pre} \sim \wedge C_2 \wedge D_2) \vee \dots \vee (S_{pre} \sim \wedge C_n \wedge D_n)) = \mathcal{G}(S_{pre} \sim \wedge C_1 \wedge D_1) \cup \mathcal{G}(S_{pre} \sim \wedge C_2 \wedge D_2) \cup \dots \cup \mathcal{G}(S_{pre} \sim \wedge C_n \wedge D_n)$.

Since the execution of a program only requires input values, test case generation from a functional scenario actually depends only on its test condition. This idea is reflected in **Criterion 2**.

Criterion 2: Let $S_{pre} \sim \wedge C_i \wedge D_i$ ($i = 1, \dots, n$) be a functional scenario of operation specification S . Then, $\mathcal{G}(S_{pre} \sim \wedge C_i \wedge D_i) = \mathcal{G}(S_{pre} \sim \wedge C_i)$.

In general, pre-condition $S_{pre} \sim$ can be in any form of predicate expression. To define \mathcal{G} to deal with test case generation from the conjunction $S_{pre} \sim \wedge C_i$, a systematic way is first to translate the conjunction into an equivalent disjunctive normal form (DNF) and then define \mathcal{G} based on it, as reflected in **Criterion 3**.

Criterion 3: Let $P_1 \vee P_2 \vee \dots \vee P_m$ be a DNF of the test condition $S_{pre} \sim \wedge C_i$. Then, we define $\mathcal{G}(S_{pre} \sim \wedge C_i) = \mathcal{G}(P_1 \vee P_2 \vee \dots \vee P_m) = \mathcal{G}(P_1) \cup \mathcal{G}(P_2) \cup \dots \cup \mathcal{G}(P_m)$.

A P_i ($i = 1, \dots, m$) is a conjunction of atomic predicate expressions, say $Q_i^1 \wedge Q_i^2 \wedge \dots \wedge Q_i^w$. An atomic predicate expression, in our context, is one of the three kinds of predicate expressions: (1) a relation (e.g., $x > y$), (2) a negation of a relation, and (3) a *strict quantified predicate*. A strict quantified predicate is a quantified predicate whose body does not contain any atomic predicate unrelated to its bound variables. For example, $\forall_{x \in X} \cdot y > x$ is a strict quantified predicate, while $\forall_{x \in X} \cdot y > x \wedge q > 0$ is not, because it contains the atomic predicate $q > 0$ that is not related to the bound variable x . This criterion indicates that generating test cases from a test condition can be achieved based on its DNF. The focus now is on the issue of how to produce test cases from a clause, such as $P_i \equiv Q_i^1 \wedge Q_i^2 \wedge \dots \wedge Q_i^w$. To discuss the generation criterion, we first need to define \mathcal{G} to deal with atomic predicate expressions, which will set up a basis for discussions on test case generation from the conjunction.

Criterion 4: Let $S_{iv} = \{x_1, x_2, \dots, x_r\}$ and $Q(x_1, x_2, \dots, x_q)$ ($q \leq r$) be a relation involving variables x_1, x_2, \dots, x_q . Then, $\mathcal{G}(Q(x_1, x_2, \dots, x_q)) = \{T_c \mid (\forall_{x \in \{x_1, x_2, \dots, x_q\}} \cdot Q(T_c(x_1), T_c(x_2), \dots, T_c(x_q))) \wedge (\forall_{x \in (S_{iv} \setminus \{x_1, x_2, \dots, x_q\})} \cdot T_c(x) = \text{anyvalue})\}$, *anyvalue* represents any value taken from the type of variable x .

For a relation involving a subset of the input variables of operation specification S , we generate a test case in which each involved variable is given a specific value in its type and all the generated values satisfy the predicate expression $Q(x_1, x_2, \dots, x_q)$, but the input variables of S that are not involved in $Q(x_1, x_2, \dots, x_q)$ are all assigned *anyvalue* from its type. For example, suppose operation specification S has two input variables x and y denoting two integers, then it is possible that $\mathcal{G}(x > 0) = \{(x, 2), (y, 8)\}$ (containing a single test case) where 8 in the pair $(y, 8)$ is *anyvalue*, but the values of variables x and y in the test set $\mathcal{G}(x < y) = \{(x, 2), (y, 5)\}$ are not *anyvalue* but the values that satisfy the relation $x < y$.

Having the above preparation, we can now define the criterion for generating a test set for a conjunction of atomic predicate.

Criterion 5: Let $Q_i^1 \wedge Q_i^2 \wedge \dots \wedge Q_i^w$ be a conjunction of w atomic predicates in a test condition of a functional scenario of S . Then, we have $\mathcal{G}(Q_i^1 \wedge Q_i^2 \wedge \dots \wedge Q_i^w) = \mathcal{G}(Q_i^1) \cap \mathcal{G}(Q_i^2) \cap \dots \cap \mathcal{G}(Q_i^w)$.

Let us take the operation *Withdraw* as an example to show how the criteria can be applied. The operation is converted into a disjunction of the following two functional scenarios:

- (1) (*exists!*[*acc: accounts*] |
acc.card_id = card_id and
acc.password = password) and
acc inset accounts and
acc.card_id = card_id and
acc.password = password and
amount <= acc.balance) and
(cash = amount and
accounts = union(diff(~accounts, {acc}),
{modify(acc, balance -> acc.balance - amount)}))
- (2) (*exists!*[*acc: accounts*] |
acc.card_id = card_id and
acc.password = password) and
acc inset accounts and
acc.card_id = card_id and
acc.password = password and
not (amount <= acc.balance)) and
(bound(*error_message*) *and*
accounts = ~accounts)

where the *let* expression is translated into the equivalent conjunction:

acc inset accounts and acc.card_id = card_id and
acc.password = password.

Following the test strategy SCS described above, we generate a test set as shown in Figure 5 that cover the two functional scenarios. Each test case consists of four values for the four input variables (i.e., the three parameters *card_id*, *password*, *amount*, and one external variable *accounts*). The test case on the first row allows us to test the situation where both the inputs *card_id* and *password* are correct and the input *amount* is smaller than the customer's account balance. The test case on the second row checks the situation where both the *card_id* and *password* are correct, but the *amount* exceeds the balance. The one on the third row tests the situation where the pre-condition of the operation is not satisfied because of the inconsistency between the input *password* and the registered *password* in the customer's account. Since the pre-condition is treated as an assumption for the operation, the last test case will not necessarily help find errors, but it will allow the tester to understand how the implementation deals with this situation and to provide a feedback to the programmer for potential improvement.

<i>card_id</i>	<i>password</i>	<i>amount</i>	<i>accounts</i>
L4513	3410	50000	{(L4513, 3410, 1000000), (A5910, 3102, 150000), (X2034, 1291, 3400000)}
A5910	3102	250000	{(G8113, 3410, 5060000), (A5910, 3102, 150000), (T2014, 1291, 2900000)}
X2034	1091	10000	{(G8113, 3410, 300000), (L0310, 3102, 35000), (X2034, 1291, 3400000)}

Fig. 5. A test set for the operation *Withdraw*.

4.3. Test Result Analysis

One of the major advantages of formal specification-based testing is that a precise test oracle can be automatically derived from the specification and can be used to determine, both automatically and manually, whether a test has found an error or not.

Definition 4. Let $f \equiv S_{pre} \sim \wedge C \wedge D$ be a functional scenario of operation specification S and P be a program implementing S . Let t be a test case generated based on f and r be the result of executing program P using t . Then, an error in P is found using t if the following condition holds:

$$S_{pre} \sim (t) \wedge C(t) \wedge \neg D(t, r)$$

This test oracle states that for a test case t satisfying both the pre-condition of S and the guard condition C of functional scenario f , if the execution result r of P does not satisfy the defining condition D (together with t), we can assert that an error in P is found by test case t .

In fact, this test oracle is derived from the refinement relation between S and P , which is formally defined as follows:

$$\forall \sim \sigma \in \Sigma \cdot S_{pre}(\sim \sigma) \implies S_{post}(\sim \sigma, \sigma),$$

where $\sim \sigma$ denotes the initial state before operation S and $\sigma (= P(\sim \sigma))$ denotes the final state after operation S that is produced by executing the corresponding program P . P is a correct refinement of specification S if and only if the final state σ produced by executing program P on the initial state $\sim \sigma$ satisfying the pre-condition meets the post-condition. In other words, if there exists any final state that does not satisfies the post-condition after the execution of the program with a satisfactory initial state, the program will not be a correct refinement of the specification, implying that a bug exists in the program.

The test for the operation *Withdraw* shown in Figure 5 can help to explain how the test oracle can be used. Assume that we obtain the actual execution result R for the test case on the first row in Figure 5 denoted by T_c^1 , which

consists of two values: updated *accounts* and *cash*. We substitute them for the variables *accounts* and *cash* in the post-condition, respectively, and evaluate the following implication:

$$Withdraw_{pre}(T_c^1) \Rightarrow Withdraw_{post}(T_c^1, R)$$

If the result is true, it means that no error is found by T_c^1 ; otherwise, it indicates the presence of some bugs in the program. Similarly, we can also perform the test result analysis for the other two test cases, which we omit here for brevity.

5. An Example

We have applied our approach to test an *IC Card software system* developed by a group of senior students at Hosei University. The system is intended to provide three services: *operations on ATM*, *purchase of railway tickets*, and *payment for shopping*. The operations on ATM includes *withdraw*, *deposit*, *show balance*, *transfer money*, *change password*, and *transactions record printing*. For purchasing railway tickets, the system provides the operations: *purchase a ticket*, *charge the card with cash*, *charge the card from the customer's bank account*, *show balance of the card* (i.e., the amount of money available in the buffer of the card), and *use the card* (for entering and existing a station). The payment for shopping is a service that allows one to pay for the goods purchased at shops using the card. It includes the following specific functions: *check card validity*, *check amount restrictions* (e.g., a card cannot be used for shopping of more than a fixed amount in Japanese yen per day at the same shop), *authenticate the card user*, and *pay for shopping*.

Since the whole example is too large to be put in a paper gracefully, we choose only the ATM sub-system as an example to show how our testing approach can be applied. As the ATM example in Section 3.3 is a simplified version of the system we use in this section, the contents of this section would be understood easily.

5.1. Application

Figure 6 shows a snapshot of the GUI scenario for an unsuccessful withdrawal from a customer's bank account. The first page of the GUI for the system shows three choices: *ATMSystem*, *ShoppingSystem*, and *TransportationSystems*. Clicking on the button named *ATMSystem*, the system transfers from the first page to the ATM GUI page given at the upper-right of the figure. Clicking on the button named *Withdraw*, the GUI changes to the page requesting a *card_id* and a *password*, which is given at the bottom-right of the figure. When a wrong password is input, the system moves to the page displaying an error message and requesting the user to re-enter the password. Clicking on the "yes" button (the left one), the system returns to the page requesting for a *card_id* and a *password*.

Using random testing based on a programmer's brief description of the interface scenarios of the system, we built a set of FSMs for the ATM system

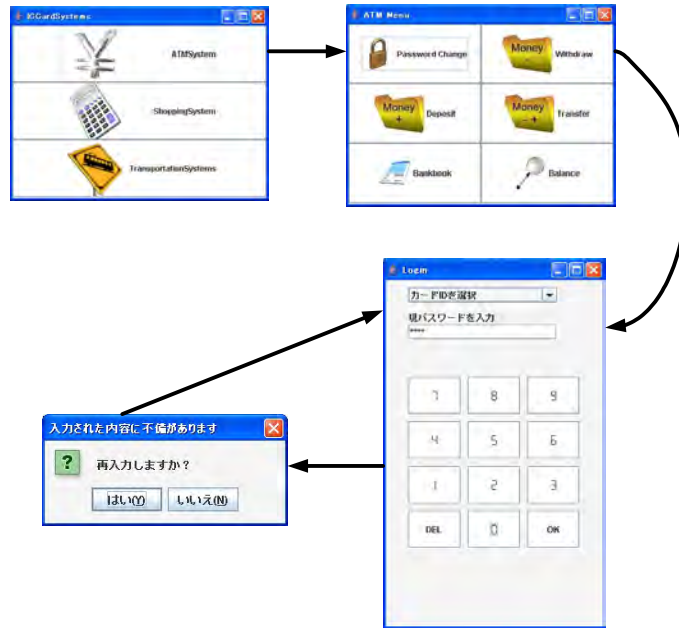


Fig. 6. A snapshot of using ATM for an unsuccessful withdrawal from the bank account

for simplicity and readability, as shown in Figure 7, each describing a single independent interface scenario, as indicated in the figure. Taking the same approach as we explained in Section 3, we construct an operation specification in pre- and post-conditions for each functional scenario. For the sake of both space and the similarity to the simplified ATM example, we omit the specifications here. We carried out testing of the program based upon the specifications to verify whether the desired behaviors are implemented correctly. The details of the testing is illustrated in Table 5.1.

Since this case study was mainly intended to test the usability of our approach (i.e., how interface scenarios can be found, how formal specifications can be constructed, and how test cases are generated) rather than conduct a rigorous

Features	No. of test cases	No. of detected defects
<i>withdraw</i>	8	3
<i>deposit</i>	7	5
<i>show balance</i>	6	2
<i>transfer money</i>	8	7
<i>change password</i>	9	6
<i>transactions record printing</i>	7	4

Table 1. Approximate data of the testing

experiment on its effectiveness, we only collected the data of defects found by our test cases rather than measured its effectiveness in terms of defect detection rate or some coverage criterion. There were total 27 defects found as the result, but the main defects that lead to major functional errors with respect to the functional requirements in the specifications are three, which are illustrated as follows:

- In the specification the cash card and password are required to provide before any service (e.g., balance inquiry, withdraw) starts, but the implementation in the program puts the selection of services before the input of card id and password. The purpose of such a requirement is to prevent people without cash card from trying to “play” with the ATM system.
- In the *Withdraw* service, if the requested amount is over the account balance, the customer should be given two choices: giving up the use of the service or re-entering the amount for withdrawal, but the implementation in the program insists only on re-entering the amount. The same error is also found in the *Transfer* service: when the amount for transferring is over the account balance, the program insists only on re-entering the amount rather than offering the choices.
- In the *Deposit* service, the amount for depositing is required not to be over 1,000,000 Japanese yen, but the program does not implement this constraint, therefore allowing any amount to be deposited in one service.

In the three errors, we found that the first one was relatively easier to find, but the second and the third ones took us more efforts to find out, because the missing functions did not cause fundamental problem in providing the services, and therefore did not easily attract the tester's attention. The tester's attention was actually raised by examining the corresponding defining condition in the post-condition of the operations. This result shows that formal specification does benefit testing in our approach.

5.2. Discussion

Our experience in the application has shown the usability and effectiveness of our testing approach for programs with no source code available. In particular, formalization of the informal requirements based upon the interface scenarios derived from the program has been found effective in helping the tester detect obscure errors and defects. Since all the potential interface scenarios need to be identified based upon executions of the program for writing interface-consistent formal specifications, a general technique called *testing-and-correction* must be taken. The technique suggests the following activities:

- No.1** Run the program using sample test cases (selected based upon the brief description of the functions of the program provided by the programmer) and record the interface scenarios experienced if the program terminates normally.

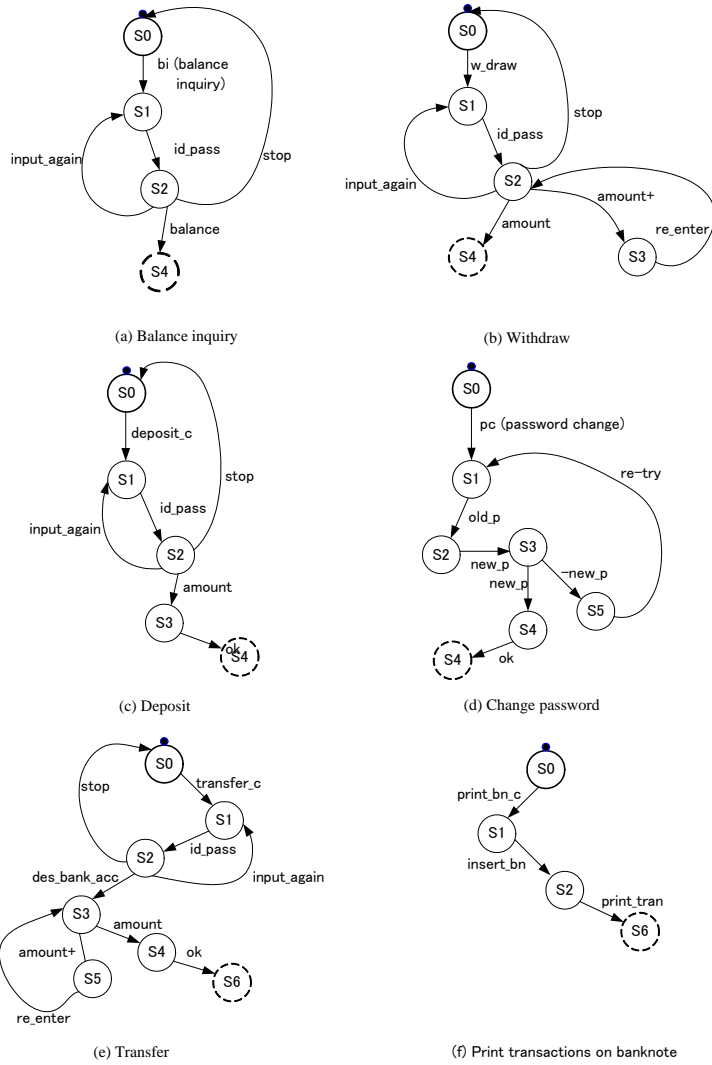


Fig. 7. FSMs for ATM interface scenarios.

No.2 If the program does not terminate normally due to some bugs, the programmer will be asked to eliminate the bugs before the next action in testing is taken.

No.3 Repeat steps 1 and 2 until all interface scenarios are found.

The process of applying the *testing-and-correction* technique helps to detect some obvious errors in the program, but it may not be effective in uncovering obscure errors, especially those violating the requirements in an ambiguous manner. Such obscure errors can be effectively revealed during the formalization of requirements and the testing process based upon the formal specifications.

To effectively use our testing approach in practice, however, the following challenges may need to be addressed.

- The program must always terminate normally or the programmer must always be available for removing bugs detected.
- The tester has sufficient ability and skill in formalizing the informal requirements into formal specifications. For this purpose, the requirements must be complete or the client must be available for consultation whenever the details of the requirements are inquired by the tester.

The first challenge is not exceptional for the application of other testing approaches, but the second one may be a major challenge in practice currently. The reason is that most software testers are not necessarily trained in formal methods and may not know how to use the technique, but this situation is changing due to three factors. The first is that many evidences have demonstrated the power of formal specification in enhancing software quality [8] and more and more companies become interested in training their employees in formal specification techniques (e.g., in Japan). The second factor is that a growing number of practitioners have become using formal specification techniques in practice, especially in the domain of dependable systems [4]; this demonstrates the feasibility and effectiveness of formal specification techniques in industry. The third is the progress made in supporting formal specification construction by software tools based on formal specification patterns [9]. With such a technique, one does not need to directly write formal specifications; instead, one only needs to express one's ideas in a structured English and the supporting tool will gradually automatically produce the formal specification. With the above progresses, it is possible for high level testing consultants or company practitioners to efficiently apply our method in practice.

6. Related Work

Our work falls into the category of model-based testing (MBT), but differs from existing approaches in the way of deriving models. In this section, we briefly overview the related work in this area and compared it to our approach.

There have been extensive research on and application of MBT over the last three decades [10], and the work can be classified into four categories: general

discussion of MBT, MBT based upon graphical models, MBT based on formal specifications, and automation in MBT. El-Far and Whittaker give a general introduction to model-based testing in [11], discussing its underlying principle, process, and techniques. Apfelbaum and Doyle argue that a finite state machine MBT can reduce cost and time, and can increase software quality, but pointed out that the technique also faces challenges in acceptance [12]. Offutt *et al* describe an approach to generating test cases from UML Statecharts for components testing [13]. Hartmann *et al* extend the approach for integration testing and provide effective tool support for test automation [14]. Bernot *et al* set up a theoretical foundation and a tool support for specification-based testing, explaining how a formal specification can serve as a base for test case generation and as an oracle for test result evaluation [15]. Dick and Faivre propose to transform pre-condition into a disjunctive normal form (DNF) and then use it as the basis for test case generation [16]. Stocks and Carrington suggest that test templates be defined as the base for test case generation and a large test template is divided into smaller templates for generating more detailed test cases [17]. In order to deal with the practical challenges MBT techniques have encountered due to the inconsistency between the component interfaces in the model and in the program, Blackburn *et al* present an interface-driven MBT approach that combines requirement modeling and component interface analysis to support automated test case and test driver generation [18]. In spite of tremendous efforts by many researchers, MBT is still difficult to be used for large scale systems because of its complexity and the potential inconsistencies in both component interface and system architectures. It is also unsuitable for testing programs, such as legacy code, which does not have source code available. To tackle this problem, Bertolino *et al* describe an anti-model-based testing approach in [19]. The essential idea of this approach shares with our testing approach presented in this paper; it suggests to use some test cases to execute the program under test and observe the traces of executions, and then try to synthesize and abstract model of the system. Compared to our work in this paper, Bertolino *et al* neither advocate the use of formalization for the abstract model, nor discuss in detail about how all the necessary traces are observed and represented and how the traces are synthesized into an abstract model. Our novel contribution in this paper is to present a systematic method for identifying all interface scenarios, formalizing requirements into formal operation specifications whose interfaces are consistent with the corresponding ones of the program, and for testing programs based upon the formal specifications.

7. Conclusion and Future Work

We present a systematic approach to testing programs with no available source code. The approach includes three steps: (1) identifying important interface scenarios by executing the program with test cases, (2) formalizing the informal requirements into formal specifications, and (3) generating test cases from the

formal specifications to test the program. The benefit of our testing approach is to allow the tester to thoroughly understand the desired behaviors of the system and to provide a precise model for testing it. We have also applied our approach to an IC Card system to gain experience and to learn potential challenges in practice.

In the future, we will continue to work along the direction set in this paper to make the testing approach more mature and effective. In particular, we will refine and improve the techniques for discovering interface scenarios, formalizing user requirements, and generating test cases. We also plan to conduct a rigorous experiment on the effectiveness of our approach in industrial environment. To enhance the efficiency of the testing activities, we will also work on the tool support.

Acknowledgments. This research is supported in part by NII Collaborative Research Program, 973 program Grant No.2010CB328102, and NSFC under Grant Nos. 61133001, 60910004. We would like to thank all the students involved in this project for their contributions to the application of the testing approach and cooperations in data collection.

References

1. B. Beizer, *Black-Box Testing*, John Wiley and Sons, Inc., 1995.
2. S. Liu, J. A. McDermid, and Y. Chen, "A Rigorous Method for Inspection of Model-Based Formal Specifications", *IEEE Transactions on Reliability*, vol. 59, no. 4, pp. 667–684, December 2010.
3. M. Li and S. Liu, "Automated Functional Scenarios-Based Formal Specification Animation", in *Proceedings of 19th Asia-Pacific Software Engineering Conference (APSEC 2012)*. December 4-7 2012, p. to appear, IEEE CS Press.
4. J. Woodcock, P. G. Larsen, J. Bicarregui, and J. Fitzgerald, "Formal Methods: Practice and Experience", *ACM Computing Surveys*, vol. 41, no. 4, pp. 1–39, 2009.
5. S. Liu, *Formal Engineering for Industrial Software Development Using the SOFL Method*, Springer-Verlag, ISBN 3-540-20602-7, 2004.
6. The VDM-SL Tool Group, *Users Manual for the IFAD VDM-SL Tools*, The Institute of Applied Computer Science, February 1994.
7. S. Liu et al, "An Automated Approach to Specification-Based Program Inspection", in *Proceedings of 7th International Conference on Formal Engineering Methods (ICFEM2005)*, Manchester, UK, 1-4 Nov. 2005, pp. 421–434, LNCS 3785, Springer-Verlag.
8. D. Craigen, S. Gerhart, and T. Ralston, *Industrial Applications of Formal Methods to Model, Design and Analyze Computer Systems: an International Survey*, Noyes Data Corporation, USA, 1995.
9. X. Wang, S. Liu, and H. Miao, "A Pattern-Based Approach to Formal Specification Construction", in *Proceedings of 2011 International Conference on Advanced Software Engineering and Its Application*, Jeju, Korea, December 13-15 2011, CCIS 257, pp. 159–168, Springer-Verlag.
10. S. R. Dalal, A. Jain, N. Karunanithi, J. M. Leaton, C. M. Lott, G. C. Patton, and B. M. Horowitz, "Model-Based Testing in Practice", in *Proceedings of Proceedings of the 21st International Conference on Software Engineering*. 1999, pp. 285–294, IEEE Computer Society Press.

11. I. K. El-Far and J. A. Whittaker, *Model-based Software Testing*, Wiley, 2001.
12. L. Apfelbaum and J. Doyle, "Model-Based Testing", in *Proceedings of Software Quality Week Conference*. May 1997, LNCS, Springer-Verlag.
13. J. Offutt and A. Abdurazik, "Generating Test Cases from UML Specifications", in *Proceedings of Second International Conference on UML'99*, Robert France and Bernhard Rumpe, Eds., Fort Collins, CO, USA, October 28-30 1999, pp. 416–429, Springer.
14. J. Hartmann, C. Imoberdorf, and M. Meisinger, "UML-Based Integration Testing", in *Proceedings of 2000 ACM SIGSOFT Symposium on Software Testing and Analysis*, Portland, Oregon, USA, 2000, pp. 60–70, ACM Press.
15. Gilles Bernot, Marie Claude Gaudel, and Bruno Marre, "Software testing based on formal specifications: a theory and a tool", *Software Engineering Journal*, pp. 387–405, November 1991.
16. J. Dick and A. Faivre, "Automating the Generation and Sequencing of Test Cases from Model-based Specifications", in *Proceedings of FME '93: Industrial-Strength Formal Methods*, Odense, Denmark, 1993, pp. 268–284, LMCS 670, Springer-Verlag.
17. P. Stocks and D. Carrington, "A Framework for Specification-Based Testing", *IEEE Transactions on Software Engineering*, vol. 22, no. 11, pp. 777–793, November 1996.
18. M. R. Blackburn, R. D. Busser, and A. M. Nauman, "Interface-Driven, Model-Based Test Automation", in *Proceedings of 2005 International Conference on Software Testing, Analysis and Review*. Nov. 14-18 2002, pp. 27–30, Software Quality Engineering.
19. A. Bertolino, A. Polini, P. Inverardi, and H. Muccini, "Towards Anti-Model-Based Testing", in *Proceedings of 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (Extended Abstract)*, June 28 - July 1 2004, pp. 124–125.

Shaoying LIU is Professor of Software Engineering at Hosei University, Japan. He received the Ph.D degree in Computer Science from the University of Manchester, U.K in 1992. His research interests include Formal Methods and Formal Engineering Methods for Software Development, Specification Verification and Validation, Specification-based Program Inspection, Automatic Specification-based Testing, and Intelligent Software Engineering Environments. He has published a book titled "Formal Engineering for Industrial Software Development Using the SOFL Method" with Springer-Verlag, four edited conference proceedings, and over 120 academic papers in refereed journals and international conferences. He has recently served as General Co-Chair of ICFEM 2012 and PC member for numerous international conferences. He is on the editorial board for the Journal of Software Testing, Verification and Reliability (STVR) and ISRN Software Engineering. He is a Fellow of British Computer Society, a Senior Member of IEEE Computer Society, and a member of Japan Society for Software Science and Technology.

Wuwei SHEN received the PhD degree in computer science from the Department of Electrical Engineering and Computer Science, the University of Michigan, Ann Arbor in 2001. He is currently an associate professor at Western Michigan University. His research interests include object-oriented software

model design and analysis, software model consistency checking, model-based software testing, software error detection, and the management of software artifacts.

Shin NAKAJIMA received Ms degree in 1981 and Ph.D degree in 2000 both from The University of Tokyo. He is currently professor at National Institute of Informatics, Tokyo, Japan. His current interests include formal methods, automated verification, and software modeling. Dr. Nakajima is a member of ACM, JSSST, and IPSJ.

Received: March 1, 2012; Accepted: December 5, 2012.

Image Denoising Using Anisotropic Second and Fourth Order Diffusions Based on Gradient Vector Convolution

Huaibin Wang, Yuanquan Wang^{*}, and Wenqi Ren

School of Computer and Communication engineering, Tianjin University of
Technology, 300384 Tianjin, China

^{*}Corresponding Author: yqwang@bit.edu.cn

Abstract. In this paper, novel second order and fourth order diffusion models are proposed for image denoising. Both models are based on the gradient vector convolution (GVC) model. The second model is coined by incorporating the GVC model into the anisotropic diffusion model and the fourth order one is by introducing the GVC to the You-Kaveh fourth order model. Since the GVC model can be implemented in real time using the FFT and possesses high robustness to noise, both proposed models have many advantages over traditional ones, such as low computational cost, high numerical stability and remarkable denoising effect. Moreover, the proposed fourth order model is an anisotropic filter, so it can obviously improve the ability of edge and texture preserving except for further improvement of denoising. Some experiments are presented to demonstrate the effectiveness of the proposed models.

Keywords: Gradient vector convolution, fourth order diffusion, anisotropic diffusion, noise removal, texture preserving.

1. Introduction

Image denoising is one of the fundamental topics of research in image processing and computer vision, whose main goal is to eliminate the noise and preserve edges in image. During the last two decades, partial differential equations (PDEs) have been justified as effective tools for image smoothing, which are able to achieve a good trade-off between noise removal and edge-preserving. Cacelles et al. generally explained the superiority of PDEs-based denoising in [1]. Besides PDE-based methods, some recently developed non-diffusion based approaches can also bring considerable improvement in denoising performance, such as the kernel regression (LARK) [2], bilateral filter [3], patch-based methods [4-5], wavelet-based methods [6-7] and the BM3D based on sparse representation [8]. In this paper, we will focus on the PDE-based diffusion methods.

The anisotropic diffusion introduced by Perona and Malik (P-M) [9] can be considered as a typical feature-preserving denoising algorithm, where diffusion is controlled by a variable coefficient in order to preserve edges. Since the

work of Perona and Malik [9], there have been extensive literatures that present a variety of PDEs-based anisotropic diffusion models and offer diverse numerical schemes to obtain the steady-state solution [10]-[18], [31-32]. Details of the behavioral of the anisotropic diffusion can be found in [10]. Scherzer et al. [11] investigated connections between regularization theory and framework of diffusion filtering. Catte et al. [12-13] proved the ill-posedness of the P-M equation and proposed a modified diffusion coefficient which is a function of a smoothed gradient. Rudin et al. introduced Shock filters [14] and total variation [15] minimization. The works in [14] opened the possibility to reformulate the image enhancement as a combination of two coupled terms that implement inverse and forward diffusion processes. Alvarez and Mazorra [16] combined the shock filter and a diffusion term for image enhancement. Gilboa et al. [17] proposed a forward and backward (FAB) adaptive diffusion process that enhances features while locally denoising smoother segments; later, they introduced the complex diffusion for ramp-preserving [18]. Black et al. [19] illustrated the relations between anisotropic diffusion and robust statistics. In [20], a method was proposed for texture preserving by adding a spatially varying fidelity term. Another interesting work [21] was the incorporation of the gradient vector flow (GVF) field [22], which is originally introduced as an external force for active contour model [23], into the anisotropic diffusion. Ghita et al. [24] introduced a new modified GVF (INGVF) field into the P-M equation in order to improve the denoising effect. Sum et al. [25] proposed a stabilization method to make the GVF-based P-M equation stable. The proposed GVF-based P-M equation can improve the denoising effect, but the computational cost is expensive.

Although the second-order anisotropic diffusion is effective for image noise removal, it can lead to staircase effect. These staircases are visually unpleasant and can be falsely detected as edges. In order to alleviate this staircase effects, a number of authors have proposed high order PDEs for image denoising [26-28], [33-34]. One of the most popular fourth-order PDEs is introduced by You-Kaveh (Y-K) [27], which seeks to approximate the noisy image with a piecewise harmonic one. Another classical fourth-order model was proposed by Lasaker, Lundervold and Tai (LLT) [28], in which two different functions have been proposed to measure the oscillations in the noisy data. The fourth-order filters damp high frequency components of images much faster than the second-order peers, this means the fourth-order models would over-smooth the step edges in images. In addition, they can also cause speckle noise in the filtered image.

In this paper, we first introduce the gradient vector convolution (GVC) model [29] into the anisotropic P-M equation. The GVC model is our previous work, which serves as external force for active contours. It is very robust to noise and can be calculated in real time. The new GVC-based P-M equation has many desirable properties, such as superior noise robustness, reduced computational cost, and the improved denoising effect. Second, the GVC field is introduced into Y-K model; the proposed GVC-based fourth order model is anisotropic because the diffusions along the directions of level set and gradient are uneven. Thus, the modified Y-K model could not only improve

peak-signal-to-noise ratio (PSNR) of the filtered image, but also keep edge and texture information in the filtered image.

This paper is organized as follows: Section 2 briefly introduces the second order and fourth order approaches for image denoising. Section 3 details the proposed method of GVC-based second order and fourth order models. The experimental results are provided in Section 4. Finally, the conclusion is given in Section 5.

2. Related Works

2.1. Second Order diffusion: P-M Model and INGVF-based P-M Model

The anisotropic diffusion proposed by Perona and Malik [9] takes the following form

$$\begin{cases} \partial u / \partial t = \operatorname{div}(c(|\nabla u|)\nabla u) \\ u(x, y, t = 0) = u_0(x, y) \end{cases} \quad (1)$$

where ∇ is the gradient operator, div is the divergence operator and $c(\cdot)$ is the diffusion coefficient. The diffusion coefficient is a positive and non-increasing function over $|\nabla u|$. The coefficient plays an important role in diffusion and it usually takes one of the following two forms:

$$c(|\nabla u|) = \frac{1}{1 + (|\nabla u|/k)^2} \quad (2)$$

$$c(|\nabla u|) = \exp\left(-(|\nabla u|/k)^2\right) \quad (3)$$

where k is so-called contrast parameter. When k is a small value, weak edges will be preserved while the denoising capability is weak. When k is a large value, the denoising capability is strong, but weak edges and fine details will be smoothed as well.

Equation (1) was associated with the following energy function [10]

$$E(u) = \int_{\Omega} f(|\nabla u|) d\Omega \quad (4)$$

where Ω is image domain, and $f(\cdot)$ is an increasing function over $|\nabla u|$.

The key idea of P-M equation is to roughly smooth out the homogeneous regions when $|\nabla u|$ is small and to enhance the boundaries instead when $|\nabla u|$ is large.

So far, much research has been devoted to improving the P-M anisotropic

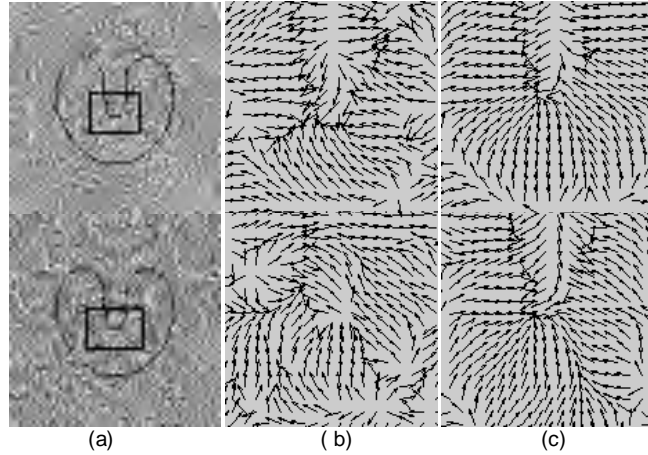


Fig.1. (a) noisy u-shape image corrupted with Gaussian noise $N(0,0.3)$ in first row and $N(0,0.4)$ in second row; (b) GVF field; (c) GVC field with $n=2, h=10$. The computing time for GVF and GVC are 0.45s and 0.048s, respectively.

diffusion method. One of the most important directions is Yu-Chua's work [21]. Although the P-M equation forms a solid foundation for other research of anisotropic diffusion models, its numerical stability still needs to be improved. Building on this concept, Yu and Chua introduced the GVF field in P-M model as follows

$$\frac{\partial u}{\partial t} = -\mathbf{V}_{GVF} \cdot \nabla u + c(|\nabla u|)\nabla^2 u \quad (5)$$

where ∇^2 is the Laplacian operator, \mathbf{V}_{GVF} is the GVF field which is proposed in [22]. The GVF-based P-M equation shows an improved performance of numerical stability when compared to original P-M equation. However, it is important to note that the (5) needs long computing time as the GVF field is calculated by iteratively solving diffusion PDEs on the whole image. Another similar method is proposed by Ghita and Whelan in [24]. In this paper, they proposed a new GVF (INGVF) which shows a better performance in the presence of impulse noise. Then, they combined the modified P-M model with adaptive median filter and gave the INGVF-based P-M model

$$\frac{\partial u}{\partial t} = (1 - IN_{Est}(u)) \left(med(u) - \frac{\partial u}{\partial (t-1)} \right) + IN_{Est}(u) \left[-\mathbf{V}_{INGVF} \cdot \nabla u + c \cdot \nabla^2 u \right] \quad (6)$$

where med denotes the adaptive median filter, IN_{Est} is the impulse estimator defined in [24] and \mathbf{V}_{INGVF} is the INGVF. The key idea of INGVF-based P-M equation is to smooth image by the first term in (6) when the image is

Image Denoising Using Anisotropic Second and Fourth Order Diffusions Based on Gradient Vector Convolution

corrupted by impulse noise and smooth image by the second term in (6) when

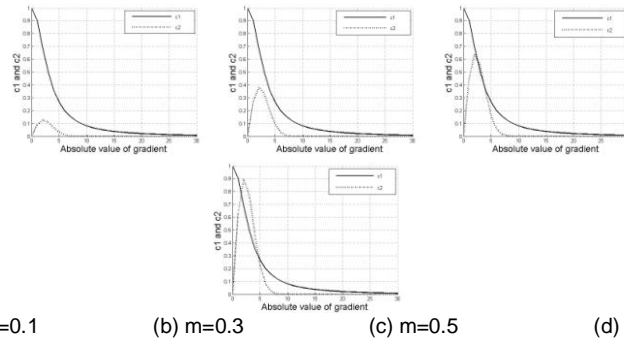


Fig.2. Diffusion functions of c_1 and c_2 with parameter m change ($k_1 = k_2 = 3$).

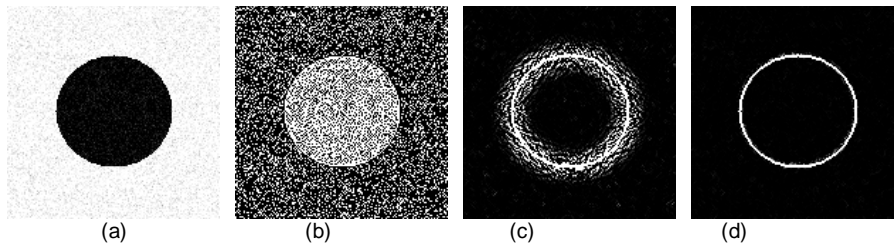


Fig.3. (a) noisy image corrupted by Gaussian noise— $N(0,20)$; (b) $u_{\eta\eta}$ calculated directly; (c) GVF-based $u_{\eta\eta}$; (d) GVC-based $u_{\eta\eta}$.

the image is corrupted by Gaussian noise. We could see that the intrinsic nature of impulse noise removal is through the median filter and the filter of Gaussian noise removal corresponding to (6) is as follow

$$\frac{\partial u}{\partial t} = -\mathbf{V}_{IN_{GVF}} \cdot \nabla u + c(|\nabla u|) \nabla^2 u \quad (7)$$

In (7), INGVF field even need a longer computing time than the GVF because it need to calculate the impulse estimator IN_{Est} besides the iteratively solving diffusion PDEs on the whole image.

2.2. Fourth Order Diffusion: Y-K model

Although the anisotropic diffusion is an effective method for image noise removal, it tends to cause staircase effect. In order to resolve this problem, You and Kaveh [27] introduced a fourth-order PDE-based denoising method in which the filtered image is obtained by minimizing the following functional

$$E(u) = \int_{\Omega} f(|\nabla^2 u|) d\Omega \quad (8)$$

where $|\nabla^2 u|$ is simply an absolute value of Laplacian of u approximated by $|u_{xx} + u_{yy}|$. By using calculus of variation, the Euler equation to minimize (8) reads

$$\frac{\partial u}{\partial t} = -\nabla^2 \left(c(|\nabla^2 u|) \nabla^2 u \right) \quad (9)$$

This fourth-order PDE favors a piecewise harmonic image, i.e., $\nabla^2 u \rightarrow 0$ locally, as $t \rightarrow \infty$. It is believed the piecewise harmonic image is a better approximate than the piecewise constant one to a natural image, and numerical results presented in [27] verified that the staircase effect is reduced and the image looks more natural.

3. GVC-based Second and Fourth Order Diffusions

3.1. GVC: Gradient Vector Convolution [29]

Gradient vector convolution (GVC) field was presented as an external force for active contour. The GVC field is motivated by gradient vector flow (GVF) field and possesses all advantages of GVF, such as enlarged capture range, initialization insensitivity and high performance on concavity convergence. However, the GVC can outperform GVF in term of computational time because the GVF is constructed by iteratively solving diffusion PDEs on the whole image while the GVC is implemented by convolving the gradient vector with a certain kernel. The GVC field $V=(u(x,y), v(x,y))$ is the solution of the following equation

$$\begin{cases} u(x, y) = f_x(x, y) * K(x, y) \\ v(x, y) = f_y(x, y) * K(x, y) \end{cases} \quad (10)$$

where $*$ denotes convolution operation, $f(x,y)$ is the edge map of an image, (f_x, f_y) is the gradient vector of image edge map, $K(x,y)$ is the convolution kernel and in practice $K(x, y) = 1/r_h^n$, $h, n \in R^+$, $r_h = \sqrt{x^2 + y^2 + h}$. The factor h plays a role analogous to scale space filtering. The GVC field has superior robustness on Gaussian noisy images. In addition, it can be implemented in real time owing to its convolution mechanism. Fig. 1 illustrates the excellent performance of GVC fields on a noisy image comparing with the GVF field. In this section, the GVF field parameters are $\mu = 0.2$, and 80 iterations, the GVC field parameters are $n=2$, $h=10$, the kernel size is the same as that of the image (image size is 64×64).

3.2. GVC-Based P-M Model

In this section, we introduce the GVC-based P-M model for image denoising.

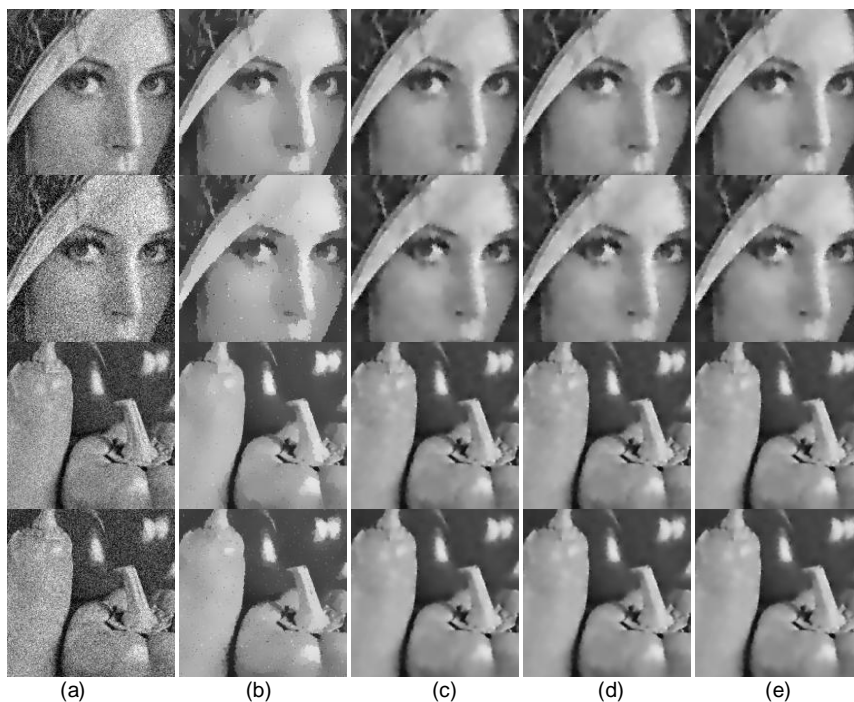


Fig.4. (a) Noisy images, results by (b) P-M equation, (c) GVF-based P-M equation, (d) INGVF-based P-M equation, (e) GVC-based P-M equation. The noise variance is 20 in the first and third rows and 30 the second and fourth rows, respectively.

Here we shall briefly mention that the two advantages of the proposed GVC-based P-M model: 1) it can improve the denoising effect; and 2) the computing time is reduced due to convolution mechanism of GVC.

As shown in Fig.1, the GVC can be implemented in real time and is more robust to noise when compared with the GVF model. As a result, it is natural to introduce the GVC into the P-M models to propose an improvement of the GVF-based anisotropic diffusion [21] and the INGVF-based one [24]. The P-M model is first expanded as

$$\frac{\partial u}{\partial t} = c(|\nabla u|)\nabla^2 u + c'\nabla|\nabla u|\cdot\nabla u \quad (11)$$

It has been pointed out in [21] that the second term is an inverse diffusion term for sharpening the boundaries while the first term is a Laplacian term for smoothing the regions that are relatively flat. Using the term $-\mathbf{V}\cdot\nabla u$ to approximate the inverse diffusion term in Eq.(11), we get the GVC-based diffusion as follows,

$$\frac{\partial u}{\partial t} = -\mathbf{V} \cdot \nabla u + c(|\nabla u|)\nabla^2 u \tag{12}$$

where \mathbf{V} denotes the GVC field. In (12), \mathbf{V} have been determined before image evolution, only ∇u needs to be computed directly from the observed images in the inverse diffusion term. The proposed model possesses all advantages of GVF-based P-M equation, such as robust estimation of the high-order derivative and improvement of numerical stability. Moreover, the GVC-based diffusion will benefit from the computational cost and noise robustness of the GVC model.

Table 1. Quantitative Comparison of the Results of Fig.4

Image(σ)	Method	PSNR	MSSIM	Iterations	Time(s) of GVF/INGVF/GVC	
Lena	20	P-M model [9]	29.28	0.78	175	
		GVF-based [21]	29.61	0.81	25	35.81
		INGVF-based [24]	29.61	0.81	25	56.49
		GVC-based(12)	30.18	0.82	25	1.96
	30	P-M model [9]	27.06	0.71	380	
		GVF-based [21]	28.33	0.78	45	35.81
		INGVF-based [24]	28.33	0.78	45	56.49
		GVC-based(12)	28.98	0.80	45	1.96
Pepper	20	P-M model [9]	29.44	0.70	175	
		GVF-based [21]	29.84	0.76	20	35.81
		INGVF-based [24]	29.84	0.76	20	56.49
		GVC-based(12)	29.90	0.78	20	1.96
	30	P-M model [9]	27.05	0.62	370	
		GVF-based [21]	27.74	0.73	40	35.81
		INGVF-based [24]	27.74	0.73	40	56.49
		GVC-based(12)	27.81	0.75	40	1.96

3.3. GVC-Based Y-K Model

The Y-K model can alleviate effectively the staircase effect, but it is isotropic and performs poor on preserving edge and texture. In this section we introduce the GVC field in the Y-K model so that it becomes an anisotropic diffusion model. To this purpose, we modify the Y-K equation as

$$\frac{\partial u}{\partial t} = -\nabla^2 (c_1 \Delta u - c_2 u_{\eta\eta}) \tag{13}$$

where η is the direction of gradient, c_1 and c_2 are the diffusion coefficients.

3.3.1. The Choice of Diffusion Coefficients

Some important analysis about the diffusion coefficients need to be highlighted. Obviously, if $c_2 = 0$, Eq.(13) becomes the isotropic one (9). If $c_1 < c_2$, the overall diffusion on the gradient direction will become backward, which damages the image feature. If $c_1 = c_2$, the diffusion on the gradient direction will disappear both in intra-region and in inter-region. In light of the above analysis, we make $c_1 > c_2$ so that the overall diffusion along the gradient direction is smaller than the one along the direction of level set.

In addition to let $c_1 > c_2$, we need to make the coefficient function c_2 close to zero for $|\nabla u| \rightarrow 0$, while c_2 close to c_1 for $|\nabla u| \rightarrow \infty$. So that the diffusion on the direction of the gradient is large in intra-region and it is small in inter-region. Equation (3) multiplied with $|\nabla u|$ can meet the above demand. So in our implementation, the diffusion functions c_1 and c_2 take the following forms

$$c_1(|\nabla u|) = \frac{1}{1 + (|\nabla u|/k_1)^2} \quad (14)$$

$$c_2(|\nabla u|) = m \cdot |\nabla u| \cdot \exp\left(-(|\nabla u|/k_2)^2\right), 0 < m \leq 1 \quad (15)$$

One can adapt the parameters k_1 , k_2 and m according to the smoothing effect. When parameter m increasing, the value of c_2 will be larger than c_1 in some region as in Fig.2, which damages the image feature. Therefore, we need set a suitable value for m so that $c_2 < c_1$ in the entire image region. In this paper, we make $m = 0.3$. This is illustrated in Fig.2.

3.3.2. The Substitute of Second-order Derivative $u_{\eta\eta}$

It is important to note that the second-order derivative $u_{\eta\eta}$ in (13) is computed from the image at each iteration. Since high-order derivative is sensitive to noise, the numerical stability of $u_{\eta\eta}$ will be reduced owing to the influence of the noise. In this paper we will replace $u_{\eta\eta}$ with $\mathbf{V}_{GVC} \cdot \mathbf{N}$, $\mathbf{N} = \nabla u / |\nabla u|$. We illustrate the improved performance of substitution of $\mathbf{V}_{GVC} \cdot \mathbf{N}$ for $u_{\eta\eta}$ and $\mathbf{V}_{GVF} \cdot \mathbf{N}$ on a synthetic image in Fig. 3. In this section the GVF field is calculated with $\mu = 0.2$ with 80 iterations and the GVC field parameters are $n=3$, $h=20$, the kernel size is 32×32 (image size 128×128). As can be seen from Figs. 3 (b)-(d), $u_{\eta\eta}$ and GVF-based $u_{\eta\eta}$ are seriously

affected by noise while the GVC-based $u_{\eta\eta}$ is near perfect and most of noise points are effectively overcome. As $\mathbf{V}_{GVC} \cdot \mathbf{N}$ can effectively approximate the second derivative $u_{\eta\eta}$, the scheme of (13) can be rewritten as

$$\frac{\partial u}{\partial t} = -\Delta(c_1 \Delta u - c_2 \mathbf{V}_{GVC} \cdot \mathbf{N}) \quad (16)$$

Since the GVC field has been determined before image evolution, only first derivative needs to be computed directly from the being filtered images. The proposed method not only possesses anisotropic characteristics, but also improves the numerical stability.

4. Experimental Results

We conduct several experiments to demonstrate the properties of the proposed methods. First, the proposed GVC-based P-M model is compared with the P-M [9], GVF-based P-M [21] and INGVF-based P-M [24] models. Then, the denoising effect and the ability of edge and texture preserving of the GVC-based Y-K model are evaluated on several real images. In order to evaluate the quality of the filtered images, the peak-signal-to-noise ratio (PSNR) and mean structure similarity (MSSIM) [30] are employed as objective indices.

4.1. Experimental Results: GVC-Based P-M Model

In order to demonstrate the desirable properties of the proposed second order model, the Lena and Pepper images are employed as test images, the dimension of both images are 512×512 . Noisy images are coined by adding zero-mean Gaussian noise of various standard deviations. The time step for all the second-order models is 0.2, and the diffusion function $c(|\nabla u|)$ takes the form in Eq.(2) with $k = 3$. The factor μ is 0.2 and the iteration number is 100 in all GVF field. The GVC field parameters are as follows: $h = 10$ and $n = 2$, the kernel size is 64×64 . Fig.4 shows the results. It is clear that the GVF, INGVF and GVC-based models successfully avoid the staircases and yield visually pleasant results. Although the results by the GVF, INGVF and GVC-based models are visually comparable, the PSNR and MSSIM indices in Table 1 manifest that the GVC-based model outperforms the other models. In this Figure, only a part of the images are shown for the sake of clarity.

4.2. Experimental Results: GVC-Based Y-K Model

4.2.1. Test of Denoising

In order to demonstrate the desirable properties of the proposed model in Eq.(16), we first use the *Lena* (512×512) image corrupted with different noise level to examine the performance of the proposed fourth order filter. The *Lena* image is corrupted by zero-mean Gaussian noise of deviation 15 and 30. In addition to the classical fourth order diffusion method Y-K model [27] and LLT model [28], the results are also compared with that of the non-local mean



Fig.5. (a) original *Lena* image; (b) noisy image: $N(0,15)$; (c) Y-K [27], Time=152.04(s); (d) LLT [28], Time=10.20(s); (e) non-local mean [4]; (f) LARK [2], Time=570.09(s); (g) BM3D [8], Time=6.99(s); (h) proposed GVC-based Y-K, Time=120.27(s).

(NLM) [4], LARK [2], and BM3D [8]. In all the following experiments, the

parameters of the proposed model are: $m = 0.3$, $k_1 = k_2 = 3$, $\Delta t = 0.03$, and $h = 10$, $n = 2$ for GVC and the kernel size is a quarter of the size of the test images. The time step for Y-K and LLT models is $\Delta t = 0.2$. The results of the NLM are yield from the IPOL¹ website, so the parameters for the NLM are



Fig.6. (a) original Lena image; (b) noisy image: $N(0,30)$; (c) Y-K [27], Time=654.21(s); (d) LLT [28], Time=26.73(s); (e) non-local mean [4]; (f) LARK [2], Time=1419.20(s); (g) BM3D[8], Time=7.04s; (h) proposed GVC-based Y-K, Time=450.28s.

default. We adopt the software packages of the LARK² and BM3D³ and the parameters for both models are unchanged as in the package. In addition to the PSNR, we also compared the computing time of these models except NLM method because the results of the NLM are yield from website and we cannot get an accurate computational time. The visual results of applying different

¹ http://www.ipol.im/pub/algo/bcm_non_local_means_denoising/

² <http://users.soe.ucsc.edu/~milanfar/software/>

³ http://www.cs.tut.fi/~foi/GCF-BM3D/index.html#ref_papers

enhancement techniques on the Lena image with different noise levels are shown in Figs.5-6. One can visualize from the Figs.5-6 that the proposed method performs better in terms of denoising effect than the other fourth order models and NLM method, though less effect than the LARK and BM3D models. Besides, the computing time of GVC-based YK model is shorter than the Y-K and LARK model. This set of experiments show that the GVC-based Y-K model performs better than Y-K, LLT and NLM models for denoising effect and shorter than Y-K and LARK models for computing time. The BM3D is the best model not only in denosing effect but also in computing time. However, we focus on diffusion-based methods in this paper, so we only compare the diffusion-based models in the following experiments.

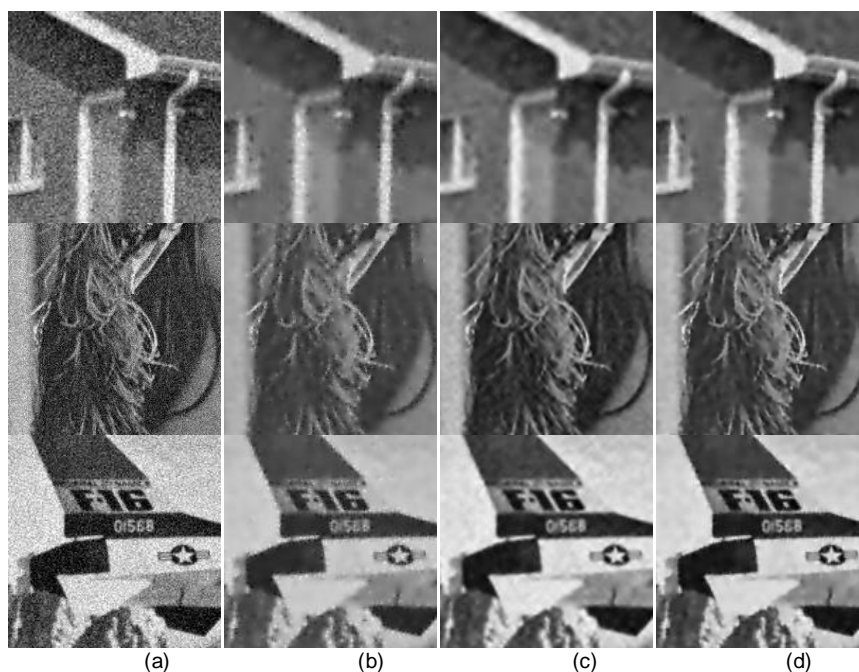


Fig.7. (a) noisy images with zero-mean Gaussian noise of deviation 20, results by (c) Y-K model[20], (d) LLT model[21], and (e) proposed fourth order model. The House, Lena and airplane images are shown in from the first to the third rows, respectively, only a part of the images are shown for the sake of clarity.

The previous two experiments are conducted for various noise levels. The next set of experiments is conducted for different images with same noise level. The *House* (256×256), *Lena* (512×512) and *Airplane* (512×512) images are employed as test images. As we can see in Figs.5-6, the methods of LARK and BM3D perform better than the diffusion model. So the results in this experiment are compared only with that of the diffusion method: Y-K model [27] and LLT model [28]. The three images are corrupted by zero-mean Gaussian noise of deviation 20. The algorithms terminated when the PSNR of

the filtered images reaches maximum.

Table 2. Quantitative Comparison of the Results of Fig.7

Image	Method	PSNR(dB)	MSSIM	Iterations
House	Y-K [27]	28.83	0.76	588
	LLT [28]	29.37	0.78	51
	GVC-based(16)	30.46	0.82	246
Lena	Y-K [27]	29.92	0.81	676
	LLT [28]	30.95	0.82	67
	GVC-based(16)	31.15	0.84	280
Airplane	Y-K [27]	28.85	0.81	567
	LLT [28]	29.71	0.82	51
	GVC-based(16)	30.94	0.87	279

The PSNR and MSSIM indices are also employed as objective index to evaluate the quality of the filtered images. Fig.7 shows the results, only a part of the images are shown for the sake of clarity. It is clear that the Y-K model suffers from speckle noise and the LLT and proposed models avoid the speckle noise successfully. Meanwhile, the Y-K and LLT models over-smooth the edges and the proposed model preserves edges much better, see the digital number '568' in the airplane image. The PSNR and MSSIM indices shown in Table 2 also manifest the high performance of the proposed model.

4.2.2. Test of Edge and Texture Preserving

In the following experiments, we illustrate the performance of the proposed model on preserving edge and texture. To evaluate the quality of the denoised image, we employ the intensity residual which is given by

$$u_{residual} = u_{denoised} - u_{noise} \tag{17}$$

where $u_{denoised}$ denotes the denoised images and u_{noise} denotes noisy image. If there is less texture information in residual, this method is considered to be better. We use two images with large amount of texture information: Barbara and Room as test images.

Figs.8-9 show the experimental results on Barbara and Room, respectively. From the intensity profiles of the selected lines, see Figs.8-9 (e), it is clear that the results by the proposed model are smoother and preserves the step edges better than the Y-K model and LLT model. From the intensity residuals, see Figs.8-9 (f)-(h), one can conclude that the proposed model largely reduced the texture information in the residuals, which means the proposed model can preserve texture much better than the Y-K model and LLT model. The experimental results indicate that the GVC-based fourth order model shows an improved performance on preserving edge and texture compared with the original Y-K model and LLT model.

Image Denoising Using Anisotropic Second and Fourth Order Diffusions Based on Gradient Vector Convolution

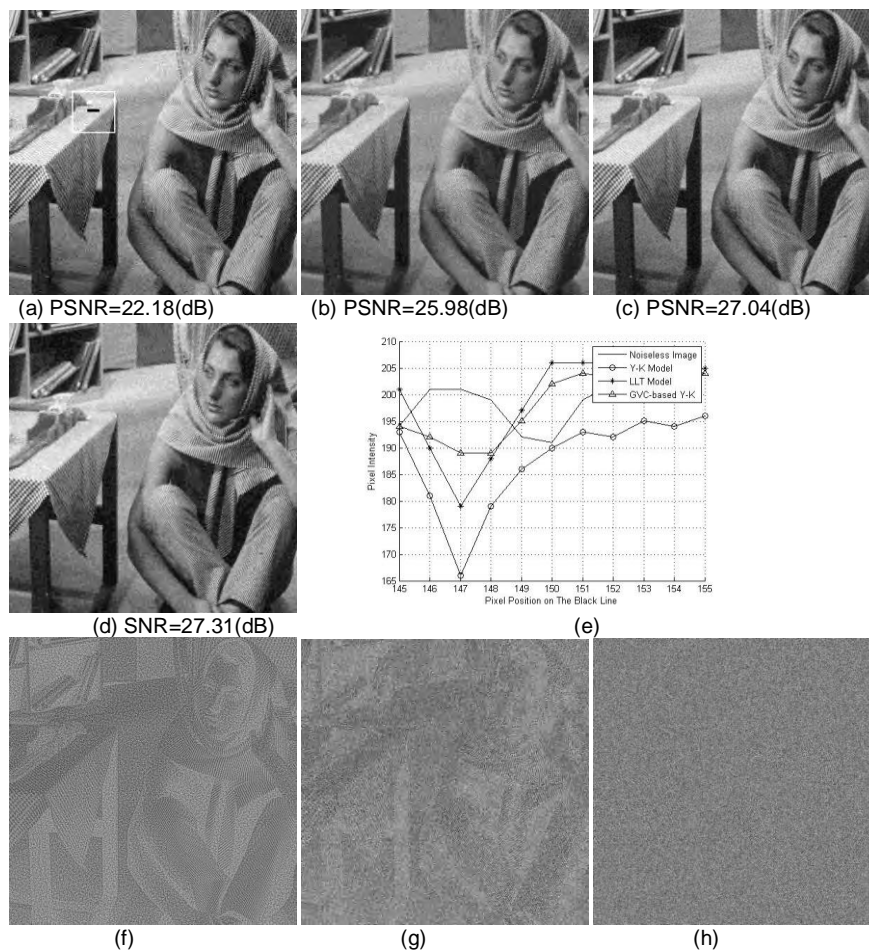


Fig.8. (a) Barbara image corrupted with Gaussian noise— $\mathcal{N}(0,20)$; (b) Y-K model; (c) LLT model; (d) GVC-based Y-K model; (e) Pixel intensity profiles for the selected black line in image (a); (f), (g), (h) The residuals corresponding to (b) (c) and (d), respectively.

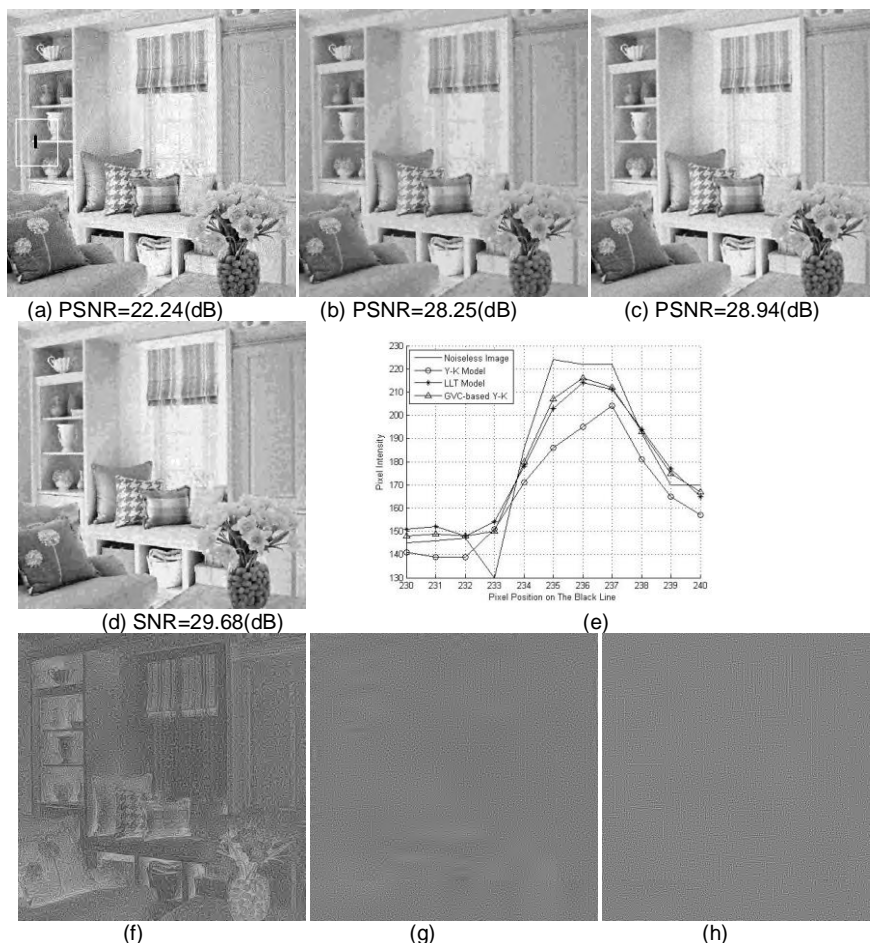


Fig.9. (a) Barbara image corrupted with Gaussian noise— $\mathcal{N}(0,20)$; (b) Y-K model; (c) LLT model; (d) GVC-based Y-K model; (e) Pixel intensity profiles for the selected black line in image (a); (f), (g), (h) The residuals corresponding to (b), (c) and (d), respectively.

5. Conclusion

In this paper, we proposed two novel diffusion models based on the GVC field. The first one is the GVC-based second order anisotropic diffusion and the second one is the GVC-based fourth order anisotropic diffusion model. Since the GVC field is very robust to noisy and it can be implemented in real time owing to its convolution mechanism, the proposed GVC-based anisotropic diffusions benefit much from the GVC models. These two proposed methods have many advantages over the existing models, such as better denoising

effects and better edge and texture preservation. Although the ability of denoising of the proposed GVC-based fourth order method is weak than the recent non-diffusion based LARK and BM3D, it is the best filter in the diffusion methods. We have conducted many experiments on different images with different noise levels. All of these qualitative and quantitative advantages of our proposed methods mean that both models can provide better image processing tools which enables noise removal, edge-preserving and staircase suppression.

Acknowledgement. This work was supported in part by the NSFC(60602050), the Key Project from the Tianjin Commission of Science and Technology(11JCZDJC15600, 10JCYBJC11000), and the Science and Technology Supported Key Project of Tianjin(11ZCKFGX00900, 10DZDZX00400)

References

1. Caselles V., Morel J. M., Sapiro G.: Tannenbaum A.: Introduction to the special issue on partial differential equations and Geometry-driven diffusion on image processing and analysis. *IEEE Trans. on Image Processing*, vol.7, no.3, 269-273. (1998)
2. Takeda H., Farsiu S., Milanfar P.: Kernel Regression for Image Processing and Reconstruction. *IEEE Trans. on Image Processing*, vol. 16, No. 2, 349-366. (2007)
3. Tomasi C., Manduchi R.: Bilateral filtering for gray and color images. *Proc. of the IEEE International Conference of Compute Vision*, 836–846. (1998)
4. Buades A., Coll B., Morel J.M.: A non-local algorithm for image denoising. *IEEE Computer Vision and Pattern Recognition*, vol. 2, 60-65. (2005)
5. Chatterjee P., Milanfar P.: Patch-Based Near-Optimal Image Denoising. *IEEE Trans. on Image Processing*, vol. 21, no. 4, 1635-1649. (2012)
6. Wang XY, Yang HY, Fu ZK: A New Wavelet-based image denoising using undecimated discrete wavelet transform and least squares support vector machine. *Expert System with Applications*, vol. 37, 7040-7049. (2010)
7. Fathi A., Naghsh -Nilchi AR.: An Efficient Image Denoising Method Based on a New Adaptive Wavelet Packet Thresholding Function. *IEEE Trans. on Image Processing*, vol. 21, no. 9, 3981-3990. (2012)
8. Dabov K., Foi A., Katkovnik V., Egiazarian K.: Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Trans. on Image Processing*, vol. 16, no. 8, 2080-2095. (2007)
9. Perona A. Malik J.: Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.12, no.7, 629 – 639. (1990)
10. You Y. L., Xu W. Tannenbaum A. Kaveh M.: Behavioral analysis of anisotropic diffusion in image processing, *IEEE Trans. on Image Processing*, vol.5, no.11, 1539-1553. (1996)
11. Scherzer O. Weickert J.: Relations between regularization and diffusion filtering, *Journal of Mathematical Image and Vision*, vol.12, no.1, 43-63. (2000)
12. Catte F., Lions P. L., Morel J.M., Coll T.: Image selective smoothing and edge detection by nonlinear diffusion, *SIAM Journal on Numerical Analysis*, vol.29, no.1, 182–193. (1992)

13. Alvarez L., Lions P. L., Morel J.M.: Image selective smoothing and edge detection by nonlinear diffusion. II, *SIAM Journal on Numerical Analysis*, vol.29, no.3. 845–846. (1992)
14. Osher S. J., Rudin L. I.: Feature-oriented image enhancement using shock filters, *SIAM Journal of Numerical Analysis*, vol.27, no.4, 919–940. (1990)
15. Rudin L. I., Osher S., Fatemi E.: Nonlinear total variation based noise removal algorithms, *Physical D*, vol.60, 259–268, (1992)
16. Alvarez L., Mazora L.: Signal and image restoration using shock filters and anisotropic diffusion, *SIAM Journal of Numerical Analysis*, vol.31, no.2, 590-605. (1994)
17. Gilboa G., Sochen N., Zeevi Y.: Forward-and-backward diffusion processes for adaptive image enhancement and denoising, *IEEE Trans. on Image Processing*, vol.11, no.7, 689-703. (2002)
18. Gilboa G., Sochen N., Zeevi Y. Y.: Image enhancement and denoising by complex diffusion processes, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.26, no.8, 1020-1036. (2004)
19. Black M. J., Sapiro G., Marimont D.H., Heeger D.: Robust anisotropic diffusion, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.7, no.3, 421 – 432. (1998)
20. Gilboa G., Zeevi Y.Y., Sochen N.: Texture preserving variational denoising using an adaptive fidelity term, *Proceedings of the VLISM, France*, 137–144. (2003)
21. Yu H., Chua C.S.: GVF-based anisotropic diffusion models, *IEEE Trans. on Image Processing*, vol.15, no.6, 1517–1524. (2006)
22. Xu C., Prince J. L.: Snakes, shapes, and gradient vector flow, *IEEE Trans. on Image Processing*, vol.7, no.3, 359-69. (1998)
23. Kass M., Witkin A., Terzopoulos D.: Snakes: active contour models, *International Journal of Computer Vision*, vol.1, no.4, 321-331. (1987)
24. Ghita O., Whelan P. F.: A new GVF-based image enhancement formulation for use in the presence of mixed noise, *Pattern Recognition*, vol.43, 2646-2658. (2010)
25. Sum A. K. W., Cheung P. Y. S.: Stabilized anisotropic diffusion, *IEEE International Conf. on Acoustics, Speech and Signal Processing*, vol.1, 709-712. (2007)
26. Wei G. W., Marimont D. H., Heeger D.: Generalized Perona-Malik Equation for Image Restoration, *IEEE Signal Processing Letters*, vol.6, no.7, 165–167. (1999)
27. You Y. L., Kaveh M., Fourth-order partial differential equations for noise removal, *IEEE Trans. on Image Processing*, vol.9, no.10, 1723–1730. (2000)
28. Lysaker M., Lundervold A., Tai X. C.: Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time, *IEEE Trans. on Image Processing*, vol.12, no.12, 1579-1590. (2003)
29. Wang Y., Jia Y.: External force for active contours: gradient vector convolution, *Pacific Rim International Conference on Artificial Intelligence*, 466-472. (2008)
30. Wang Z., Bovik A. C., Sheikh H.R.: Image quality assessment: from error visibility to structural similarity, *IEEE Trans. on Image Processing*, vol. 13, no.4, 1-14. (2004)
31. Krissian K., Westin C.F., Kikinis R., Vosburgh K.: Oriented speckle reducing anisotropic diffusion, *IEEE Trans. on Image Processing*, vol.16, no.5, 1412-1424. (2007)
32. Carmona R.A., Zhong S.: Adaptive smoothing respecting feature denoising, *IEEE Trans. on Image Processing*, vol.7, no.3, 353-358. (1998)
33. Lysaker M., Tai X. C.: Iterative image restoration combining total variation minimization and a second-order functional, *International Journal of Computer Vision*, vol.66, no.1, 5-18. (2006)

34. Blomgren P., Chan T.F., Mulet P.: Extensions to total variation denoising, Proceedings of SPIE, San Diego, vol. 3162, (1997)

Huaibin Wang is a professor and the vice dean of the School of Computer Science and Communication, Tianjin University of Technology, where he first joined by the end of 1988 and promoted to be a professor in 2000. He got his Bachelor degree at Jilin University in 1982 and completed his Master thesis in 1988 at Harbin Institute of Technology. During 1982-1986, he worked at China Agricultural Machinery Design and Research Institute. Prof. Wang has published more than one hundred papers in reputable journals and conference proceedings, covering the areas of computer vision, image processing and network security. Prof Wang has received many Chinese national prestigious academic awards, and has also been serving as an expert of information science panel, Tianjin committee of science and technology, China.

Yuanquan Wang is a professor of Computer Science and Communication, Tianjin University of Technology, where he was recruited as a Talent Professor in 2006, became yet a professor in 2012. He completed his Master and Ph.D. thesis at Nanjing University of Science and Technology (NUST) in 1998 and 2004, respectively, then joined the Beijing Institute of Technology (BIT) in 2004 as a postdoc. He won the most outstanding postgraduate award from NUST in 1998. During 1998-2001, he worked as a software engineer in industry. Prof. Wang has published dozens of papers in reputable journals and conference proceedings, covering the areas of Computer Vision, Image Processing and Medical Image Analysis, and his papers have gotten over 200 citations, including the citation by Nature Method journal.

Wenqi Ren received the B.S. degrees from the School of Information Engineering, Hebei University of Technology, in 2010. He is currently pursuing the Master degree in School of Computer Science and Communication Engineering, Tianjin University of Technology, working closely with Prof. Y. Wang and H. Wang. His current research interests is image processing focusing on PDEs and sparse representation.

Received: February 19, 2012; Accepted: Decmber 06, 2012.

Active Semi-supervised Framework with Data Editing

Xue Zhang^{1,2} and Wangxin Xiao^{2,3}

¹ Key Laboratory of High Confidence Software Technologies, Ministry of Education, Peking University, Beijing 100871, China

¹School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

² Department of Computer Science, Jingtangshan University, Ji'an 343009, China
jane_zhang@pku.edu.cn

³School of Traffic and Transportation Engineering, Changsha University of Science and Technology, Changsha 410114, China
wx.xiao@rioh.cn

Abstract. In order to address the insufficient training data problem, many active semi-supervised algorithms have been proposed. The self-labeled training data in semi-supervised learning may contain much noise due to the insufficient training data. Such noise may snowball themselves in the following learning process and thus hurt the generalization ability of the final hypothesis. Extremely few labeled training data in sparsely labeled text classification aggravate such situation. If such noise could be identified and removed by some strategy, the performance of the active semi-supervised algorithms should be improved. However, such useful techniques of identifying and removing noise have been seldom explored in existing active semi-supervised algorithms. In this paper, we propose an active semi-supervised framework with data editing (we call it ASSDE) to improve sparsely labeled text classification. A data editing technique is used to identify and remove noise introduced by semi-supervised labeling. We carry out the data editing technique by fully utilizing the advantage of active learning, which is novel according to our knowledge. The fusion of active learning with data editing makes ASSDE more robust to the sparsity and the distribution bias of the training data. It further simplifies the design of semi-supervised learning which makes ASSDE more efficient. Extensive experimental study on several real-world text data sets shows the encouraging results of the proposed framework for sparsely labeled text classification, compared with several state-of-the-art methods.

Keywords: sparsely labeled text classification; active learning; semi-supervised learning; data editing

1. Introduction

Automatic text classification is of great importance due to the large volume of text documents in many real-world applications. The goal of automatic text classification is to automatically assign documents to a number of predefined categories. A supervised classification model often needs a very large number of training data to enable the classifier's reliable performance. As we know, manually labeling the training data for a machine learning algorithm is a tedious and time-consuming process, and even unpractical (e.g., online web-page recommendation). Correspondingly, one important challenge for automatic text classification is how to reduce the number of labeled documents that are required for building a reliable text classifier.

In order to reduce the effort involved in acquiring labeled examples, there are two major strategies, active learning and semi-supervised learning. The aim of active learning is to select most informative unlabeled examples for manually labeling so that a good classifier can be learned with significantly fewer labeled examples. Active learning has been extensively studied in machine learning for many years and has already been employed for text classification in the past [1-3]. Semi-supervised learning tries to learn a classification model from the mixture of labeled and unlabeled instances, which also has been employed for text classification [4-5]. The fusion of active learning with semi-supervised learning can further bring advantage, thus several combination algorithms have been proposed for text classification [6-7].

Sparsely labeled classification is a special form of classification in which only very few labeled instances are available. It exists in many real-world applications such as content-based image retrieval, online web-page recommendation, object identification and text classification, where the abundant unlabeled instances are available but the labeled ones are fairly expensive to obtain. The sparsity of training data often leads to severe distribution bias between the training data and the unlabeled data (we call it training data bias). It is very difficult to learn a weak useful hypothesis with the extremely few labeled instances. Existing semi-supervised learning and active learning algorithms, which often need quite a number of labeled instances to learn an initial weak useful predictor for further learning, cannot perform well for sparsely labeled text classification [8].

Due to the poor performance of the initially learned hypothesis based on the very few training data, it is unavoidable to contain much noise in the self-labeled instances. Extremely few labeled training data in sparsely labeled text classification aggravate such situation. If such noise could not be identified and removed from the new training data set, they may snowball themselves in the following learning process and thus hurt the generalization ability of the final hypothesis. On the other hand, if such noise could be identified and removed by some strategy, the performance of the active semi-supervised algorithms should be improved. However, such useful techniques of identifying and removing noise have been seldom explored in existing active semi-supervised algorithms, especially using the advantage of active learning to do this. This is one motivation of this work.

In this paper, we propose an active semi-supervised framework with data editing (we call it ASSDE) to improve sparsely labeled text classification. ASSDE conducts in a self-training style process. In order to efficiently integrate active learning, we extend the standard self-training by substituting ensemble classifiers for its single classifier. Furthermore, we introduce a data editing technique into ASSDE by fully utilizing the advantage of active learning. Data editing technique is used to identify and remove the noise contained in self-labeled instances. ASSDE iterates the steps of self-labeling, active labeling and data editing until satisfying some stopping criteria.

The main contributions of this paper may be summarized as follows:

- We propose an active semi-supervised framework with data editing which is more effective and efficient for sparsely labeled text classification compared with state-of-the-art algorithms.
- We carry out a data editing technique to identify and remove the noise contained in self-labeled instances by fully utilizing the advantage of active learning, which is novel and incurs very little computation complexity while improving the classification accuracy.
- We propose a novel parameter ensemble strategy to extend the standard self-training algorithm in order to efficiently integrate active learning while incurring less computation complexity.
- We empirically demonstrate that data editing can simplify the design of semi-supervised learning for efficiency reason, while not degrading the performance.
- We conduct extensive experiments on three benchmark real-world text data sets to evaluate its performance with different parameters.

The rest of this paper is organized as follows. We discuss some of the related work in Section 2. Section 3 describes the algorithm in detail. Section 4 presents the results of the experiments. A short conclusion and future work are presented in Section 5.

2. Related Work

A variety of algorithms for text classification have been proposed [1-14], including supervised methods, semi-supervised methods, active methods and the combinations. In the following, we only focus on the study and techniques related to sparsely labeled text classification.

There are several techniques which are beneficial to sparsely labeled text classification. These techniques includes: 1) semi-supervised learning and active learning, 2) transfer learning [12-13], 3) feature extension with semantic concepts [14], 4) clustering aided methods [15-17], 5) data editing [18]. Semi-supervised learning and active learning are two common used techniques to address the problem of insufficient training data. However, they often need quite a number of training data to train a weak useful predictor for further learning. In sparsely labeled classification, it is very challenging, if not impossible, to generate such weak useful predictor, which makes existing

semi-supervised and active learning algorithms cannot be applied. The second technique beneficial to sparsely labeled text classification is transfer learning. It refers to the problem of retaining and applying the knowledge learned in one or more tasks to efficiently develop an effective hypothesis for a new task. Transfer learning techniques can be used to improve sparsely labeled classification by transferring the useful knowledge for the problem. The third technique beneficial to sparsely labeled text classification is to utilize the world knowledge (e.g. Wiki, WordNet). By using the world knowledge, the feature representation of an instance is enhanced by semantic concepts which can weaken the feature sparsity in some degree. Clustering aided methods beneficial to sparsely labeled classification including both expanding the training data from unlabeled data [15] and augmenting the data set with new features [17]. Another technique beneficial to sparsely labeled text classification is data editing. It can be explored to identify and remove the noise contained in self-labeled instances. In this paper, we address the problem of sparsely labeled text classification by active semi-supervised learning with data editing.

In sparsely labeled classification, the generalization ability of the hypothesis learned on the initial training data is often very poor. Thus there may contain much noise in the self-labeled instances. In self-training style algorithms, the early introduced noise by semi-supervised learning may snowball themselves, which often makes the final hypothesis of very poor performance. If such noise could be identified and removed by exploring some useful techniques, the classification accuracy should be improved. Data editing technique could be used for this end. In conventional studies, data editing aims to remove noisy instances from the original training data set with the goal to improve classification accuracy by producing smooth decision boundaries. A new self-training style algorithm, SETRED, is proposed in [19] by introducing a data editing technique to the self-training process to filter out the noise in the self-labeled instances. SETRED outperforms the standard self-training, which indicates that the performance of semi-supervised learning can be further improved by introducing proper data editing technique. This paper shows that it is encouraging to fuse active learning with data editing to identify and remove the noise in the self-labeled instances. It incurs very little computation complexity while improving the classification accuracy.

Although sparsely labeled text classification is a very significant problem in many real-world applications, there have been very limited researches on it. A clustering based classification method, CBC [15], is one such work. It combines transductive support vector machines (TSVM) with k-means and iterates these two steps alternatively. Although clustering can help to overcome the sparsity and the training data bias, CBC has a very high computation complexity because both TSVM and k-means are very time-consuming for sparsely labeled high dimensional text classification. Based on kernel canonical component analysis, OLV (learning with One Labeled example and Two Views) [20] and ALESLE (Active Learning with Extremely Sparse Labeled Examples) [21] algorithms have been proposed for sparsely labeled classification. While OLV works in semi-supervised setting, ALESLE works in

active setting. OLTV and ALESLE require two sufficient views, which is not practical for many real-world applications.

Phan et al. 2008 [22] propose a general framework for building classifiers that deal with short and sparse text & Web segments by making the most of hidden topics discovered from large-scale data collections. This framework can be viewed as a semi-supervised learning technique, but it is flexible in that the universal data are not necessary to have the same format as the labeled training or future unseen data. Cai et al. 2003 [23] propose a framework for text categorization which attempts to analyze topics from both training and test data using probabilistic latent semantic analysis (PLSA) and uses both the original data and resulting topics to train two different weak classifiers for boosting. Xu et al 2008 [24] propose a web-assisted text categorization framework which automatically identifies important keywords from the available labeled documents to form the queries and then uses search engines to retrieve from the Web a multitude of relevant documents. These retrieved documents are then exploited by a semi-supervised framework.

3. ASSDE Framework

3.1. Problem Description and Notation

Let $D=L \cup U$ denote the set of instances in a p -dimensional Euclidean space R^p , where $L=\{<x_i, y_i>\}_{1 \leq i \leq l}$ is the set of labeled instances and $U=\{x_i\}_{l+1 \leq i \leq l+u}$ the set of unlabeled ones. Here y_i is the class label of instance x_i and $l < u$. This paper only considers single-label classification that exact one label should be assigned to each instance in D . The set of classes is denoted by $C=\{c_r\}_{1 \leq r \leq |C|}$ and $|C|$ is the cardinality of C and r is an integer. Each instance in L has been labeled or assigned to one class in C , while the class label of each instance in U is unknown and needs to be determined.

3.2. ASSDE Framework

The aim of this work is to improve sparsely labeled text classification by introducing data editing technique into active semi-supervised framework. As we know, one of the problems of semi-supervised learning is its learning efficiency. To make the framework more efficient, we also expect to simplify the design of semi-supervised learning by fully utilizing the advantage brought by the fusion of active learning with data editing. We show empirically that data editing can make up for the deficiency of simplifying the design of semi-supervised learning. Table 1 gives the Pseudo-code description of ASSDE.

Table 1. Pseudo-code describing ASSDE algorithm

```

Algorithm: ASSDE
Input: the labeled set  $L$ , the unlabeled set  $U$ ,  $cpNum$ ,  $Maxcplter$ ,  $n$ 
Output: the full labeled set  $D=L \cup U$ 
Progress:
  Offline:
    Computing distance matrix( $L \cup U$ )
  Online:
     $h_i \leftarrow \text{Learn}(L), i=1,2,\dots$ 
     $Iter=0$ 
     $L \leftarrow \emptyset, SL \leftarrow \emptyset, SP \leftarrow \emptyset$ 
    Repeat until no data in  $U$  can be put into  $SP$  or  $U = \emptyset$ 
      ( $SP, CP$ )= $\text{partition}(U, h_1, h_2, \dots)$ 
       $SL = \text{choose}(SP, n)$ 
       $SL = SL \cup SL$ 
       $U = U - SL$ 
      If  $Iter < maxcplter$ 
         $L = \text{BMAL}(CP, cpNum)$ 
         $L = L \cup L$ 
         $U = U - L$ 
         $h_i \leftarrow \text{Learn}(L), i=1,2,\dots$ 
         $SL = \text{recheck}(SL, h_1, h_2, \dots)$ 
         $Iter = Iter + 1$ 
      End
       $h_i \leftarrow \text{Learn}(L \cup SL), i=1,2,\dots$ 
    End
    If  $U \neq \emptyset$  then for each instance in  $U$ ,
       $c = \text{majority voting}(h_1, h_2, \dots)$ 
    End

```

ASSDE works as follows. Firstly ensemble classifiers are trained on the initial training data set L . The trained ensemble classifiers are used to predict the label of each instance in U . According to the labels predicted by the ensemble classifiers, instances in U are partitioned into two sets, that is, contention points set CP and consistent points set SP . CP contains the instances whose labels predicted by the ensemble classifiers are inconsistent, while SP consists of the instances which have consistent predicted labels. Secondly, n most confident examples, say SL , are selected from SP and labeled with their predicted labels. Then $cpNum$ instances, say L , from CP are selected by a batch mode active learning algorithm for manually labeling. Due to the small size of L , the generalization ability of the hypothesis learned by ensemble learning may be poor. Consequently, SL may contain much noise which will hurt the generalization ability of the final hypothesis with the accumulation of such noise in the following self-training processes. Therefore, we employ a data editing technique to identify and remove the noise. We use

the ensemble classifiers retrained on $L \cup L$ to predict the labels of self-labeled instances in SL . For each instance in SL , if the newly predicted label is inconsistent with its current label, then it will be removed from SL and thrown into unlabeled data set U again. Now the training data set consists of instances with ground-truth labels and self-labeled instances (denoted by SL).

ASSDE conducts in a self-training style process in which the steps of self-labeling, active labeling and data editing are iterated alternatively. After the completion of active learning, all the self-labeled instances labeled in former iterations will be rechecked again. ASSDE iterates the self-training style process until almost all the unlabeled instances are labeled with high confidence. If there is any instance in U , we use the majority voting strategy to label it.

3.3. Ensemble Strategy

Several useful ensemble techniques have been proposed, such as the well-known training data resampling [25] and input feature resampling [26]. However, we only use a simple ensemble strategy (we construct the ensemble classifiers using k-nearest neighbor (kNN) with different k parameter) in ASSDE for two main reasons. One reason is that the well-known ensemble techniques are either not suitable for sparsely labeled text classification or incurring large computation/storage complexity. Intuition and empirical experiments indicate that training data resampling technique is not suitable for sparsely labeled case, since it may further aggravate the sparsity of training data for each component classifier. Input feature resampling technique may be helpful, but it can increase the computation and storage complexity because we have to compute the nearest neighbors of an instance for each component classifier and store the corresponding distance matrix. Furthermore, our empirical experiments with FASBIR (Filtered Attribute Subspace based Bagging with Injected Randomness) [27] also indicate that ensemble learning itself can hardly address the sparsely labeled classification problem well. Based on this fact, the aim of ensemble strategy in ASSDE is to make efficient integration of active learning and the overall efficiency. This is the next reason that we use a simple ensemble strategy in ASSDE.

From table 1 we can see that the efficiency of ASSDE is mainly determined by the ensemble part. In order to improve the efficiency of ASSDE, we can simplify the design of ensemble part while not greatly degrading the overall accuracy. We use kNN as the base learner and construct the ensemble classifiers using kNN with different k parameter. The component classifiers are trained in parallel training style. Therefore it can generate the ensemble predictions by only computing the maximal k nearest neighbors for an instance and it only needs to store one distance/similarity matrix for all ensemble classifiers. Our ensemble strategy only has the similar computation and storage complexity with that of one kNN with the maximal k parameter in the component classifiers. This is the main reason that we select kNN as the base classifier and use the simple ensemble strategy. We can also compute the

distance/similarity matrix beforehand to further make the online learning and classifying procedures efficient, which make kNN an efficient base classifier for algorithms conducted in iterated mode.

3.4. Batch Mode Active Learning

The key of batch mode active learning (BMAL) is to ensure the selected instances of both informativeness and diversity. BMAL method [3] based on farthest-first traversal (we call it BMAL_FFT) is based on the intuition that for two examples, the larger the distance between them, the smaller redundancy the information they provide.

BMAL_FFT works as follows. First, it selects an instance x from CP randomly or according to its uncertainty for the learning model, and adds x to query set Q . Then it selects the next instance x_i according to equation (2) and adds x_i to Q . BMAL_FFT repeats the above selection procedure until the needed number of instances has been selected.

$$d(x_i, Q) = \min_{y \in Q} \|x_i - y\| \quad (1)$$

$$x_i = \operatorname{argmax}_{x_i \in CP} d(x_i, Q) \quad (2)$$

BMAL_FFT is a global search method which may be not efficient for very large-scale text classification problem. In this paper, BMAL_FFT selects instances from CP set, which has a much smaller search space and whose instances are more informative than the whole unlabeled data set U .

3.5. Data Editing Strategy

The sparsity of training data in sparsely labeled classification often makes the generalization ability of the initial hypothesis very poor. There may contain much noise in the self-labeled data set SL because the classifiers may incorrectly assign labels to some unlabeled instances. Such noise may accumulate in the following iterations which will hurt the generalization ability of the final hypothesis. It is obvious that if the mislabeled instances in SL could be identified and removed, especially in the early iterations, the learned hypothesis is expected to be better. This is the basis that we introduce data editing technique into the proposed framework to identify and remove the mislabeled instances.

After each active learning process, the training data with ground-truth labels increase. In general terms, the classifiers trained on the enlarged training data set will generate more accurate hypothesis. It may be helpful using this hypothesis to identify the noise contained in the self-labeled data set labeled by former less accurate hypotheses. Our data editing strategy is based on this intuition. It works as follows. After each active learning process, the ensemble classifiers are retrained on the enlarged training data set $L \cup L$. Then they are used to predict the label of each instance in SL . If any inconsistency exists

between the newly predicted label and its current label for an instance, the instance will be removed from SL and added to U again.

In ASSDE, the introduction of data editing technique provides many chances for self-labeled instances to mend their ways, which is different from the traditional semi-supervised learning. Furthermore, we carry out data editing technique by fully utilizing the advantage of active learning, which incurs very little computation complexity while improving the classification accuracy. This is novel in active semi-supervised learning community, according to our knowledge.

We think that data editing technique should be very useful especially in sparsely labeled text classification since the extremely few labeled training data make the mislabeling unavoidable. Data editing technique makes the mistakes made in earlier stages not as severe as that in traditional semi-supervised learning algorithms. Based on this fact, we can simplify the design of the module that determines the algorithm's overall efficiency, and use data editing technique to make up for the deficiency. In ASSDE, we design the ensemble strategy just based on this mind.

4. Experiments

4.1. Data Sets

For a consistent evaluation, we conduct our empirical experiments on three benchmark data sets, 20NewsGroups, Reuters-21578 and email spam filtering data set. The details of data sets are given in table 2.

20 Newsgroups is one famous Web-related data collection. From the original 20 Newsgroups data set, same-2, consisting of 2 very similar newsgroups (comp.windows.x, comp.os.ms -windows), and diff-2, consisting of 2 very different newsgroups (alt.atheism and comp.windows.x), are used to evaluate the performance of the algorithms on data sets with different separability. Same-2 and diff-2 both contain 2000 instances, 1000 for each class. We use Rainbow software¹ to preprocess the data (removing stop words and words whose document frequency less than 3, stemming) and we get 7765 and 8599 unique terms for same-2 and diff-2, respectively. Then terms are weighted with their TFIDF (Term Frequency-Inverse Document Frequency) values.

The Reuters-21578 corpus contains Reuters news articles from 1987. We only show the experimental results of train1.svm in LWE² (Locally Weighted Ensemble framework) since the algorithms have the similar performance on other Reuters data sets. Train1.svm contains 1239 documents (two class) and 6889 unique terms.

¹ <http://www.cs.cmu.edu/~mccallum/bow/>

² <http://ews.uiuc.edu/~jinggao3/kdd08transfer>

The email spam data set, released by ECML/PKDD 2006 discovery challenge, contains a training set of publicly available messages and three set of email messages from individual users as test sets. The algorithms have similar performance on different email spam data sets in LWE^2 , therefore we only show the experimental results of test1.svm for the limited space. test1.svm contains 2500 messages and 83636 unique terms.

Table 2. The details of data sets

Data sets	#classes	#total instances	#features	#training instances for each class
same-2	2	2000	7765	5
diff-2	2	2000	8599	5
Reuters	2	1239	6889	5
Spam	2	2500	83636	5

4.2. Performance Evaluation

Macro_F1 is used as the performance measurement. $F1$ metric is defined as $F1 = 2 \cdot P \cdot R / (P + R)$, where P and R are precision and recall for a particular class. $F1$ metric takes into account both the precision and the recall, thus is a more comprehensive metric than either precision or recall when separately considered.

Macro_F1 is a measurement which evaluates the overall performance of the classification model. Macro_F1 is defined in equ. (3), where F_1^i is the $F1$ value of class i .

$$\text{Macro_F1} = \frac{1}{|C|} \sum_{i=1}^{|C|} F_1^i \quad (3)$$

4.3. Experimental Results and Analysis

In ASSDE, one important parameter is n , which affects both the effectiveness and efficiency of the proposed method. $cpNum$ and $Maxcplter$ are pairwise parameters which also affect the effectiveness and efficiency of the proposed method. In the following, we mainly conduct experiments with respect to parameters n , $cpNum$ and $Maxcplter$, to study their influence on the performance of the proposed method. At the same time, we conduct experiments with several degenerated variants of ASSDE to see the contribution of each component technique. Furthermore, we compare our method with those of several state-of-the-art algorithms both in effectiveness and in efficiency.

In the following experiments, the ensemble size is 3 and we set $k=1, 3, 5$ for the three component kNN classifiers in ASSDE and its degenerated variants. We use random strategy to select the first instance in the batch mode active learning. We conduct each experiment 40 runs and the average results are

given. In each run, the algorithms perform on the same randomly chosen training data set which has 5 positive labeled instances and 5 negative labeled instances. The runtime given below for all algorithms is the average online time.

Robustness with different parameter n . Since ASSDE integrates several techniques, its two degenerated variants can be easily derived. First, if only ensemble learning and self-training are employed, then EnST (ensemble style self-training) algorithm is obtained. That is, EnST algorithm substitutes ensemble classifiers for one base classifier in standard self-training. Second, if EnST algorithm is augmented by active learning (BMAL_FFT), then AcEnST algorithm is obtained. Note that, although ASSDE integrates ensemble learning, active learning and data editing together, we can only obtain its two degenerated variants, EnST and AcEnST. This is because of the dependent relationships among the component techniques in ASSDE. From table 1 we can find that active learning is based on the ensemble learning and data editing is based on the active learning.

In this subsection, we conduct experiments for ASSDE and its degenerated variants with different parameter n in order to see the contribution of each component technique and the robustness of the algorithms with parameter n . The standard self-training algorithm (ST) is taken as the baseline and it also refers to kNN as its base classifier. Here we set $cpNum=30$, $Maxcplter=6$ for algorithms with batch mode active learning.

Figures 1, 2, 3, and 4 give the Macro_F1 performance with different parameter n for all algorithms on four data sets. For the standard self-training (ST) algorithm, we only give the performance with $k=3$ since it has similar performance with $k=3$ and $k=5$, but it performs relatively worse with $k=1$.

Figure 1 presents the Macro_F1 performance of the algorithms with different parameter n on same-2. It could be found that ASSDE significantly outperforms the other three algorithms, which verify the usefulness of the fusion of active semi-supervised learning with data editing technique. Furthermore, ASSDE is more robust than the other three algorithms to parameter n . EnST outperforms ST slightly with most parameter n , which accords with our former analysis that the simple ensemble strategy in ASSDE is designed to make efficient integration of active learning and to achieve overall efficiency while without degrading the performance. AcEnST outperforms EnST with all parameter n , which verify the usefulness of active learning. Note that, ASSDE performs better with larger values of parameter n , which makes it very efficient since larger n means lower computation complexity (less iterations).

Figure 1 also shows that the algorithms except ASSDE perform better with the increase of parameter n first, then worse with the increase of parameter n . We think this is due to the paradox that larger parameter n makes larger training data set available for retraining the classifiers in next iteration which is very useful for sparsely labeled classification especially in the earlier iterations, but larger parameter n also means that more noise may be introduced into training data set which will hurt the learning results in the following iterations. Therefore there should be a balance point on which the classification model achieves its best performance. Different algorithms achieve this balance point

with different parameter n . For example, ST achieves its balance point with about $n=250$, EnST and AcEnST with about $n=150$. It seems that data editing technique could delay such balance point (with the increase of n) by identifying and removing the noise contained in self-labeled training data.

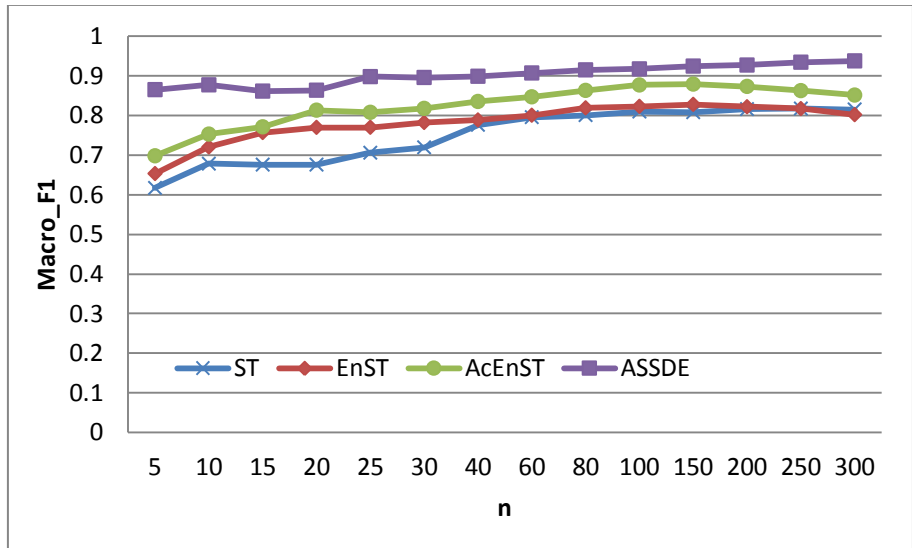


Fig.1. Macro_F1 performance with different parameter n on same-2

Figure 2 presents the Macro_F1 performance of the algorithms with different parameter n on diff-2. From figure 2 it could be found that ASSDE outperforms the other three algorithms with all parameter n and it is very robust to parameter n . EnST and ST perform similarly. AcEnST outperforms EnST and ST, and it achieves the best performance with $n=5$. For ST and EnST, there also exists the balance point phenomenon. ST achieves its best performance with about $n=80$, EnST with about $n=60$. The results in figure 2 also accords with our analysis in former sections.

Figure 3 gives the Macro_F1 performance of the algorithms with different parameter n on Reuters. ASSDE outperforms the other three algorithms with all parameter n and it is more robust to parameter n . EnST is outperformed by ST with small n , but it outperforms ST when $n>80$. AcEnST outperforms ST and EnST with all parameter n and its performance degrades with the increase of n when $n>30$. ASSDE achieves its best performance with about $n=80$, ST with about $n=15$, EnST with about $n=10$, and AcEnST with about $n=20$. From the balance point of the four algorithms, we can see that ASSDE is most efficient for its least iterations, which must benefit from data editing technique.

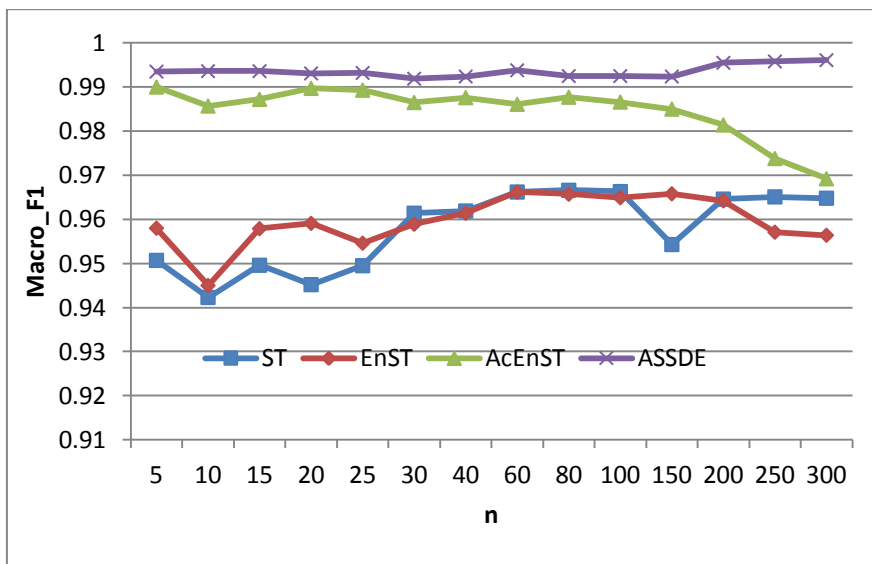


Fig.2. Macro_F1 performance with different parameter n on diff-2

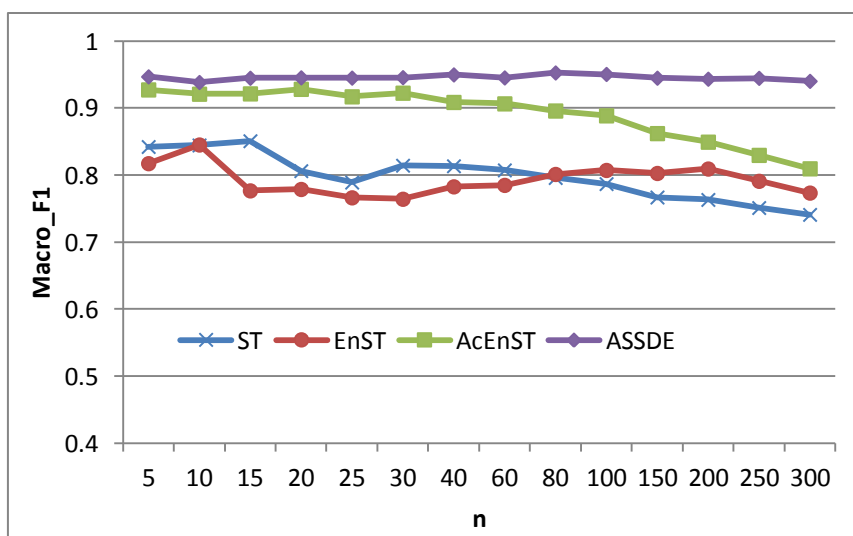


Fig.3. Macro_F1 performance with different parameter n on Reuters

Figure 4 presents the Macro_F1 performance of the algorithms with different parameter n on Spam data set. It could be found that ASSDE outperforms ST and EnST with all parameter n , and outperforms AcEnST when $n > 60$. ASSDE ties with AcEnST when $n < 60$, but it is more robust with parameter n than other

three algorithms. EnST performs slightly better than ST with almost all parameter n . AcEnST significantly outperforms ST and EnST with all parameter n . It seems that active learning plays a leading role for performance improvement in ASSDE on Spam data set. ASSDE achieves its best performance with about $n=200$, AcEnST with about $n=25$ or $n=40$, EnST with about $n=80$, and ST with about $n=100$.

In general, from figures 1 to 4, we can conclude that active learning in ASSDE is always beneficial to improve the overall performance, and that data editing technique plays a key role in the robustness and improving the performance for ASSDE. Furthermore, data editing technique makes ASSDE more efficient since it can make ASSDE achieve its best performance with larger values of n (correspondingly less iterations). It seems that data editing technique brings larger advantage for data set of lower separability (e.g. same-2). This may be due to the fact that there should be more noise in self-labeled instances for data set of lower separability. Thus data editing technique can greatly improve the overall performance by identify and remove such noise contained in self-labeled training data. This accords with our intuition.

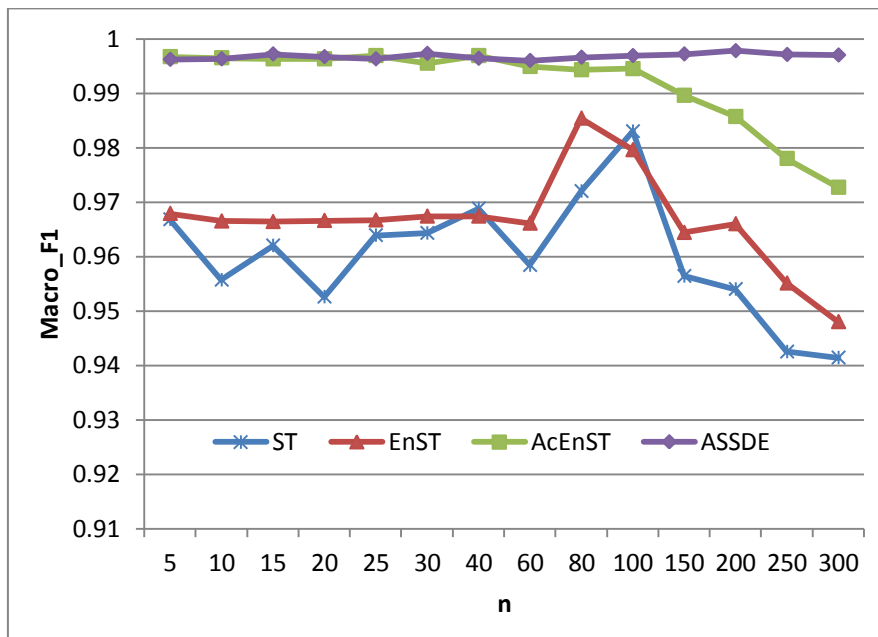


Fig.4. Macro_F1 performance with different parameter n on Spam

Robustness with $cpNum$ & $Maxcplter$. Parameters of $cpNum$ and $Maxcplter$ determine the amount of manual effort involved. In general, for an active learning algorithm, the more the manual effort involved, the better the performance it achieves. The aim of active learning is to reduce the effort involved without degrading the performance. In ASSDE, the performance will be improved with the increase of the product of $cpNum$ and $Maxcplter$, which is more intuitive. Therefore we only conduct experiments in the case of $cpNum * Maxcplter = \text{constant}$, while $cpNum$ and $Maxcplter$ may take different values.

The following experiments are also conducted on the basic training data set which contains 5 training data for each class and is sampled at random in each run. We test the performance of ASSDE with $cpNum = 10, 20, 30, 60$ and 90 , correspondingly $Maxcplter = 18, 9, 6, 3$, and 2 . Since the iterations of ASSDE are determined by parameter n , the iterations may be less than $Maxcplter$ with larger value of parameter n . That is, in the following experiments, ASSDE can at most actively label 180 instances in each run. We set $n = 300$.

Figure 5 shows the performance of ASSDE with $cpNum$ on four data sets. In the case of $cpNum * Maxcplter = \text{constant}$, the performance declines slightly with the increase of $cpNum$, and at the same time the runtime also declines (please see figure 6). We think this is because that the redundancy among the actively selected instances increases with the increase of $cpNum$. In general, $cpNum = 30$ (correspondingly $Maxcplter = 6$) is best for ASSDE when considering both the effectiveness and the efficiency on the four data sets.

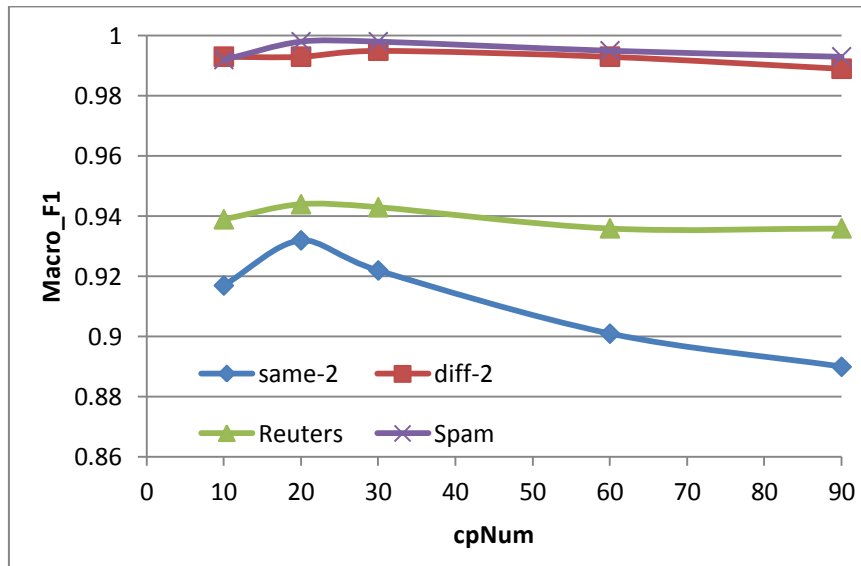


Fig.5. Performance with $cpNum$ on four data sets

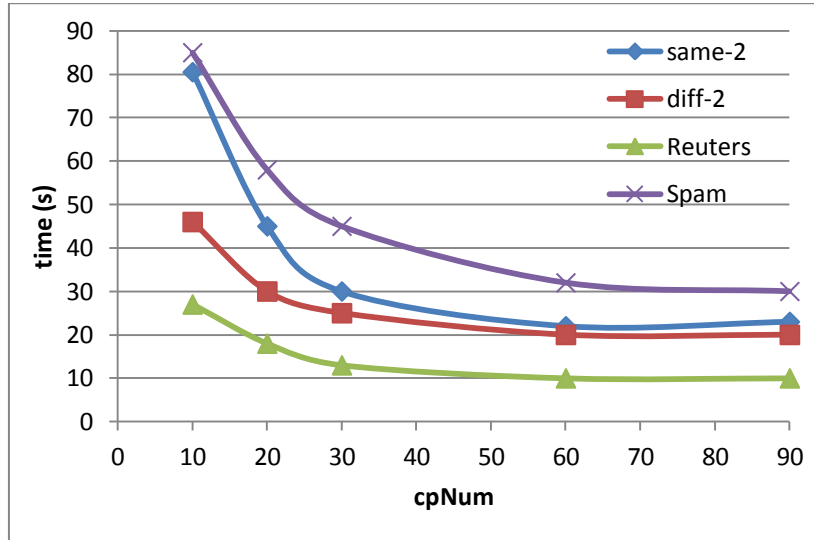


Fig.6. Runtime with cpNum on four data sets

Performance Comparison. To evaluate the performance of ASSDE with the increase of labeled training data, we conduct the following experiments to compare the performance of ASSDE, Support Vector Machines (SVM, one of the most successful supervised algorithm) and TSVM (one of the most successful semi-supervised algorithm) with different labeling rate. We set $n=300$, $cpNum=30$ and $Maxcplter=6$ for ASSDE. Since the iterations of ASSDE are determined by parameter n , in the following experiments, ASSDE can at most actively label (selected by active learning and labeled manually by an expert) 180 instances in each run.

For fair comparison between the active algorithm (e.g. ASSDE) and non-active algorithm (e.g. SVM and TSVM), we conduct experiments with two training data sets, that is, the basic training data set (5 training data for each class) for all three algorithms and the extended training data set (basic training data set extended with $cpNum * Maxcplter$ randomly selected training data) for SVM (denoted as SVM*) and TSVM (denoted as TSVM*). The SVMlight package³ is used in our experiments for the implementation of SVM and TSVM using default configurations.

Since ASSDE performs very well on diff-2 and Spam (see figures 2 and 4), there is very little room for it to improve performance with the increase of training data size. Therefore, we only conduct the experiments on same-2 and Reuters. For each figure, the x-axis represents the number of training data in each class in basic training data set. Figures 7 and 8 give the performance comparison results.

³ <http://svmlight.joachims.org/>

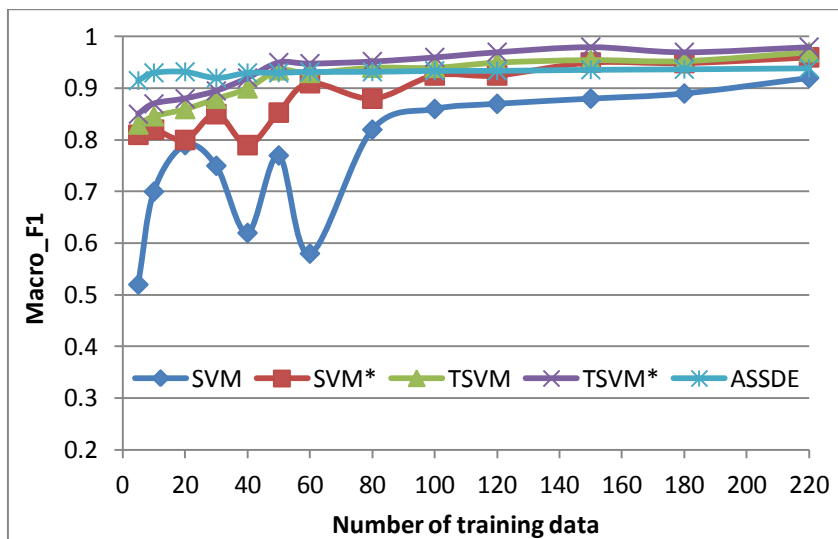


Fig.7. Performance comparison on same-2

From figure 7, we can see that ASSDE outperforms other algorithms when training data are less than 40 (4%) on same-2. From figure 8, we can see that ASSDE outperforms other algorithms with all training data size on Reuters. Compared with other algorithms, ASSDE performs better and more robust for sparsely labeled text classification.

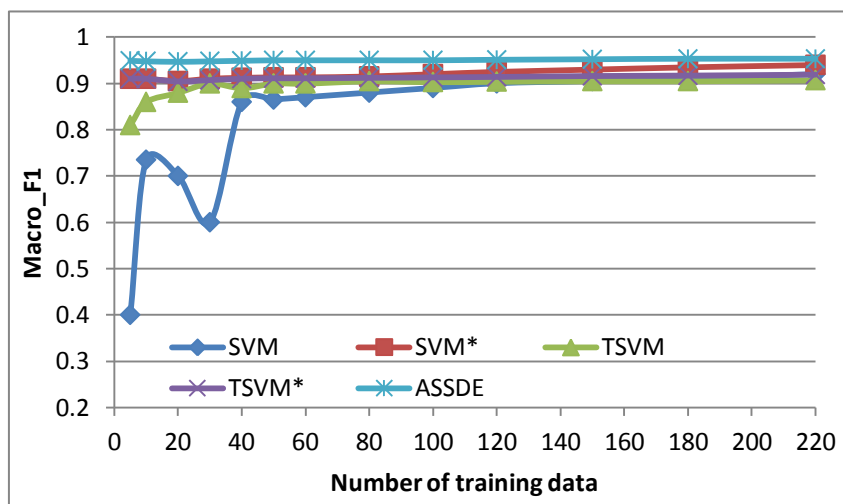


Fig.8. Performance comparison on Reuters

5. Conclusion

In this paper, an active semi-supervised framework with data editing is proposed to improve the performance of sparsely labeled text classification. The aim of data editing in ASSDE is to identify and remove the noise contained in self-labeled training data and thus to improve the overall performance. Our basic consideration is to implement data editing technique by fully utilizing the advantage of active learning in order to incur less computation complexity while improving the accuracy. At the same time, we expect to simplify the design of key component which determines ASSDE's efficiency and use data editing to make up for the deficiency. Therefore we use a very simple but efficient ensemble strategy in ASSDE. Extensive experiments on four text data sets show that data editing is a very useful technique for improving the performance of sparsely labeled text classification, and it makes the algorithm more efficient. This accords with our expectation.

For future work, we will explore more suitable active learning and data editing techniques which may further improve the performance of sparsely labeled text classification. More efficient and effective ensemble strategy for sparsely labeled text classification will be another research direction. Moreover, we will further explore new techniques to cope with the training data sparsity and training data bias for sparsely labeled text classification, e.g. semantic feature extension and clustering aided techniques.

Acknowledgment. The authors would like to thank Dr. Xu Zhu and the anonymous reviewers for their critical advice. This work is partially supported by the National Natural Science Foundation of China (No.61127005, No.61133010, No.50708085, and No.50978127), the special scientific research funding of Research Institute of Highway, Ministry of Transport (No.1206030211003), and the Project of Education Department of Jiangxi Province (No.GJJ08415).

References

1. Liere, R., Tadepalli, P.: Active learning with committees for text categorization. In Proceedings 14th Conference of the American Association for Artificial Intelligence (AAAI). MIT Press, Providence, Rhode Island, 591–596 (1997).
2. Steven, C. H. Hoi, Jin, R., Lyu, M. R.: Large-scale text categorization by batch mode active learning. In Proceedings of the 15th international conference on World Wide Web. Edinburgh, Scotland, May 23-26, 633-642 (2006).
3. Zhang, X., Zhao, D. Y., Chen, L.W., Min, W.H.: Batch Mode Active Learning based Multi-View Text Classification. In Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, China, Aug.14-16, 2009.
4. Joachims, T.: Transductive inference for text classification using support vector machines. In Proceedings of the 16th international conference on machine learning (ICML1999). Bled, Slovenia, June 27-30, 200-209 (1999).
5. Nigam, K., McCallum, A. K., Thrun, S., Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2-3): 103-134, 2000.

6. McCallum, A., Nigam, K.: Employing EM and pool-based Active Learning for text classification. In Proceedings of the 15th International Conference on Machine Learning. Madison, Wisconsin, USA, July 24-27, 350–358 (1998).
7. Gu, P., Zhu, Q. S., Zhang C.: A multi-view approach to semi-supervised document classification with incremental naïve bayes. *Computers and Mathematics with Applications*, 57(6):1030-1036, 2009.
8. Zhou, Z.H., Zhan, D.C., Yang, Q.: Semi-supervised learning with very few labeled training examples. Association for the advancement of artificial intelligence, 2007.
9. Zhuang, D., Zhang, B.Y., Yang, Q., Yan, J., Chen, Z., Chen, Y.: Efficient text classification by weighted proximal SVM. In Proceedings of the Fifth IEEE International Conference on Data Mining. Houston, Texas, USA, November 27 - 30 538-545 (2005).
10. Liu, T., Chen, Z., Zhang, B.Y., Ma, W.Y., Wu, G.Y.: Improving text classification using local latent semantic indexing. In Proceedings of the Fourth IEEE International Conference on Data Mining. Brighton, UK, November 01–04, 162-169 (2004).
11. Yang, Y., Liu, X.: A re-examination of text categorization methods. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. University of California at Berkeley, USA, August 15-19, 42-49 (1999).
12. Dai, W., Xue, G., Yang, Q., Yu, Y.: Transferring Naïve Bayes Classifiers for Text Classification. Association for the Advancement of Artificial Intelligence, 2007, 540-545.
13. Banerjee, S.: Boosting Inductive Transfer for Text Classification Using Wikipedia. In Proceedings of the sixth international conference on machine learning and applications. Kingsgate Marriott, December 13-15, 148-153 (2007).
14. Wang, P., Domeniconi, C.: Building Semantic Kernels for Text Classification using Wikipedia. In Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. Las Vegas, NV, USA, August 24-27, 713-721 (2008).
15. Zeng, H. J., Wang, X. H., Chen, Z., Ma, W. Y.: CBC: Clustering based text classification requiring minimal labeled data. In Proceedings of the 3rd IEEE International Conference on Data Mining. Melbourne, Florida, USA, November 19 – 22, 2003.
16. Raskutti, B., Ferrá, H., Kowalczyk, A.. Combining clustering and co-training to enhance text classification using unlabelled data. In Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining. Edmonton, Alberta, Canada, July 23-26, 620-625 (2002).
17. Raskutti, B., Ferrá, H., Kowalczyk, A.: Using Unlabelled Data for Text Classification through Addition of Cluster Parameters. In Proceedings of the Nineteenth International Conference on Machine Learning. Sydney, Australia, July 8-12, 514 – 521 (2002).
18. Wilson, D.L.: Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Transactions on Systems, Man, and Cybernetics*, 2:408-420, 1972.
19. Li, M., Zhou, Z.H.: SETRED: Self-training with editing. In Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05), Hanoi, Vietnam, May 18-20, 611-621 (2005).
20. Zhou, Z. H., Zhan, D. C., Yang, Q.: Semi-supervised learning with very few labeled training examples. Association for the advancement of artificial intelligence, 2007.
21. Sun S. L.: Active learning with extremely sparse labeled examples. In Proceedings of Neural Information Processing Systems Workshop on Learning from Multiple Sources. Vancouver, British Columbia, Canada, December 8-11, 2008.

Xue Zhang and Wangxin Xiao

22. Phan, X. H., Nguyen, L. M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In Proceedings of International World Wide Web Conference. Beijing, China, April 21 – 25, 91-100 (2008).
23. Cai, L., Hofmann, T.: Text categorization by boosting automatically extracted concepts. Proc. ACM SIGIR, Toronto, Canada. July 28 - August 1, 2003.
24. Xu, Z., Jin, R., Huang, K., Lyu, M. R., King, I.: Semi-supervised text categorization by active search. In Proceedings of ACM 17th Conference on Information and Knowledge Management (CIKM 2008). Napa Valley, California, October 26-30, 1517-1518 (2008).
25. Breiman L.: Bagging predictors. Machine Learning, 24(2):123–140, 1996.
26. Bryll, R., Ricardo, G. O., Quek, F.: Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. Pattern Recognition 36(2003):1291-1302.
27. Zhou, Z. H., Yu, Y.: Ensemble local learners through multimodal perturbation. IEEE transactions on systems, man, and cybernetics-Part B: Cybernetics, 2005, 35(4): 725-735.

Xue Zhang, received the BS degree in electronic engineering from XiDian University, Xian, China, in 1999. She received the MS degree in control theory and control engineering from Southwest University of Science and Technology, Mianyang, China, in 2003, and received the PhD degree in computer science from Southeast University, Nanjing, China, in 2007. From 2008 to the present, she is a postdoctoral fellow in Peking University. Her research interests include data mining and machine learning, with emphasis on the applications to text mining and bioinformatics.

Wangxin Xiao, received the PhD degree in traffic information and control engineering from Southeast University, Nanjing, China, in 2004. From 2005 to 2007, he engaged in postdoctoral research in Wuhan University of Technology. Since 2008 he has been an associate professor in Research Institute of Highway Ministry of Transport. From 2009 to 2011, he was also a postdoctoral fellow in Changsha University of Science and Technology. His research interests include pattern recognition, Intelligent Transport Systems (ITS) and data mining with applications to traffic data.

Received: February 02, 2012; Accepted: November 29, 2012.

An Ant System based on Moderate Search for TSP

Ping Guo and Zhujin Liu

College of Computer Science, Chongqing University, Chongqing, China
Chongqing Key Laboratory of Software Theory & Technology, Chongqing 400044,
China
guoping@cqu.edu.cn

Abstract. Ant Colony Optimization (ACO) algorithms often suffer from criticism for the local optimum and premature convergence. In order to overcome these inherent shortcomings shared by most ACO algorithms, we divide the ordinary ants into two types: the utilization-oriented ants and the exploration-oriented ants. The utilization-oriented ants focus on constructing solutions based on the learned experience like ants in many other ACO algorithms. On the other hand, inspired by the adaptive behaviors of some real-world *Monomorium* ant species who tend to select paths with moderate pheromone concentration, a novel search strategy, that is, a completely new transition rule is designed for the exploration-oriented ants to explore more unknown solutions. In addition, a new corresponding update strategy is also employed. Moreover, applying the new search strategy and update strategy, we propose an improved version of ACO algorithm—Moderate Ant System. This improved algorithm is experimentally turned out to be effective and competitive.

Keywords: Ant Colony Optimization; Adaptive behavior; Traveling Salesman Problem; Local optimum; Premature convergence

1. Introduction

Combinational optimization (CO) problems exist widely in the real world and many of them are non-deterministic polynomial time hard (NP-hard). NP problems are decision problems where the solutions can be recognized in polynomial time by a non-deterministic Turing machine. And NP-hard problems are at least as hard as the hardest problems in NP. Such problems need not be in NP; indeed, they may not even be decision problems [1,2]. As a particular kind of optimization problems, CO problems play an important role be it in theory or in practice. Traditional algorithms can obtain the optimal solution to NP-hard CO problems but often at the cost of great computational time, which is infeasible for any practical application [3]. Swarm intelligence, inspired by the social behaviors of real animals, provides a novel way to deal with such complicated optimization problems [4]. Swarm intelligence algorithms, which usually are approximation algorithms, can compute

satisfactory solution in markedly less time. Particularly, ant colony optimization (ACO) is a typical sort of SI algorithms. For many problems, ACO algorithms generate competitive results that are very close to the best known solutions while on some problems like quadratic assignment problem (QAP) they are the state-of-the-art [5].

ACO takes inspiration from the foraging behaviors of real ants which can always find the shortest path between their nests and food sources. The famous experiment, known as “shortest bridge” experiment, simulates the behaviors and explains the optimization ability [6,7]. When there are more than one path between a nest and a food source, each ant will choose one path by chance. And the ants will return to their nest as soon as they arrive at the food source. Meanwhile, they lay pheromone on the paths they have passed. It is obvious that it takes less time for the ants that choose the shorter path to get back to the nest. Accordingly, the shorter path gains more pheromone from ants than longer paths because of the high round-trip frequency. After that, ants in the nest tend to choose the path with higher pheromone concentration instead of random selection when departing for the food source. Thus, an increasing number of ants will be attracted to the shorter path and those ants will lay more pheromone on the shorter path over time. Eventually, the path between the nest and the food source selected by almost all of the ants, that is, the shortest path, is found. An individual ant has little intelligence. However, a swarm of ants achieves incredible high intelligence [8]. The term “stigmergy” is used to illustrate the distinguished capability that owned by ants as well as other animals. Specifically, stigmergy, refers to a self-organized mechanism which describes such an activity: a number of individuals interact with each other indirectly via modifying the common environment which serves as a medium; in other words, ants affect the local environment and the environment feeds back to ants, thus, ants exchange information; as a consequence, the interaction contributes to the update of the environment which affords a new platform for new interaction [5]. Drawing on this mechanism, a multitude of ant algorithms have been created and ACO meta-heuristic is used to describe these algorithms in a more general way [9], however, ant algorithms are not confined to ACO.

The usual method to examine the effectiveness of a new algorithm is to apply it to some problems and compare its performance with that of already known algorithms. Traveling Salesman Problem (TSP), probably the best known instance of NP-hard problems [100], is firstly used to evaluate ACO's performance, which indicates its central role in the development of ACO [11]. Subsequently, TSP has served as a benchmark for the sake of comparison with other optimization methods. TSP stems from a real-world problem: given a set of cities and their pairwise distances, the goal is to find the shortest tour that visits each city exactly once. In more formal terms, the task is to find a Hamiltonian circuit of minimal length on a fully connected graph [12]. There are quite a few TSP data sets that are available, and our experiments are based on these data sets as well.

Overall, the existing ACO algorithms share similar search and update strategies. More precisely, the transition rules that they employ render them in favor of components with higher pheromone concentration during the construction of feasible solutions. And update rules are more likely to give the better solutions more pheromone when updating the pheromone trails [13]. It is evident that transition rules and update rules supplement each other. The average quality of solutions benefits from the selection bias and update bias indeed. On the other hand, taking into account that ACO is an iterative approach, the biases may lead to premature convergence and local optimum, that is, limit the discovery of new or better answers, which is the starting point of our research. As a matter of fact, not all species of ants in the real world tend to select paths with more pheromone. According to some new research [14], some kinds of ants will strike a balance between paths with different quantity of pheromone. They prefer paths with moderate pheromone rather than paths with overly high or overly low pheromone concentration. Inspired by this, we present an improved ACO algorithm by bring in an extra transition rule and the counterpart update rule. The improved ACO algorithm is experimentally proven that it can obtain better solutions than other ACO algorithms in most cases.

The rest of this paper is organized as follows: Section 2 introduces several main ACO algorithms and discusses their design ideas. In Section 3, the original idea of Moderate Ant System (MAS) is explained and we present the search strategy and update strategy of MAS as well as the concrete algorithm. Section 4 gives a performance comparison and analysis between our improved algorithm and main ACO algorithms. Finally, Section 5 concludes the paper.

2. Related Work

A lot of ACO algorithms have been proposed since Dorigo et al. introduced the first ant algorithm in 1996. Here we present typical ones of them: Ant System (AS) [11,15], Max-Min Ant System (MMAS) [16] and Ant Colony System (ACS) [17,18]. In order to be in lines with our experiments, we use TSP as a specific instance to describe the algorithms.

For the convenience, Let G_{best} is the global best solution.

2.1. Ant system

As mentioned above, AS is the first ACO algorithm presented, and meanwhile, it is the original version of various ant algorithms as well. Therefore, we give a relatively complete but brief introduction of AS first, and the same details are omitted when introducing other ACO algorithms. Generally, AS and most of other ACO algorithms share the analogous

framework in terms of implementation. The main framework of ACO algorithms including AS is shown in Algorithm 1.

```

Algorithm 1 The framework of ACO algorithms
  Initialize the parameters and pheromone trails;
  while termination conditions not satisfied do
    Step 1: Each ant constructs a solution
    Step 2: Update the pheromone trails
  end while
end
    
```

It can be seen that the core of the algorithms is a loop body and each loop consists of two crucial steps: the construction of solutions and the update of pheromone trails. The loop body, in fact, is an iterative process through which the accumulation of pheromone can play a role.

Iteration is defined as the interval in $(t, t+1)$ during which each of the m ants constructs a solution. In other words, every ant traverses all of the n cities in each iteration. AS together with other ant algorithms employs the tabu table to record the already visited cities of an iteration in order to obtain feasible answers. In Step 1 of Algorithm 1, during the construction of a solution, ants choose the next city to be visited via certain search strategy which is a stochastic mechanism [19]. When ant k is in city i , the probability of going to city j is given by:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{l \notin tabu_k} \tau_{il}^\alpha \eta_{il}^\beta} & \text{if } j \notin tabu_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where, $tabu_k$ is the tabu table of ant k which is comprised of the cities that have been visited in the current iteration. The parameters α and β control the relative importance of the pheromone τ_{ij} versus the visibility η_{ij} . The visibility represents the heuristic information, which is given by:

$$\eta_{ij} = 1/d_{ij} \quad (2)$$

Where, d_{ij} denotes the distance between city i and city j . This transition rule is also shared by other algorithms.

After each iteration, Step 2 of Algorithm 1 is performed. The pheromone τ_{ij} linked to the path connecting city i and j , is updated as Eq. (3).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3)$$

Where ρ is the evaporation rate, m is the number of ants, and $\Delta\tau_{ij}^k$ is the quantity of pheromone laid on path(i,j) by ant k :

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_K & \text{if ant } k \text{ used path}(i,j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Where, Q is a constant and L_k is the tour length of the solution obtained by ant k .

2.2. Ant system with elitist strategy

AS_{elite} is the first improved version of the original AS. In order to make the best-so-far solution more attractive to ants in the following iterations, the best-so-far solution is given extra pheromone. The best-so-far solution is called global-best solution and the ants who obtain this solution are called elitist ants. The update formula of pheromone is given by:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k + n_e \Delta\tau_{ij}^e \quad (5)$$

Where e denotes the elitist ants, n_e denotes the number of elitist ants and $\Delta\tau_{ij}^e$ is defined as follows:

$$\Delta\tau_{ij}^e = \begin{cases} Q / L^{gb} & \text{if } path(i, j) \in G_{best} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Where L^{gb} is the tour length of global-best solution.

The elitist strategy enables AS to improve the quality of average solution and to find better solutions at earlier time. However, the strategy narrows the difference between selections of different ants.

2.3. Rank-based Ant System

AS_{rank} draws on the concept of ranking and adopts it into the pheromone update procedure. The n ants are ranked according to the quality of their solutions in decreasing order (i.e., $L_1 \leq L_2 \leq \dots \leq L_m$). σ is a parameter of AS_{rank} . Only the paths included in the solutions of $\sigma-1$ best ants acquire extra pheromone and the amount depends directly on the ant's rank μ and the quality of its solution. Moreover, the paths belong to global-best solution L^{gb} receive an additional amount of pheromone which depends on the length of L^{gb} , weighted by parameter σ . The pheromone update function is given by:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sigma\Delta\tau_{ij}^e + \sum_{\mu=1}^{\sigma-1} (\sigma - \mu)\Delta\tau_{ij}^\mu \quad (7)$$

where $\Delta\tau_{ij}^e$ as Eq.(6) and $\Delta\tau_{ij}^\mu$ as Eq.(8).

$$\Delta\tau_{ij}^{\mu} = \begin{cases} \frac{Q}{L_{\mu}} & \text{if the } \mu\text{th best ant travels path}(i, j) \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

2.4. Max-Min Ant System

MMAS differs from the AS in two main aspects: only the best ant is allowed to update the pheromone trails, and the value of pheromone on the paths is bound. The pheromone update function is implemented as follows:

$$\tau_{ij}(t+1) = [(1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best}]_{\tau_{min}}^{\tau_{max}} \quad (9)$$

Where τ_{max} and τ_{min} are the upper bound and lower bound of the amount of pheromone on the paths, respectively. And $[x]_b^a$ as Eq.(10) and $\Delta\tau_{ij}^{best}$ as Eq.(11).

$$[x]_b^a = \begin{cases} a & \text{if } x > a \\ b & \text{if } x < b \\ x & \text{otherwise} \end{cases} \quad (10)$$

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{1}{L_{best}} & \text{path}(i, j) \in G_{best} \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

L_{best} is the length of the tour obtained by the best ant. It can be the best solution found in the current iteration—iteration-best, L_{ib} or the best solution found since the beginning of the algorithm—global-best, L_{gb} , or a combination of both.

With respect to the upper and lower bounds on the pheromone values, τ_{max} and τ_{min} tend to be set empirically and are determined on a case-by-case basis. Nevertheless, there are still some guide lines that have been proposed for defining τ_{max} and τ_{min} on the basis of analytical considerations.

2.5. Ant Colony System

The most significant contribution of ACS is the introduction of a local pheromone update in addition to the pheromone update applied at the end of the construction process, which is more in line with the natural behavior of real ants.

The local pheromone update is executed by all the ants after each construction step of the n steps of an iteration. Moreover, each ant applies it only the last path which has just been traversed:

$$\tau_{ij} = (1 - \varphi)\tau_{ij} + \varphi\tau_0 \quad (12)$$

Where $\varphi \in (0, 1]$ is the pheromone decay coefficient, and τ_0 is the initial value of the pheromone trail on the path.

Similar to the MMAS algorithm, the pheromone is also updated at the end of each iteration by the iteration-best or the global-best ant only. However, the update function is slightly different:

$$\tau_{ij}(t+1) = \begin{cases} (1 - \rho)\tau_{ij}(t) + \rho\Delta\tau_{ij} & path(i, j) \in G_{best} \\ \Delta\tau_{ij} & otherwise \end{cases} \quad (13)$$

where

$$\Delta\tau_{ij} = 1/L_{best} \quad (14)$$

As in MMAS, L_{best} can be either L_{ib} or L_{gb} .

ACS also uses a different transisiton rule, called the pseudorandom proportional rule. If let k be an ant located on city i , $q_0 \in [0, 1]$ be a parameter and q be a random value in $[0, 1]$, then the next city j is selected according to Eq.(15) with $q \leq q_0$, else Eq.(1).

$$p_{ij}^k = \begin{cases} 1 & \text{if } j = \arg \max \{ \tau_{ij} \eta_{ij}^\beta \mid j \notin tabu_k \} \\ 0 & otherwise \end{cases} \quad (15)$$

Additionally, ACS employs the candidate lists to make the selections prefer some nearer cities during the construction process. In the TSP instance, each of the n has a candidate list. The city i 's candidate list consists of a certain number of cities which are the closest ones to city i . Obviously, the candidate lists of the cities can be built prior to the loop body and they remain stable when performing the algorithm. When an ant is at city i , it will choose the next city among those of city i 's candidate list which are not visited yet in the current iteration. Only if all of the cities in the candidate list are included in tabu table, would the other cities be selected according the transition rule.

3. Moderate Ant System

3.1. Design Ideas of MAS

Typically, the designs of existing ant algorithms manifest two common apparent principles: ants prefer paths with higher pheromone concentration

when constructing a feasible solution at an iteration; the paths belong to the good solutions tends to gain greater amount of pheromone when updating the pheromone trails. Clearly, the two principles strengthen each other. These two principles are able to improve the quality of the solutions and speed up the convergence to some extent.

However, if ants select the paths at a probability in proportion to the pheromone concentration, the chances of finding new solutions would decrease over time. As can be seen from Figure 1, suppose that there is only one food source in the upper path at the beginning, definitely, all the ants will converge to the upper path, and new ants will make the same choice as well because of the great amount of pheromone laid on the upper path, even when a new food source appears in the lower path. According to Figure 2, there are two food sources in the two paths. The upper food source is nearer to the nest than the lower food source. Accordingly, ants go there and back between the upper food source and the nest more frequent and deposit more pheromone on the upper path. Thus, ants will converge to the upper path eventually. Under both the two circumstances, the ability to find new food sources (solutions) are limited by selection bias and update bias of traditional principles which are illustrated above when solving a concrete problem such as TSP.

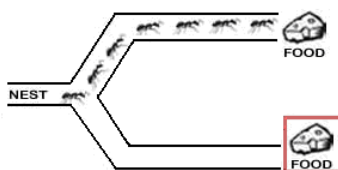


Fig. 1. A new food source appears after convergence

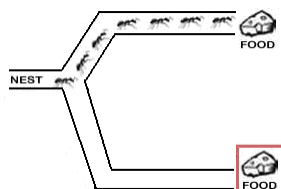


Fig. 2. Two food sources in different length of paths

Actually, the advantages that acquired by applying the principles are accompanied by premature convergence and local optimum which are highly criticized. In consideration of this, we intend to explore other principles to overcome these drawbacks. In particular, some research in animal behavior field offers us an alternative way to design new ACO algorithms.

In the natural world, there are some Monomorium ant species (e.g., *M.niloticum*, *M.najrane*, *M.mayri*) that dislike overly high pheromone concentration. Instead, moderate pheromone concentration renders them more active and paths with too high or too low pheromone concentration

induce either no response or repellency [14]. This is the adaptive behavior of the three Monomorium ant species. They regard paths with moderate amount of pheromone as promising sources and tend to choose them, which seems to run contrary to the traditional principles used in ant algorithms.

In the view of traditional design ideas, ants have a better chance to choose paths with higher pheromone concentration. There are two main reasons that cause the high pheromone. Firstly, if there are numerous ants traveling the same path, it indicates that ants for the food source on the path are enough. Thus, a multitude of ants travel the path and lays a great amount of pheromone on the path. Secondly, if ants can frequently travel the same path, it indicates that the food source on the path is close to the nest and do not require too much ants. In this case, a few ants travel the path frequently and also deposit a great amount of pheromone. Two different behaviors end with the same result. At a deeper level, it involves an important problem to ants, that is, distributing the workforce. Be it sufficient ants selecting a path or ants traveling a path frequently enough, it is unnecessary to allocate more ants to the path whose pheromone concentration is already high enough. Otherwise, it would be a waste of workforce. Therefore, the Monomorium ant species' adaptive behavior (tend to choose the paths with moderate pheromone concentration) is understandable and reasonable. It equips the ants with the ability to distribute their workforce more probably and efficiently.

Inspired by the adaptive behavior of the Monomorium ant species, we develop a novel ACO algorithm -- Moderate Ant System (MAS). In MAS, a new transition rule which can enhance the capability to find new solutions is designed. Furthermore, in order to match the new transition rule and improve the performance of the algorithms, new update rule is designed as well.

3.2. Concrete Algorithm

MAS is an improved version based on AS, differs from AS in search and update strategy (i.e., transition rule and update rule) corresponding to the two main steps of Algorithm 1.

Search strategy of MAS. Selection bias contributes to the average quality of solutions while it brings the premature convergence and local optimum at the same time. In order to conquer the disadvantages, more explorative operations are needed. Meanwhile, as a sort of iterative algorithms, the solutions are improved gradually, so the average quality of solutions is supposed to be guaranteed as well. Therefore, a balance should be struck between exploring new knowledge and exploiting the already-known knowledge. In the ACO algorithms introduced in Section II, all the ants fulfill the same obligation. To be more exact, all of the ants construct the solutions according to the same transition rule. For that reason, in MAS, the ants are divided into two categories: the utilization-oriented ants and the exploration-oriented ants. The two sorts of ants are entrusted with different tasks, respectively.

First, actually, the utilization-oriented ants share the same business with that of the ants in most ACO algorithms. This kind of ants mainly concentrates on utilizing the search experience accumulated by the iterative process in previous iterations. They choose the paths with higher pheromone concentration at a higher probability. During the construction of solutions, the utilization-oriented ants make their choices according to the probability distribution given by Equation 1 which is same to the ordinary ants.

Second, the design of exploration-oriented ants is inspired by the Monomorium ant species and is the core of MAS. The utilization-oriented ants perform a biased search and they can choose poor paths (with lower pheromone concentration) but at a low probability. Unlike the utilization-oriented ants, in order to explore more possibilities, the exploration-oriented ants employ a more aggressive transition rule which is discussed in detail.

The pheromone concentration determines the choices of ants in real world, while in ACO algorithms, the combination of pheromone and visibility determines the choices of ants for the sake of better performance. Here, $info_{ij}$ is used to represent the integrated factors on the path from city i to city j , and $info_{ij}$ is given by:

$$info_{ij} = \tau_{ij}^\alpha \eta_{ij}^\beta \tag{16}$$

Drawing on the search strategy of Monomorium ant species, the new transition rule should make the ants tend to select paths with moderate *information* instead of too much or too little *information*.

Suppose that ant k is at city i at step t of an iteration. When selecting the next city as part of the solution, ant k has $n-t$ adjacent cities available. According to Figure 3, from city i 's point of view, the *information* values on the $n-t$ adjacent paths can be regarded as a stochastic variable follow a normal distribution $N(\mu, \sigma^2)$ since n in TSP is usually a big enough number.

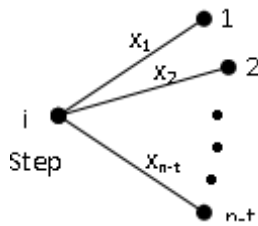


Fig. 3. $n-t$ choices of an ant at city i at step t

The parameters of $N(\mu, \sigma^2)$ are estimated as follow:

$$\mu = \frac{1}{n-t} \sum_{j \in tabu_k} info_{ij} \tag{17}$$

$$\sigma^2 = \frac{1}{n-t} \sum_{j \notin tabu_k} (info_{ij} - \mu)^2 \quad (18)$$

In addition, attraction_{ij} is defined as follows:

$$attraction_{ij} = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(info_{ij} - \mu)^2}{2\sigma^2}} & \text{if } j \notin tabb_k \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

Which denotes the attraction of path from city *i* to city *j* for the exploration-oriented ants in MAS. If let *k* be an ant located on city *i*, $q_0 \in [0, 1]$ be a parameter, and *q* be a random variable uniformly distributed in $[0, 1]$, then the exploration-oriented ants will choose the next city *j* according to Eq.(20) with $q \leq q_0$, else Eq.(19) is used.

$$P_{ij}^k = \begin{cases} 1 & \text{if } j = \arg \max_{j \notin tabu_k} attraction_{ij}, \\ 0 & \text{otherwise,} \end{cases} \quad (20)$$

With probability q_0 , the exploration-oriented ants select the most moderate path in terms of *attraction*, while with probability $(1 - q_0)$, the exploration-oriented ants incline to select the more moderate paths but have a chance to select other paths.

The number of the utilization-oriented ants and the number of the exploration-oriented ants are important parameters of MAS. Tuning their numbers can affect the degree of utilization and exploration (i.e. decide whether to attach more importance to utilizing the learned search experience or exploring new solutions). If there are far more exploration-oriented ants than utilization-oriented ants, the algorithm is about to lose the benefit brought by the iterative process on the basis of relatively good solutions and achieve poor performance.

Update strategy of MAS. In order to achieve the greatest effect of our new transition rule, the counterpart update strategy is designed and employed in MAS. The update strategy in MAS consists of two parts: the evaporation of pheromone trails and the accumulation of pheromone trails. The evaporation lessens the pheromone value on paths while the accumulation adds the pheromone value on paths.

In MAS, similar to the phenomenon in the real world, the pheromone on all of the paths evaporates over the time, which is implemented by:

$$\tau_{ij}'(t) = (1 - \rho)\tau_{ij}(t) + \rho\tau_0 \quad (21)$$

Where $\tau_{ij}'(t)$ represents the pheromone concentration on the path between city *i* and city *j* after evaporation but before given extra pheromone at the *t*th iteration. Additionally, τ_0 is initialized by a very small value. It is evident that the lower bound of the pheromone value on a path is τ_0 . The evaporation can

avoid unlimited accumulation of the pheromone trails. Moreover, if a path does not gain extra pheromone in several consecutive iterations, its corresponding pheromone concentration decreases rapidly until the pheromone trail value is equal to τ_0 .

After evaporation, the accumulation of pheromone trails is performed. Not all of the paths gain extra pheromone. As in AS_{rank} , only paths belonging to the best n_b-1 solutions in an iteration are given extra pheromone in MAS and the amount of extra pheromone that paths gain depends on the rank of solutions. The better the rank is, the more pheromone paths of the solutions gain. These paths are provided with strong additional reinforcement and in the following iterations will attract more utilization-oriented ants. However, the update rule is quite different from that of AS_{rank} , which is given by:

$$\tau_{ij}(t+1) = \tau_{ij}'(t) + \sum_{\mu=1}^{n_b-1} \frac{n_b - \mu}{n_b} (\Delta\tau_{ij} - \tau_{ij}'(t)) \quad (22)$$

It means that n_b-1 ants (be it utilization-oriented ants or exploration-oriented ants) will deposit extra pheromone to the paths they visited at the current iteration. It is notable that these n_b-1 ants accumulate the pheromone trails one by one and $\tau_{ij}'(t)$ is updated at the same time, so $\tau_{ij}'(t)$ may differ from ant to ant. Hence, it is easy to prove that the upper bound of the pheromone value on a path is $\Delta\tau_{ij}$. According to our research, using L_{gb} to calculate $\Delta\tau_{ij}$ in MAS achieves better performance than using L_{ib} , especially when MAS is applied to TSP data sets with more than 200 cities.

MMAS employs pheromone limits to prevent the pheromone value on the paths from overly low or overly high, thus, MMAS is able to avoid algorithm stagnation. In particular, MAS implements the similar limits in an implicit manner. The pheromone trails in MAS are guaranteed that $\forall(i,j): \tau_0 \leq \tau_{ij} \leq \Delta\tau_{ij}$, which will contribute to the performance of MAS as well.

Pseudo-code of MAS. On the basis of the designed search strategy and update strategy, as can be seen in Algorithm 2, the pseudo-code of MAS is given.

```

Algorithm 2 Moderate Ant System
Initialize the parameters and pheromone trails;
while termination conditions not satisfied do
  begin
    Step 1:
    For each utilization-oriented ant
      Construct a solution according to Eq. 1.
    End For
    For each exploration-oriented ant
      Construct a solution according to Eq. 19, 20.
    End For
    Step 2:
    For all the paths
      Update the pheromone trail as Eq. 21.
  end

```

```

End For
For the paths belonging to the  $n_b$  best solutions
  Update the pheromone trail use Eq. 22.
End For
end while;
end.

```

4. Experimentation

To show the performance of MAS and compare it with other ACO algorithms (i.e., AS, AS_{elite}, AS_{rank}, MMAS, and ACS), these experimentations are completed that based on the TSPLIB's data sets (<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>).

4.1. Data Sets Used

In the following experimentations, 10 test data sets are employed. Table 1 illustrates the data sets and their size used in the experiments.

Table 1. TSP test data sets

Data set	Number of cities
att48.tsp	48
eil76.tsp	76
u159.tsp	159
ts225.tsp	225
pr439.tsp	439
rat575.tsp	575
p654.tsp	654
d1291.tsp	1291
vm1748.tsp	1748
u2152.tsp	2152

4.2. Parameters Setup

Our experimentation is based on ACOTSP which offers a high-quality implementation of various ACO algorithms for TSP [20]. With the common code, MAS is developed and compared in the same framework. And the parameters adopted in the experimentation are the default parameters set in ACOTSP. The main parameters used in the experiments are set as shown in Table 2.

As the solutions that ACO algorithms obtain may vary from try to try due to the use of probability. To be fair and get accurate results, we run each

algorithm 10 times, respectively. And in each try, the maximum runtime is set to 10 seconds. Because ACO algorithms are iterative algorithms, the runtime should not be too short. 10 seconds is an acceptable time for most application and the obtained results are relatively stable. And we also do the experiments with runtime set to 15s and 20s, however, the difference of solutions' quality over the three runtimes is quite small, not greater than 0.5%. In addition, in order to examine the usefulness of our new search strategy and update strategy, local search is not employed in all tested algorithms.

Table 2. Main parameters

Parameters	Value	Memo
α	1.0	For all
β	2.0	For all
ρ	0.5	For all
m	25	For all
tries	10	For all
runtime	10s	For all
n_e	100	For AS_{elit}
σ	6	For AS_{rank}
q	0.5	For ACS
φ	0.1	For ACS
size of candidate list	20	For ACS
number of utilization-oriented ants	13	For MAS
number of exploration-oriented ants	12	For MAS
q_0	0.8	For MAS
n_b	6	For MAS

4.3. Experimental Results

The results obtained are presented in Table 3 and Table 4. Table 3 illustrates the shortest tour lengths (best solutions in 10 tries) obtained by the six tested ant algorithms on the 10 test data sets, while Table 4 illustrates the average tour lengths (average solutions of 10 tries) obtained by the six tested ant algorithms on the test data sets. Table 5 and Table 6 give the results of the best performer for each algorithm regarding the shortest tour and the average tour, respectively.

In order to show the performance of different ACO algorithms more clearly, Relative Solution Quality (RSQ) of algorithm A, achieved on data set D, is defined as follows:

$$RSQ_{D,A} = \left(1 - \frac{solution_A - Best}{Worst - Best}\right) \times 100\% \quad (23)$$

Where $solution_A$ denotes the solution (the shortest tour length or average tour length) obtained by algorithm A. $Best$ denotes the best solution (the

shortest tour length or average tour length) obtained by the six tested ant algorithms, and *Worst* denotes the worst solution (the shortest tour length or average tour length) obtained by the six tested ant algorithms. Obviously, the greater the value of RSQ is, the better the solution.

Given the definition of RSQ, two line graphs are created. According to Figure 4, the RSQ of MAS stabilizes at 100% or very close to 100%. To be more exact, MAS obtains the best solution on all data sets except att48.tsp (RSQ= 96.55%), rat575.tsp (RSQ= 95.39%) and vm1748.tsp (RSQ= 89.99%). MMAS and ACS are commonly regarded as very effective ACO algorithms. Both of them achieve good performance but second to MAS in terms of finding the shortest tour. As shown in Figure 5, in terms of average solution quality, MMAS performs best. MAS also obtains decent average solutions. Specifically, MAS get best average solutions in four TSP data sets. And it performs better than ACS and only second to MMAS. By comparison, AS's performance is the poorest regarding both the best solutions and the average solutions.

Table 3. The shortest tour lengths obtained

Data set	AS	AS _{elite}	AS _{rank}	MMAS	ACS	MAS
att48.tsp	11438	11572	11300	11021	11021	11040
eil76.tsp	557	561	564	554	554	551
u159.tsp	47850	47301	47671	45319	45423	43270
ts225.tsp	133006	131776	131386	130814	130711	129167
pr439.tsp	123827	122977	121706	118915	120870	118274
rat575.tsp	7889	7877	7822	7668	7607	7620
p654.tsp	41935	41585	41408	40618	41234	39975
d1291.tsp	58890	57140	57357	55728	56244	55698
vm1748.tsp	405534	398635	396208	396144	397473	397084
u2152.tsp	75140	75173	75002	73809	74058	73773

MAS's quality of average solutions is not as good as that of best solutions. It is understandable because the new search strategy allocate part of the ants (the exploration-oriented ants) to focus on founding new tours and diversify MAS's choices rather than finding solutions around best-so-far tours. By and large, in lines with the design intent, MAS is able to find better solutions than other ant algorithms in most cases. To a large extent, MAS addresses the inherent problems (i.e., premature convergence and local optimum) existing widely in ACO algorithms by applying new search strategy as well as the counterpart update strategy. Although the average solutions of MAS are not the best among all of the tested algorithms, MAS obtains satisfactory average solutions as well due to the reservation of the ordinary ants (the utilization-oriented ants in MAS). Considering that TSP aims at finding the shortest tour, all other solutions become worthless as soon as a shorter tour is found. Therefore, in conclusion, MAS is very competitive and significant.

Table 4. The average tour lengths obtained

Data set	AS	AS _{elite}	AS _{rank}	MMAS	ACS	MAS
att48.tsp	11889.3	11682.3	11581.7	11071.7	11067.8	11584.8
eil76.tsp	567.9	586.8	581.5	559.9	561.2	572.2
u159.tsp	49168.4	48348.6	48396.2	46208.5	46378.9	43340.6
ts225.tsp	135751.7	134408.1	132394.3	132575.9	132289.4	132198.4
pr439.tsp	126033.8	125511.9	124594.4	123619.4	124463.8	124065.4
rat575.tsp	7950	7921.8	7902.6	7830.9	7834.7	7800.5
p654.tsp	42682.7	42818.6	42700.6	41873.2	42233.9	41661.8
d1291.tsp	59487.7	58324.2	58073.3	57397.4	57725.6	57605.1
vm1748.tsp	409406.6	404314.6	401487.7	401273.0	403220.4	402692.3
u2152.tsp	76209.4	75608.9	75621.3	74486.6	74733.5	74759.0

Table 5. Results of best performers in terms of the shortest tour

Data set	Best Performer	Tour Length
att48.tsp	MMAS, ACS	11021
eil76.tsp	MAS	551
u159.tsp	MAS	43270
ts225.tsp	MAS	129167
pr439.tsp	MAS	118274
rat575.tsp	ACS	7607
p654.tsp	MAS	39975
d1291.tsp	MAS	55698
vm1748.tsp	MMAS	396144
u2152.tsp	MAS	73773

Table 6. Results of best performers in terms of the average tour

Data set	Best Performer	Tour Length
att48.tsp	ACS	11067.8
eil76.tsp	MMAS	559.9
u159.tsp	MAS	43340.6
ts225.tsp	MAS	132198.4
pr439.tsp	MMAS	123619.4
rat575.tsp	MAS	7800.5
p654.tsp	MAS	41661.8
d1291.tsp	MMAS	57397.4
vm1748.tsp	MMAS	401273.0
u2152.tsp	MMAS	74486.6

An Ant System based on Moderate Search for TSP

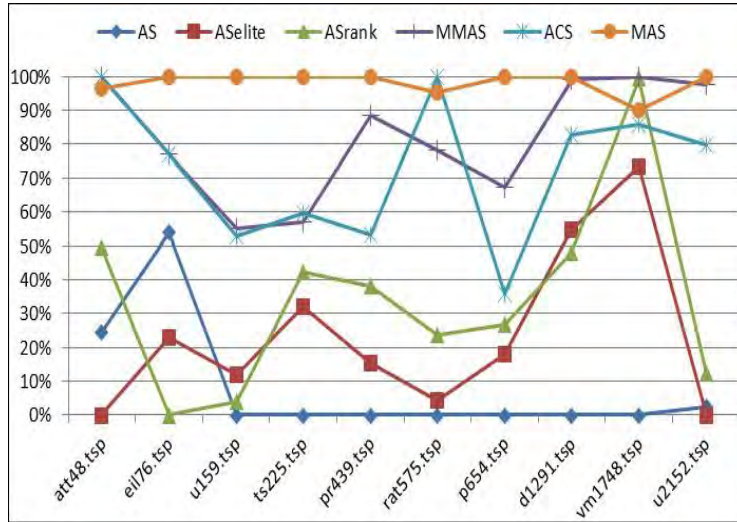


Fig. 4. Best solutions obtained by the tested algorithms. The data sets are sorted by the size in ascending order.

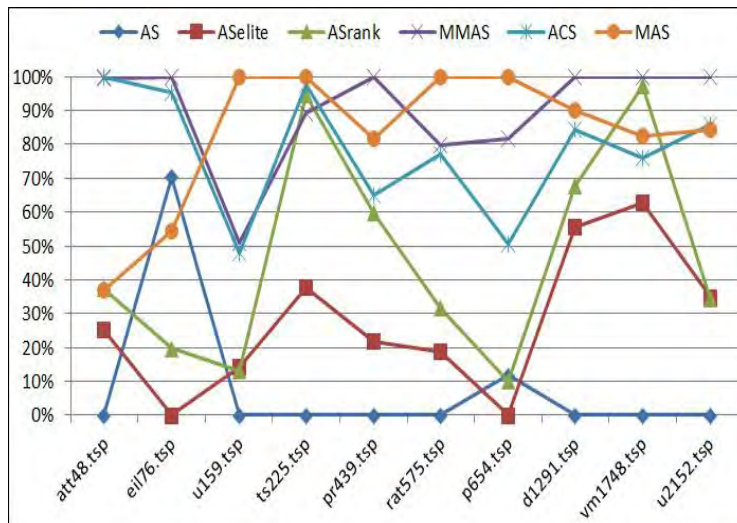


Fig. 5. Average solutions obtained by the tested algorithms. The data sets are sorted by the size in ascending order.

5. Conclusion

This paper discussed the shortcomings of several main ACO algorithms which often get in trouble in local optimum and premature convergence when searching the optimum solution owing to the selection bias and update bias. To overcome these obvious defects, we propose a new ACO algorithm named Moderate Ant System. In MAS, the ordinary ants are divided into two groups, the utilization-oriented ants and the exploration-oriented ants. As for the exploration-oriented ants, taking inspiration from the adaptive behavior of the *Monomorium* ant species, a novel transition rule is designed and adopted by the exploration-oriented ants. The new transition rule requires a little more time and space, but the extra overhead is linear complexity. Moreover, we also design new update rules to match MAS's search strategy. Taking TSP as an example, ten standard TSP test data sets are employed to examine the performance of MAS algorithm in the experiments. As a result, MAS does achieve our design intent and conquer the known defects to some extent. Overall, the best solution of MAS is better than that of other ant algorithms at the expense of slightly losing the average solution quality.

In the future work, more thorough experimentation and further investigation into MAS algorithm are required in order to explore more characteristics of MAS and perfect the performance of MAS. Moreover, we are planning to apply MAS to other classical problems and give a further study about the performance of MAS.

References

1. R. Demontis, "A simple NP-hard problem", ACM SIGACT News, vol. 40, no. 2, pp. 45-48, 2009.
2. D. S. Hochba, "Approximation algorithms for NP-hard problems" ACM SIGACT News, vol. 28, no. 2, pp. 40-52, 1997.
3. K. Socha, and M. Dorigo, "Ant colony optimization for continuous domains," European Journal of Operational Research, vol. 185, no. 3, pp. 1155-1173, 2008.
4. Z. Wu, N. Zhao, G. Ren, T. Quan, "Population declining ant colony optimization algorithm and its application", Expert Systems with Applications, vol. 36, no. 3, pp. 6276-6281, 2009.
5. M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," Computational Intelligence Magazine, IEEE, vol. 1, no. 4, pp. 28-39, 2006.
6. O. Cordon, F. Herrera, and T. Stützle, "A review on the ant colony optimization metaheuristic: basis, models and new trends," Mathware & Soft Computing, vol. 9, pp. 141-175, 2002.
7. B. Chandra Mohan, R. Baskaran, "A survey: Ant Colony Optimization based recent research and implementation on several engineering domain", Expert Systems with Applications, vol. 39, no. 4, pp. 4618-4627, 2012.
8. J. Yang, Y. Zhuang, "An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem", Applied Soft Computing, vol. 10, no. 2, pp. 653-660, 2010.

9. M. Dorigo, and G. Di Caro, "Ant colony optimization: a new meta-heuristic," Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, July, 1999.
10. Uğur, D. Aydın, "An interactive simulation and analysis software for solving TSP using Ant Colony Optimization algorithms", Advances in Engineering Software, vol. 40, no. 5, pp. 341-349, 2009.
11. M. Dorigo, V. Maniezzo, and A. Coloni, "Ant system: optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 26, no. 1, pp. 29-41, 1996.
12. S. M. Chen, C. Y. Chien, "Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques", Expert Systems with Applications, vol. 39, no. 12, pp. 14439-14450, 2011.
13. R. J. Mullen, D. Monekosso, S. Barman, and P. Remagnino, "A review of ant algorithms," Expert Systems with Applications, vol. 36, no. 6, pp. 9608-9617, 2009.
14. M. A. Mashaly, "Monomorium ant's trail pheromones: Glandular source, optimal concentration, longevity and specificity," Journal of Asia-Pacific Entomology, vol. 13, no. 1, pp. 23-26, 2010.
15. Bullheimer, R. F. Hartl, and C. Strauss, "A new rank-based version of the Ant System: a computational study," Central European Journal for Operations Research and Economics, vol. 7, no. 1, pp. 25-38, 1999.
16. T. Stützle, and H. H. Hoos, "MAX-MIN ant system," Future Generation Computer Systems, vol. 16, no. 8, pp. 889-914, 2000.
17. M. Dorigo, and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 53-66, 1997.
18. M. Dorigo, and T. Stützle, Ant Colony Optimization. Cambridge: The MIT Press, 2004, pp. 76-80.
19. O. Baskan, S. haldenbilen, H. Ceylan, H. Ceylan, "A new solution algorithm for improving performance of ant colony optimization", Applied mathematics and computation, vol. 211, no. 1, pp. 75-84, 2009.
20. T. Stützle, ACOTSP, Version 1.0. Available from <http://www.aco-metaheuristic.org/aco-code>, 2004.

Ping Guo (b. September 7, 1963) received his M.Sc. (1989) and PhD (2004) in Computer software and theory from Chongqing University. Now he is full professor of Computer Science at School of Computer Science, Chongqing University, China. His current research interests include different aspects of Artificial Intelligence and Biological computing model. He has (co-) authored 1 books and more than 130 papers.

Zhujin Liu (b. November 30, 1986) received her M.Sc. (2012) in Computer science and technology from Chongqing University of Chongqing. Now he is an engineer of access network in Huawei Company. His current research interests include multicast, gpon and Qos. He has (co-) authored more than 5 papers.

Received: March 02, 2012; Accepted: December 03, 2012.

Optimal Node Placement of Industrial Wireless Sensor Networks Based on Adaptive Mutation Probability Binary Particle Swarm Optimization Algorithm

Ling Wang^{1,2}, Wei Ye¹, Haikuan Wang¹, Xiping Fu¹, Minrui Fei¹, and Muhammad Ilyas Menhas¹

¹Shanghai Key Laboratory of Power Station Automation Technology, School of Mechatronics Engineering and Automation, Shanghai University, 200072 Shanghai, China

{wangling, yeweyish, hkwang, fuxp, mrfei, Ilyasmenhas}@shu.edu.cn

²Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, 32611 Florida, USA

Abstract. Industrial Wireless Sensor Networks (IWSNs), a novel technique in industry control, can greatly reduce the cost of measurement and control and improve productive efficiency. Different from Wireless Sensor Networks (WSNs) in non-industrial applications, the communication reliability of IWSNs has to be guaranteed as the real-time field data need to be transmitted to the control system through IWSNs. Obviously, the network architecture has a significant influence on the performance of IWSNs, and therefore this paper investigates the optimal node placement problem of IWSNs to ensure the network reliability and reduce the cost. To solve this problem, a node placement model of IWSNs is developed and formulized in which the reliability, the setup cost, the maintenance cost and the scalability of the system are taken into account. Then an improved adaptive mutation probability binary particle swarm optimization algorithm (AMPBPSO) is proposed for searching out the best placement scheme. After the verification of the model and optimization algorithm on the benchmark problem, the presented AMPBPSO and the optimization model are used to solve various large-scale optimal sensor placement problems. The experimental results show that AMPBPSO is effective to tackle IWSNs node placement problems and outperforms discrete binary Particle Swarm Optimization (DBPSO) and standard Genetic Algorithm (GA) in terms of search accuracy and the convergence speed with the guaranteed network reliability.

Keywords: industrial wireless sensor networks, node placement, binary particle swarm optimization, adaptive mutation.

1. Introduction

In harsh industrial environments, a large number of hazards ranging from strong mechanical vibrations, high temperatures, fragile surfaces, noisy electrical effects and even explosive gases may occur. Although wired industrial communication systems, such as fieldbus systems and wired Highway Addressable Remote Transducer (HART), have been used successfully in the field of factory automation and process automation [1], [2], on one hand, it is still difficult to install the wiring in these harsh industrial environments. On the other hand, the installation and maintenance of cables and sensors are much more expensive than sensors themselves for the traditional wired control systems. In recent years, wireless sensor networks (WSNs) have been applied in numerous fields, such as space exploration and border protection. By deploying wireless sensors to operate unattended in harsh industrial environments, it would be possible to avoid the risks to human lives and decrease the cost of the applications [3]. Thus, the industrial wireless communication technology has drawn increasing attention and been applied to factory automation and industrial process control [4], [5], [6], [7]. With Industrial Wireless Sensor Networks (IWSNs), engineers can collect information from where it previously has been economically or technologically infeasible, and therefore the process would be enhanced with respect to quality and quantity further. Now, IWSNs have been applied in food industry, beverages industry, pharmaceuticals industry, oil industry, gas industry, chemical industry, mining industry, refining industry, power plants, pulp industry, etc. The industrial applications of IWSNs demonstrate that wireless installation typically costs as much as 50 percent less than the wired alternative. Meanwhile, IWSNs can improve efficiency and be applied to the fields where the wired system cannot be installed.

However, there are still obstacles for the large-scale applications of IWSNs, among which the reliability of networks is the absolute requirement for most industrial systems. Thus this paper develops an optimal node placement model for the cluster-based IWSNs in which the reliability, cost and scalability of the network are taken into account. To obtain the best performance, a novel adaptive mutation probability binary Particle Swarm Optimization algorithm (AMPBPSO) is proposed to solve the optimal node placement problem of large-scale IWSNs with the model.

The rest of this paper is organized as follows. Section 2 introduces the related research work. Section 3 presents the node placement model of IWSNs where the objective function is formulated and introduced in detail. Section 4 specifies the optimization mechanism and procedure of AMPBPSO. The implementation of solving the optimal node placement based on AMPBPSO is addressed in Section 5. In Section 6, AMPBPSO and the developed model are first verified on the benchmark problem. Then AMPBPSO is utilized to solve the optimal node placement problems of IWSNs, and the results are compared with those of discrete binary Particle Swarm Optimization (DBPSO) and Genetic Algorithm (GA). Finally, the conclusions and future work are drawn in Section 7.

2. Related Work

Nowadays, Industrial Wireless Sensor Networks (IWSNs) have become a new research hotspot in industrial control technologies. To improve the performance of IWSNs and meet requirements in industrial applications, researchers developed various strategies to enhance and extend IWSNs. Heo et al. [8] proposed a new energy aware routing protocol for IWSNs in which real-time and reliable delivery as well as energy saving were considered. Park et al. [9] presented an adaptive protocol for IWSNs, called Breath, which ensured a desired packet delivery and delay probabilities while minimizing the energy consumption of network where the reliability of the packet were treated as the constraint. Bertocco et al. [10] developed a suitable testbed enlisting IEEE 802.14.5 wireless sensor nodes for studying the interference to optimize the network setup. Considering the high bit error rate characteristics of wireless channel due to harsh conditions like attenuation, noise and channel fading, Balasubramanian et al. [11] offered a novel real-time Medium Access Control (MAC) protocol that was specifically tailored to the requirement of the industrial environments. Villaverde et al. [12] introduced a route selection algorithm, named InRout, where local information was shared among neighboring nodes of IWSNs to enable efficient route selection. In their work, route selection is described as a multi-armed bandit task and Q-learning techniques are adopted to gain the best solution with low overhead, and thus it can improve the reliability and typical quality of service. Chen et al. [13] depicted a new distributed estimation and collaborative control scheme for wireless sensor and actuator networks to make up for the problems caused by the unreliable wireless and multi-hop communication among sensors and actuators in IWSNs. Yu et al. [14] discussed the use of forward error correction (FEC) codes in IWSNs to improve the link reliability and reduce the number of retransmissions in harsh industrial environments. They proposed a FEC scheme suitable for MAC level protection in which the packet was divided into groups and encoded using systematic FEC codes. Xing et al. [15] optimized the MAC protocol and presented a modified multi-channel one for IWSNs which could achieve more reliable communication in an energy and bandwidth efficient way.

Although the previous work has improved the reliability of IWSNs based on protocol modification and routing optimization, it is still necessary to carefully design and optimize the node placement scheme of IWSNs from the system point of view as it directly determines the final performance of networks especially for large-scale IWSNs. Actually, node placement has drawn increasing attention in non-industrial applications of WSNs, and various deployment strategies and algorithms have been proposed to construct sensor networks for different design goals, such as the minimum cost, highest energy efficiency, network coverage and connectivity of WSNs. However, the optimal node placement of WSNs has been proven to be an NP-hard problem [16], [17], and therefore evolutionary algorithms have been studied and applied to node deployment problems, which have shown the outstanding performance in solving a wide variety of NP-hard problems. As the most

widely used evolutionary algorithm, Genetic Algorithms (GAs) have been successfully used to design the node placement of WSNs. Ferentinos and Tsiligiridis [18] used a GA to design WSNs to fulfill the existent connectivity constraints and incorporate energy-conservation to guarantee maximum life span of the network. Jia et al. [19] investigated the coverage control scheme based on a multi-objective GA in which the minimum number of sensors was selected in a densely deployed environment while preserving full coverage of networks. Hu et al. [20] developed a hybrid approach of combining a genetic algorithm with schedule transition operation to maximizing the lifetime of sensor networks. To achieve better results, various meta-heuristic algorithms, such as Particle Swarm Optimization (PSO) [21], [22], Differential Evolution (DE) [23], Harmony Search [24] and Ant Colony Optimization (ACO) [25], [26] have been applied to the optimal design of WSNs. However, the previous works mainly focus on optimizing node placement for maximizing the lifetime of networks due to the characteristics of WSN applications. Different from general WSNs in the non-industrial applications [27], the demand of energy saving in IWSNs is alleviated as the network is maintained and the battery of nodes can be replaced. But IWSNs have stringent requirement of the communication reliability because the real-time field data are transmitted to the control system through IWSNs. Therefore, the network reliability should be especially concerned in the node placement of IWSNs, and the node placement strategies for the WSNs mentioned above are unsuitable for IWSNs. Now only few attempts have been made on optimizing the node deployment in IWSNs to satisfy the network reliability needs. And as far as we know, no one has addressed this problem utilizing intelligent optimization algorithms.

3. Node Placement of IWSNs

3.1. Communication model of cluster-based IWSNs

The node placement problem of IWSNs with the two-tiered clustering architecture is addressed in this work. The lower-layer is the single-hop communication between the sensor node and its cluster-head node. The upper-layer is the routing among cluster-heads to the base station via multi-hopping. In IWSNs, each node (including the sensor node and cluster-head node) has a maximum communication range. The communication radii of the sensor node and cluster-head node are denoted as R_s and R_{CH} , respectively. Generally, the communication capacity of the cluster-head node is more powerful than that of the regular sensor node. For instance, assuming that three sensor nodes (S1, S2 and S3) and three cluster-head nodes (CH1, CH2 and CH3) placed in the field as Fig. 1, the communication radii are constant and the communication ranges of them can be determined. The distance

Optimal Node Placement of Industrial Wireless Sensor Networks Based on Adaptive Mutation Probability Binary Particle Swarm Optimization Algorithm

between the sensor node S1 and the cluster-head node CH1 or CH2 is below R_S , that is, CH1 and CH2 are covered by the communication circle of S1, and thus the monitored data of S1 can be reliably transmitted to CH1 or CH2. However, the data transmitted from S2 will not be received by CH2 as the location of CH2 is beyond the maximum communication range of S2. In case of CH1 running out of its energy and the data of any sensor cannot be transmitted to the base station, which will make the whole network break down and cause the shutdown of the industry system or even hazard. Thus the extra nodes have to be deployed for redundancy to guarantee the essential reliability and robustness of IWSNs. Besides, the additional cluster-head nodes can improve the balance of load and extend the lifetime of batteries, which reduces the maintenance work of IWSNs. In summary, two questions, i.e. how many cluster-head nodes are needed at least and how to place them, should be considered carefully to satisfy the design goals of IWSNs.

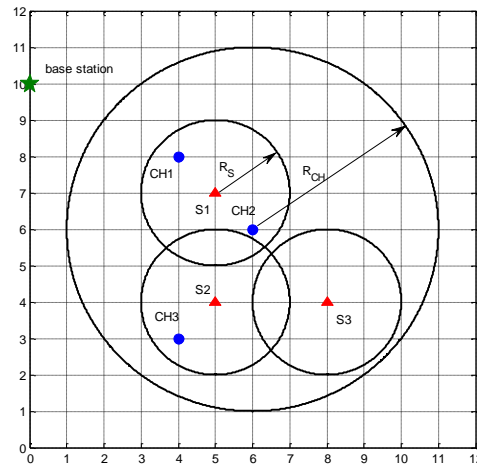


Fig. 1. A sample on the communication model of sensor nodes (▲) and cluster-head nodes (●)

3.2. Optimal node placement model of IWSNs

The main objective of designing the architecture of IWSNs is to guarantee the reliability of IWSNs as well as minimize the investment cost and the maintenance cost of networks. In addition, the maximum communication load of cluster-head nodes also need be considered for the scalability and real time of the system. Therefore, the optimal node placement model of IWSNs is a constrained multi-objective optimization problem and can be formulated as (1)-(2).

$$f = \alpha \times C + \beta \times SD_{CL} . \tag{1}$$

$$s.t. \begin{cases} MIN_{CH}^S = L_{SN} \geq 2 \\ MIN_{CH}^{CH} = L_{CN} \geq 2 \\ LCH_j \leq MCL - 1 \end{cases} . \tag{2}$$

where C is the setup cost of IWSNs; SD_{CL} is the standard deviation of cluster-head communication load which indicates the maintenance cost of IWSNs; α and β are the weighting factors with $\alpha + \beta = 1$. MIN_{CH}^S is the minimum communicating cluster-head number of all the sensors, and MIN_{CH}^{CH} is the least communicating cluster-head number of all the cluster-head nodes. L_{SN} , L_{CN} and MCL represent the sensor node reliability constraint, the cluster-head node reliability constraint and the maximum load constraint, respectively. LCH_j is the load number of the j -th cluster-head.

Reliability Constraint. To ensure the communication reliability of IWSNs, each node must have at least one redundant cluster-head node; that is, the cluster-head number for each node in IWSNs should be more than two as (3)

$$\begin{cases} MIN_{CH}^S = L_{SN} \geq 2 \\ MIN_{CH}^{CH} = L_{CN} \geq 2 \end{cases} . \tag{3}$$

where L_{SN} and L_{CN} are the pre-defined least numbers of cluster-heads for sensor nodes and cluster-head nodes, respectively.

As seen in Fig. 2, the sensor node S1 and the cluster-head node CH4 have one redundant cluster-head node, i.e. CH2 and CH6, respectively. When CH1 or CH5 fails, the data of S1 still can be transmitted to the base station successfully due to the assigned redundant cluster-head nodes, and therefore the reliability of IWSNs is improved and guaranteed.

Suppose that there are N_s sensor nodes and N_{ch} cluster-head nodes placed in the industry field and N_i ($i = 1, 2, \dots, N_s$) and M_j ($j = 1, 2, \dots, N_{ch}$) are the cluster-head node numbers of the i -th sensor node and the j -th cluster-head, respectively, MIN_{CH}^S and MIN_{CH}^{CH} can be calculated as (4)-(5).

$$MIN_{CH}^S = \min \{N_i, | i = 1, 2, \dots, N_s \} . \tag{4}$$

$$MIN_{CH}^{CH} = \min \{M_j, | j = 1, 2, \dots, N_{ch} \} . \tag{5}$$

However, a node need not transmit its data to every cluster-head node in its communication range. Thus, in this work every sensor node or cluster-head node only communicates with the nearest L_{SN} or L_{CN} cluster-head nodes. The nearest cluster-head node is used as the regular working one and the others are reserved as the redundant ones.

Maximum Load. Although the energy and communication capacity of cluster-head nodes are more powerful than those of sensor nodes, their load-driven capacity is still limited owing to the finite computation ability. Thus, the total load of cluster-head nodes should not be beyond the maximum communication load MCL . Furthermore, at least one configuration point should be reserved for the upgrading or maintenance of the system. Therefore, the maximum number of load nodes for each cluster-head node is defined as (6).

$$LCH_j \leq MCL - 1. \quad (6)$$

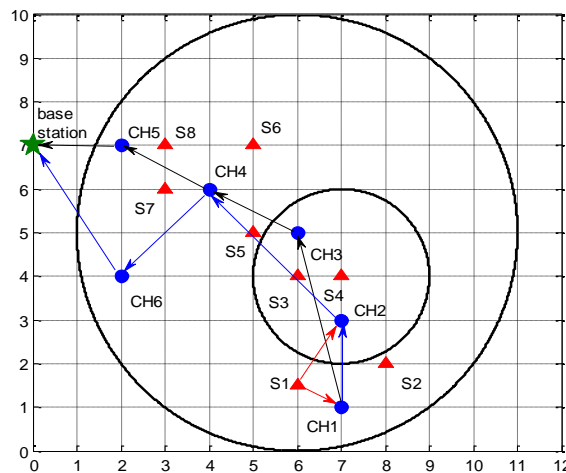


Fig. 2. An instance of the reliable IWSNs (the symbol \blacktriangle represents the sensor node and the symbol \bullet represents the cluster-head node)

Setup Cost. When constructing IWSNs, the setup cost should be reduced without decreasing the performances of the system, especially in the very large-scale applications. For the node placement in IWSNs, the change of the cost mainly lies in the investment of cluster-head nodes as the sensor nodes are already determined by the requirements of monitoring. Thus, the setup cost can be defined as (7)

$$C = P_{ch} \times N_{ch} . \quad (7)$$

where N_{ch} is the number of the cluster-head nodes used and P_{ch} is the weight of the cluster-head node number .

Maintenance Cost. The communication load equilibration of cluster-head nodes plays a key role in balancing the energy consumption. As the communication of node needs energy supply, the high-load cluster-head node consumes energy faster than the others. To insure the normal running of IWSNs, the battery of cluster-head node need be replaced before its

energy totally runs out. Thus, the unbalanced load uniformity will bring the frequent maintenance of IWSNs, which should be avoided. Therefore, the load uniformity of IWSNs is considered as the main metric of the maintenance cost to be optimized in this paper, which can be indicated by the standard deviation of the cluster-head load (SDCL) as (8)-(9),

$$SD_{CL} = \sqrt{\frac{\sum_{j=1}^{Nch} (LCH_j - ML_{CH})^2}{Nch - 1}} . \quad (8)$$

$$ML_{CH} = \frac{\sum_{j=1}^{Nch} LCH_j}{Nch} . \quad (9)$$

where ML_{CH} is the mean load of all the cluster-head nodes.

4. Adaptive Mutation Probability Binary Particle Swarm Optimization

Particle Swarm Optimization (PSO) algorithm was first proposed by Kennedy and Eberhart in 1995 [28], which has been studied and applied to solve numerous science and engineering problems successfully. It works with a group of particles. Each particle can be considered as a candidate solution and represented by a position vector $x_i = (x_{i1}, \dots, x_{ij}, \dots, x_{iD})$ in a D dimensional search space, which keeps on moving toward new points in the search space with the addition of a velocity vector $v_i = (v_{i1}, \dots, v_{ij}, \dots, v_{iD})$ to facilitate the search procedure. During the search process all particles move toward the areas of potential solutions by utilizing the cognitive and social learning components. The process is repeated until any prescribed stopping criterion is reached. After any iteration, each particle updates its position and velocity to achieve a better fitness value according to (10)-(11)

$$v_{ij}^{G+1} = w \times v_{ij}^G + c_1 \times r_1 \times (P_{ij}^G - x_{ij}^G) + c_2 \times r_2 \times (Pg_j^G - x_{ij}^G) . \quad (10)$$

$$x_{ij}^{G+1} = v_{ij}^{G+1} + x_{ij}^G . \quad (11)$$

where v_{ij}^G represents the velocity of the j -th element of the i -th individual at iteration G ; w is the inertia weight; c_1 and c_2 are called as the acceleration factors, usually valued as 2.0; r_1 and r_2 are the uniform random numbers between 0 and 1; x_{ij}^G represents the position of the j -th decision variable of the i -th individual at iteration G ; P_i^G and Pg^G denote the local best position of

the i -th individual and the global best position of the group until iteration G , respectively.

However, the standard PSO and majority of its variants are developed for optimization problems in continuous space, which cannot be used to solve discrete binary optimization problems directly. To extend its application fields, Kennedy and Eberhart proposed a discrete binary PSO (DBPSO) algorithm in 1997 [29]. DBPSO reserves the velocity updating formula (10), but the velocity vector is transformed into the probability of being “1” of each bit through the sigmoid limiting function as (12). Finally, the binary solution is generated based on this probability as (13)

$$\text{sigmoid}(v_{ij}^{G+1}) = \frac{1}{1 + e^{-v_{ij}^{G+1}}} \quad (12)$$

$$x_{ij}^{G+1} = \begin{cases} 1, & \text{if } \text{rand}() \leq \text{sigmoid}(v_{ij}^{G+1}) \\ 0, & \text{else} \end{cases} \quad (13)$$

where $\text{rand}()$ is a random number uniformly distributed in $[0,1]$.

However, DBPSO is easy to be trapped in the local optima. To make up for it, a probability binary Particle Swarm Optimization (PBPSO) algorithm was presented in our previous work, which has been proven to be efficient and effective for solving various optimization problems [30], [31], [32]. Considering that the optimal node placement of large-scale IWSNs is a high-dimensional constrained multi-objective NP-hard problem and its fitness evaluation is very time-consuming, a novel adaptive mutation PBPSO (AMPBPSO) is proposed in this paper into which an adaptive mutation operator is introduced to improve the search ability and convergence speed of the algorithm.

4.1. Initialization

AMPBPSO adopts the binary encoding. Assume the population size of AMPBPSO and the dimension of solutions are N and D , respectively. The i -th individual, i.e. a candidate solution, is denoted as $bx_i = (bx_{i1}, \dots, bx_{ij}, \dots, bx_{iD})$, where $i = 1, 2, \dots, N$; $j = 1, 2, \dots, D$; $bx_{ij} \in \{0,1\}$. The local best position of the i -th individual and the global best position of the group are denoted as $bp_i = (bp_{i1}, \dots, bp_{ij}, \dots, bp_{iD})$ and $bp_g = (bp_{g1}, \dots, bp_{gj}, \dots, bp_{gD})$, respectively. In the search process, the whole updating mechanism of PSO is reserved. However, the original position vector $x_i = (x_{i1}, \dots, x_{ij}, \dots, x_{iD})$ of PSO is redefined as the pseudo-probability of being “1” in AMPBPSO, and the velocity $v_i = (v_{i1}, \dots, v_{ij}, \dots, v_{iD})$ represents the change of the pseudo-probability x_i .

Ling Wang et al.

bx_{ij} , v_{ij} and x_{ij} of AMPBPSO is initialized as (14)-(16),

$$bx_{ij}^0 = \begin{cases} 1, & \text{if } rand() \leq 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

$$v_{ij}^0 = (v_{\max} - v_{\min}) \times rand() + v_{\min} \quad (15)$$

$$x_{ij}^0 = (x_{\max} - x_{\min}) \times rand() + x_{\min} \quad (16)$$

where $rand()$ is a random number uniformly distributed in $[0,1]$; v_{\min} and v_{\max} are the lower and upper velocity bound; x_{\min} and x_{\max} are the pre-defined lower and upper pseudo-probability bound, respectively.

4.2. Population Updating

Like PSO, the velocity v_i and the pseudo-probability x_i of AMPBPSO is dynamically updated according to (17)-(18),

$$v_{ij}^{G+1} = w \times v_{ij}^G + c_1 \times r_1 \times (bP_{ij}^G - bx_{ij}^G) + c_2 \times r_2 \times (bPg_j^G - bx_{ij}^G) \quad (17)$$

$$x_{ij}^{G+1} = v_{ij}^{G+1} + x_{ij}^G \quad (18)$$

Note that the binary solution bx_{ij}^G , bP_{ij}^G and bPg_j^G are used here instead of x_{ij}^G , P_{ij}^G and Pg_j^G in the standard PSO. If $x_{ij}^{G+1} > x_{\max}$ or $x_{ij}^{G+1} < x_{\min}$, we fix x_{ij}^{G+1} to x_{\max} or x_{\min} , respectively. After the updating of the pseudo-probability, the real probability pr_{ij} is calculated and adopted to generate a new binary solution through the probability estimation operator as (19)-(20).

$$pr_{ij}^{G+1} = (x_{ij}^{G+1} - x_{\min}) / (x_{\max} - x_{\min}) \quad (19)$$

$$bx_{ij}^{G+1} = \begin{cases} 1, & \text{if } rand() \leq pr_{ij}^{G+1} \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

4.3. Adaptive Mutation

Although the standard PBPSO is efficient and effective, the performance of PBPSO is not quite satisfactory on high-dimensional problems. To improve the search ability and avoid premature convergence, an adaptive mutation operator is adopted which operates as (21)-(22)

$$\text{if } \text{rand}() < p_m, bx_{ij}^G = \begin{cases} 1, & \text{if } bx_{ij}^G = 0 \\ 0, & \text{if } bx_{ij}^G = 1 \end{cases} \quad (21)$$

$$p_m = \frac{1}{D} \times (0.05 + \frac{1.45}{G_{\max}} \times G) \quad (22)$$

where p_m is the mutation rate of each bit. In AMPBPSO, p_m adaptively varies based on the solution dimension D and the iteration number G ; that is, p_m linearly increases from $\frac{0.05}{D}$ to $\frac{1.5}{D}$ in the search process. The introduction of the dimension D is to enhance the scalability of the algorithm, and the linear increment of p_m can retain the diversity of the population. The final value of p_m , i.e. $\frac{1.5}{D}$, is adopted to ensure the accuracy of local search.

The evolutionary mechanism of AMPBPSO can be described as Fig. 3. Unlike DBPSO discarding the position updating formula and using the velocity as the probability to generate solutions, AMPBPSO reserves the original position of PSO by re-defining it as the pseudo-probability and therefore the entire updating mechanism of PSO is inherited. All the binary individuals of AMPBPSO have the pseudo-probability states as well as velocity states, which are updated following the framework of PSO and adopted to yield new candidate solutions using the probability estimation operator. For one thing, as the pseudo-probability of each individual is updated from generation to generation according to the information from its historical and global best individuals, AMPBPSO has the strong global exploration ability. For another, AMPBPSO adopts the pseudo-probability vector to create the new solution, thus it can maintain population diversity effectively. Moreover, the adaptive mutation operator enhances further the capability of local search as well as escaping from the local optima. Therefore, AMPBPSO can achieve an appropriate balance between exploration and exploitation.

After the offspring solutions are generated, the fitness of each individual is calculated and the local best position as well as the global best position is replaced if the fitness value of the new one is better. In summary, the procedure of AMPBPSO can be depicted as Fig. 4.

5. Implementation of AMPBPSO for the Node Placement in IWSNS

5.1. Solution Representation

For industrial applications, the sensor nodes are deployed for monitoring and their locations are already determined. Thus, the variables included in the optimal node placement of IWSNs are the number and positions of the cluster-head nodes used. Each individual of AMPBPSO specifies the number and positions of cluster-head nodes encoded as a binary vector, where a bit “1” indicates that a cluster-head node is placed at the corresponding grid conjunction while a bit “0” denotes the opposite. Fig. 5 displays an example individual, which represents a grid with four rows and four columns. The grid junctions are encoded row by row into the binary vector, and therefore a 16-bits binary solution can be used to delineate a corresponding candidate deployment scheme.

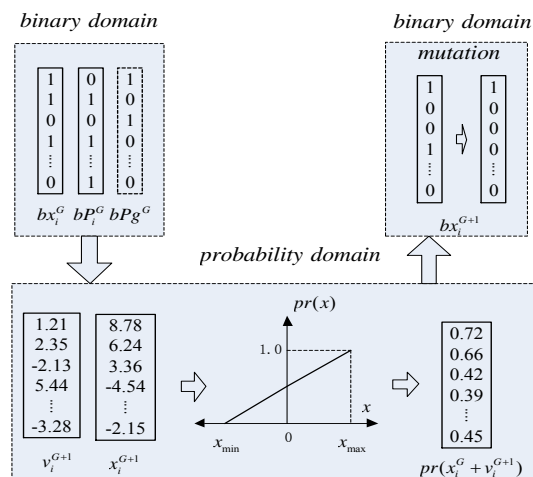


Fig. 3. Evolutionary mechanism of AMPBPSO

5.2. Fitness Function

As mentioned above, the optimal node placement of IWSNs is a constrained optimization problem. To deal with the constraints, the penalty function method is introduced to fix the fitness and lead the algorithm to search in the feasible areas effectively as (23)-(26).

$$f_p = f + p_1 + p_2 + p_3 . \quad (23)$$

$$p_1 = \sum_{i=1}^{N_S} \max \{0, c_1 \times (L_{SN} - N_i)\} . \quad (24)$$

$$p_2 = \sum_{j=1}^{N_{CH}} \max \{0, c_1 \times (L_{CN} - M_j)\} . \quad (25)$$

$$p_3 = \sum_{j=1}^{N_{CH}} \max \{0, c_2 \times (LCH_j + 1 - MCL)\} . \quad (26)$$

where f_p is the final fitness value; p_1 , p_2 and p_3 are the penalty items of the sensor node reliability constraint, the cluster-head node reliability constraint and the maximum load number constraint, respectively. c_1 is the penalty coefficient of the reliability constraints, and c_2 is the penalty coefficient for violating the limitation of the communication load.

5.3. Optimal node placement of IWSNs based on AMPBPSO

Based on the pre-defined fitness function, AMPBPSO iteratively generates new candidate solutions to search the optimal node placement scheme of IWSNs. The whole procedure of designing the optimal node placement with AMPBPSO can be described as follows:

Step 1: Set the model parameters of IWSNs, such as R_S , R_{CH} and MCL .

Step 2: Initialize AMPBPSO including weight factor w , learning factors c_1 and c_2 , the maximum velocity v_{max} , the binary population bx , the local best individual bP_i , the global best individual bP_g , the velocity v , the pseudo-probability x and its lower/upper bound x_{min}/x_{max} .

Step 3: Update the velocity vector and pseudo-probability vector of each individual according to (17)-(18).

Step 4: Generate offspring individuals, i.e. new binary solutions, by performing the probability estimation operator as (19)-(20) and the adaptive mutation operator as (21)-(22).

Step 5: Calculate the fitness value of each individual according to (23)-(26).

Step 6: Update the local best individual bP_i and the global best individual bP_g .

Step 7: Stop the iteration and output the optimal solution if the termination criteria are met, otherwise go to Step 3.

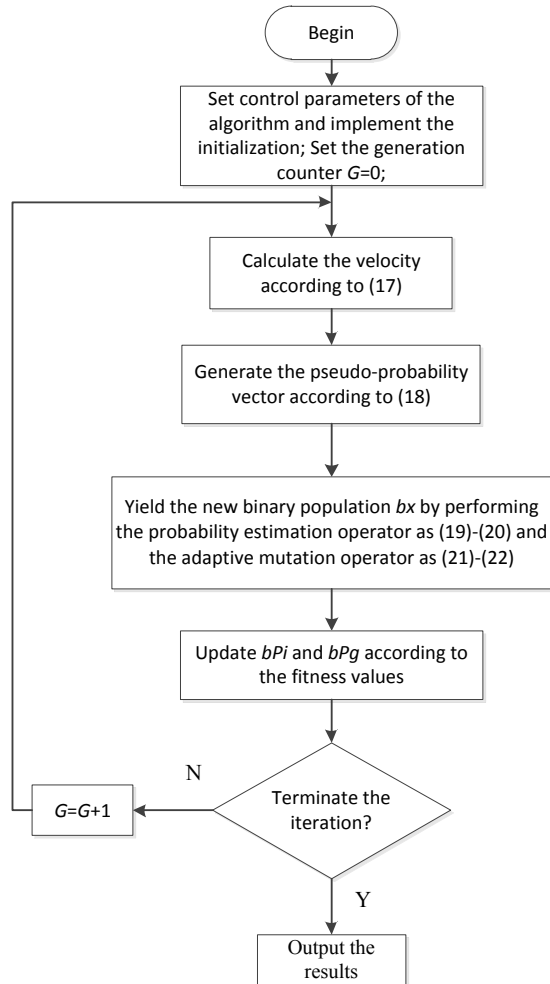


Fig. 4. The flowchart of AMPBPSO

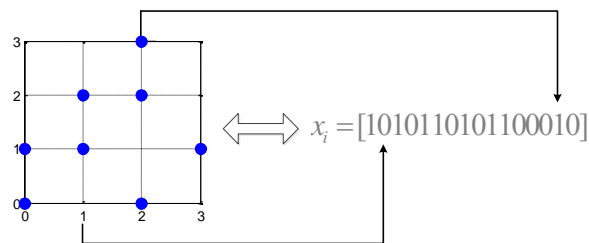


Fig. 5. Binary encoding for the node placement of IWSNs

6. Experiments and Analysis

6.1. Node Placement Benchmark Problem of IWSNs

To test the effectiveness of the developed node placement model for IWSNs, a small-scale node deployment benchmark problem is designed as Fig. 6. The grid of this problem is 10×10 and the positions of the sensor nodes are pre-placed and represented as the triangles in the figure. The communication radii of the sensor node and the cluster-head node are set as $R_S=2$ and $R_{CH}=5$, respectively. The reliability constraints are $L_{SN}=2$ and $L_{CN}=2$. Therefore, the optimal solution of this benchmark is unique, and the corresponding positions of the deployed cluster-heads are marked as the circles. AMPBPSO, DBPSO [29] and GA [18] with the recommended parameters in Table 1 were adopted to tackle the benchmark problem. The population size and the maximum generation of each algorithm were set as $N=200$ and $G_{max}=100$, respectively.

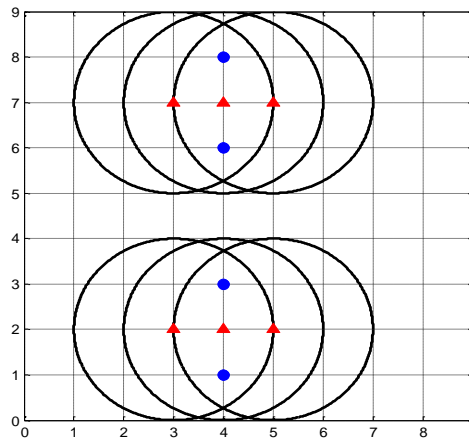


Fig. 6. The optimal placement scheme of the standard benchmark problem (\blacktriangle represents the sensor node, \bullet represents the cluster-head node, the circles are the communication ranges of the sensor nodes)

The experiments were run 10 times independently. The experimental results, i.e., the best fitness (Best), the mean fitness (Mean) and the standard deviation (Std_Dev) are listed in Table 2. “+” of the t-test results indicates that AMPBPSO is significantly better than the compared algorithm at the 95% confidence; “-” represents that AMPBPSO is significantly worse than the compared algorithm; and “ \approx ” denotes that the difference is not significant. Table 2 indicates that the proposed model as well as AMPBPSO is effective for the node placement of IWSNs as the unique optimal solution of the

problem was found out successfully. However, AMPBPSO did not find the best solution with 100% success rate while DBPSO and GA totally failed to reach the global best solution, which illustrates that the optimal sensor placement problem is very complicated. The t-test results show that AMPBPSO is significantly superior to GA and DBPSO on this benchmark which demonstrates that AMPBPSO has the better global search ability and the adaptive mutation operator effectively enhances the performance of the algorithm.

Table 1. Parameters of AMPBPSO, DBPSO AND GA

Algorithms	Parameters
AMPBPSO	$w = 0.8, c_1 = c_2 = 2.0, v_{\max} = 6.0, x_{\min} = -20, x_{\max} = 20,$ $p_m = \frac{1}{D} \times (0.05 + \frac{1.45}{G_{\max}} \times G)$
DBPSO [29]	$w = 0.8, c_1 = c_2 = 2.0, v_{\max} = 6.0$
GA [18]	$p_s = 1.0, p_c = 0.8, p_m = 0.005$

Table 2. Results of AMPBPSO, DBPSO and GA on the node placement benchmark problem

Algorithm		AMPBPSO	DBPSO	GA
Best	fp	3.32	12.90	15.30
	p1	0	0	0
	p2	0	0	0
	p3	0	0	0
Mean±	fp	4.01±0.51	14.18±1.13	17.93±1.61
	p1	0±0	0±0	0±0
	p2	0±0	0±0	0±0
Std_Dev	p2	0±0	0±0	0±0
	p3	0±0	0±0	0±0
Sucess rate		90%	0%	0%
t-test		/	+	+

6.2. Large-scale Node Placement of IWSNs

The number and positions of sensor nodes in IWSNs are designed and determined for monitoring industrial systems, and therefore they depends on the specific application and vary for the different systems. Thus, without loss of generality, sensor nodes are assumed to be randomly distributed in the industrial field, that is, the positions of sensor nodes are generated randomly in this paper. Six large-scale node placement problems, i.e. 200m×200m, 400m×400m and 600m×600m with nodes densities $\delta = 0.2$ and $\delta = 0.5$, were

Optimal Node Placement of Industrial Wireless Sensor Networks Based on Adaptive Mutation Probability Binary Particle Swarm Optimization Algorithm

yielded and adopted as the benchmarks. The communication radii of the sensor node and the cluster-head node were set as $R_S=50m$ and $R_{CH}=100m$, respectively. The reliability constraints were $L_{SN}=2$ and $L_{CN}=2$. The required accuracy of the node location information is application-dependent. On the one hand, the communication reliability is extremely important for industrial applications and the radius of wireless sensors is not exact due to various kinds of interferences in the industry environment. Thus, the high accurate location of cluster-head nodes for IWSNs is unnecessary. On the other hand, with the improvement on the accuracy of the grid onto which the industrial field is mapped, the search space of the node placement problem will increase exponentially. Based on the above factors, the grid cell was set as 10m in this paper.

Table 3. Results of AMPBPSO, DBPSO and GA on the 200m×200m node placement problems

Density		0.2			0.5		
Algorithm		AMPBPSO	DBPSO	GA	AMPBPSO	DBPSO	GA
Best	fp	16.15	72.06	78.48	27.27	36.91	40.96
	p1	0	0	0	0	0	0
	p2	0	0	0	0	0	0
	p3	0	0	0	0	0	0
Mean ± Std_Dev	fp	18.18 ±1.77	74.55 ±2.36	85.13 ±2.77	27.26 ±0	39.09 ±2.16	46.44 ±2.39
	p1	0±0	0±0	0±0	0±0	0±0	0±0
	p2	0±0	0±0	0±0	0±0	0±0	0±0
	p3	0±0	0±0	0±0	0±0	0±0	0±0
t-test		/	+	+	/	+	+

Table 4. Results of AMPBPSO, DBPSO and GA on the 400m×400m node placement problems

Density		0.2			0.5		
Algorithm		AMPBPSO	DBPSO	GA	AMPBPSO	DBPSO	GA
Best	fp	200.09	397.70	422.50	112.90	228.89	246.50
	p1	0	0	0	0	0	0
	p2	0	0	0	0	0	0
	p3	0	0	0	0	0	0
Mean± Std_Dev	fp	209.52 ±7.20	404.02 ±4.22	430.34 ±12.57	118.90 ±3.35	235.77 ±3.23	254.34 ±9.38
	p1	0±0	0±0	0±0	0±0	0±0	0±0
	p2	0±0	0±0	0±0	0±0	0±0	0±0
	p3	0±0	0±0	0±0	0±0	0±0	0±0
t-test		/	+	+	/	+	+

AMPBPSO, DBPSO and GA with the recommended parameters were used to solve the large-scale node placement problems for a fair comparison. All the algorithms were repeated 10 times on each problem independently. Tables 3-5 list the experimental results which show that AMPBPBO surpasses DBPSO and GA on all the problems and the average results of AMPBPSO are even better than the best ones of DBPSO and GA. The t-test results also demonstrate that AMPBPSO is significantly superior to GA and DBPSO.

Table 5. Results of AMPBPSO, DBPSO and GA on the 600m×600m node placement problems

Density		0.2			0.5		
Algorithm	AMPBPSO	DBPSO	GA	AMPBPSO	DBPSO	GA	
Best	fp	662.47	969.70	1009.70	359.28	580.90	600.10
	p1	0	0	0	0	0	0
	p2	0	0	0	0	0	0
	p3	0	0	0	0	0	0
Mean ± Std_Dev	fp	671.83 ±13.28	986.10 ±5.36	1026.74 ±13.04	366.59 ±6.62	591.14 ±4.11	611.94 ±8.18
	p1	0±0	0±0	0±0	0±0	0±0	0±0
	p2	0±0	0±0	0±0	0±0	0±0	0±0
	p3	0±0	0±0	0±0	0±0	0±0	0±0
t-test	/	+	+	/	+	+	

The average convergence curves of each algorithm on the six problems are drawn in Fig. 7-12. From Fig. 7-12, we can clearly observe that AMPBPSO has the best search ability and outperforms DBPSO and GA in terms of solution quality and the convergence speed for the optimal node placement of IWSNs.

7. Conclusions and Future Work

In this paper, we have investigated the optimal node placement problem of IWSNs. Different from the non-industrial applications of WSNs, the reliability of the wireless communication has to be guaranteed as IWSNs are utilized to monitor the manufacturing process. To meet the specific need on the reliability of IWSNs, a node placement model of IWSNs is developed and formulized in which the reliability, the setup cost, the maintenance cost and scalability of the system are taken into account. Given the high dimension and NP-hard characteristic of the large-scale IWSNs node placement, an improved adaptive mutation probability binary Particle Swarm Optimization algorithm (AMPBPSO) is proposed to solve the problems where the adaptive mutation operator is introduced to keep the diversity of the population as well

Optimal Node Placement of Industrial Wireless Sensor Networks Based on Adaptive Mutation Probability Binary Particle Swarm Optimization Algorithm

as enhance local search. The presented algorithm and model were firstly verified on the small-scale node placement benchmark problem. The results of the benchmark demonstrate the effectiveness and efficiency of the model and algorithm. Finally, the proposed model and AMPBPSO were used to tackle the large-scale node placement problems of IWSNs. DBPSO and GA were also applied to all the problems for a comparison. The simulation results indicate that AMPBPSO can effectively solve the optimal node placement problems with the guarantee of the reliability and outperforms DBPSO and GA in terms of search accuracy and the convergence speed.

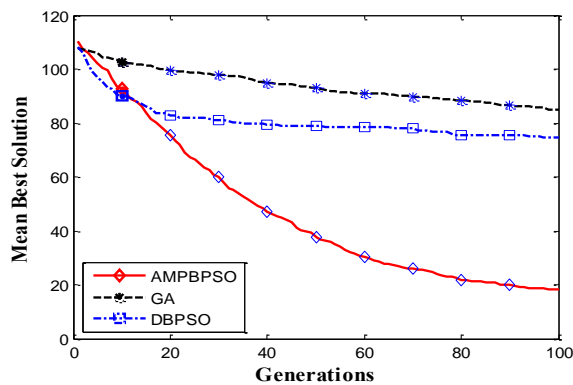


Fig. 7. Convergence curves of the 200m×200m node placement problem with $\delta = 0.2$

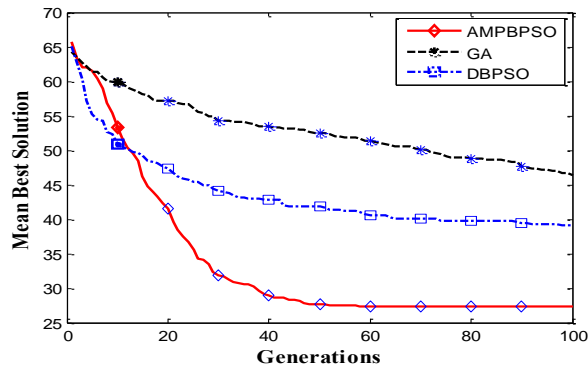


Fig. 8. Convergence curves of the 200m×200m node placement problem with $\delta = 0.5$

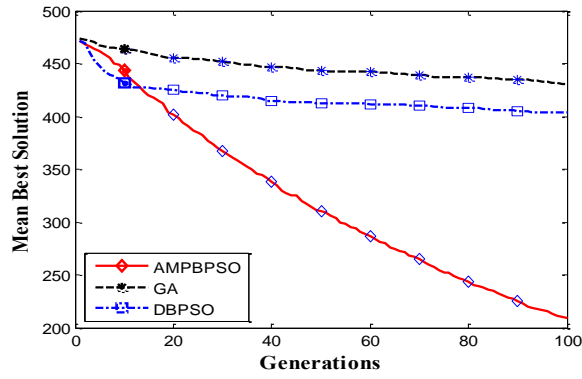


Fig. 9. Convergence curves of the 400m×400m node placement problem with $\delta=0.2$

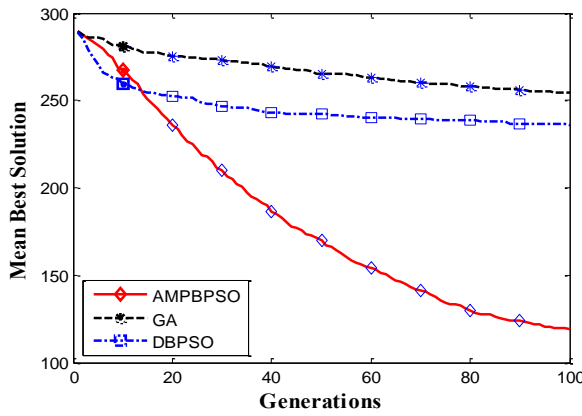


Fig. 10. Convergence curves of the 400m×400m node placement problem with $\delta=0.5$

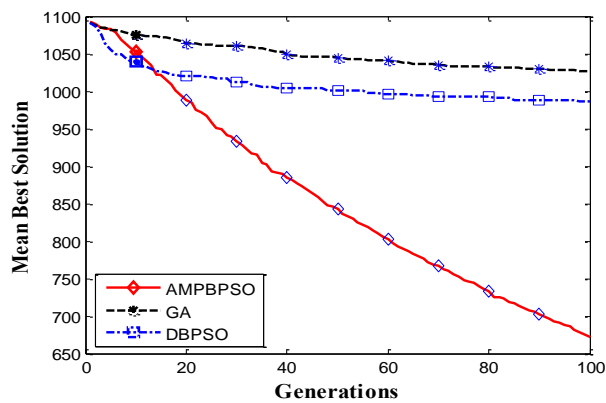


Fig. 11. Convergence curves of the 600m×600m node placement problem with $\delta=0.2$

Optimal Node Placement of Industrial Wireless Sensor Networks Based on Adaptive Mutation Probability Binary Particle Swarm Optimization Algorithm

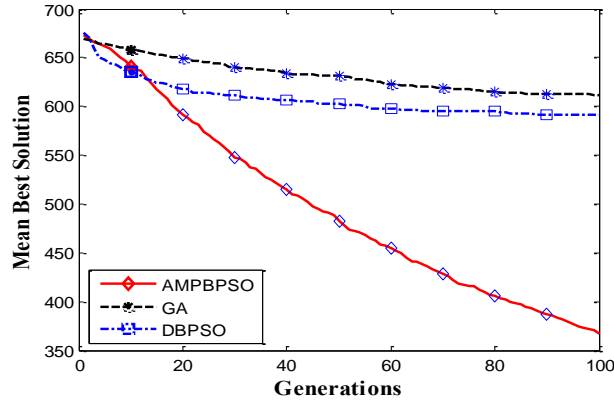


Fig. 12. Convergence curves of the 600m×600m node placement problem with $\delta=0.5$

The optimal node placement of IWSNs is a multi-objective optimization problem, and the setting of weighting factors of each optimization goal is a challenge for real applications. Thus, the Pareto-based approach has the advantage, which is a direction for the future research. Another direction of the future research is to improve the optimization model to meet the more requirements of industrial applications.

Acknowledgments. This work is supported by research funds for the Doctoral Program of Higher Education of China (20103108120008), the Projects of Shanghai Science and Technology Community (10ZR1411800 & 10JC1405000), National Natural Science Foundation of China (Grant No. 60834002 & 61074032), Shanghai University “11th Five-Year Plan” 211 Construction Project and the Mechatronics Engineering Innovation Group project from Shanghai Education Commission.

References

1. Zhuang, L. Q., Goh, K. M., Zhang, J. B.: The wireless sensor networks for factory automation: Issues and challenges. *IEEE Conference on Emerging Technologies and Factory Automation*, 141-148. (2007)
2. Lee, J. H., Kwon, T., Song, J. H.: Group Connectivity Model for Industrial Wireless Sensor Networks. *IEEE Transactions on Industrial Electronics*, Vol. 57, 1835-1844. (2010)
3. Senel, F., Younis, M. F., Akkaya, K.: Bio-inspired Relay Node Placement Heuristics for Repairing Damaged Wireless Sensor Networks. *IEEE Transactions on Vehicular Technology*, Vol. 60, 1835-1848. (2011)
4. Gungor, V. C., Hancke, G. P.: Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches. *IEEE Transactions on Industrial Electronics*, Vol. 56, 4258-4265. (2009)
5. Ramanjaneyulu, B. S., Gopinathan, E.: Wireless sensor networks in industrial automation. *IETE Technical Review*, Vol. 22, 139-149. (2005)

Ling Wang et al.

6. Flammini, A., Ferrari, P., Marioli, D., Sisinni, E., Taroni, A.: Sensor networks for industrial applications. The 2nd International Workshop on Advances in Sensors and Interface, 1-15. (2007)
7. Flammini, A., Ferrari, P., Marioli, D., Sisinni, E.: Wired and wireless sensor networks for industrial applications. *Microelectronics Journal*, Vol. 40, 1322-1336. (2009)
8. Heo, J., Hong, J., Cho, Y.: EARQ: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks. *IEEE Transactions on Industrial Informatics*, Vol. 5, 3-11. (2009)
9. Park, P., Fischione, C., Bonivento, A., Johansson, K. H., SangiovanniVincent, A.: Breath: an adaptive protocol for industrial control applications using wireless sensor networks. *IEEE Transactions on Mobile Computing*, Vol. 10, 821-838. (2011)
10. Bertocco, M., Gamba, G., Sona, A., Vitturi, S.: Experimental characterization of wireless sensor networks for industrial applications. *IEEE Transactions on Instrumentation and Measurement*, Vol. 57, 1537-1546. (2008)
11. Balasubramanian, K., Anil Kumar, G. S., Manimaran, G., Wang, Z.: A novel real-time MAC protocol exploiting spatial and temporal channel diversity in wireless industrial networks. *Lecture Notes in Computer Science*, Vol. 4297, 534-546. (2006)
12. Villaverde, B. C., Rea, S., Pesch, D.: InRout—A QoS aware route selection algorithm for industrial wireless sensor networks. *Ad Hoc Networks*, Vol. 10, 458-478. (2011)
13. Chen, J., Cao, X., Cheng, P., Xiao, Y., Sun, Y.: Distributed collaborative control for industrial automation with wireless sensor and actuator networks. *IEEE Transactions on Industrial Electronics*, Vol. 57, 4219-4230. (2010)
14. Yu, K., Gidlund, M., Akerberg, J., Bjorkman, M.: Reliable and Low Latency Transmission in Industrial Wireless Sensor Networks. *Procedia Computer Science*, Vol. 5, 866-873. (2011)
15. Xing, Z., Zeng, P., Wang, H.: Reliability, Capacity, and Energy Efficiency: A Comprehensively Optimized MAC Protocol for Industrial Wireless Sensor Networks. *Advances in Communication Systems and Electrical Engineering*, Vol. 4, 139-154. (2008)
16. Ke, W. C., Liu, B. H., Tsai, M. J.: Constructing a Wireless Sensor Network to Fully Cover Critical Grids by Deploying Minimum Sensors on Grid Points Is NP-Complete. *IEEE Transactions on Computers*, Vol. 56, pp. 710-715. (2007)
17. Wu, Q., Rao, N. S. V., Du, X., Iyengar, S. S.: On efficient deployment of sensors on planar grid. *Computer Communications*, Vol. 30, 2721-2734. (2007)
18. Ferentinos, K. P., Tsiligiridis, T. A.: Adaptive design optimization of wireless sensor networks using genetic algorithms. *Computer Networks*, Vol. 51, 1031-1051. (2007)
19. Jia, J., Chen, J., Chang, G., Tan, Z.: Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm. *Computers & Mathematics with Applications*, Vol. 57, 1756-1766. (2009)
20. Hu, X. M., Zhang, J., Yu, Y., Chung, H. S. H., Li, Y. L., Shi, Y. H., Luo, X. N.: Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks. *IEEE Transactions on Evolutionary Computation*, Vol. 14, 766-781. (2010)
21. Pradhan, P. M., Baghel, V., Panda, G., Bernard, M.: Energy efficient layout for a wireless sensor network using multi-objective particle swarm optimization. *IEEE International Advance Computing Conference*, 65-70. (2009)

Optimal Node Placement of Industrial Wireless Sensor Networks Based on Adaptive Mutation Probability Binary Particle Swarm Optimization Algorithm

22. Aziz, N. A. B. A., Moheemmed, A. W., Alias, M. Y.: A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram. *International Conference on Networking, Sensing and Control*, 602-607. (2009)
23. Hojjatoleslami, S., Aghazarian, V., Aliabadi, A.: DE Based Node Placement Optimization for Wireless Sensor Networks. *The 3rd International Workshop on Intelligent Systems and Applications*, 1-4. (2011)
24. Nezhad, S. E., Kamali, H. J., Moghaddam, M. E.: Solving K-Coverage Problem in Wireless Sensor Networks Using Improved Harmony Search. *2010 International Conference on Broadband, Wireless Computing, Communication and Applications*, 49-55. (2010)
25. Nematy, F., Rahmani, A. M., Teshnelab, M., Rahmani, N.: Ant Colony Based Node Deployment and Search in Wireless Sensor Networks. *2010 International Conference on Computational Intelligence and Communication Networks*, 363-366. (2010)
26. Li, D., Liu, W., Cui, L.: EasiDesign: an Improved Ant Colony Algorithm for Sensor Deployment in Real Sensor Network System. *2010 IEEE Global Telecommunications Conference*, 1-5. (2010)
27. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., Anderson, J.: Wireless Sensor Networks for Habitat Monitoring. *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 88-97. (2002)
28. Kennedy, J., Eberhart, R.: Particle swarm optimization. *The Proceedings of the 1995 IEEE International Conference on Neural Networks*, Vol. 4, 1942-1948. (1995)
29. Kennedy, J., Eberhart, R. C.: A Discrete Binary Version of the Particle Swarm Algorithm. *The Proceedings of IEEE International Conference on Computation, Cybernetics and Simulation, System, Man and Cybernetics*, Vol. 5, 4104-4108. (1997)
30. Wang, L., Wang, X., Fu, J., Zhen, L.: A novel probability binary particle swarm optimization algorithm and its application. *Journal of software*, Vol. 3, 28-35. (2008)
31. Zhen, L. L., Wang, L., Wang, X.T., Huang, Z.: A Novel PSO-Inspired Probability-based Binary Optimization Algorithm. *International Symposium on Information Science and Engineering*, 248-251. (2008)
32. Menhas, M. I., Wang, L., Fei, M.R., Pan, H.: Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design. *Expert Systems with Applications*, Vol. 39, 4390-4401. (2011)

Ling Wang is an associate professor in Control Theory and Control Engineering at Shanghai University. He received B.Sc. degree in Automation from East China University of Science and Technology in 2002 and Ph.D. degree in control theory and control engineering from East China University of Science and Technology in 2007. His research interests include intelligent optimization algorithm design and wireless sensor networks.

Ling Wang et al.

Wei Ye is a postgraduate student major in Control Science and Engineering at Shanghai University. He received B.Sc. degree in Measurement and Control Technology from Shanghai University in 2010. His research interests include industrial wireless sensor networks and evolutionary computation.

Haikuan Wang is a post doctor in Control Theory and Control Engineering at Shanghai University. He received Ph.D. degree in control theory and control engineering from Shanghai University in 2011. His research interests include wireless sensor networks and industrial engineering.

Xiping Fu received his B.Sc. and Master degree in Automation from Shanghai University in 2008 and 2011, respectively. His research interests include particle swarm optimization algorithm, differential evolution algorithm and industrial wireless sensor networks.

Minrui Fei received his Ph.D. degree in Control Theory and Control Engineering from Shanghai University in 1997. Since 1998, he has been a professor and doctoral supervisor in Shanghai University. His current research interests are in the areas of intelligent control, networked learning control systems, wireless sensor networks, etc. He is a vice-chairman of Chinese Association for System Simulation, a standing director of China Instrument & Control Society, and a director of Chinese Artificial Intelligence Association.

Muhammad Ilyas Menhas received his B.Sc in electrical engineering from University of Azad Jammu and Kashmir in 2002. He has been an assistant engineer with Azad Jammu and Kashmir Electricity Department. Currently he is doctoral student at Shanghai University. His research interests include intelligent optimization algorithms, automation and control.

Received: January 17, 2012; Accepted: November 15, 2012.

Automatic T-S fuzzy model with application to designing predictive controller¹

Zhi-gang Su, Pei-hong Wang, and Yu-fei Zhang

School of energy & environment, Southeast University, Nanjing, Jiangsu 210096,
China
Zhigangsu@seu.edu.cn

Abstract. A novel methodology is proposed to automatically extract T-S fuzzy model with enhanced performance using VABC-FCM algorithm, a novel Variable string length Artificial Bee Colony algorithm (VABC) based Fuzzy C-Mean clustering technique. Such automatic methodology not requires a priori specification of the rule number and has low approximation error and high prediction accuracy with appreciate rule number. Afterward, a new predictive controller is then proposed by using the automatic T-S fuzzy model as the dynamic predictive model and VABC as the rolling optimizer. Some experiments were conducted on the superheated steam temperature in power plant to validate the performance of the proposed predictive controller. It suggests that the proposed controller has powerful performance and outperforms some other popular controllers.

Keywords: T-S fuzzy model, fuzzy c-means; automatic; Artificial Bee Colony; predictive control, superheated steam temperature

1. Introduction

System modeling and (rolling) optimization are the two main issues to design a predictive controller for an uncertain nonlinear system [27]. In this paper, we aim to propose a novel methodology used to automatically extract model from data and then to design a predictive controller using such novel methodology.

Fuzzy system modeling approach is an effective tool for the approximation of uncertain nonlinear systems on the basis of observed data [13]. Among the different fuzzy modeling techniques, the Takagi-Sugeno (T-S) model [34] has attracted most attention and is popularly applied [12, 14, 28, 30, 32]. The construction of a T-S fuzzy model is generally done in two steps. In the first step, the fuzzy sets in the rule antecedents are determined. This can be done manually, using expert knowledge of the process or by some data-driven techniques. In the second step, the parameters of the consequent functions

¹ This article is supported by the National natural science foundation of China (51106025). This is an extended version of the paper [38] originally presented at the international conference on ICNC-FSKD, 25-28, July, 2011.

are estimated. As these functions are usually chosen to be linear in their parameters, standard linear least-squares methods can be used. The bottleneck of the construction procedure is the identification of the fuzzy sets in the antecedents. In the traditional T-S fuzzy model, the expert knowledge is used to define linguistically the fuzzy sets, which leads to the drawbacks of subjectivity and suffers lack of generalization. To overcome these issues, reduce expert knowledge intervention and rather build self learning systems, more objective T-S fuzzy models are developed, such as the first family of population stochastic optimization based methods [4, 9, 10, 11, 16, 19, 25, 39] and the second family of clustering based methods [1, 6, 20, 21, 33, 37, 40]. Because of the powerful global search ability of the population stochastic optimization algorithms, this first family of method has high prediction accuracy [15]. Attractive features of the second family of method are the simultaneous identification of the antecedent membership functions along with the consequent local linear models and the implicit regularization [15].

Although the aforementioned approaches improve the efficiency of the T-S fuzzy model, they have various challenges that should not be neglected [35]. The two main challenges among these are how to automatically determine the rule number and to obtain a global optimal model. For the above first family of method, they usually can get a global structure of T-S fuzzy model due to their powerful global search ability. However, they cannot automatically determine the rule number. In some literature [39], the rule number is labeled as a cell in the fixed length of individual so as to automatically determine the rule number. It is crisp in nature. This drawback also exists for the second family of method due to the reason that the cluster number is assumed to be fixed *a priori*. In addition, the second family of method usually cannot get global optimization solutions, because the traditional fuzzy clustering techniques often get stuck at suboptimal solutions based on the initial configuration. A methodology, used to extract T-S fuzzy model with enhanced performance and without knowing the rule number as a priori, is desirable. In [31], we viewed the fuzzy clustering as an optimization problem where the length of the optimal solution(s) is not known as a priori, and we thus proposed a novel version of Fuzzy C-Means clustering technique, called VABC-FCM algorithm, based on the so-called Variable string length Artificial Bee Colony (VABC) algorithm, another contribution in [31]. By using such novel VABC-FCM algorithm, in this paper, we further propose a novel methodology to automatically extract T-S fuzzy models from data. Use of VABC-FCM allows such methodology not require a priori specification of the rule number and have low approximation error and high prediction accuracy.

For the later issue in traditional predictive controller, the optimal control sequences are computed using the derivation of a given quadratic objective function that is usually represented in the form of vector or matrix. However, it is hard to separate the variables to be optimized from the given quadratic objective function. Thus, it is difficult to obtain analytic solution by using such derivation methods. In addition, the derivation methods exist defects on dealing with control problem with constraint. Therefore, some population stochastic optimization algorithms are used instead of the traditional

derivation methods [14, 22, 27, 32]. Here, we apply the (V)ABC (i.e., VABC or ABC) algorithm as the optimizer, due to the following two reasons [18]: (1) ABC algorithm is recently a popular swarm intelligent algorithm, which outperforms the evolutionary algorithms and other swarm intelligent algorithm in the most of cases; and (2) The number of control parameters involved in ABC is less than that of other population stochastic optimization algorithms. Note that, the (rolling) optimization in predictive control is a constrained optimization problem where the length of optimal solutions is fixed and known as a priori. So, the VABC algorithm and ABC algorithm are equivalent in this case, which will be discussed later in the paper. By using the automatic T-S fuzzy model as the dynamic predictive model and the VABC algorithm as the rolling optimizer, we design a new predictive controller in this paper.

The rest of this paper is organized as follows. Section 2 introduces VABC algorithm and VABC-FCM algorithm. Section 3 presents the automatic T-S fuzzy model. In Section 4, a new fuzzy model predictive controller is designed. The last section concludes this paper.

2. VABC algorithm and VABC based Fuzzy clustering

2.1. The VABC algorithm

In VABC algorithm, the foraging artificial bees are categorized into three groups: employed bees, onlookers and scouts. The task of each category of bees is also the same as that in the ABC algorithm [17, 18] except for some differences. The main difference is the string or food source representation. In ABC, all the lengths of strings are fixed to be equal, whereas those in VABC are variable. This difference is shown in a d -dimensional space in Fig. 1.

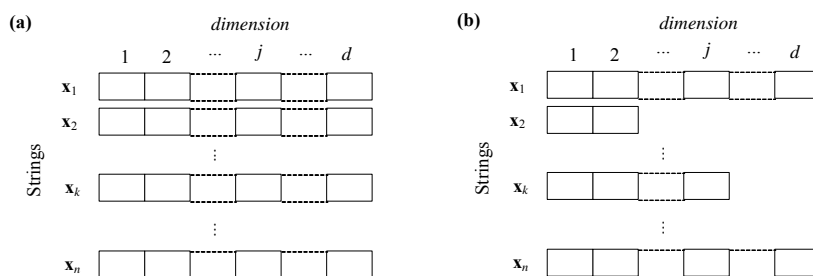


Fig. 1. String representation for (a) ABC and (b) VABC

Firstly, a randomly distributed initial population of n solutions is generated according to mechanism (1).

$$x_{i,j} = x_{\min,j} + \text{rand}[0,1] \times (x_{\max,j} - x_{\min,j}), \quad j \in \{1, 2, \dots, J_i\} \quad (1)$$

where J_i is the length of the i^{th} string and suppose its value is no less than 2. Thus, given an upper bound d_{\max} for J_i ($i = 1, 2, \dots, n$), it can be randomly derived according to:

$$J_i = \text{round}(\text{rand}[0,1] \times (d_{\max} - 2)) + 2 \quad (2)$$

where each solution \mathbf{x}_i ($i = 1, 2, \dots, n$) is a d -dimensional vector.

After initialization, the population is subjected to repeated cycles of four major steps: updating feasible solutions, selecting feasible solutions, avoiding suboptimal solutions and solutions mutation.

All employed bees select a new candidate food source position. The choice is based on the neighborhood of the previously selected food source. The position of such new candidate food source is calculated from equation below:

$$x_{\text{new},j} = x_{i,j} + \text{rand}[-1,1] \times (x_{i,j} - x_{k,q}) \quad (3)$$

where $k \in \{1, 2, \dots, n\}$ and $k \neq i$, and $\text{rand}[-1, 1]$ is a uniformly distributed real number in interval $[-1, 1]$. Subscript j and q denotes respectively the j^{th} and q^{th} dimension of \mathbf{x}_i and \mathbf{x}_k , determined as

$$\begin{cases} q \leftarrow j = \text{round}(\text{rand}[0,1] \times (J_i - 1)) + 1, & \text{if } J_k \geq J_i \\ j \leftarrow q = \text{round}(\text{rand}[0,1] \times (J_k - 1)) + 1, & \text{otherwise} \end{cases} \quad (4)$$

with $\text{round}(\cdot)$ function rounding a number to the nearest integer and the symbol " $q \leftarrow j$ " assigning value of variable j to variable q .

The old food source position in the employed bees' memory will be replaced by the new candidate food source position, if the new position has a better fitness value. Employed bees will return to their hive and share the fitness value of their new food sources with the onlooker bees.

In the next step, each onlooker bee selects one of the proposed food sources depending on the fitness value obtained from the employed bees. The probability that a food source will be selected can be obtained from following equation:

$$p_i = \text{fit}_i / \sum_{i=1}^n \text{fit}_i \quad (5)$$

where fit_i is the fitness value of the i^{th} food source \mathbf{x}_i . Obviously, the higher the fit_i is, the more probability the i^{th} food source is selected.

The probability of a food source being selected by the onlooker bees increases as the fitness value of a food source increases. After the food source is selected, onlooker bees will go to the selected food source and select a new candidate food source position in the neighborhood of the selected food source. The new candidate food source can be also calculated by Eq. (3).

In the third step, any food source position that does not improve the fitness value will be abandoned and replaced by a new position that is randomly determined by a scout bee. This helps avoid suboptimal solutions. The new randomly chosen position will be calculated according to Eq. (1).

The final step is a mutation operation on the strings/food sources that are selected for mutation. The mutation operator is defined of three types:

1. If the length of the selected string is longer than that of the string holding the best fitness, one randomly generated cell (i.e., a real-number in a random dimension) is removed from the string.
2. If the length of the selected string is shorter than that of the string holding the best fitness, one randomly generated cell is added to the string.
3. Otherwise, the selected string is hold on.

The mutation probability is selected adaptively for each string as in [29]. Let fit_{max} be the maximum fitness, \overline{fit} be the average fitness of the population and fit be the fitness value of the solution to be mutated. The expression for mutation probability, p_m , is given below:

$$p_m = \begin{cases} k_1 (fit_{max} - fit) / (it_{max} - \overline{fit}), & \text{if } fit > \overline{fit} \\ k_2, & \text{otherwise} \end{cases} \quad (6)$$

where k_1 and k_2 are constant. Here, both k_1 and k_2 are kept equal to 0.5.

Given a threshold p_0 for the mutation, the string is selected when its mutation probability is bigger than p_0 , i.e., $p_m > p_0$. It is evident that p_0 cannot be higher than the value of k_2 , otherwise the mutation operation is useless: once p_0 is higher than the value of k_2 , the length of each string in the population will keep the same as its initial length when the VABC is terminated. In such viewpoint, the VABC algorithm degenerates to ABC algorithm once the lengths of string x_i are equally initialized and/or p_0 is set to be bigger than k_2 .

The maximum number of cycles (C_{max}) is used to control the number of iterations and is a termination criterion. The process will be repeated until the number of iteration equals the C_{max} .

2.2. VABC-FCM: VABC based fuzzy c-means clustering approach

Let $Z = \{z_1, z_2, \dots, z_N\}$ be a collection of vectors in R^d describing N objects. Let c ($2 \leq c \leq N$) be the desired cluster number. Each cluster is represented by a prototype or a center $v_k \in R^d$. Let V denotes a matrix of size $(c \times d)$ composed of the coordinates of the cluster centers such that V_{kq} is the q^{th} component of the cluster center v_k . FCM looks for a partition matrix $U = (u_{ik})$ of size $(N \times c)$ and for the matrix V by minimizing the following objective function:

$$J_{FCM}(U, V, Z) = \sum_{i=1}^N \sum_{k=1}^c u_{ik}^m D_{ik}^2 = \sum_{i=1}^N \sum_{k=1}^c u_{ik}^m \|z_i - v_k\|^2 \quad (7)$$

subject to the following constraints (8) and (9):

$$\sum_{k=1}^c u_{ik} = 1, \quad \forall i \in \{1, 2, \dots, N\} \tag{8}$$

$$\sum_{i=1}^N u_{ik} > 0, \quad \forall k \in \{1, 2, \dots, c\} \tag{9}$$

For a given cluster number c , the algorithm starts from an initial guess for either the partition matrix or the cluster centers and iterates until convergence. Convergence is guaranteed but it may lead to local minimum [2]. To choose optimal cluster number c , clustering validity index is necessary. A sufficient review for fuzzy clustering validity indexes can refer to Refs. [23, 24]. Among them, the most popular two are XB-index [36] and PBM-index [23], defined as:

$$XB(U, V, Z) = \sum_{k=1}^c \sum_{i=1}^N u_{ik}^2 D_{ik}^2 / N \times \min_{i \neq j} \{ \|v_i - v_j\|^2 \} \tag{10}$$

$$PBM(U, V, Z) = E_1 \times \max_{i,j} \{ \|v_i - v_j\| \} / c \sum_{k=1}^c \sum_{i=1}^N u_{ik} D_{ik} \tag{11}$$

where E_1 is the constant avoiding the PBM to be too smaller. Here, $E_1 = 100$.

As can be seen from validity indexes (10) and (11), the cluster number minimizing XB-index leads to the optimal partition of data set Z , and on the contrary, proper fuzzy partitioning can be obtained for a data set by maximizing the PBM-index. Thus, the fuzzy c-means clustering can be viewed as an optimization problem where the best partition is considered to be the one that corresponds to the minimum value of the XB-index or to the maximum value of the PBM-index. Therefore, the framework of the VABC-FCM is the same as that of the VABC algorithm except for some differences, which are interpreted as followings.

The first difference may be the string representation and population of food sources initialization. In VABC-FCM, center based encoding is used where centers are considered to be indivisible. Real numbers are encoded in the strings which represent the coordinates of the centers of the partitions. The length of a particular string x_i is now J'_i , calculated by

$$J'_i = d \times \lceil \text{round}(\text{rand}[0,1] \times (d_{\max} - 2)) + 2 \rceil \tag{12}$$

Note that d_{\max} here is the upper bound of the cluster number, not the maximal dimension of data space as interpreted in VABC algorithm. The initial cluster number will therefore range from two to d_{\max} . The centers encoded in a string are randomly selected distinct points from the data set. The selected points are distributed randomly in the string.

The membership value u_{ik} in VABC-FCM algorithm is updated by using the same criterion as the FCM algorithm. After obtaining the membership value u_{ik} , the centers encoded in a string are further updated using the same

criterion as the FCM algorithm. Centers are further updated so as to incorporate a limited amount of local search for speeding up the convergence of VABC-FCM.

The second difference between VABC-FCM algorithm and VABC algorithm is the fitness computation. The fitness in the VABC-FCM algorithm can be calculated as:

- XB-index case: $fit = 1 / (1 + XB(U, V, Z))$
- PBM-index case: $fit = PBM(U, V, Z)$

The final difference exists in mutation operation on the selected string(s). In VABC algorithm, once the length of the selected string(s) is not equal to that of the string holding the best fitness, one random dimension is removed or increased. However, in VABC-FCM algorithm, the indivisible cell is not one dimension but a cluster center. Thus, one random cluster center is removed or increased when the selected string(s) do not have the best fitness.

With the above interpretations, the VABC-FCM algorithm can be directly derived from VABC algorithm.

3. Automatic T-S fuzzy model based on VABC-FCM

The typical T-S fuzzy model rule can be described as follow:

$$R_i: \text{ If } \mathbf{x} \text{ is } A^i, \text{ Then } y^i = f^i(\mathbf{x}) = a_0^i + a_1^i x_1 + a_2^i x_2 + \dots + a_d^i x_d \quad (13)$$

where $i = 1, 2, \dots, N_R$, N_R is the number of fuzzy rules; $\mathbf{x} = [x_1, x_2, \dots, x_d]$ is the input variable vector for the i^{th} rule, d is the dimension of input variable; $\theta_i = [a_0^i, a_1^i, a_2^i, \dots, a_d^i]$ are the consequent parameters; y^i is an output of the i^{th} fuzzy rule; A^i is a fuzzy variable on product variable space.

Given the input variables \mathbf{x} , the fuzzy model output is computed by a weighted mean defuzzification as follows:

$$\hat{y} = \frac{\sum_{i=1}^{N_R} w_i y^i}{\sum_{i=1}^{N_R} w_i} \quad (14)$$

where the weight strength w_i of the i^{th} rule is calculated as

$$w_i = \mu_{A^i}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{v}_i^x\|^2}{\sigma_i^2}\right) \quad (15)$$

where μ_{A^i} is the membership function of A^i , \mathbf{v}_i^x and σ_i denote the center (or mean) and width (or standard deviation) of the fuzzy set A^i , respectively.

The VABC-FCM based T-S fuzzy model, called *automatic T-S fuzzy model*, is proposed in the following ways. Firstly, the VABC-FCM algorithm is used to automatically partition the input-output data space. In VABC-FCM, only the centers of clusters are encoded. Secondly, the membership matrix is used to indicate the fuzzy partition. Thus, the *automatic T-S fuzzy model* can be represented as below:

$$R_i : \text{IF } \mathbf{x} \text{ is in cluster } C_i(v_i^x, u_i) \text{ Then} \quad (16)$$

$$y^i = a_0^i + a_1^i x_1 + a_2^i x_2 + \dots + a_d^i x_d$$

where vector \mathbf{u}_i , the i^{th} row of membership degree matrix U , is calculated as:

$$\mathbf{u}_i = \left\{ u_{ij} \mid u_{ij} = D_{ji}^{-2/(m-1)} / \sum_{i=1}^{N_R} D_{ji}^{-2/(m-1)}, j = 1, 2, \dots, N \right\} \quad (17)$$

The following Algorithm 1 depicted the procedure used to automatically extract T-S fuzzy model.

Algorithm 1 VABC-FCM based automatic T-S fuzzy model

Identification stage

Input: Input-output $Z = [X, \mathbf{y}]$, where X is input matrix and \mathbf{y} is output vector.

Output: Structure of automatic T-S fuzzy model: $\{V, U, \theta\}$.

Step 1: Initialization.

Step 2: Apply the VABC-FCM to automatically determine the rule number N_R , the center coordinates $V = \{\mathbf{v}_k \mid k = 1 \text{ to } N_R\}$ of the N_R rules, and the membership matrix $U = \{u_{ik} \mid i = 1 \text{ to } N\}$.

Step 3: Determine the local model in each rule by minimizing:

$$\min_{\theta_k} \frac{1}{N} (\mathbf{y} - X_e \theta_k)^T \Phi_k (\mathbf{y} - X_e \theta_k)$$

where $X_e = [1, X]$ is the matrix extended by a unitary column and the diagonal matrix Φ_k is defined as:

$$\Phi_k = \text{diag}[u_{k1}, u_{k2}, \dots, u_{kN}]$$

Thus, the consequent parameters of the k^{th} local model is given by:

$$\theta_k = (X_e^T \Phi_k X_e)^{-1} X_e^T \Phi_k \mathbf{y}$$

Evaluation stage

Input: An arbitrary input $\mathbf{z} = [\mathbf{x}, y]$ and T-S fuzzy model: $\{V, U, \theta\}$.

Output: Prediction \hat{y} of \mathbf{x} .

Step 1: Calculate the distances between \mathbf{z} and center vectors in V ;

Step 2: Calculate the membership degrees of sample \mathbf{z} belonging to each local model; take the membership degrees as the weights of \mathbf{z} belong to the N_R local models

Step 3: Calculate the response y^k of \mathbf{x} for local model θ_k , and obtain the prediction \hat{y} of \mathbf{x} .

To validate the performance of the automatic T-S fuzzy model, the nonlinear static system [33] is used to validate the prediction accuracy of the proposed model, whereas the Box-Jenkins gas furnace system [3] is used to validate the approximation ability of the proposed model. In the two examples, the parameters of automatic T-S fuzzy model are initialized as follows: food number $n = 100$, $C_{\max} = 500$, $P_0 = 0.3$ and $d_{\max} = 15$. The Mean Square Error (MSE) is used as a performance measure, defined as:

$$MSE = \frac{1}{N} \sum_{k=1}^N (\hat{y}_k - y_k)^2$$

To show the powerful performance, our proposed method is compared with other advanced T-S fuzzy models as mentioned in the Introduction.

Example 1: Nonlinear static system [33]

The nonlinear static system with two inputs, x_1 and x_2 , and a single output, y :

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \leq x_1, x_2 \leq 5$$

As in [33], 50x2 input-output data, consisting of 50 training and testing samples respectively, are used. With the 50 input-output training samples, the automatic T-S fuzzy model can be constructed. The experiments are run for 50 times. The automatic T-S fuzzy model for random one of the 50 running times is presented as follows.

R_1 : IF \mathbf{x} is in C_1 ((1.2721, 1.4796), \mathbf{u}_1)

Then $y^1 = 2.6648 + 0.7508x_1 + 0.7408x_2$

R_2 : IF \mathbf{x} is in C_2 ((2.6415, 2.0677), \mathbf{u}_2) Then $y^2 = 4.1955 - 0.2718x_1 - 0.6389x_2$

R_3 : IF \mathbf{x} is in C_3 ((4.302, 4.3758), \mathbf{u}_3)

Then $y^3 = 0.9269 + 0.0274x_1 + 0.0542x_2$

where \mathbf{u}_1 , \mathbf{u}_2 and \mathbf{u}_3 among the rules R_1 , R_2 and R_3 are indicated in Fig. 2.

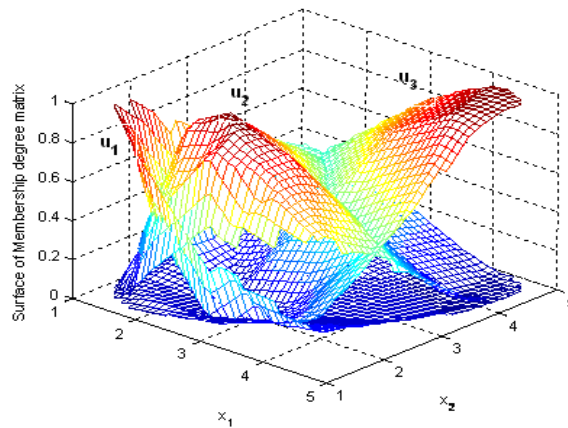


Fig. 2. Membership matrix for the nonlinear static system by using VABC-FCM algorithm

Fig. 3 shows the prediction of the T-S fuzzy model and the prediction errors for the 50 testing samples. In this case, the training MSE is 0.024911, and the prediction/testing MSE is 0.055616.

The average MSE of 50 running times is used as the total performance criterion. The average training MSE is $0.0249 \pm 3.851e-5$, and average prediction MSE is $0.0556 \pm 1.8136e-5$, as shown in Table 1. For comparison,

the other three advanced methods [20, 33, 40] are also run here. The results are presented in Table 1. It suggests from Table 1 that our method has high prediction accuracy with appreciated rules.

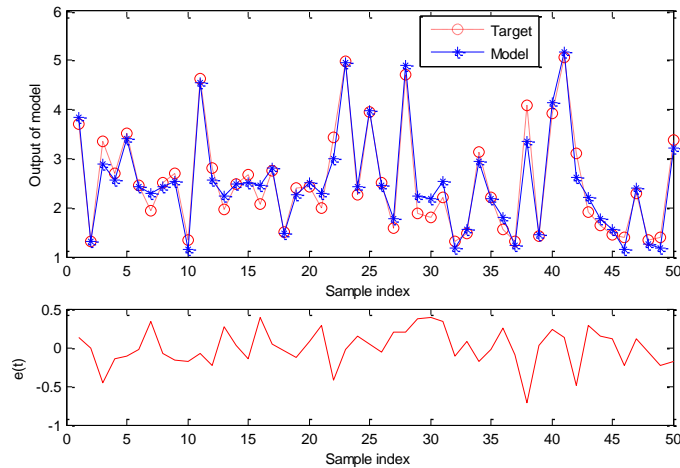


Fig. 3. Predictions of the 50 testing samples using automatic T-S fuzzy model

Table 1. Comparison results for nonlinear static system*1

Method	Rules (mean)	Prediction MSE (mean ± std.)
Literature [33]	6	0.079
Literature [20]	4/5	0.114 / 0.102
VGA-FCM [40]	4	0.069 ± 0.012
Our method	3	0.0556 ± 1.8136e-5

*1 In literatures [20 33], the Fuzzy T-S models are identified using clustering technique, thus the Std. is absence.

Example 2: Box-Jenkins (BJ) gas furnace [3]

This example uses BJ gas furnace data to show the approximation ability of the proposed model. The model holding lower approximation error with fewer rules is the best one. The BJ dataset, recorded from a combustion process of a methane-air mixture, has been used widely to validate the performance of the new proposed modeling methods. There are originally 296 data points, $t = 1$ to 296. $y(t)$ is the output CO₂ concentration and $u(t)$ is the input gas flowing rate. It has been found in the most literatures that three popular sets of input variables for predicting $y(t)$: (1) $y(t-1)$ and $u(t-4)$, (2) $y(t-1)$, $y(t-2)$, $u(t-3)$ and $u(t-4)$, and (3) $y(t-1)$, $y(t-2)$, $y(t-3)$, $u(t-1)$, $u(t-2)$ and $u(t-3)$. All these three kinds of input variables are considered in our study.

As that done in Example 1, the experiment is implemented for 50 times, and the average MSE is taken as the performance criterion. The average training/Approximation MSE for different type of input variables are presented

in Table 2, as well as the Std. values of the MSE. In Table 2, we also present the results obtained by using other advanced methods. It shows that our proposed method outperforms other methods.

Table 2. Comparison results for BJ gas furnace data set

Methods	Input variables	rules (mean)	MSE (mean \pm std)
PSO [4, 19, 39]	$y(t-1), u(t-4)$	3.64	0.1550 ± 0.0422
CRPSO [39]	$y(t-1), u(t-4)$	3.96	0.1428 ± 0.0138
GA [9, 10, 16, 39]	$y(t-1), u(t-4)$	4.32	0.1478 ± 0.0109
DE [11, 39]	$y(t-1), u(t-4)$	4.96	0.1768 ± 0.0602
VGA-FCM [40]	$y(t-1), u(t-4)$	3.74	0.1515 ± 0.0312
Literature [37]	$y(t-1), u(t-4)$	25	0.328
Literature [6]	$y(t-1), u(t-4)$	3	0.2678
Literature [20]	$y(t-1), u(t-4)$	4	0.426
Literature [21]	$y(t-1), y(t-2),$ $u(t-3), u(t-4)$	5	0.248
Literature [33]	$y(t-1), y(t-2),$ $y(t-3), u(t-1),$ $u(t-2), u(t-3)$	2	0.068
Our method	$y(t-1), u(t-4)$	4	$0.1256 \pm 2.7755e-5$
	$y(t-1), y(t-2),$ $u(t-3), u(t-4)$	3	$0.0581 \pm 1.5160e-4$
	$y(t-1), y(t-2),$ $y(t-3), u(t-1),$ $u(t-2), u(t-3)$	3	$0.0548 \pm 1.5164e-4$

4. Application to design fuzzy predictive controller

Section 4.1 shows the procedures used to design predictive controller. Consequently, some experiments are conducted to validate the performance of the designed predictive controller in section 4.2.

4.1. Automatic T-S model and (V)ABC based predictive controller

There are various families of strategies used to design predictive controller since the basic concept of predictive control is proposed [26]. Although there exist differences in assumptions and designing ideas, the key idea of these strategies are similar. More precisely, for a given Single Input (control input or controller output u) Single Output (process output or plant output y) system (SISO) illustrated in Fig. 4, such key idea can be interpreted as follows:

At time t , the current plant state is sampled. The (corrected) plant outputs $\{\hat{y}(t), \hat{y}(t+1), \dots, \hat{y}(t+N_p-1)\}$ are computed for a time horizon $[t, t+N_p-1]$ in the further. These predicted outputs depend on the information that has been

collected, the dynamic predictive model and the assumed control input trajectory $\{v(t), v(t+1), \dots, v(t+N_u-1)\}$ for a appreciated time horizon $[t, t+N_u-1]$ (i.e., N_u is generally smaller than the prediction horizon N_p);

Select optimal control sequence $\{v^*(t), v^*(t+1), \dots, v^*(t+N_u-1)\}$ in the assumed input trajectory v to make the plant outputs \hat{y} approximate the reference trajectory y_r as possible. Such approximation can be realized by minimizing a objective function, generally defined as:

$$\min J = \sum_{k=1}^{N_p} [\hat{y}(t+k) - y_r(t+k)]^2 + \lambda \sum_{k=0}^{N_u-1} (\Delta v(t+k))^2$$

where, $\Delta v(t+k) = v(t+k) - v(t+k-1)$, y_r is reference trajectory and λ is constant weight vector.

Apply the first element of the optimal control sequence v^* as the actual control input signal to the plant, i.e., $u(t) = v^*(t)$;

Repeat the above operations, once the later sampling time comes.

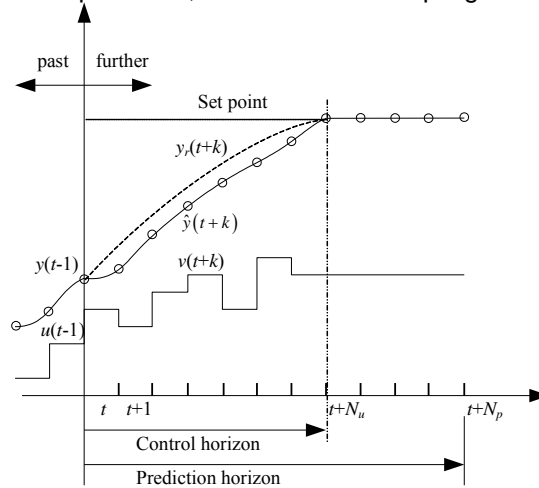


Fig. 4. Key idea of predictive control strategy by using SISO system

In practice, the plant (i.e., object) to be controlled is much more complex and Multiple Inputs Multiple Outputs (MIMO) system is usually faced. In this case, the MIMO can be viewed as the composition of some subsystems with multiple inputs and single output (MISO). By comparing with SISO, note that the MISO only has some more inputs. To avoid confusion with scalar variables such as u, v, y, y_r and \hat{y} in the SISO, the corresponding vector variables are denoted by $\underline{u}, \underline{v}, \underline{y}, \underline{y}_r$ and $\underline{\hat{y}}$ in MIMO system. In what follows, the predictive controller is designed for MIMO system in general.

According to the above key idea, we can see that there are three significant characteristics for the model predictive control algorithm: dynamic predictive model, rolling optimization and feedback correction. In our study, the above key idea is also applied. The predictive controller is designed according to the following four steps.

- (1) Structure design for the predictive controller

For a given MIMO system, the predictive controller is designed as shown in Fig. 5. It can be seen that the structure maintains the three significant characteristics, holding by the traditional predictive controller. The two main differences are the applications of automatic T-S fuzzy model and (V)ABC algorithm respectively as the dynamic prediction model and rolling optimizer.

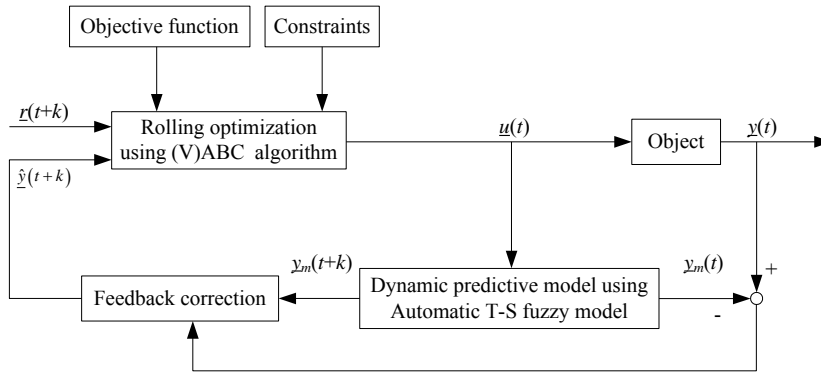


Fig. 5. Structure diagram of the designed predictive controller

(2) Dynamic predictive model by using automatic T-S fuzzy model

For MIMO system with n_i inputs and n_o outputs, the dynamic predictive model holds the following form in general:

$$\underline{y}_m(t) = f(\underline{y}(t-1), \underline{y}(t-2), \dots, \underline{y}(t-n_y), \underline{u}(t-1), \underline{u}(t-2), \dots, \underline{u}(t-n_u)) \quad (18)$$

where $f(\cdot)$ is the (nonlinear) function describing the plant system, n_u and n_y are the orders of inputs and outputs respectively, $\underline{u}(t-k)$ and $\underline{y}(t-k)$ are respectively the control input vector and plant/object output vector, defined respectively as

$$\underline{u}(t-k) = [u_1(t-k), \dots, u_j(t-k), \dots, u_{n_u}(t-k)]^T$$

$$\underline{y}(t-k) = [y_1(t-k), \dots, y_i(t-k), \dots, y_{n_o}(t-k)]^T$$

Such general form (18) can be decomposed into number of n_o subsystems with n_i inputs and single output. Thus, we have

$$\underline{y}_m(t) = \begin{bmatrix} f_1(y_1(t-1), \dots, y_1(t-n_y), \underline{u}(t-1), \dots, \underline{u}(t-n_u)) \\ f_2(y_1(t-1), \dots, y_2(t-n_y), \underline{u}(t-1), \dots, \underline{u}(t-n_u)) \\ \vdots \\ f_{n_o}(y_{n_o}(t-1), \dots, y_{n_o}(t-n_y), \underline{u}(t-1), \dots, \underline{u}(t-n_u)) \end{bmatrix} \quad (19)$$

where $f_i(\cdot)$ is the dynamic predictive model for the i^{th} MISO subsystem.

The dynamic predictive model $f_i(\cdot)$ ($1 \leq i \leq n_o$) can be in various forms such as Neural network, fuzzy model and statistic regression model. In our study,

we apply the automatic T-S fuzzy model as the dynamic predictive model $f(\cdot)$ due to its powerful approximation ability and prediction ability, as already discussed in Section 3. Once the input variables of the automatic T-S fuzzy model are determined, the automatic T-S model can be identified to predict the outputs of the object in the further by using the information of these input variables collected at the past and current states. The way how to establish and identify the automatic T-S fuzzy model can refer to Section 3.

(3) Feedback correction strategy

As we known, the performance of automatic T-S fuzzy model may be degenerated along with the running times, although the automatic T-S fuzzy model holds powerful approximation and prediction abilities. This is because the automatic T-S fuzzy model may be established under limited information, i.e., the information is collected at the past and current states. Along with the running times, some new useful information maybe occurs. As well, the characteristics of the object maybe change along with the running times. For instance, characteristics of the object will degenerate when the service life of object is closed. In addition, it is unfeasible to frequently identify the dynamic predictive model in practice. Therefore, feedback correction is necessary.

In this study, we feed back the prediction error and apply it to correct the output of the dynamic predictive model. More precisely, we have

$$\begin{aligned} \hat{y}(t+k) &= \underline{e}(t) + \underline{y}_m(t+k) \\ &= \underline{e}(t) + f\left(\begin{array}{c} \hat{y}(t+k-1), \dots, \hat{y}(t+1), \underline{y}(t), \dots, \underline{y}(t+k-n_y), \\ \underline{v}(t+k-1), \dots, \underline{v}(t), \underline{u}(t-1), \dots, \underline{u}(t+k-n_u) \end{array}\right), k=1, \dots \end{aligned} \quad (20)$$

where $\underline{u}(t+k-n_u), \dots, \underline{u}(t-1)$ are the actual input trajectory at the past cases, $\underline{v}(t), \dots, \underline{v}(t+k-1)$ are the assumed control input trajectory at the current and further cases, and $\underline{e}(t)$ is the feedback prediction error defined as:

$$\underline{e}(t) = \underline{y}(t) - \hat{y}(t) = \underline{y}_m(t) - f(\underline{y}(t-1), \dots, \underline{y}(t-n_y), \underline{u}(t-1), \dots, \underline{u}(t-r)) \quad (21)$$

(4) Rolling optimization by using (V)ABC algorithm

The rolling optimization aims to find the optimal control sequence $\{\hat{v}^*(t+k) \mid k = 0, 1, \dots, N_u-1\}$ under the following two constraints:

$$\text{Input constraint:} \quad \underline{u}_{\min} \leq \underline{v}(t+k) \leq \underline{u}_{\max} \quad (22)$$

$$\text{Input move constraint:} \quad -\Delta \underline{u}_{\max} \leq \Delta \underline{v}(t+k) \leq \Delta \underline{u}_{\max} \quad (23)$$

where \underline{u}_{\min} and \underline{u}_{\max} are the lower and upper constraints of the control input \underline{u} (Note that \underline{v} is the assumed control inputs. Therefore, we use \underline{u}_{\min} and \underline{u}_{\max} instead of \underline{v}_{\min} and \underline{v}_{\max}), $\Delta \underline{v}(t+k) = \underline{v}(t+k) - \underline{v}(t+k-1)$ and $\Delta \underline{u}_{\max}$ is the maximal variation in the control input.

To apply the (V)ABC as the optimizer, we should encode the assumed control inputs \underline{v} as population of food sources in advance. The l^{th} food source can be encoded in the following form:

$$\mathbf{x}_l = [\underline{v}'(t)^T, \underline{v}'(t+1)^T, \dots, \underline{v}'(t+k-1)^T, \underline{v}'(t+k)^T, \dots, \underline{v}'(t+N_u-1)^T] \quad (24)$$

where $\underline{v}'(t+k) = [v'_1(t+k), v'_2(t+k), \dots, v'_{n_i}(t+k)]^T, k = 0, 1, \dots, N_u - 1$

To satisfy the two constraints in Eqs. (22) and (23) simultaneously, it is unfeasible to produce the food sources by applying operation in Eq. (1). Here, the initial population of control inputs at time t is determined according to

$$v'_j(t) = \min\{u_{j,\max}, \max\{u_{j,\min}, u_j(t-1) + \text{rand}[-1,1] \times \Delta u_{j,\max}\}\}, j = 1, 2, \dots \quad (25)$$

and the control inputs at other times are defined according to

$$v'_j(t+k) = \min\{u_{j,\max}, \max\{u_{j,\min}, v'_j(t+k-1) + \text{rand}[-1,1] \times \Delta u_{j,\max}\}\} \quad (26)$$

where, $k = 1$ to N_u , $u_{j,\min}$, $u_{j,\max}$ and $\Delta u_{j,\max}$ are the j^{th} element of the \underline{u}_{\min} , \underline{u}_{\max} and $\Delta \underline{u}_{\max}$, respectively.

Note that, the new produced candidate food source \mathbf{x}_{new} by applying operation (3) might not satisfy the input move constraint. This situation can be avoided by using the following correction mechanism for each input variable j .

If $\Delta = v_j^{new}(t+k) - v_j^{new}(t+k-1) > \Delta u_{j,\max}$, we have

$$v_j^{new}(t+k) = v_j^{new}(t+k) - (\Delta - \Delta u_{j,\max});$$

If $\Delta = v_j^{new}(t+k) - v_j^{new}(t+k-1) < -\Delta u_{j,\max}$, we have

$$v_j^{new}(t+k) = v_j^{new}(t+k) - (\Delta + \Delta u_{j,\max}).$$

Once the assumed control inputs are encoded as population of food sources, the rest task is to design an objective function for the rolling optimization. For convenience, we apply the quadratic function, defined as:

$$J_{MIMO} = \sum_{i=1}^{n_o} \gamma_i J_i \quad (27)$$

where γ_i are constant weights, and J_i is the quadratic objective function of the i^{th} MISO system and is defined as

$$J_i = \sum_{k=N_0}^{N_p} \|y_{r,i}(t+k) - \hat{y}_i(t+k)\|^2 + \lambda_i \sum_{k=0}^{N_u-1} \|\Delta \underline{v}_i(t+k)\|^2 \quad (28)$$

where N_0 is time delay, reference trajectory $y_{r,i}(t+k) = \alpha_{i,k} y(t) + (1 - \alpha_{i,k}) y_{sp,i}$, $y_{sp,i}$ is set point and $\alpha_{i,k}$ is constant weight, $\alpha_{i,k} = 0.1$ is used in general.

The following Algorithm 2 shows the steps to realize the proposed predictive controller.

Algorithm 2: Automatic T-S model & (V)ABC based predictive controller

1. Initialization
Predetermine the parameters such as prediction horizon N_p , control horizon N_u , weight coefficients γ_i and λ_i , the limits of control input \underline{u}_{\min} , \underline{u}_{\max} and $\Delta\underline{u}_{\max}$, and the control parameters n , d_{\max} , P_0 and C_{\max} in VABC.
 2. At time t , read the information such as the plant outputs $\underline{y}(t-1), \dots, \underline{y}(t-n_y)$ and actual control inputs $\underline{u}(t-1), \dots, \underline{u}(t-n_u)$.
 3. Generate the population of food sources for the current time t .
 4. By taking Eq. (27) as the objective function, the optimal assumed control input trajectory $\{\underline{v}(t+k) \mid k = 0, 1, \dots, N_u-1\}$ are obtained by using the (V)ABC algorithm.
 5. Apply the first element of the optimal control sequence \underline{v}^* as the actual control input to the plant, i.e., $\underline{u}(t) = \underline{v}^*(t)$.
 6. Go to the second step and repeat the operations interpreted in steps 2 ~ 5, when the new state comes, i.e., $t = t + 1$.
-

To implement the Algorithm 2, some parameters should be predetermined such as N_p , N_u , γ_i , λ_i , \underline{u}_{\min} , \underline{u}_{\max} , $\Delta\underline{u}_{\max}$, n , d_{\max} , P_0 and C_{\max} . Among them, the latter four parameters can be initialized as that done in [31]. The weight coefficients λ_i are used to reduce the influence of large changing in control input. If the control system is stable and the changing in control input is not large, the coefficients λ_i can take a relatively smaller value. On the contrary, large value can be taken by λ_i . In our method, the coefficients λ_i can take a small value because the control input has been restricted in a certain interval in advance, as shown in Eqs. (22) and (23). Here, coefficients $\lambda_i = 0.5$ for $i = 1$ to n_o . The weight coefficients γ_i are used to indicate the importance of each sub objective function J_i . Here, $\gamma_i = 1$ for $i = 1$ to n_o , because we consider all the objective functions J_i play the same contribution.

The prediction horizon N_p and control horizon N_u play important roles for the predictive controller. The prediction horizon controls the stability and speediness of the dynamic response. The smaller N_p is, the speediness is faster but the stability is worse; on the contrary, the stability is better but the speediness is slower. In addition, N_p taking large value will lead to the increasing computational complexity of the controller. The control horizon N_u is critical to the sensitivity of the controller: the higher N_u is, the controller is more sensitive. However, large N_u will lead to the worse stability. In the most of cases, the control horizon plays opposite role to the controller as that of the prediction horizon. Therefore, we can firstly determine the control horizon according to the dynamic characteristics of the object and then determine the prediction horizon according to the simulation results. Usually, $N_u = 1 \sim 2$ is used for the object with simple dynamic characteristics, and $N_u = 4 \sim 8$ is used for the complex object.

4.2. Superheated steam temperature control system

Continuous process in power plant and power station are complex systems characterized by nonlinearity, uncertainty and load disturbance. The superheater is an important part of the steam generation process in the boiler-turbine system, where steam is superheated before entering the turbine that drives the generator.

Superheater is made of material that have characteristic of high temperature resistant during normal operation, and the average temperature of superheater is close to highest permitted temperature of material. It will result in malfunction and damage when superheater is overheated. On the other hand, when superheated steam temperature is too low, it will reduce the thermal efficiency of power plant and influence the safe operation of turbine. Therefore, superheated steam temperature is one of the most important factor determining the safety and economic of thermal power plant.

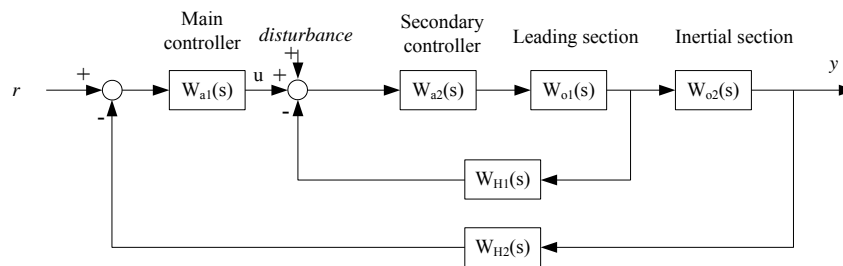


Fig. 6. Structure of superheated steam temperature control system

Sources of disturbances, affecting the superheated steam temperature, are various, such as, the steam flow, combustion conditions of boiler, enthalpy of steam entering the superheater, the change of temperature and flow speed of fume that flows through the superheater, etc. It is difficult to reach small dynamic deviation. Moreover, the temperature adjustment of export steam is required to prevent the superheater been overheated. At present the Spray Water control strategy is adopted as steam temperature adjustment approach for most thermal power plants. In such case, the input, denoted as $u(t)$, is the amount of Spray Water (kg/s), and the output, denoted as $y(t)$, is the superheated steam temperature ($^{\circ}\text{C}$). Thus, the structure of superheated steam temperature control system can be interpreted in Fig. 6.

Some parameters of the superheated steam system in Fig. 6 are presented as follows [5]:

$$W_{o1}(s) = \frac{8}{(1+15s)^2} \quad ^{\circ}\text{C} / \text{mA}, \quad W_{o2}(s) = \frac{1.125}{(1+25s)^3} \quad ^{\circ}\text{C} / \text{mA}$$

$$W_{H1}(s) = 0.1 \quad \text{mA} / ^{\circ}\text{C}, \quad W_{H2}(s) = 0.1 \quad \text{mA} / ^{\circ}\text{C}, \quad W_{a2}(s) = 25$$

According to these parameters above, the generalized transfer function of the superheated temperature can be calculated as:

$$G(s) = \frac{W_{a1}(s)W_{o1}(s)W_{o2}(s)W_{H2}(s)}{1 + W_{a1}(s)W_{o1}(s)W_{H2}(s)}$$

$$= \frac{22.5}{3516000s^5 + 890625s^4 + 401250s^3 + 41850s^2 + 1605s + 21}$$

To simulate for the superheated steam temperature system (see in Fig. 6), suppose the input $u(t) \in [-10, 10]$, i.e., $u_{\min} = -10$, $u_{\max} = 10$, is generated from a pseudo random signal generator. The sampling period is set as 5 second (s) and we continuously record a set of samples for 2000 seconds. In other word, four hundreds of points of signals $u(t)$ are collected. Corresponding to the input $u(t)$, the output $y(t)$ can be obtained by using some ways. An intuitionistic way is to deduce $y(t)$ in the SIMULINK environment in MATLAB® software according to the calculus $G(s)$. Fig. 7 shows the input $u(t)$ and output $y(t)$ generated by using this way.

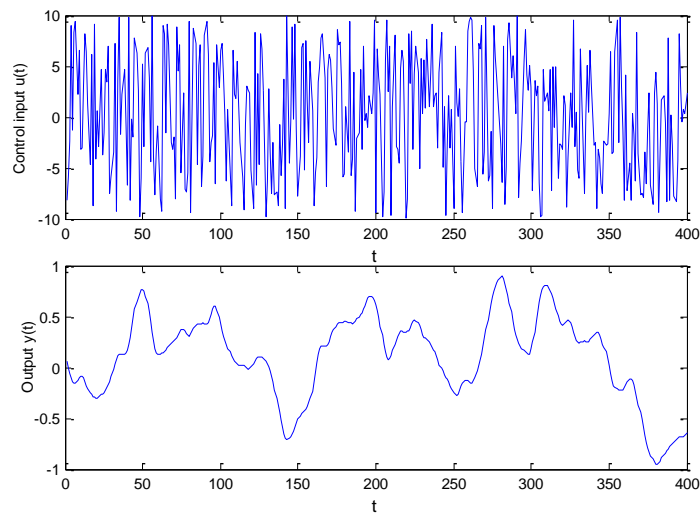


Fig. 7 The output and input of training samples

By selecting $\mathbf{x} = [y(t-1), y(t-2), y(t-3), u(t-1), u(t-2), u(t-3)]$ as the input variables (i.e., $n_y = n_u = 3$) and $y(t)$ as the output, thus the amount of 400 training samples can be constructed. By using these samples, the automatic T-S fuzzy model can be constructed for the superheated steam temperature, as follows.

R_1 : If \mathbf{x} is in C_1 ((-0.9556, -0.9556, -0.9556, -2.7559, -3.0635, -3.6739), \mathbf{u}_1)
Then

$$y_m(t) = -0.0015 + 2.718y(t-1) - 2.5562y(t-2) + 0.8403y(t-3)$$

$$+ 0.0003u(t-1) + 0.0001u(t-2) - 0.0001u(t-3)$$

R_2 : If \mathbf{x} is in C_2 ((0.2858, 0.2865, 0.2864, 0.6995, 0.7949, 0.8041), \mathbf{u}_2) Then

$$y_m(t) = -0.0015 + 2.718y(t-1) - 2.5562y(t-2) + 0.8403y(t-3) \\ + 0.0003u(t-1) + 0.0001u(t-2) - 0.0001u(t-3)$$

where the membership degrees u_1 and u_2 are indicated in Fig. 8.

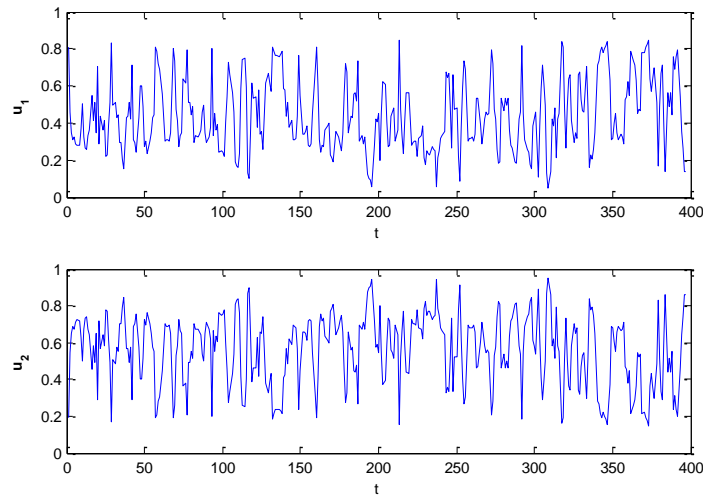


Fig. 8 Membership degrees distribution of 2 clusters for the superheated steam temperature

With the above established dynamic predictive model, we still should predetermine the prediction horizon and control horizon. According to the discussions about these two parameters in Section 4.1, here we assume $N_u = 1$ and $N_p = 10$ by considering the fact that the superheated steam temperature system is not too complex and does not have system time delay, i.e., $N_0 = 0$.

To validate the performance of our method, the generalized predictive controller (GPC) [7, 8] and the traditional PI controller are also studied. The prediction horizon N_p and control horizon N_u taken by GPC are the same as that in our method. The main controller for the traditional PI is designed as

$$W_{a1}(s) = \frac{1}{0.5} \left(1 + \frac{1}{74s} \right).$$

Four typical study cases are conducted to show whether our method is robust or not when the disturbance, prediction horizon, inertial coefficient and gain increase respectively.

Case 1: Study on the influence of step disturbance

In this case, we analyze the laws how the step changes of reference value (r) and disturbance (d) influence on the performance of our method. Fig. 9 shows the results when $r = +1 \text{ mA}$, and Fig. 10 shows the results when $d = +4 \text{ mA}$. It can be seen from Figs. 9 and 10 that our controller can rapidly approach to the new set point. As well, we can see that the performance of our method outperforms the traditional PI and GPC.

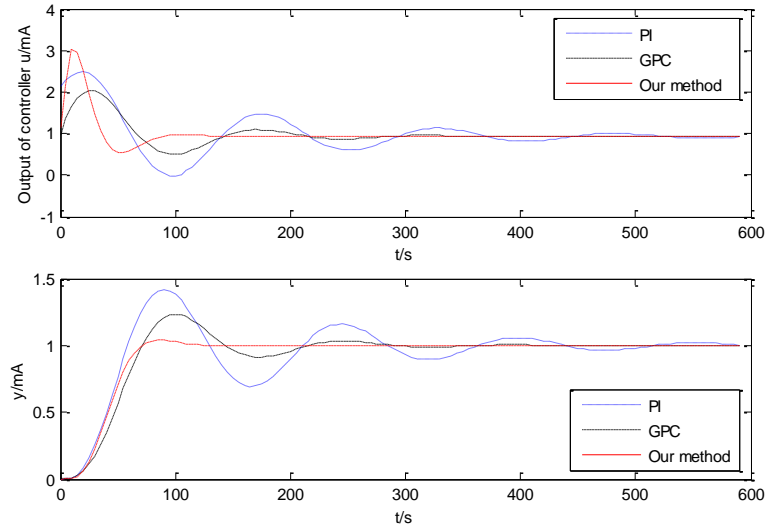


Fig. 9 The response curve in the case $r = 1 \text{ mA}$

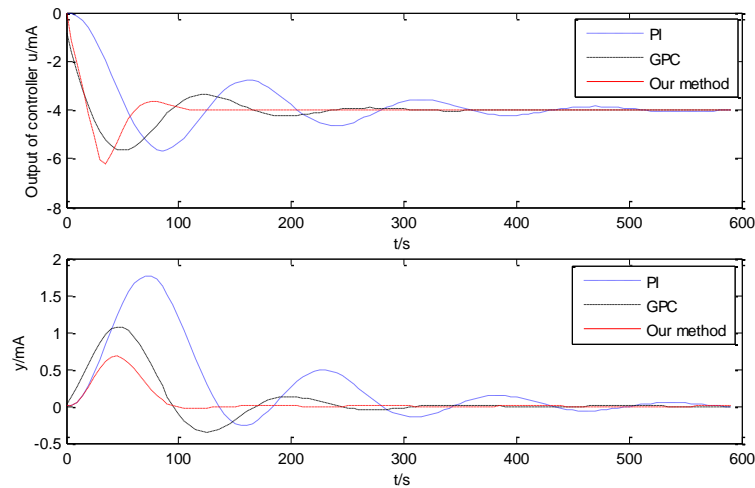


Fig. 10 The simulations under the condition disturbance = +4 mA

Case 2: Study on the influence of prediction horizon

In this case, we analyze the laws how the prediction horizon N_p influence on the performance of our method. Here, we only consider the following three cases: $N_p = 6, 10,$ and 16 under the condition that disturbance $d = +4 \text{ mA}$. The results are shown in Fig. 11. From Fig. 11, we can see that our method has powerful ability on tracking and anti-jamming. The higher the N_p is, the controller is more robust in stability.

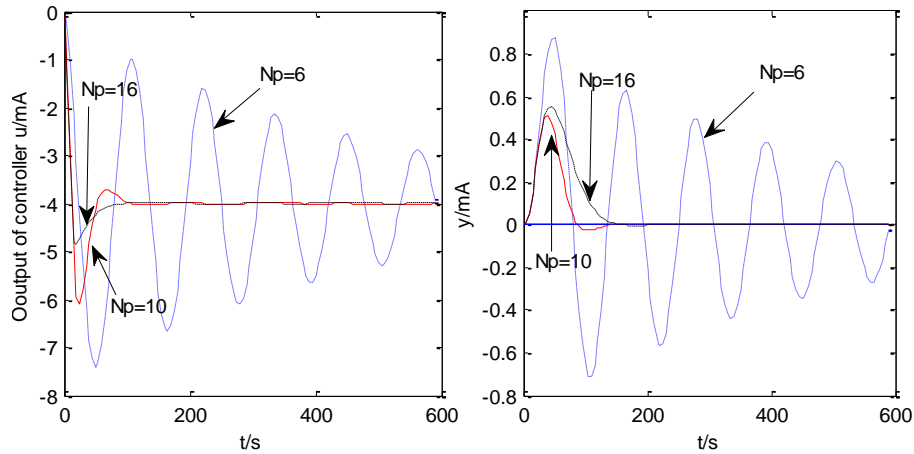


Fig. 11 Influence of N_p on results under the condition disturbance = 4 mA and

Case 3: Study on the influence of inertial coefficient

The dynamic characteristics of superheated steam temperature always vary with the changing working conditions. By considering this fact, we want to see whether our method is valid or not when the parameters (i.e., the inertial coefficient in W_{o2}) of the superheated steam temperature system change in a large range. In other words, we analyze whether our method is sensitive to the inertia of the object or not. To study on the influence of inertial coefficient, we suppose $W_{o1}(s)$ is unchanged and the inertial coefficient in $W_{o2}(s)$ increases from 25 to 35, i.e., the $W_{o2}(s)$ is redefined as follows (Note that the established automatic T-S fuzzy model is not repeatedly identified according to training samples derived from the following redefined transfer function):

$$W'_{o2}(s) = \frac{1.125}{(1 + 35s)^3} \text{ } ^\circ\text{C} / \text{mA}$$

The simulation results are shown in Fig. 12. We can see from Fig. 12 that our method has small oscillation, and the performance of our method outperforms the traditional PI and GPC.

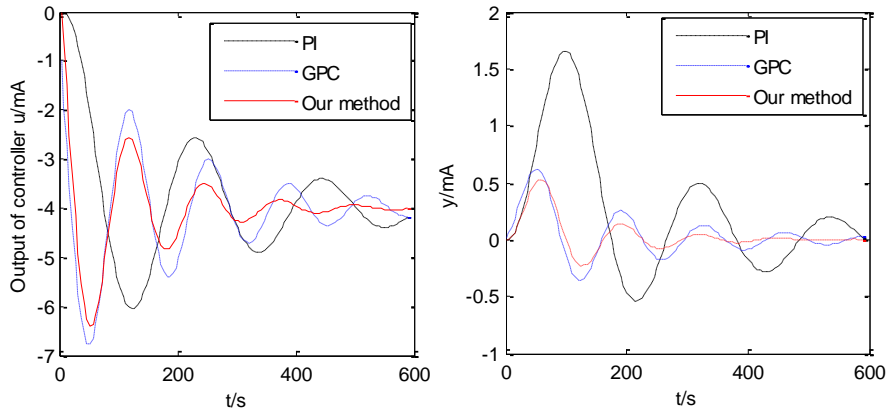


Fig. 12 Results in the case when the inertia of the object increases

Case 4: Study on the influence of gain

In this case, we want to see whether the proposed method is sensitive to the change of gain or not. We suppose $W_{o1}(s)$ is unchanged and gain in $W_{o2}(s)$ increases from 1.125 to 1.875, i.e., the $W_{o2}(s)$ is redefined as:

$$W'_{o2}(s) = \frac{1.875}{(1 + 25s)^3} \text{ } ^\circ\text{C} / \text{mA}$$

The simulation results are shown in Fig. 13. It can be seen from Fig. 13 that our method is robust in the changes of gain and its performance outperforms the traditional PI and GPC.

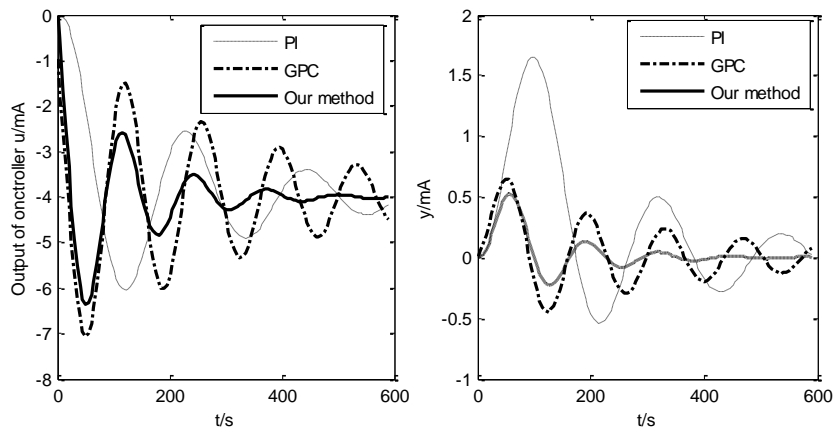


Fig. 13 Results in the case when proportional gain increases from 1.125 to 1.875

In the study cases 3 and 4, the inertial coefficient and gain can also decrease and we have the similar results as above two study cases.

Form the above four study cases, we can see that our method has powerful performance and outperforms the traditional PI strategy and GPC strategy.

5. Conclusions

In this study, a novel methodology for automatically extracting T-S fuzzy model with enhanced performance is proposed by using a novel swarm intelligent fuzzy clustering technique, i.e., the Variable string length Artificial Bee Colony (ABC) algorithm based Fuzzy C-Mean clustering (VABC-FCM). Use of the VABC-FCM algorithm makes the proposed methodology can automatically evolve the rule number from data without knowing the rule number as a priori. Moreover, the output fuzzy partition matrix of VABC-FCM is sufficiently applied in the identification process of T-S fuzzy model, which brings the convenience for model construction and reduces the approximation error. Some numerical examples are used to validate the proposed T-S fuzzy model. The results suggest that it has high prediction accuracy with appreciated rule number.

Consequently, a new fuzzy model predictive controller is designed by using the proposed T-S fuzzy model as its dynamic predictive model and by using (V)ABC algorithm as its rolling optimizer. Taking the superheated steam temperature in power plant as the example, some experiments were conducted to validate the performance of the proposed fuzzy model predictive controller. The experimental results show that the proposed fuzzy model predictive controller has powerful performance. In addition, we compare our method with the popular generalized predictive controller and PI. It shows that our method outperforms the generalized predictive controller and traditional PI controller.

References

1. Babuska, R.. Fuzzy modeling for control. Kluwer Academic Publisher, Boston. (1998)
2. Bezdek, J.C.. Pattern recognition with fuzzy objective function algorithms. Plenum, New York. (1981)
3. Box, G.E.P., Jenkins, G.M.. Time series analysis, forecasting and control. Sanfrancisco, CA: Holden Day. (1970)
4. Chen, C.C.. A PSO-based method for extracting fuzzy rules directly from numerical data. *Cybern. Syst.* Vol. 37, No. 7, 707-723. (2006)
5. Chen, L.J.. Principle of automatic control in thermal process (in chinese). Water Resources and Electric Power Press, Beijing. (1982)
6. Chen, J.Q., Xi, Y.G., et al.. A clustering algorithm for fuzzy model identification. *Fuzzy sets and systems*, Vol. 38, 319-329. (1998)
7. Clarke, D.W., Mohtadi, C., Tuffts, P.S.. Generalized predictive control: Part 1, the basic algorithm. *Automatica*, Vol. 23, 137-148. (1987)
8. Clarke, D.W., Mohtadi, C., Tuffts, P.S.. Generalized predictive control: Part 2, extensions and interpretations. *Automatica*, Vol. 23, 149-160. (1987)
9. Cordon, O., Herrera, F.. A two-stage evolutionary process for designing TSK fuzzy rule based systems. *IEEE Transaction on Systems, Man and Cybernetic B*, Vol. 29, No. 6, 703-715. (1999)

10. Cordon, O., Herrera, F., Gomide, F., Hoffmann, F., Magdalena, L.. Ten years of genetic fuzzy systems: current framework and new trends. In: proceedings of IFSA and NAFIPS, Vol. 3, 1241-1246. (2001)
11. Eftekhari, M., Katebi, S.D., Karimi, M., Jahanmiri, A.H.. Eliciting transparent fuzzy model using differential evolution. Applied soft computing, Vol. 8, 466-476. (2008)
12. Fisher, M., Nelles, O., Isermann, R.. Adaptive predictive control of a heat exchanger based on a fuzzy model. Control Eng. Practice, Vol. 6, 259-269. (1998)
13. Hellendoorn, H., Driankov, D.. Fuzzy model identification: selected approaches. Springer, Berlin, Germany. (1997).
14. Jiang, H., Kwong, C.K., Chen, Z., Ysim, Y.C.. Chaos particle swarm optimization and T-S fuzzy modelling approaches to constrained predictive control. Expert systems with applications, Vol. 39, 194-201. (2012)
15. Johansen, T.A., Babuska, R.. On multi-objective identification of Takagi-Sugeno fuzzy model parameters. In preprint 15th IFAC congress, Barcelona, Spain. (2002).
16. Kang, S.J., Woo, C.H., Hwang, H.S., Woo, K.B.. Evolutionary design of fuzzy rule base for nonlinear system modeling and control. IEEE Transaction on Fuzzy Systems, Vol. 8, No. 1, 37-45. (2000)
17. Karaboga, D.. An idea based on honey bee swarm for numerical optimization. Technical report-TR06, Erciyes University, Engineering faculty, Computer Engineering Department. (2005)
18. Karaboga, D., Basturk, B.. On the performance of artificial bee colony (ABC) algorithm. Applied soft computing, Vol. 8, No. 1, 687-697. (2008)
19. Khosla, A., Kumar, S., Aggarwal, K.K.. A framework for identification of fuzzy models through particle swarm optimization. In: IEEE Indicon Conference, Chennai, India, 388-391. (2005)
20. Li, N., Li S.Y., Xi, Y.G.. Multi-model modeling method based on satisfactory clustering. Control theory & application, Vol. 20, No. 5, 783-787. (2003)
21. Lv, J.H., Chen, J.Q., Liu, Z.Y., et al., A study on fuzzy rules based and nonlinear models for thermal processes. Proceedings of the Chinese society for electrical engineering, Vol. 22, No. 11, 132-137. (2002)
22. Martinez, M., Senent, J.S., Blasco, X.. Generalized predictive control using genetic algorithms. Engineering applications of artificial intelligence, Vol. 11, 355-367. (1998)
23. Maulik, U., Bandyopadhyay, S.. Performance evaluation of some clustering algorithms and validity indices. IEEE Transactions on pattern Anal. Machine intelligent, Vol. 24, No. 12, 1650-1654. (2002)
24. Pakhira, M.K., Bandyopadhyay, S., Maulik, U.. Validity index for crisp and fuzzy clusters. Pattern recognition, Vol. 37, No. 3, 487-501. (2004)
25. Pedrycz, W., Reformat, M.. Evolutionary fuzzy modeling. IEEE Transaction on Fuzzy Systems, Vol. 11, No. 5, 652-665. (2003)
26. Richalet, J., Rault, A., Testud, J.L., Papon, J.. Model predictive heuristic control Application to industrial processes. Automatica, Vol. 14, 413-428. (1978)
27. Sarimveis, H., Bafas, G.. Fuzzy model predictive control of non-linear processes using genetic algorithms. Fuzzy sets and systems, Vol. 139, 59-80. (2003)
28. Sousa, J.M., Babuska, R., Verbuggen, H.B.. Fuzzy predictive control applied to an air-conditioning system. Control Eng. Practice, Vol. 5, 1395-1406. (1997)
29. Srinivas, M., Patnaik, L.M.. Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics, Vol. 24, No. 4, 656-667. (1994)

30. Su, B., Chen, Z., Yuan, Z.. Constrained predictive control based on T-S fuzzy model for nonlinear system. *Journal of systems engineering and electronics*, Vol. 18, 95-100. (2007)
31. Su, Z.G., Wang, P.H., Shen, J., Li, Y.G., Zhang, Y.F., Hu, E.J.. Automatic fuzzy partitioning approach using Variable string length Artificial Bee Colony (VABC) algorithm. *Applied soft computing*, Vol. 12, No. 11, 3421-3441. (2012)
32. Sugeno, M., Kang, G.T. Fuzzy modeling and control of multilayer incinerator. *Fuzzy sets and systems*, Vol. 1, No. 8, 329-346. (1986)
33. Sugeno, M., Yasukawa, T.. A fuzzy logic based approach to qualitative modeling. *IEEE Transaction on Fuzzy Systems*, Vol. 1, No. 1, 7–31. (1993)
34. Takagi, T., Sugeno, M.. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, No. 1, 116–132. (1985)
35. Turksen, I.B., Celikyilmaz, A.. Comparison of fuzzy functions with fuzzy rule base approaches. *Int. J. Fuzzy Syst*, Vol. 8, No. 3, 137-149. (2006)
36. Xie, X.L., Beni, G.. A validity measure for fuzzy clustering. *IEEE Transaction on pattern Anal. Machine intelligent*, Vol. 13, No. 8, 841-847. (1991)
37. Xu C.W., Yong, Z.. Fuzzy model identification and self-learning for dynamic systems. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 17, No. 4, 683–689. (1987)
38. Zhang, Y.F., Su, Z.G., Wang, P.H.. A convenient version of T-S fuzzy model with enhanced performance. In: *proceedings of the 8th international conference on Fuzzy Systems and Knowledge Discovery*, 1074-1079. (2011)
39. Zhao, L., Qian, F., Yang, Y.P., et al.. Automatically extracting T-S fuzzy models using cooperative random learning particle swarm optimization. *Applied soft computing*, Vol. 10, 938-944. (2010)
40. Zhu, H.X.. A study on fuzzy modeling and control based on immune optimization algorithms for thermal processes. MS thesis, Southeast University at Nanjing, (2005)

Zhi-gang Su received his M.S. and Ph.D. degrees from Southeast University, Nanjing, China in 2006 and 2010, respectively. He worked with Southeast University from 2010. His research interests are focused on intelligent modeling, machine learning, soft computing and optimization algorithm, as well as their applications.

Pei-hong Wang is a full professor with School of Energy & Environment, Southeast University. His research interests are focused on intelligent modeling, diagnosis and optimization algorithm, as well as the applications of these methods in power plant.

Yu-fei Zhang is an associate professor with School of Energy & Environment, Southeast University. Her research interests are focused on intelligent modeling and control, as well as their applications.

Received: February 22, 2012; Accepted: November 26, 2012.

Superior Performance of Using Hyperbolic Sine Activation Functions in ZNN Illustrated via Time-Varying Matrix Square Roots Finding

Yunong Zhang, Long Jin, and Zhende Ke

School of Information Science and Technology, Sun Yat-sen University
Guangzhou 510006, Guangdong, China
zhynong@mail.sysu.edu.cn, spaformind@qq.com, kezhende@126.com

Abstract. A special class of recurrent neural network, termed Zhang neural network (ZNN) depicted in the implicit dynamics, has recently been proposed for online solution of time-varying matrix square roots. Such a ZNN model can be constructed by using monotonically-increasing odd activation functions to obtain the theoretical time-varying matrix square roots in an error-free manner. Different choices of activation function arrays may lead to different performance of the ZNN model. Generally speaking, ZNN model using hyperbolic sine activation functions may achieve better performance, as compared with those using other activation functions. In this paper, to pursue the superior convergence and robustness properties, hyperbolic sine activation functions are applied to the ZNN model for online solution of time-varying matrix square roots. Theoretical analysis and computer-simulation results further demonstrate the superior performance of the ZNN model using hyperbolic sine activation functions in the context of large model-implementation errors, in comparison with that using linear activation functions.

Keywords: Zhang neural network, global exponential convergence, hyperbolic sine activation functions, time-varying matrix square roots, implementation errors.

1. Introduction

The problem of solving for matrix square roots is considered to be an important special case of nonlinear matrix equation problem, which widely arises in many scientific and engineering fields; e.g., control theory [1], optimization [2], and signal processing [3]. In general, the solution of matrix square roots, which can usually be a fundamental part of many solutions, can be achieved via matrix equations solving. Thus, many numerical algorithms/methods have been presented and developed for online solution of matrix square roots [1–6]. Generally speaking, it may not be efficient enough for most numerical algorithms due to their serial-processing nature performed on digital computers [2, 3]. For large-scale online or real-time applications, the minimal arithmetic operations of such numerical algorithms are usually proportional to the cube of the matrix dimension n , i.e., $O(n^3)$ operations [7]. To remedy inherent weaknesses

of such numerical approaches, many parallel-processing computational methods, including various dynamic-system approaches, have been developed and implemented on specific architectures [8–14]. The neural-dynamic approach is thus now regarded as a powerful alternative to online computation in view of its high-speed parallel-distributed processing property and the convenience of hardware realization [8]. Besides, it is worth mentioning that most reported computational schemes are theoretically/intrinsically designed for time-invariant (or termed, static, constant) problems solving currently, which are usually related to the traditional gradient-based methods [8, 14–18].

Since March 2001, a special recurrent neural network, termed Zhang neural network (ZNN), has been formally proposed by Zhang *et al* [9, 11, 12] for time-varying problems solving (e.g., time-varying Sylvester equation solving, time-varying matrix inversion and optimization). The design of the ZNN is based on a matrix/vector-valued error-function, instead of a norm-based scalar-valued energy-function usually associated with gradient-based neural networks (GNN). In addition, ZNN is depicted in an implicit dynamics which arises frequently in analog electric circuits and systems due to Kirchhoff's rules [11], instead of an explicit dynamics that usually depicts a GNN model.

In the hardware implementation of neural networks, there always exist some realization errors, which are more complicated than the ideal situation. For example, the incapacity of electronic components would limit the performance of the ZNN, and generate various errors (e.g., differentiation error and model-implementation error [15]). Due to these realization errors, the solution of the circuit-implemented ZNN may not be accurate. In this case, robustness analysis of the proposed ZNN would be important and necessary. This paper presents the general framework of the ZNN model solving for time-varying matrix square roots, and, by using a special type of monotonically-increasing odd activation functions (e.g., hyperbolic sine activation functions), the ZNN model with superior robustness in the context of large implementation errors, is analyzed and investigated, which is compared with the ZNN model activated by linear functions.

The rest of this paper is organized as follows. Section 3 introduces the novel problem formulation and investigates the convergence properties of the ZNN model for online solution of time-varying matrix square roots. Section 4 presents the robustness analysis of the ZNN model with hyperbolic sine activation functions in the context of (very) large model-implementation errors. In Section 5, illustrative simulative results are shown to verify the superior convergence and robustness of the ZNN model using hyperbolic sine activation functions for time-varying matrix square roots finding, which further substantiate the theoretical analysis. Finally, conclusions are drawn in Section 6. To the best of the authors' knowledge, there is almost no others' literature dealing with such a specific problem of online solution of time-varying matrix square roots at present stage, and the main contributions of the paper lie in the following facts.

- This paper presents a recurrent neural network (i.e., ZNN) for online solution of time-varying matrix square roots. To pursue the superior conver-

gence and robustness properties, a special type of activation functions (i.e., hyperbolic sine activation functions), which is compared with the linear activation functions, is applied to the ZNN model. This is also the main motivation of the work.

- Together with the theoretical analysis, this paper presents the convergence properties of the proposed ZNN model, which substantiate the superior performance of the ZNN model using hyperbolic sine activation functions, in comparison with that using linear activation functions.
- For potential hardware/circuit implementation and considering its related uncertain realization errors, the robustness property of ZNN model is investigated in the context of large implementation errors, which demonstrates the efficacy and superiority of the proposed ZNN model activated by hyperbolic sine functions.
- The simulative results of different illustrative and representative examples are presented, where different activation functions [i.e., hyperbolic sine functions and linear functions] are exploited in the ZNN model for online solution of time-varying matrix square roots, and the superior convergence and robustness of the proposed ZNN model are thus verified evidently.

2. Related work

Newton iteration [2] has been investigated as an useful method for matrix square roots finding. Similar iterations have been designed and studied differently, such as the Denman and Beavers (DB) method [19] based on the matrix sign function iteration, the Meini iteration based on a cyclic reduction (CR) algorithm [1] and the iteration derived from Newton (IN) method [20]. However, all of those methods, which are designed intrinsically for static matrix square roots finding, may generate large lagging errors in time-varying applications. As a novel class of recurrent neural network (RNN), ZNN has been formally proposed and investigated for the online solution of various time-varying problems. Detailed comparisons between the ZNN model and the four numerical methods (i.e., Newton iteration, DB iteration, CR iteration, and IN iteration) for online solution of static matrix square roots can be found in [21]. Furthermore, the traditional GNN model, which is also designed intrinsically for static matrix square roots finding, is simulated and compared with the ZNN model in [22] for online solution of time-varying matrix square roots.

Moreover, this paper investigates the matrix-valued nonlinear time-varying problem, i.e., the time-varying matrix square roots finding, which is quite different from the authors' previous work (e.g., the linear time-varying problems solving [9, 11, 12, 23, 24]). To further pursue the superior convergence and robustness properties, a special type of activation functions, i.e., hyperbolic sine activation functions (which is compared with the linear activation functions), is applied to the ZNN model. It is worth pointing out that the method of using hyperbolic sine activation functions has seldom been proposed and studied before, which is one of the main motivations of the work.

3. Problem formulation and neural-network solver

In the ensuing subsections, the problem formulation of time-varying matrix square roots is introduced firstly, and then the ZNN model is developed and analyzed for time-varying matrix square roots finding.

3.1. Problem formulation

Let us consider the following time-varying matrix square roots (TVMSR) problem (which can also be viewed as a time-varying nonlinear matrix equation problem):

$$X^2(t) - A(t) = 0, \quad t \in [0, +\infty), \quad (1)$$

where $A(t) \in R^{n \times n}$ denotes a smoothly time-varying positive-definite matrix, which, together with its time derivative $\dot{A}(t)$, are assumed to be known numerically or could be measured accurately. In addition, $X(t)$ is the time-varying unknown matrix to be solved for, and our objective in this work is to find $X(t) \in R^{n \times n}$ so that (1) holds true for any $t \geq 0$.

Before solving (1) in real time, the following preliminaries [1, 3, 25, 26] are provided as a basis for further discussion.

DEFINITION 2.1. *Given a smoothly time-varying matrix $A(t) \in R^{n \times n}$, if matrix $X(t) \in R^{n \times n}$ satisfies the time-varying nonlinear equation $X^2(t) = A(t)$, then $X(t) \in R^{n \times n}$ is a time-varying square root of matrix $A(t) \in R^{n \times n}$ [or to say, $X(t)$ is a time-varying solution to nonlinear equation (1)].*

Square-root existence condition. *If smoothly time-varying matrix $A(t) \in R^{n \times n}$ is positive-definite (in general sense [26]) at any time instant $t \in [0, +\infty)$, then there exists a time-varying matrix square root $X(t) \in R^{n \times n}$ for $A(t)$.*

In addition, it follows from Kronecker-product and vectorization technique [9, 27] that time-varying nonlinear matrix equation (1) could be written as

$$(I \otimes X(t))\text{vec}(X(t)) - \text{vec}(A(t)) = 0, \quad (2)$$

where symbol \otimes denotes the Kronecker product [9, 27], and operator $\text{vec}(\cdot) : R^{n \times n} \rightarrow R^{n^2 \times 1}$ [e.g., $\text{vec}(A(t))$] generates a column vector obtained by stacking all column vectors of the input matrix argument [e.g., $A(t)$] together.

3.2. Zhang neural network

To solve for time-vary matrix square root $A^{1/2}(t)$, by Zhang *et al's* method [9, 11, 12], the following matrix-valued error function could be defined firstly:

$$E(t) = X^2(t) - A(t) \in R^{n \times n}.$$

Secondly, the error-function's time-derivative $\dot{E}(t) \in R^{n \times n}$ could be made such that every entry $e_{ij}(t) \in R$ of $E(t) \in R^{n \times n}$ converges to zero, $i, j =$

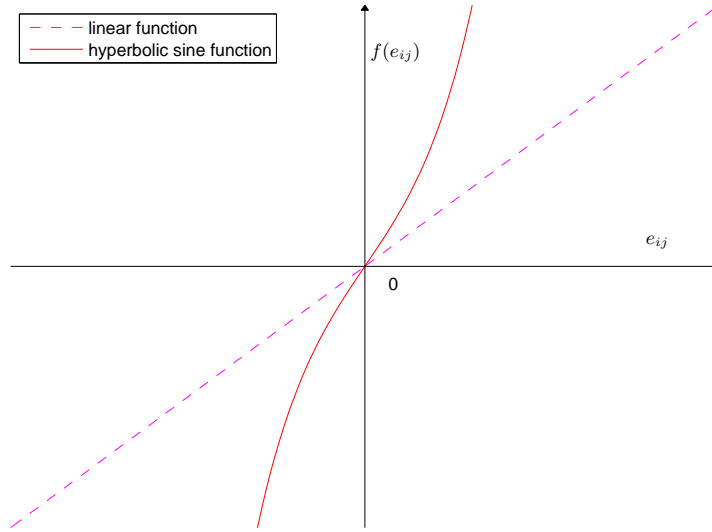


Fig. 1. Activation function $f(\cdot)$ used in the neural network.

$1, 2, \dots, n$; in mathematics, to choose $\dot{e}_{ij}(t)$ such that $\lim_{t \rightarrow \infty} e_{ij}(t) = 0, \forall i, j \in \{1, 2, \dots, n\}$. A general form of $\dot{E}(t)$ is given by Zhang *et al* as

$$\frac{dE(t)}{dt} = -\Gamma \mathcal{F}(E(t)), \quad (3)$$

where design parameter Γ and activation-function array $\mathcal{F}(\cdot)$ are described as follows.

- $\Gamma \in R^{n \times n}$ is a positive-definite (diagonal) matrix used to scale the convergence rate of the neural network. For simplicity, Γ can be γI with scalar $\gamma > 0 \in R$ and I denoting the identity matrix. Γ (or γI), being a set of reciprocals of capacitance parameters in the hardware implementation, should be set as large as the hardware would permit (e.g., in analog circuits or VLSI [15]), or selected appropriately for experimental and/or simulative purposes.
- $\mathcal{F}(\cdot) : R^{n \times n} \rightarrow R^{n \times n}$ denotes an activation-function matrix array of the neural network. In general, any monotonically-increasing odd activation function $f(\cdot)$, being the ij th element of $\mathcal{F}(\cdot)$, can be used for the construction of the neural network. In this paper, the following two types of activation functions are discussed and compared (which are shown in Figure 1):
 - linear activation function $f(e_{ij}) = e_{ij}$, and
 - hyperbolic sine activation function $f(e_{ij}) = \exp(\xi e_{ij})/2 - \exp(-\xi e_{ij})/2$ with design parameter $\xi \geq 1$.

Thirdly, expanding ZNN design formula (3) leads to the following implicit dynamic equation of the ZNN model for online matrix square roots finding via nonlinear time-varying equation (1) solving:

$$X(t)\dot{X}(t) + \dot{X}(t)X(t) = -\gamma\mathcal{F}(X^2(t) - A(t)) + \dot{A}(t), \quad (4)$$

where $X(t)$, starting from an initial condition $X(0) \in R^{n \times n}$, is the activation state matrix corresponding to theoretical time-varying matrix square root $X^*(t)$ of $A(t)$. It is worth mentioning that, when using a linear activation function array $\mathcal{F}(\cdot)$, the general nonlinearly-activated ZNN (4) reduces to the following linearly-activated one:

$$X(t)\dot{X}(t) + \dot{X}(t)X(t) = -\gamma X^2(t) + \gamma A(t) + \dot{A}(t). \quad (5)$$

For simulative purposes, based on the Kronecker-product and vectorization technique [9, 27], the matrix differential equation (4) can be transformed to a vector differential equation. We thus obtain the following theorem.

THEOREM 2.1. *The matrix-form differential equation (4) can be reformulated as the following vector-form differential equation:*

$$(I \otimes X + X^T \otimes I)\text{vec}(\dot{X}) = -\gamma\mathcal{F}((I \otimes X)\text{vec}(X) - \text{vec}(A)) + \text{vec}(\dot{A}), \quad (6)$$

where superscript T denotes the transpose of a matrix or a vector, and activation-function array $\mathcal{F}(\cdot)$ in (6) is defined as before except that its dimensions are changed to be $R^{n^2 \times 1} \rightarrow R^{n^2 \times 1}$. In the simulation, $M(t, x) := I \otimes X + X^T \otimes I$ denotes the nonsingular mass matrix of a standard ordinary-differential-equation (ODE) problem.

Proof. See Appendix A for details.

In addition, for ZNN (4) which solves for the time-varying matrix square root of $A(t)$, we have the following theorems on its convergence.

THEOREM 2.2. *Consider smoothly time-varying matrix $A(t) \in R^{n \times n}$ in nonlinear equation (1), which satisfies the square-root existence condition. If a monotonically increasing odd activation-function array $\mathcal{F}(\cdot)$ is used, then error-function $E(t) = X^2(t) - A(t) \in R^{n \times n}$ of ZNN (4), starting from randomly-generated positive-definite (or negative-definite) diagonal initial state-matrix $X(0) \in R^{n \times n}$, can converge to zero [which implies that state matrix $X(t) \in R^{n \times n}$ of ZNN (4) can converge to theoretical positive-definite (or negative-definite) time-varying matrix square root $X^*(t)$ of $A(t)$].*

Proof. From the compact form of ZNN design formula $\dot{E}(t) = -\Gamma\mathcal{F}(E(t))$, a set of n^2 decoupled differential equations can be written equivalently as follows:

$$\dot{e}_{ij}(t) = -\gamma f(e_{ij}(t)), \quad (7)$$

for any $i \in \{1, 2, 3, \dots, n\}$ and $j \in \{1, 2, 3, \dots, n\}$. Thus, to analyze the equivalent ij th subsystem (7), we define a Lyapunov function candidate $v_{ij}(t) = e_{ij}^2(t)/2 \geq 0$ with its time-derivative

$$\frac{dv_{ij}(t)}{dt} = e_{ij}(t)\dot{e}_{ij}(t) = -\gamma e_{ij}(t)f(e_{ij}(t)).$$

As mentioned previously, $f(\cdot)$ is a monotonically-increasing odd activation function; i.e., $f(-e_{ij}(t)) = -f(e_{ij}(t))$. Then, the following result is obtained:

$$e_{ij}(t)f(e_{ij}(t)) \begin{cases} > 0, & \text{if } e_{ij}(t) \neq 0, \\ = 0, & \text{if } e_{ij}(t) = 0, \end{cases}$$

which guarantees the final negative-definiteness of \dot{v}_{ij} (i.e., $\dot{v}_{ij} < 0$ for $e_{ij} \neq 0$ while $\dot{v}_{ij} = 0$ for $e_{ij} = 0$ only). By Lyapunov theory [28], equilibrium point $e_{ij} = 0$ of (7) is asymptotically stable; i.e., $e_{ij}(t)$ converges to zero, for any $i \in \{1, 2, 3, \dots, n\}$ and $j \in \{1, 2, 3, \dots, n\}$. In other words, the matrix-valued error-function $E(t) = [e_{ij}(t)] \in R^{n \times n}$ is convergent to zero. In addition, we have $E(t) = X^2(t) - A(t)$; or equivalently, $X^2(t) = A(t) + E(t)$. Since $E(t) \rightarrow 0$ as $t \rightarrow +\infty$, we have $X^2(t) \rightarrow A(t)$ [i.e., $X(t) \rightarrow X^*(t)$] as $t \rightarrow +\infty$. That is, state matrix $X(t)$ of ZNN model (4) can converge to the theoretical time-varying matrix square root $X^*(t)$ of $A(t)$.

Furthermore, when state matrix $X(t)$ of ZNN model (4) starting from a randomly-generated positive-definite diagonal initial state-matrix $X(0)$, it can converge to the positive-definite time-varying matrix square root $A^{1/2}(t)$ [i.e., a form of $X^*(t)$]. This can be proofed by contradiction. Suppose that state matrix $X(t)$ starting from a positive-definite diagonal initial state-matrix $X(0)$ converges to the negative-definite TVMSR $-A^{1/2}(t)$ [i.e., the other form of $X^*(t)$], then state-matrix $X(t)$ must pass through at least one 0-eigenvalue, which leads to the contradiction that the left and right hand sides of ZNN (4) can not hold. So, starting from a randomly-generated positive-definite diagonal initial state-matrix $X(0)$, state matrix $X(t)$ of ZNN model (4) can converge to the positive-definite time-varying matrix square root $A^{1/2}(t)$. Similarly, it can proved that, starting from a randomly-generated negative-definite diagonal initial state-matrix $X(0)$, state matrix $X(t)$ of ZNN model (4) can converge to the negative-definite time-varying matrix square root $-A^{1/2}(t)$. The proof is thus complete.

THEOREM 2.3. *In addition to Theorem 2.2, if a linear activation function array $\mathcal{F}(\cdot)$ is used, then the matrix-valued error-function $E(t) = X^2(t) - A(t) \in R^{n \times n}$ of ZNN (4), starting from a randomly-generated positive-definite (or negative-definite) diagonal initial-state-matrix $X(0) \in R^{n \times n}$, can exponentially converge to zero with convergence rate γ , which corresponds to the convergence of state matrix $X(t) \in R^{n \times n}$ of ZNN (4) to $X^*(t)$. Moreover, if the hyperbolic sine activation function array is used, then the superior convergence can be achieved for ZNN (4), as compared to the linear activation case.*

Proof. 1) If the linear activation function $f(e_{ij}) = e_{ij}$ is used, from the ij th subsystem (7), we have $\dot{e}_{ij} = -\gamma e_{ij}$, and thus $e_{ij}(t) = \exp(-\gamma t)e_{ij}(0)$. In other words, the matrix-valued error-function $E(t) \in R^{n \times n}$ can be expressed explicitly

as

$$E(t) = \begin{bmatrix} e_{11}(0) & e_{12}(0) & \cdots & e_{1n}(0) \\ e_{21}(0) & e_{22}(0) & \cdots & e_{2n}(0) \\ \vdots & \vdots & \ddots & \vdots \\ e_{n1}(0) & e_{n2}(0) & \cdots & e_{nn}(0) \end{bmatrix} \exp(-\gamma t) = E(0) \exp(-\gamma t).$$

This evidently shows that the error function $E(t)$ exponentially converges to zero with convergence rate γ for the ZNN (4) activated by the linear activation function array. In addition, we have $E(t) = X^2(t) - A(t)$ and then $X^2(t) = A(t) + E(t)$, i.e., $X^2(t) = A(t) + E(0) \exp(-\gamma t)$. Since $E(0) \exp(-\gamma t) \rightarrow 0$ exponentially as $t \rightarrow +\infty$, we have again $X^2(t) \rightarrow A(t)$ and $X(t) \rightarrow X^*(t)$ as $t \rightarrow +\infty$. That is, state matrix $X(t)$ of ZNN (4) can converge to the theoretical time-varying matrix square root $X^*(t)$ of $A(t)$.

2) Define a Lyapunov function candidate

$V := \|E(t)\|_F^2/2 = \text{trace}(E^T(t)E(t))/2 = \text{vec}^T(E(t))\text{vec}(E(t))/2 \geq 0$ for ZNN (4). Since

$$\text{vec}(E(t)) := [e_{11}, \dots, e_{n1}, e_{12}, \dots, e_{n2}, \dots, e_{1n}, \dots, e_{nn}]^T \rightarrow 0$$

is equivalent to $E(t) \rightarrow 0$, then we use $\text{vec}(E(t))$ instead of $E(t)$ to analyze the Lyapunov function candidate and its time derivative. Thus, defining e_i as the i th element of $\text{vec}(E(t))$, we have

$$V(t) = \sum_{i=1}^{n^2} e_i^2(t)/2, \text{ and } \dot{V}(t) = \sum_{i=1}^{n^2} e_i(t)\dot{e}_i(t) = -\gamma \sum_{i=1}^{n^2} e_i(t)f(e_i(t)).$$

So, if linear activation functions are used, we have

$$V(t)_{\text{lin}} = \sum_{i=1}^{n^2} e_i^2(t)/2 \geq 0, \text{ and } \dot{V}(t)_{\text{lin}} = -\gamma \sum_{i=1}^{n^2} e_i^2(t) \leq 0.$$

This implies that the matrix-valued error-function $E(t)$ of linearly-activated ZNN model (5) can converge to zero according to the aforementioned discussion and Lyapunov theory [28].

On the other hand, if hyperbolic sine functions are used, the corresponding Lyapunov function candidate is still $v(t)_{\text{hs}} = v(t)_{\text{lin}} \geq 0$. By Taylor series expansion, the aforementioned hyperbolic sine function is formulated as

$$\begin{aligned} f(e_i) &= [\exp(\xi e_i) - \exp(-\xi e_i)]/2 \\ &= [2(\xi e_i) + 2 \cdot \frac{(\xi e_i)^3}{3!} + 2 \cdot \frac{(\xi e_i)^5}{5!} + \dots]/2 \\ &= \xi e_i + \frac{(\xi e_i)^3}{3!} + \frac{(\xi e_i)^5}{5!} + \dots \\ &= \sum_{r=1}^{+\infty} \frac{\xi^{2r-1} (e_i)^{2r-1}}{(2r-1)!}. \end{aligned}$$

Thus, we have the following derivation:

$$\begin{aligned}
 \dot{v}(t)_{\text{hs}} &= -\gamma \sum_{i=1}^{n^2} e_i f(e_i) \\
 &= -\gamma \sum_{i=1}^{n^2} e_i \sum_{r=1}^{+\infty} \frac{\xi^{2r-1} (e_i)^{2r-1}}{(2r-1)!} \\
 &= -\gamma \sum_{i=1}^{n^2} \sum_{r=1}^{+\infty} \frac{\xi^{2r-1} (e_i)^{2r}}{(2r-1)!} \\
 &\ll -\gamma \sum_{i=1}^{n^2} e_i^2 = \dot{v}(t)_{\text{lin}} \leq 0.
 \end{aligned}$$

From the aforementioned analysis procedure, we see that Lyapunov function candidate $v(t)_{\text{hs}}$ can diminish to zero when hyperbolic sine activation functions are used, with much faster convergence rate than that using linear activation functions. This implies that, when hyperbolic sine functions are exploited, nonlinearly-activated ZNN model (4) possesses superior convergence in comparison with linearly-activated ZNN model (5). The proof is thus completed.

REMARK 2.1. The construction of ZNN (4) allows us to have many more choices of different activation functions. In many engineering applications, it may be necessary to investigate the impact of different activation functions in the RNN, in view of the fact that nonlinearity always exists. Even if the linear activation function is used, the nonlinear phenomenon may appear in its hardware implementation, e.g., in the form of saturation and/or inconsistency of the linear slope, or due to truncation and round-off errors of digital realization [15]. The investigation of different activation functions may give more insights into the imprecise-implementation problem of neural networks.

REMARK 2.2. One more advantage of using the hyperbolic sine function over the linear function lies in the extra design parameter ξ , which is an effective factor of the convergence rate. When there is an upper bound on γ due to hardware implementation, the parameter ξ will be another effective factor expediting the ZNN (4) convergence. The convergence for the hyperbolic sine activation functions can be much faster than that for the linear activation functions, when using the same level of design parameters ξ and γ . This is because the error signal $e_{ij} = [X^2 - A]_{ij}$ in (4) is amplified by the hyperbolic sine activation function for the whole error range $(-\infty, +\infty)$ (i.e., the larger slope and absolute value of the hyperbolic sine activation function in the whole error range as shown in Figure 1).

4. Robustness analysis

In the analog implementation or simulation of recurrent neural networks, we usually assume that it is under ideal conditions. However, there always exist

some realization errors in hardware implementation. The differentiation errors of $A(t)$, and the model-implementation error appear most frequently in the hardware realization. For these realization errors possibly appearing in model (4), we investigate the ZNN robustness by considering the following matrix-valued ZNN design formula perturbed with a large model-implementation error:

$$X(t)\dot{X}(t) + \dot{X}(t)X(t) = -\gamma\mathcal{F}(X^2(t) - A(t)) + \dot{A}(t) + \Delta\omega, \quad (8)$$

where $\Delta\omega \in R^{n \times n}$ denotes the general model-implementation errors [including the differentiation errors of matrix $A(t)$ as a part]. For these errors, the following lemmas on the robustness of largely-perturbed ZNN model (8) could be achieved [23, 24].

LEMMA 3.1. *Consider the above perturbed ZNN model with a large model implementation error $\Delta\omega \in R^{n \times n}$ finally depicted in equation (8). If $0 \leq \|\Delta\omega\|_F \leq \varepsilon < \infty$ for any $t \in [0, +\infty]$, then the steady-state residual error $\lim_{t \rightarrow \infty} \|E(t)\|_F$ is always uniformly upper bounded by some positive scalar, provided that the design parameter $\gamma > 0$ is large enough (the so-called design-parameter requirement). Furthermore, the steady-state residual error $\lim_{t \rightarrow \infty} \|e(t)\|_F$ decreases to zero as γ tends to positive infinity.*

LEMMA 3.2. *In addition to the general robustness results given in Lemma 3.1, the largely-perturbed ZNN model (8) possesses the following properties.*

- *With linear activation functions used, the steady-state error $\lim_{t \rightarrow \infty} \|E(t)\|_F$ can be written out as a positive scalar under the design parameter requirement.*
- *With hyperbolic sine activation functions used, the superior convergence and robustness properties exist for the whole error range $(-\infty, +\infty)$; i.e., the design-parameter requirement can be removed in this case and the steady-state residual error $\lim_{t \rightarrow \infty} \|E(t)\|_F$ can be made (much) smaller [which can be further done by increasing γ and/or ξ], as compared to the situation of using linear activation functions.*

5. Illustrative examples

The previous sections have presented the convergence and robustness results of ZNN model (4) [together with a largely perturbed ZNN model (8)] for online solution of time-varying matrix square root problem (1). In this section, computer-simulation results and observations are provided to verify the superior characteristics of using hyperbolic sine activation functions to those of using linear activation functions.

Example 1. For illustration and comparison, let us consider equation (1) with the following time-varying matrix $A(t)$:

$$A(t) = \begin{bmatrix} 16 + \sin 4t \cos 4t & 7 \sin 4t \\ 7 \cos 4t & 9 + \sin 4t \cos 4t \end{bmatrix}. \quad (9)$$

Superior Performance of ZNN with Hyperbolic Sine Activation Functions

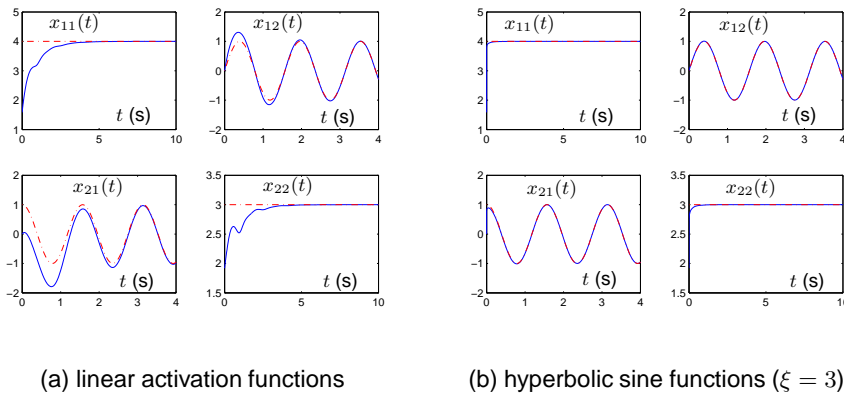


Fig. 2. Online solution of time-varying matrix square root problem (9) by ZNN (4) with $\gamma = 1$, where the neural-network solutions $X(t)$ are denoted by solid blue curves, and the theoretical time-varying square root $A^{1/2}(t)$ is denoted by red dash-dotted curves.

Simple manipulations may verify that a theoretical time-varying matrix square root $X^*(t)$ of $A(t)$ could be

$$X^*(t) = \begin{bmatrix} 4 & \sin 4t \\ \cos 4t & 3 \end{bmatrix},$$

which is used to check the correctness of the neural-network solution $X(t)$. Based on the aforementioned vector-form ZNN model (6) proposed in Theorem 2.1, we can obtain the simulated ZNN state $X(t)$ by using ODE routine “ode23t” [25]. As illustrated in Figure 2, starting from randomly-generated positive-definite diagonal initial-state $X(0) \in [0, 2]^{2 \times 2}$, neural state $X(t)$ of ZNN (4) using hyperbolic sine activation functions converges to the theoretical time-varying solution much faster, compared to that using linear activation functions. Furthermore, in order to further investigate the convergence performance, we monitor and show the residual error $\|E(t)\|_F$ during the problem solving process of ZNN. As seen from Figure 3, the residual errors $\|E(t)\|_F$ of ZNN (4) all decrease rapidly to zero, where the convergence rate of ZNN (4) using hyperbolic sine activation functions appears to be 5 times faster than that using linear activation functions. From these figures, we can confirm well the theoretical results given in Theorems 2.2 and 2.3.

To show the robustness characteristics of the largely-perturbed ZNN model, the following large model implementation error $\Delta\omega$ is specially added in (8):

$$\Delta\omega = \begin{bmatrix} 10^2 & 10^2 \\ 10^2 & 10^2 \end{bmatrix}.$$

As we can see from Figure 4, with the large model-implementation error, the steady-state residual error $\lim_{t \rightarrow \infty} \|E(t)\|_F$ of the perturbed ZNN (8) is still

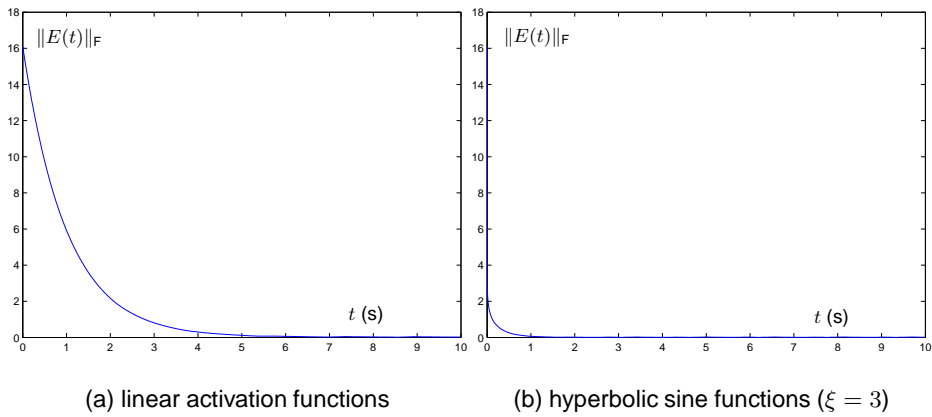


Fig. 3. Residual errors of ZNN (4) with $\gamma = 1$ for TVMSR finding of (9).

bounded. In addition, with $\gamma = 1$, ZNN (8) using linear activation functions results in large residual errors. This is shown evidently in Figure 4(a). In contrast, as shown comparatively in Figure 4(a) and (b), when hyperbolic sine activation functions are used, the convergence time of the perturbed ZNN (8) is faster than that using linear activation functions, and the steady-state residual error of the perturbed ZNN (8) is around 50 times smaller than that using linear activation functions. Furthermore, Figure 4(b) is about using hyperbolic sine activation functions with design parameter $\xi = 3$, while Figure 5(a) and (b) are about $\xi = 5$ and $\xi = 7$, respectively. It is observed that, by increasing ξ , the steady-state residual error $\lim_{t \rightarrow \infty} \|E(t)\|_F$ of ZNN (8) is decreased very effectively (e.g., for $\xi = 7$, which is around 133 times smaller than that using linear activation functions). In addition, comparing Figure 4(b) and Figure 6, we can see that, as the design parameter γ increases from 1 to 10 and then to 100, the upper bound of the steady-state residual error is decreased effectively from around 3.53 to 1.99 and then to 0.59. Note that, when γ is much larger (e.g., 10^6 or 10^9), the steady-state residual error would become tiny. Besides, if the very large model-implementation error (e.g., $\Delta\omega_{ij} = 10000$ with $i, j = 1, 2$) is added to the ZNN model, the robustness results are shown in Figure 7, from which the same conclusion can be drawn; i.e., using hyperbolic sine activation functions results in a much smaller steady-state residual error than using linear activation functions [e.g., around 5000 times smaller, as seen comparatively from Figure 7(a) and (b)].

Example 2. In order to further investigate the efficacy of ZNN models [including (4) and (8)] using hyperbolic sine activation functions, let us consider equation (1) with the following symmetric positive-definite time-varying matrix $A(t)$ with its theoretical time-varying square root $X^*(t)$ given as well for com-

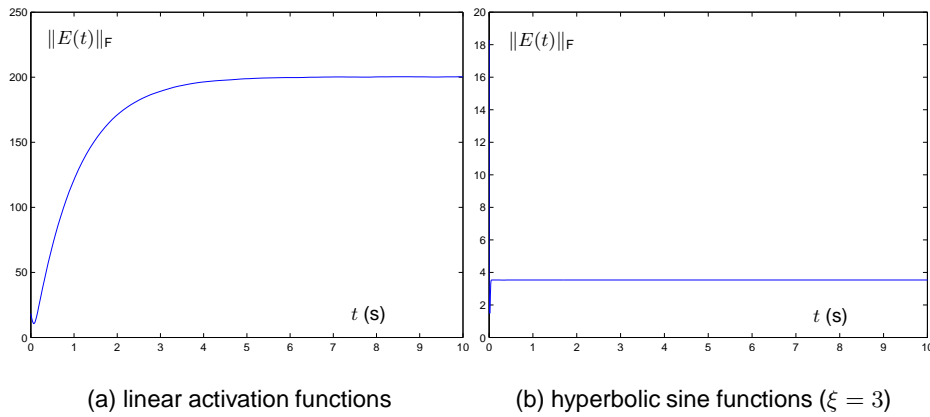


Fig. 4. Residual errors of largely-perturbed ZNN (8) for time-varying matrix square roots finding of (9) (with $\gamma = 1$ and $\Delta\omega_{ij} = 10^2, i, j \in \{1, 2\}$).

parison purposes:

$$A(t) = \begin{bmatrix} 5 + 0.25s^2 & 2s + 0.5c & 4 + 0.25s \times c \\ 2s + 0.5c & 4.25 & 2c + 0.5s \\ 4 + 0.25s \times c & 2c + 0.5s & 5 + 0.25c^2 \end{bmatrix}, \quad (10)$$

$$X^*(t) = \begin{bmatrix} 2 & 0.5s & 1 \\ 0.5s & 2 & 0.5c \\ 1 & 0.5c & 2 \end{bmatrix},$$

where s and c denote $\sin(6t)$ and $\cos(6t)$, respectively.

As illustrated in Figure 8, starting from randomly-generated positive-definite diagonal initial-state $X(0) \in [0, 2]^{3 \times 3}$, state matrix $X(t)$ of ZNN (4) using hyperbolic sine activation functions converges faster than that using linear activation functions. In order to further investigate the convergence performance, we monitor and show the residual error $\|E(t)\|_F$ during the problem solving process of ZNN. From Figure 9, we observe that the residual errors $\|E(t)\|_F$ of ZNN (4) all decrease rapidly to zero, where the convergence rate of ZNN (4) using hyperbolic sine activation functions is also 5 times faster than that using linear activation functions. These figures substantiate again the theoretical results given in Theorems 2.2 and 2.3.

To comparatively show the robustness characteristics of the largely-perturbed ZNN model, we exploit once more the large model-implementation error $\Delta\omega_{ij} = 10^2, i, j \in \{1, 2, 3\}$, which is added in ZNN model (8). With design parameter $\gamma = 1$ and two types of activation functions used, the robustness performance of the largely-perturbed ZNN model (8) is shown in Figure 10, where the steady-state residual errors $\lim_{t \rightarrow \infty} \|E(t)\|_F$ are all still bounded. In addition, as shown in Figure 10(a) and (b), when hyperbolic sine activation functions with

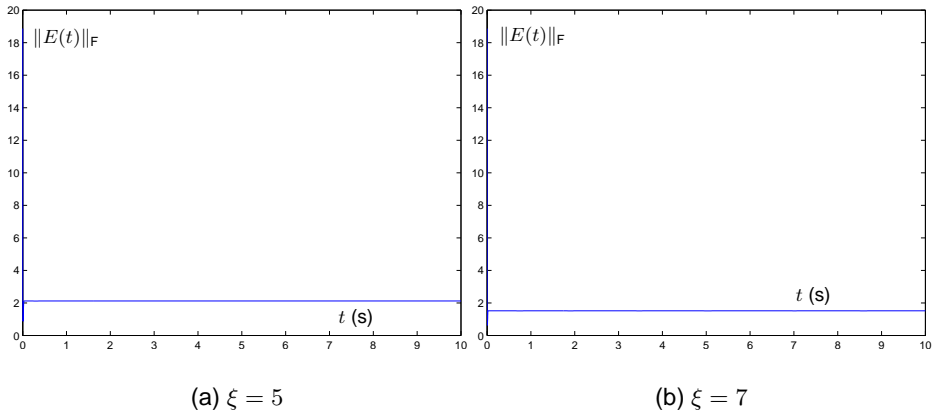


Fig. 5. Residual errors of largely-perturbed ZNN (8) using hyperbolic sine activation functions with different values of ξ (still with $\gamma = 1$ and $\Delta\omega_{ij} = 10^2, i, j \in \{1, 2\}$).

$\xi = 5$ are used, the convergence time of the largely-perturbed ZNN (8) is faster than that using linear activation functions, and the steady-state residual error of the largely-perturbed ZNN (8) is around 100 times smaller than that using linear activation functions. In summary, compared to the situation of using linear activation functions, superior robustness performance is achieved for the ZNN models by using hyperbolic sine activation functions.

Example 3. In order to further investigate the efficacy of ZNN model (4) using hyperbolic sine activation functions for larger-dimension matrices, let us consider equation (1) with the following time-varying Toeplitz matrix $A(t)$:

$$A(t) = \begin{bmatrix} a_1(t) & a_2(t) & a_3(t) & \cdots & a_n(t) \\ a_2(t) & a_1(t) & a_2(t) & \cdots & a_{n-1}(t) \\ a_3(t) & a_2(t) & a_1(t) & \cdots & a_{n-2}(t) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_n(t) & a_{n-1}(t) & a_{n-2}(t) & \cdots & a_1(t) \end{bmatrix} \in R^{n \times n}. \quad (11)$$

Let $a_1(t) = n + \sin(5t)$, and $a_k(t) = \cos(5t)/(k-1)$ with $k = 2, 3, \dots, n$. Figure 11 shows the simulation results of ZNN model (4) using hyperbolic sine activation functions for time-varying square roots finding of the above Toeplitz matrix $A(t)$ in the situation of $n = 4$ and $n = 10$. As seen from Figure 11(a) and (b), the residual errors $\|E(t)\|_F$ of ZNN (4) for factorizing Toeplitz matrices with different dimensions (i.e., $R^{4 \times 4}$ and $R^{10 \times 10}$) both diminish to zero, which implies that their corresponding state matrices always converge to the time-varying square roots of $A(t)$. These further substantiate the efficacy of the ZNN model (4) using hyperbolic sine activation functions on solving for time-varying square roots of larger-dimension matrices.

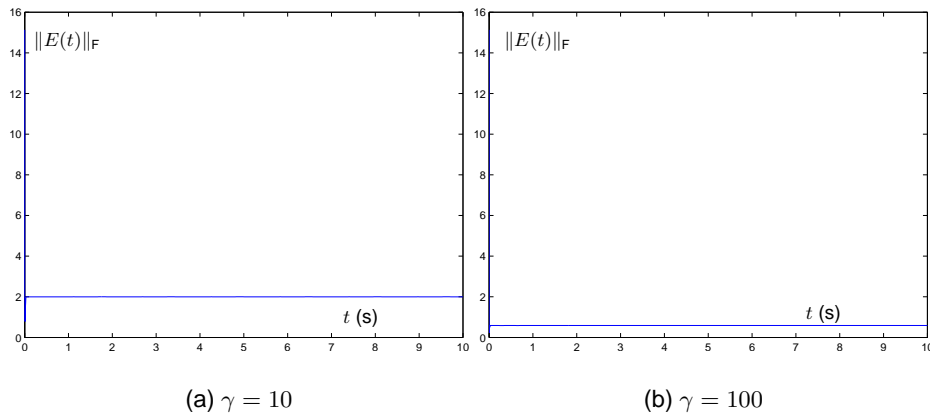


Fig. 6. Residual errors of largely-perturbed ZNN (8) using hyperbolic sine activation functions with different values of γ (with $\xi = 3$ and $\Delta\omega_{ij} = 10^2, i, j \in \{1, 2\}$).

Before ending this section, it is worth noting that, because of the similarity of the results and figures to the above ones, the corresponding simulations of ZNN models starting from negative-definite initial states are not presented (though they have done successfully and consistently with the theoretical results given in the theorems and lemmas of the paper). Besides, comparisons between ZNN model (4) and other methods can be seen in Appendix B.

6. Conclusions

In this paper, the convergence and robustness properties of Zhang neural network using hyperbolic sine activation functions have been investigated, analyzed and verified for the online solution of time-varying matrix square roots. Other computer-simulation results of using different activation functions (e.g., sigmoid activation functions, hard-limiting activation functions, piecewise-linear activation functions, and hyperbolic tangent activation functions) have been omitted due to the similarity and the space limitation, and thus hyperbolic activation functions have only been compared with linear activation functions in this paper. Theoretical analysis has demonstrated that superior convergence and robustness can be achieved readily for ZNN models even in the context of (very) large model-implementation errors by using hyperbolic sine activation functions, as compared to those using linear activation functions. Computer-simulation results have further substantiated the efficacy and superiority of the ZNN models using hyperbolic sine activation functions for time-varying matrix square roots finding.

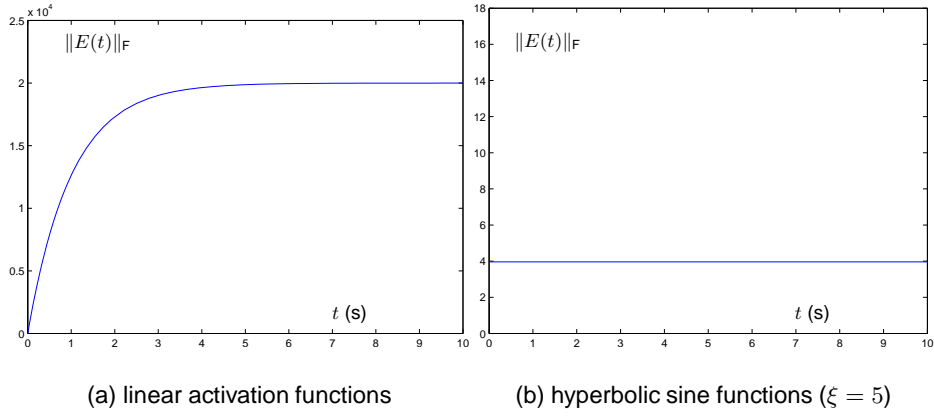


Fig. 7. Residual errors of very largely perturbed ZNN (8) for time-varying matrix square roots finding of (9) (with $\gamma = 1$ and $\Delta\omega_{ij} = 10^4, i, j \in \{1, 2\}$).

Appendix A

Proof of Theorem 2.1

By vectorizing ZNN model (4) based on the Kronecker-product and vectorization technique, the left hand side of equation (4) is

$$\begin{aligned} \text{vec}(X\dot{X} + \dot{X}X) &= \text{vec}(X\dot{X}) + \text{vec}(\dot{X}X) \\ &= (I \otimes X)\text{vec}(\dot{X}) + (X^T \otimes I)\text{vec}(\dot{X}) \\ &= (I \otimes X + X^T \otimes I)\text{vec}(\dot{X}), \end{aligned}$$

where argument t is dropped for presentation convenience. The right hand side of (4) is

$$\begin{aligned} \text{vec}(-\gamma\mathcal{F}(X^2 - A) + \dot{A}) &= -\gamma\text{vec}(\mathcal{F}(X^2 - A) + \dot{A}) \\ &= -\gamma\text{vec}(\mathcal{F}(X^2 - A)) + \text{vec}(\dot{A}). \end{aligned} \quad (12)$$

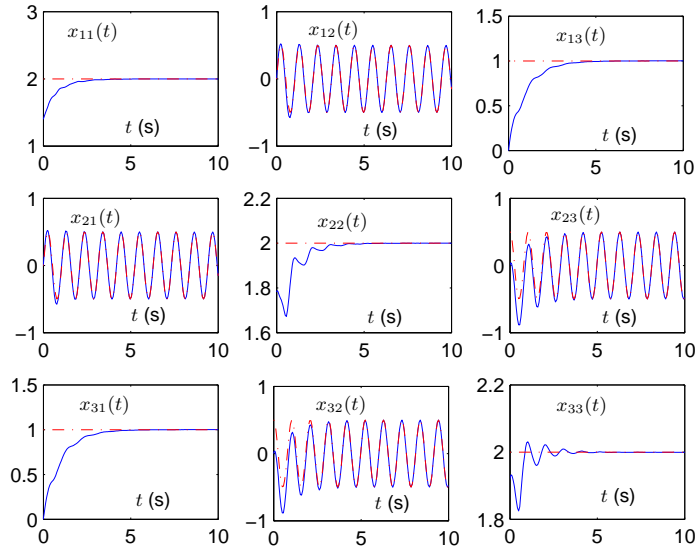
Note that the aforementioned activation function array $\mathcal{F}(\cdot)$ could also be vectorized, i.e., being from $R^{n^2 \times 1}$ to $R^{n^2 \times 1}$. Thus, we have

$$\begin{aligned} \text{vec}(\mathcal{F}(X^2 - A)) &= \mathcal{F}(\text{vec}(X^2 - A)) \\ &= \mathcal{F}(\text{vec}(X^2) - \text{vec}(A)) \\ &= \mathcal{F}((I \otimes X)\text{vec}(X) - \text{vec}(A)). \end{aligned} \quad (13)$$

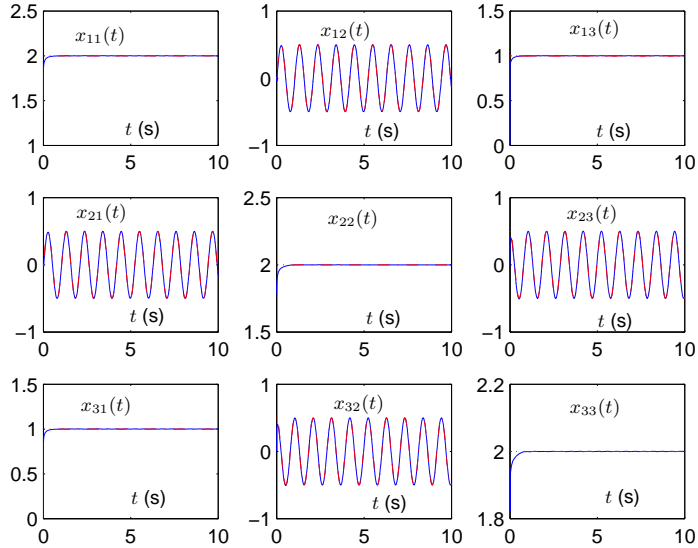
Combining (12) and (13) yields the vectorization of the right hand side of equation (4):

$$\text{vec}(-\gamma\mathcal{F}(X^2 - A) + \dot{A}) = -\gamma\mathcal{F}((I \otimes X)\text{vec}(X) - \text{vec}(A)) + \text{vec}(\dot{A}).$$

Superior Performance of ZNN with Hyperbolic Sine Activation Functions



(a) linear activation functions



(b) hyperbolic sine functions with $\xi = 5$

Fig. 8. Online solution of time-varying matrix square root problem (10) by ZNN (4) with $\gamma = 1$, where the neural-network solutions $X(t)$ are denoted by solid blue curves, and the theoretical time-varying square root $A^{1/2}(t)$ is denoted by red dash-dotted curves.

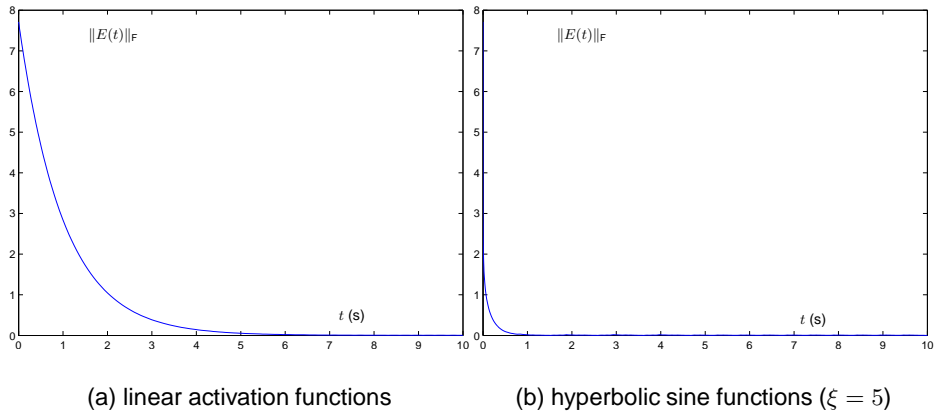


Fig. 9. Residual errors of ZNN (4) with $\gamma = 1$ for TVMSR finding of (10).

Evidently, the vectorization of both sides of matrix-form differential equation (4) should be equal, which generates the vector-form differential equation (6). The proof of Theorem 2.1 is thus completed.

Appendix B

To keep the completeness of the paper and make interesting comparisons with ZNN model (4), numerical methods are presented and investigated here for the online solution of time-varying matrix square roots as well. Based on the results in [5, 6], four numerical methods (i.e., Newton iteration, DB iteration, CR iteration, and IN iteration) are used frequently to solve for static matrix square roots, which is the constant case of (1). For comparison, the important iterative formulas of the numerical methods are listed as follows in order to solve for the time-varying matrix square roots.

– Newton iteration:

$$\begin{cases} X_k H_k + H_k X_k = A_k - X_k^2, \\ X_{k+1} = X_k + H_k, \end{cases} \quad (14)$$

where iteration index $k = 0, 1, 2, 3, \dots$. In addition, $A(t)$ is discretized by the standard sampling method, of which the sampling gap is denoted by $\tau = t_{k+1} - t_k$. For convenience and for consistency with X_k , we use A_k standing for $A(t = k\tau)$.

– DB iteration:

$$\begin{cases} X_{k+1} = (X_k + Y_k^{-1})/2, & X_0 = A_0, \\ Y_{k+1} = (Y_k + X_k^{-1})/2, & Y_0 = I, \end{cases} \quad (15)$$

where $A_0 = A(t = 0)$.

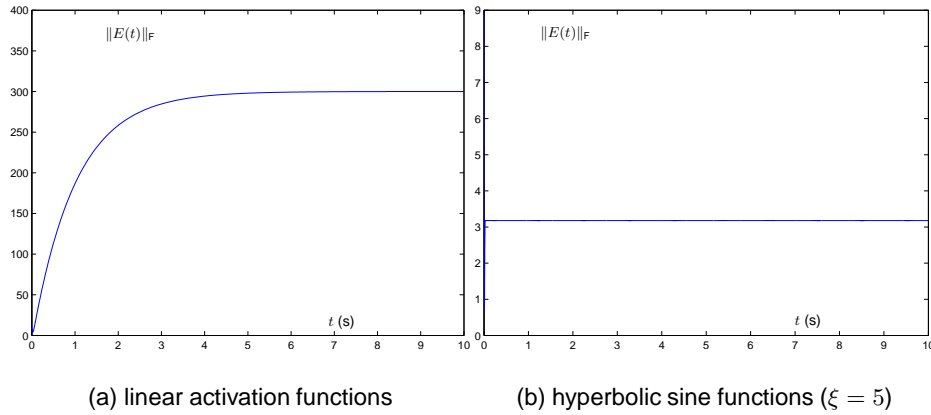


Fig. 10. Residual errors of largely-perturbed ZNN (8) for time-varying matrix square roots finding of (10) (with $\gamma = 1$ and $\Delta\omega_{ij} = 10^2, i, j \in \{1, 2, 3\}$).

– CR iteration:

$$\begin{cases} Y_{k+1} = -Y_k Z_k^{-1} Y_k, & Y_0 = I - A_0, \\ Z_{k+1} = Z_k + 2Y_{k+1}, & Z_0 = 2(I + A_0), \\ X_{k+1} = Z_{k+1}/4. \end{cases} \quad (16)$$

– IN iteration:

$$\begin{cases} X_{k+1} = X_k + E_k, & X_0 = A_0, \\ E_{k+1} = -E_k X_{k+1}^{-1} E_k/2, & E_0 = (I - A_0)/2. \end{cases} \quad (17)$$

According to the formulas above, the solving process of DB iteration (15), CR iteration (16) and IN iteration (17) do not contain the information of A_k in the iteration process [i.e., without A_k in the formulas], which implies that the sequences $\{X_k\}$ generated by these methods converge to the matrix square roots of A_0 (in view of the initial conditions) and then do not change [though the matrix $A(t)$ changes]. The three numerical methods are theoretically/intrinsically designed for online solution of time-invariant (or termed, static, constant) matrix square roots, but not aimed at finding the time-varying matrix square roots in real time t . The three methods may be approximately effective, when the time-varying problem can be divided into many static problems to solve via a short-time invariance assumption [i.e., to find the matrix square roots of A_k separately and successively], but this may not be accurate and may not be what solving a time-varying problem in real time t means. Therefore, DB iteration (15), CR iteration (16) and IN iteration (17) are, generally speaking, not applicable to online solution of time-varying matrix square roots.

Different with the aforementioned three iterations, Newton iteration (14) contains A_k in the formula, so the sequence $\{X_k\}$ generated by Newton iteration

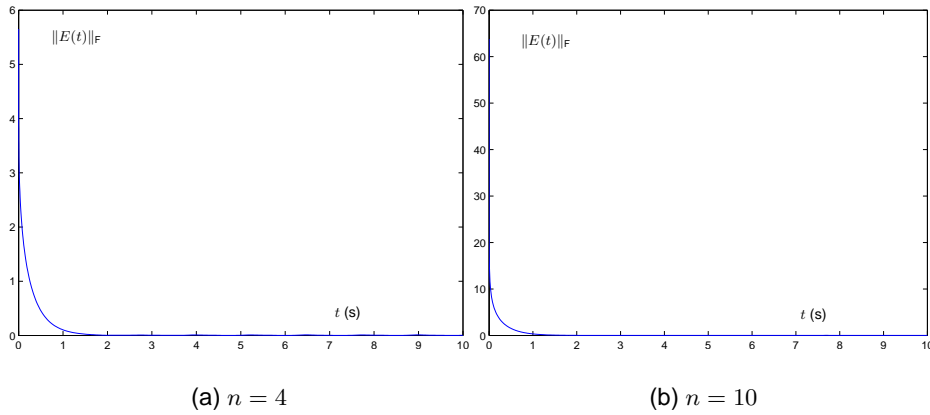


Fig. 11. Residual errors of ZNN (4) using hyperbolic sine activation functions for time-varying square roots finding of the Toeplitz matrix $A(t)$ with different dimensions.

(14) can change with A_k correspondingly [e.g., X_2 corresponds to the estimation of the matrix square root of $A_2 = A(t = 2\tau)$] and try to find the time-varying matrix square roots in real time t . In addition, the relationship between the ZNN model and Newton iteration for online solution of static matrix square roots is presented in [21]. Considering Example 1 again, we apply Newton iteration (14) to online solution of the time-varying matrix square root of (9). As illustrated in Figure 12(a), the sequence $\{X_k\}$ generated by Newton iteration (14) with the sampling gap $\tau = 0.01$ could not fit well with the theoretical square root even after a long period of time. Its residual error $\|E_k\|_F$ is depicted in Figure 12(b), which shows that the error of the solution computed by Newton iteration (14) is considerably large. Comparing Figures 2 and 3 with Figure 12, we can see that the ZNN model (4) is more effective and more accurate than Newton iteration (14) for online time-varying matrix square roots finding.

In addition to the above, the detailed comparison between the ZNN model and the four numerical methods (i.e., Newton iteration, DB iteration, CR iteration, and IN iteration) for online solution of static matrix square roots can refer to [21]. Furthermore, the gradient neural network (GNN) model for online solution of time-varying matrix square roots is simulated and compared with the ZNN model in [22]. These results substantiate that the ZNN model (4), which is theoretically/intrinsically designed for online solution of time-varying matrix square roots, is thus more effective than the four numerical methods (i.e., Newton iteration, DB iteration, CR iteration, and IN iteration) and the GNN model for online time-varying matrix square roots finding (or termed, time-varying matrix quadratic factorization).

Superior Performance of ZNN with Hyperbolic Sine Activation Functions

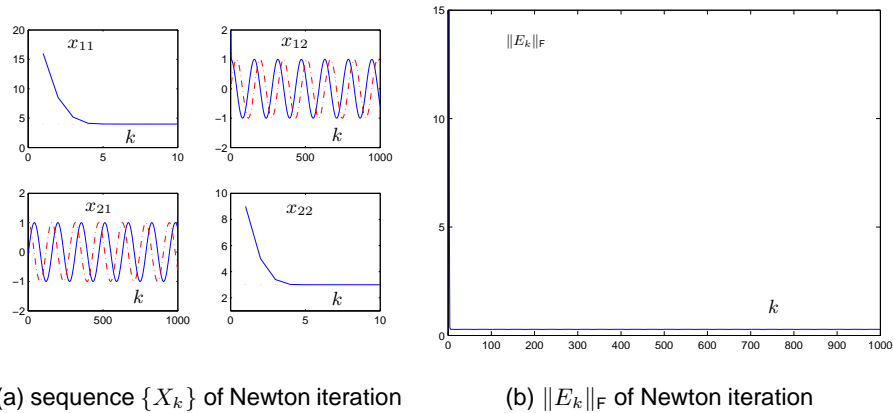


Fig. 12. Convergence performance of Newton iteration (14) for online solution of time-varying matrix square root problem (9).

References

1. Higham, N.J.: Stable iterations for the matrix square root. *Numerical Algorithms* 15(2), 227–242 (1997)
2. Long, J., Hu, X., Zhang, L.: Newton's method with exact line search for the square root of a matrix. *Journal of Physics: Conference Series* 96(1), 1–5 (2008)
3. Hasan, M.A., Rahman, S.: Fixed point iterations for computing square roots and the matrix sign function of complex matrices. In: *Proceedings of the 39th IEEE Conference on Decision and control*. pp. 4253–4258. Sydney, Australia (2000)
4. Johnson, C.R., Okubo, K., Reams, R.: Uniqueness of matrix square roots and an application. *Linear Algebra and Its Applications* 323(1–3), 51–60 (2001)
5. Higham, N.J.: *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, SIAM, England (2008)
6. Higham, N.J.: *Accuracy and Stability of Numerical Algorithms*. Second Edition. Society for Industrial and Applied Mathematics, SIAM, England (2002)
7. Zhang, Y., Leithhead, W.E., Leith, D.J.: Time-series Gaussian process regression based on Toeplitz computation of $O(N^2)$ operations and $O(N)$ -level storage. In: *Proceedings of the 44th IEEE Conference on Decision and Control*. pp. 3711–3716. Seville, Spain (2005)
8. Zhang, Y.: Revisit the analog computer and gradient-based neural system for matrix inversion. In: *Proceedings of the IEEE International Symposium on Intelligent control*. pp. 1411–1416. Limassol, Cyprus (2005)
9. Zhang, Y., Jiang, D., Wang, J.: A recurrent neural network for solving Sylvester equation with time-varying coefficients. *IEEE Transactions on Neural Networks* 13(5), 1053–1063 (2002)
10. Belair, J., Campell, S.A., Driessche, P.V.: Frustration, stability, and delay-induced oscillations in a neural network model. *SIAM Journal on Applied Mathematics* 56(1), 245–255 (1996)

11. Zhang, Y., Ge, S.S.: Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Transactions on Neural Networks* 16(6), 1477–1490 (2005)
12. Ma, W., Zhang, Y., Wang, J.: MATLAB Simulink modeling and simulation of Zhang neural networks for online time-varying Sylvester equation solving. In: *Proceedings of IEEE International Joint Conference on Neural Networks*. pp. 285–289. Hong Kong, China (2008)
13. Wang, J.: Recurrent neural networks for computing pseudoinverses of rank-deficient matrices. *SIAM Journal on Applied Mathematics* 18(5), 1479–1493 (1997)
14. Driessche, V.D., Zou, X.: Global attractivity in delayed Hopfield neural network models. *SIAM Journal on Applied Mathematics* 58(6), 1878–1890 (1998)
15. Mead, C.: *Analog VLSI and Neural Systems*. Addison-Wesley, Reading, Boston, MA, USA (1989)
16. El Emary, I., Emar, W., Aqel, M.J.: The adaptive fuzzy designed PID controller using wavelet network. *Computer Science and Information Systems* 6(2), 141–163 (2009)
17. Zhang, Y., Wang, J.: Global exponential stability of recurrent neural networks for synthesizing linear feedback control systems via pole assignment. *IEEE Transactions on Neural Networks* 13(3), 633–644 (2002)
18. Zhang, Y., Wang, J.: A dual neural network for convex quadratic programming subject to linear equality and inequality constraints. *Physics Letters A* 298(4), 271–278 (2002)
19. B Meini.: The matrix square root from a new functional perspective: theoretical results and computational issues. *SIAM Journal on Matrix Analysis and Applications* 26(2), 362–376 (2004)
20. B Iannazzo.: A note on computing the matrix square root. *Calcolo* 40, 273–283 (2003)
21. Zhang, Y., Yang, Y., Cai, B., Guo, D.: Zhang neural network and its application to Newton iteration for matrix square root estimation. *Neural Computing and Applications* 21(3) 1–8 (2010)
22. Zhang, Y., Yang, Y., Yi, C., Chen, K.: Simulation and comparison of Zhang neural network and gradient neural network solving for time-varying matrix square roots. In: *Proceedings of the 2nd International Symposium on Intelligent Information Technology Application*. pp. 966–970. Shanghai, China (2008)
23. Zhang, Y., Peng, H.: Zhang neural network for linear time-varying equation solving and its robotic application. In: *Proceedings of International conference on Machine Learning and Cybernetics*. pp. 3543–3548 (2007)
24. Zhang, Y., Chen, K., Ma, W.: MATLAB simulation and comparison of Zhang neural network and gradient neural network for online solution of linear time-varying equations. In: *Proceedings of International Conference on Life System Modeling and Simulation*. pp. 450–454. Shanghai, China (2007)
25. Chapman, J.S.: *MATLAB Programming for Engineers*. Thomson Learning, USA (2002)
26. Zhang, Y.: On the LVI-based primal-dual neural network for solving online linear and quadratic programming problems. In: *Proceedings of the American Control Conference*. pp. 1351–1356 (2005)
27. Horn, R.A., Johnson, C.R.: *Topics in Matrix Analysis*. Cambridge University Press, Cambridge (1991)
28. Zhang, Y.: *Dual Neural Networks: Design, Analysis, and Application to Redundant Robotics*. *Progress in Neurocomputing Research*. Nova Science Publishers. pp. 41–81. New York (2007)

Yunong Zhang received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, in 1996, the M.S. degree from South China University of Technology, Guangzhou, China, in 1999, and the Ph.D. degree from Chinese University of Hong Kong, Shatin, Hong Kong, China, in 2003.

He is currently a professor with the School of Information Science and Technology, Sun Yat-sen University (SYSU), Guangzhou, China. Before joining the SYSU in 2006, he had been with the National University of Ireland, Maynooth, Ireland, the University of Strathclyde, Glasgow, U.K., and the National University of Singapore, Singapore, since 2003. His main research interests include neural networks, robotics, computation and optimization. His web-page is now available at <http://sist.sysu.edu.cn/~zhynong>.

Long Jin was born in Lanzhou, China, in 1988. He received the B.S. degree in Automation from Sun Yat-sen University, Guangzhou, China, in 2011. He is currently pursuing the Ph.D. degree in Communication and Information Systems at the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. His main research interests include neural networks, robotics and intelligent information processing.

Zhende Ke received the B.S. degree in Electronic Information Science and Technology and M.S. degree in Communication and Information Systems from Sun Yat-sen University, Guangzhou, China, in 2010 and 2012, respectively. His research interests include neural networks and numerical methods.

Received: January 21, 2012; Accepted: December 1, 2012.

Clustering based Two-Stage Text Classification Requiring Minimal Training Data

Xue Zhang^{1,2} and Wangxin Xiao^{3,4}

¹ Key Laboratory of High Confidence Software Technologies, Ministry of Education,
Peking University, Beijing 100871, China

¹School of Electronics Engineering and Computer Science,
Peking University, Beijing 100871, China

²Department of Physics, Shangqiu Normal University, Shangqiu 476000, China
jane_zhang@pku.edu.cn

³ Department of Computer Science, Jinggangshan University, Ji'an 343009, China

⁴School of Traffic and Transportation Engineering, Changsha University of Science and
Technology, Changsha 410114, China
wx.xiao@rioh.cn

Abstract. Clustering has been employed to expand training data in some semi-supervised learning methods. Clustering based methods are based on the assumption that the learned clusters under the guidance of initial training data can somewhat characterize the underlying distribution of the data set. However, our experiments show that whether such assumption holds is based on both the separability of the considered data set and the size of the training data set. It is often violated on data set of bad separability, especially when the initial training data are too few. In this case, clustering based methods would perform worse. In this paper, we propose a clustering based two-stage text classification approach to address the above problem. In the first stage, labeled and unlabeled data are first clustered with the guidance of the labeled data. Then a self-training style clustering strategy is used to iteratively expand the training data under the guidance of an oracle or expert. At the second stage, discriminative classifiers can subsequently be trained with the expanded labeled data set. Unlike other clustering based methods, the proposed clustering strategy can effectively cope with data of bad separability. Furthermore, our proposed framework converts the challenging problem of sparsely labeled text classification into a supervised one, therefore, supervised classification models, e.g. SVM, can be applied, and techniques proposed for supervised learning can be used to further improve the classification accuracy, such as feature selection, sampling methods and data editing or noise filtering. Our experimental results demonstrated the effectiveness of our proposed approach especially when the size of the training data set is very small.

Keywords: text classification, clustering, active semi-supervised clustering, two-stage classification.

1. Introduction

The goal of automatic text classification is to automatically assign documents to a number of predefined categories. It is of great importance due to the ever-expanding amount of text documents available in digital form in many real-world applications, such as web-page classification and recommendation, email processing and filtering. Text classification has once been considered as a supervised learning task, and a large number of supervised learning algorithms have been developed, such as Support Vector Machines (SVM) [1], Naïve Bayes [2], Nearest Neighbor [3], and Neural Networks [4]. A comparative study was given in [5]. SVM has been recognized as one of the most effective text classification methods. Furthermore, a number of techniques suitable for supervised learning have been proposed to improve classification accuracy, such as feature selection, data editing or noise filtering, and sampling methods against bias.

A supervised classification model often needs a very large number of training data to enable the classifier's good generalization. The classification accuracy of traditional supervised text classification algorithms degrades dramatically with the decrease of the number of training data in each class. As we know, manually labeling the training data for a machine learning algorithm is a tedious and time-consuming process, and even unpractical (e.g., online web-page recommendation). Correspondingly, one important challenge for automatic text classification is how to reduce the number of labeled documents that are required for building reliable text classifier. This leads to an active research problem, semi-supervised learning. There have been proposed a number of semi-supervised text classification methods, including Transductive SVM (TSVM) [6], Co-Training [7] and EM [8]. A comprehensive review could be found in [9]. By exploring information contained in unlabeled data, these methods obtain considerable improvement over supervised methods with relatively small size of training data set. However, most of these methods adopt the iterative approach which train an initial classifier based on the distribution of the labeled data. They still face difficulties when the labeled data set is extremely small since they will have a poor starting point and cumulate more errors in iterations when the extremely few labeled data are far apart from corresponding class centers due to the high dimensionality.

To address the problem of sparsely labeled text classification, we present a clustering based two-stage text classification method with both labeled and unlabeled data. Experimental results on several real-world data sets validate the effectiveness of our proposed approach. Our contributions can be summarized as follows.

- We propose a novel clustering based two-stage classification approach that requires minimal training data to achieve high classification accuracy.
- In order to improve the accuracy of the self-labeled training data by clustering, we propose an active semi-supervised clustering method to cope with data sets of bad separability.
- On the basis of clustering, we convert the challenging problem of sparsely labeled text classification into supervised one. Thus supervised

classification models and techniques suitable for text classification can be used to further improve the overall performance.

—We conduct extensive experiments to validate our approach and study related issues.

The rest of this paper is organized as follows. Section 2 reviews several existing methods. Our clustering method is given in Section 3 with some analysis. The detailed algorithm is then presented in Section 4. Experimental results are presented in Section 5. Section 6 concludes this paper.

2. Related Work

Clustering has been applied in many sub-domains of the problem of text classification, including feature compression or extraction [10], semi-supervised learning [11], and clustering in large-scale classification problems [12,13]. The following will review several related work about clustering aiding classification in the area of semi-supervised learning. A comprehensive review for text classification aided by clustering can be found in [14].

Clustering has been used to extract information from unlabelled data in order to boost the classification task. There are roughly four cases of semi-supervised classification aided by clustering. In particular, clustering is used: (1) to create a training set from the unlabelled set [15], (2) to augment an existing labeled set with new documents from the unlabeled data set [11], (3) to augment the data set with new features [8,16], and (4) to co-train a classifier [17,18]. More recently, simultaneous learning frameworks for clustering and classification have been proposed [19,20].

To make use of unlabeled data, one assumption which is made, explicitly or implicitly, by most of the semi-supervised learning algorithms is the so-called cluster assumption that two points are likely to have the same class label if there is a path connecting them passing through regions of high density only. That is, the decision boundary should lie in regions of low density. Based on the ideas of spectral clustering and random walks, a framework for constructing kernels which implement the cluster assumption was proposed in [21]. Also based on cluster assumption, [22] applied spectral clustering to represent the labeled and unlabeled data. By clustering unlabeled data with labeled data using probabilistic and fuzzy approaches, [23] proposed a framework to improve the performance of base classifier with unlabeled data. In text classification, there are often many low-density areas between positive and negative labeled examples because of the high dimensionality and data sparseness. This situation will be worsened with the decrease of the number of training data in each class.

The most related work is the clustering based text classification (CBC) approach [11]. In CBC, firstly, semi-supervised soft k-means is used to cluster the labeled and unlabeled data into k clusters, where k is set to the number of classes in the classification task. $p\%$ most confident unlabeled examples from each cluster (i.e. the ones nearest to the cluster's centroid) are added to the

training data set. Then TSVM is trained on the augmented training data set and unlabeled data set. Similarly, $p\%$ most confident unlabeled examples from each class (i.e. the ones with the largest margin) are added to the training data set. CBC iterates the step of clustering and the step of classification alternatively until there is no unlabeled data left. In CBC, in order to guarantee the labeling accuracy, the value of p should be small enough. That is, after the clustering step in each iteration, the training data set is augmented with very few examples. Therefore, the classifier in the following classification step should have an accepted performance with small size of training data set. This put a strong constraint on the selected classification models. CBC can hardly perform well with supervised classification models, e.g. SVM, which will be demonstrated later.

The success of CBC is based on the assumption that even when some of the data points are wrongly classified, the most confident data points, i.e. the ones with largest margin under classification model and the ones nearest to the centroids under clustering model, are confidently classified or clustered. This assumption guarantees the high accuracy of the self-labeled training data and correspondingly the good performance of the algorithm. We separate this assumption into clustering assumption and classification assumption for convenience. However, our empirical experiments show that the assumptions are often violated on data sets of bad separability. Firstly, clustering assumption can't be hold in this case, at least for the soft-constraint k-means [11]. In fact, each cluster's centroid may locate in: 1) the domain of its corresponding true class, 2) the border of its true class and other classes, 3) the domain of other class. The probability that the last two cases occur increases with the degrading of data separability. In the last two cases (we call them cluster bias), CBC will introduce more noise into the training data set in its clustering step, which might make the classification assumption also be violated since the noise will have a big effect due to the initially very few truly labeled training data. Then the following iterative steps will further cumulate more errors.

In sparsely labeled text classification, the extremely few training data make many techniques which are useful for ameliorating data separability, e.g. feature selection, not effective, because the training data can not characterize the whole data set well. When the size of training data set is extremely small, unsupervised learning gives better performance than supervised and semi-supervised learning algorithms. In this paper, we develop an active semi-supervised clustering based two-stage approach to address the problem of sparsely labeled text classification. Different from CBC, our aim is to convert the problem of sparsely labeled text classification into a supervised one by using clustering. Therefore, supervised classification models, e.g. SVM, can be applied, and techniques proposed for supervised learning can be used to further improve the classification accuracy, such as feature selection, sampling methods and data editing or noise filtering. Furthermore, our proposed active semi-supervised clustering method aims to cope with data sets with any separability. The goal of clustering here is to generate enough training data for supervised learning with high accuracy.

3. Active Semi-supervised Clustering

Using clustering to aid semi-supervised classification, the key point lies in that the clustering results can to some extent characterize the underlying distribution of the whole data set. Only in this case, clustering is helpful to augment training data set or extract useful features to improve the performance of classification. Although clustering methods are more robust to the bias caused by the initial sparsely labeled data, empirical experiences show that the results of clustering might also be biased (e.g. the cluster bias of cases 2) and 3)), sometimes heavily, especially on data sets of bad separability. The soft-constrained k-means in CBC can reduce bias in the labeled examples by basing the constraints (the guidance of the initially labeled data) not on exact examples but on their centroid. But it still cannot cope with the bias well in training data on data sets with bad separability.

Table 1 gives two clustering based algorithms to augment training data. They implement the iterative reinforcement strategy. In each iteration, a clustering method is used to cluster the whole data set with the guidance of labeled training data, and then several examples are selected according to some criteria and labeled with the labels of the centroids they belong to. In SemiCC algorithm, the clustering method is soft-constrained k-means adopted in CBC algorithm. The clustering method (we call it as active soft-constrained k-means) in SemiCCAc algorithm is proposed in order to address the cluster-bias problem.

Table 1. Two Clustering Algorithms: *SemiCC* and *SemiCCAc*

<p>Input: Labeled data set D_l and unlabeled data set D_u, the number of iterations $maxIter$, p</p> <p>Output: Augmented labeled data set D_l'</p> <p>Initialize: $D_l' = D_l$, $D_u' = D_u$, $iter = 0$</p> <p>Algorithm <i>SemiCC</i>: While $iter < maxIter$ and $D_u \neq \Phi$</p> <ul style="list-style-type: none"> ● $iter = iter + 1$ ● Calculate initial centroids: $o_i = \frac{1}{n_i} \sum_{\forall j, t_j = i} x_j, i = 1, \dots, c, x_j \in D_l'$ and set current centroids $o_i^* = o_i$. n_i is the number of examples in D_l' whose label is i. ● The labels of the centroids $t(o_i) = t(o_i^*)$ are equal to labels of the corresponding examples.
--

- Repeat until cluster result doesn't change any more
 - Assign $t(o_i^*)$ to each $x_i \in D_l + D_u$ that are nearer to o_i^* than to other centroids.
 - Update current centroids:

$$o_i^* = \frac{1}{n_i} \sum_{\forall j, t_j=i} x_j, i=1, \dots, c, x_j \in D_l + D_u, n_i \text{ is the}$$
 number of examples in $D_l + D_u$ whose label is i .
 - Calculate the nearest centroids o_j^* for each o_i , if $t(o_i) \neq t(o_i^*)$, exit the loop.
- From each cluster, select $p\%$ examples $x_i \in D_u'$ which is nearest to o_i^* , add them to D_l' , and delete them from D_u' .

Algorithm SemiCCAc:

While $iter < maxIter$ and $D_u \neq \Phi$

- $iter = iter + 1$
- Calculate initial centroids:

$$o_i = \frac{1}{n_i} \sum_{\forall j, t_j=i} x_j, i=1, \dots, c, x_j \in D_l', \text{ and set current}$$
 centroids $o_i^* = o_i$. n_i is the number of examples in D_l' whose label is i .
- The labels of the centroids $t(o_i) = t(o_i^*)$ are equal to labels of the corresponding examples.
- Repeat until cluster result doesn't change any more
 - Assign $t(o_i^*)$ to each $x_i \in D_l + D_u$ that are nearer to o_i^* than to other centroids.
 - Update current centroids:

$$o_i^* = \frac{1}{n_i} \sum_{\forall j, t_j=i} x_j, i=1, \dots, c, x_j \in D_l + D_u, n_i \text{ is the}$$
 number of examples in $D_l + D_u$ whose label is i .
 - Calculate the nearest centroids o_j^* for each o_i , if $t(o_i) \neq t(o_i^*)$, exit the loop.
- From each cluster, select $p\%$ examples $x_i \in D_u'$ which is nearest to o_i^* and sort them with descending order of confidences, x_1, \dots, x_m

```

■ If the true label of  $x_1$  and  $x_m$  equals to  $t(o_i^*)$ :
    add the  $m$  examples to  $D'_i$ 
    delete them from  $D'_u$ 
else
    add  $x_1$  and  $x_m$  with their true labels to  $D'_i$ 
    delete  $x_1$  and  $x_m$  from  $D'_u$ 
end

```

SemiCCAc is different from SemiCC in the labeling strategy for the selected examples of highest confidences according to the clustering results. In soft-constrained k-means, it doesn't take the found centroids' location into consideration. It just labels the selected examples nearest to each centroid. Therefore, it will introduce much noise into the training data set with the presence of cluster bias. In active soft-constrained k-means, it first estimates the location of each centroid. Only for clusters whose centroids locate within their true classes, it labels all the selected examples nearest to the corresponding centroids with the labels of their centroids. For the clusters with the presence of cluster bias, it just labels two examples with their true label for each cluster.

An important problem in active soft-constrained k-means is how to estimate the location of each cluster's centroid. The strategy used here is to inquire the true labels of two examples (the nearest and the farthest examples to the centroid in the selected $p\%$ examples) by resorting to an oracle or expert for each cluster. If the two examples have the same label with that of their centroid, then all the $p\%$ selected examples are labeled with the label of the centroid. Otherwise, only the two examples are added to training data set with their true labels. The strategy is based on the intuition that cluster bias is more likely happened when the two examples have different labels with that of their centroid. When the two examples have the same label, but different from their centroid, the cluster's centroid is most likely locate in the domain of other classes. When one of the two examples has the same label with that of the centroid, the cluster's centroid is most likely locate in the border of the true class and other classes. We can filter out much noise by using this strategy. It is also delightful that cluster bias can be rectified in the following iterations in SemiCCAc by estimating the location of clusters' centroids, which property guarantees the high accuracy of self-labeled training data in spite of the poor starting points.

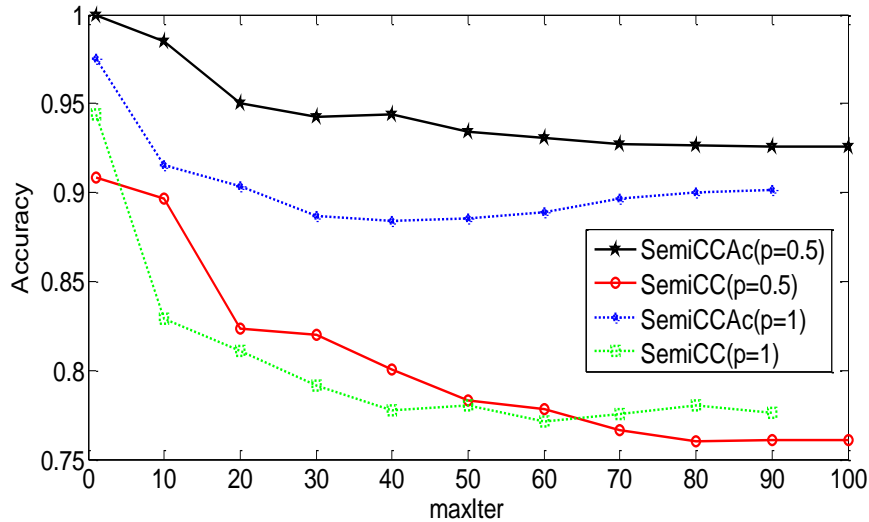


Fig.1. Accuracy of self-labeled training data with iterations

In figure 1, we depict the average accuracy of self-labeled training data by applying the two clustering algorithms to a text classification problem in 20 runs (same2, consisting of two most similar classes in 20Newsgroups, 5 training data for each class). From figure 1, it could be found that the accuracy of self-labeled training data by SemiCCAc is significantly higher than that by SemiCC. With the increase of p value, the accuracy degrades first, but then it rises with the increase of iterations.

When $maxIter=1$, *SemiCC* degrades to the soft-constrained k -means. We can also see that the average accuracy in *SemiCC* is below 0.95 when $maxIter=1$, which indicates that soft-constrained k -means introduces noise into the training data set with a certain probability. This noise will hurt the following classifier's learning, especially when the size of the initial training data set is very small. We think the phenomenon of cluster bias can partially explain why the performance of CBC improves slowly than those of TSVM and co-training with the increase of training data. With the increase of iterations, the accuracy of self-labeled training data by *SemiCC* degrades much faster than that of *SemiCCAc*. Therefore, techniques to cope with the cluster bias are very important for clustering based semi-supervised classification. This also tells us that the proposed active semi-supervised clustering method is effective for addressing the problem of cluster bias.

4. Two-Stage Classification Framework: ACTC

In this section, we present the detail of the Active semi-supervised Clustering based Two-stage text Classification algorithm (ACTC). All documents are tokenized into terms and we construct one component for each distinct term. Thus each document is represented by a vector $(w_{i1}, w_{i2}, \dots, w_{ip})$ where w_{ij} is weighted by TFIDF. The cosine function is used in the clustering algorithm to calculate the distance from an example to the centroid. In the classification stage, we use a SVM classifier trained with the augmented training data to classify the whole data set. The detailed algorithm is presented in table 2.

ACTC consists of two stages: clustering stage and classification stage. In the clustering stage, SemiCCAc is used to augment the training data set. Users can set the values of *maxIter* and *p* to determine how many new documents should be labeled by SemiCCAc. At the second stage, discriminative classifiers can subsequently be trained with the expanded labeled data set. Soft-constrained k-means is in fact a generative classifier [11]. According to [24], generative classifiers reach their asymptotic performance faster than discriminative classifiers, but usually lead to higher asymptotic error than discriminative classifiers. This motivates us to combine clustering with discriminative classifiers together to address the problem of sparsely labeled text classification.

ACTC in fact converts the problem of sparsely labeled text classification into a supervised one, thus supervised classification models suitable for text classification can be used. Moreover, the techniques proposed for supervised learning can be used to improve the performance. For instance, it is unavoidable to falsely label some examples in the clustering stage, then data editing or noise filtering techniques are expected to improve the performance of ACTC. Other techniques also can be used to improve the performance, such as feature selection and sampling.

Table 2. ACTC and CBCSVM

<p>Input: Labeled data set D_l and unlabeled data set D_u and the number of iterations <i>maxIter</i>, <i>p</i></p> <p>Output: The full labeled set $D_l' = D_l + D_u$ A classifier L</p> <p>Algorithm ACTC:</p> <ol style="list-style-type: none"> 1. Clustering Stage Use SemiCCAc (repeat <i>maxIter</i> iterations) to augment the training data set and we get an augmented training data set D_l' 2. Classification Stage

Train a SVM classifier L based on D'_i . And use the learned classifier to classify the whole data set.

Algorithm CBCSVM:

$iter=0$

1. while $iter < maxIter/2$

$iter=iter+1$

 1.1 Clustering step

 Use soft-constrained k-means to clustering the whole data set, and select $p\%$ unlabeled examples nearest to its centroid for each cluster and add them to D'_i

 1.2 Classification step

 Train a SVM classifier based on D'_i .

 From each class, select $p\%$ unlabeled examples with the largest margin, and add them to D'_i

2. Train a SVM classifier L based on D'_i . And use the learned classifier to classify the whole data set.

In order to verify the two-stage framework performs better than CBC algorithm in supervised learning, we substitute SVM for TSVM in CBC, named CBCSVM. Note that, if we use TSVM as the classifier, the performance of both algorithms will be expected to get improved. However, the time complexity of a TSVM classifier is much higher than that of a SVM classifier, because it repeatedly switches estimated labels of unlabeled data and tries to find the maximal margin hyperplane. The more unlabeled data are, the more time it requires. The worse of the data separability is, the more time it requires. For example, on same2, which consists of two most similar classes of 20Newsgroups and 1000 examples in each class, TSVM requires several hours to complete when 5 training data for each class are used. SVM only needs about 1 second. With enough training data, the performance of SVM is expected to be similar with that of TSVM, but it requires much less time. This motivates us to propose the two-stage classification method, which converts the problem of sparsely labeled text classification into a supervised one.

CBCSVM is also given in table 2 for convenience. Since CBC selects $p\%$ unlabeled examples both in clustering and classification steps in each iteration, we set the number of iterations to half of that in ACTC in order to make them have the same selection times.

The difference between our approach and CBC is that, we expand the training data set by a self-training style clustering process and resorting to an oracle or expert to evaluate the clusters' centroids. After completion of the training data expansion, discriminative classifiers could be trained on the expanded training data set. Therefore, ACTC puts less constraint on the

classification model, which enables us to treat the following classification stage as a supervised learning problem.

5. Performance Evaluation

5.1. Data sets

For a consistent evaluation, we conduct our empirical experiments on two benchmark data sets, 20NewsGroups and Reuters-21578. 20Newsgroups is one famous Web-related data collection. From the original 20 Newsgroups data set, same2, consisting of 2 very similar newsgroups (comp.windows.x, comp.os.ms-windows) is used to evaluate the performance of the algorithms. Same2 contains 2000 instances, 1000 for each class. We use Rainbow software¹ to preprocess the data (removing stop words and words whose document frequency are less than 3, stemming) and we get 7765 unique terms for same2. Then terms are weighted with their TFIDF values.

The Reuters-21578 corpus contains Reuters news articles from 1987. We only show the experimental results of train1.svm in LWE² since the algorithms have the similar performance on other Reuters data sets. Train1.svm contains 1239 documents (two class) and 6889 unique terms.

5.2. Evaluation Metric

We use macro-averaging of F1 measure among all classes to evaluate the classification result.

For each class $i \in [1, c]$, let A_i be the number of documents whose real label is i , and B_i the number of documents whose label is predicted to be i , and C_i the number of correctly predicted documents in this class. The precision and recall of the class i are defined as $P_i = C_i / B_i$ and $R_i = C_i / A_i$ respectively.

For each class, the $F1$ metric is defined as $F_1 = 2 \cdot P \cdot R / (P + R)$ where P and R are precision and recall for a particular class. $F1$ metric takes into account both precision and recall, thus it is a more comprehensive metric than either precision or recall when separately considered.

The macro-averaging $F1$ is a measurement which evaluates the overall performance of the classification model. It is defined as:

$$Macro_F1 = \frac{1}{c} \sum_{i=1}^c 2 \times P_i \times R_i / (P_i + R_i) \quad (1)$$

¹ <http://www.cs.cmu.edu/~mccallum/bow/>

² <http://ews.uiuc.edu/~jinggao3/kdd08transfer>

5.3. Experimental Results

The SVM^{light} package³ is used in our experiments for the implementation of SVM using default configurations.

We first compare ACTC and CBCSVM with different iterations on two data sets. SVM and SemiCCAc are used as the baseline in order to see the benefits brought by our two-stage classification framework and CBCSVM. We set $p=0.5$. We conduct the experiments 30 runs and the average results are given. The number of training data is 5 for each class and randomly sampled in each run.

Figures 2 and 3 give the Macro_ *F1* performance with different iterations on same2 and Reuters respectively. In ACTC and CBCSVM, parameter *maxIter* determines the number of self-labeled training data. Larger value of *maxIter* means more self-labeled training data and larger size of the training data set for the final SVM training. That is, the size of training data set for the final SVM increases with the increase of *maxIter*.

From figure 2, we can see that ACTC significantly outperforms the other algorithms with any value of *maxIter* and its performance improves with the increase of the *maxIter*. This indicates the following two aspects. One is that SVM classifier significantly benefits from the augmented training data set by comparing its performance with that of SVM trained on the initial training data set. The other is that the self-labeled training data are of high accuracy so that the benefit from the self-labeled training data exceeds the negative effect of the noise contained in the self-labeled training data. This accords with that shown in figure 1 in section 3. The performance of CBCSVM degrades slightly with the increase of *maxIter*. Because soft-constrained k-means cannot cope with cluster bias well, it introduces more noise into the self-labeled training data which further put negative effect on the SVM training. Such noise cumulates in the following iterations, which make the final SVM perform worse than that in ACTC. SemiCCAc outperforms SVM, which accords with the former conclusion that unsupervised learning gives better performance than supervised learning when the size of training data set is extremely small.

³ <http://svmlight.joachims.org/>

Clustering based Two-Stage Text Classification Requiring Minimal Training Data

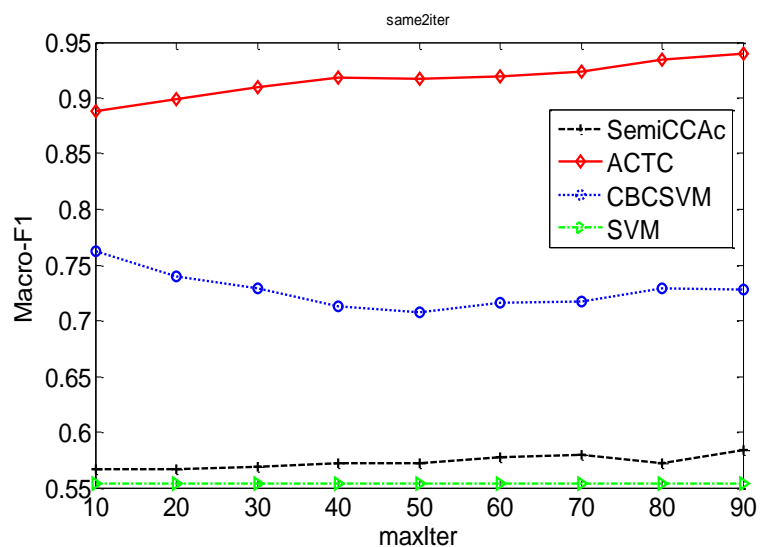


Fig.2. Performance with *maxIter* on same2

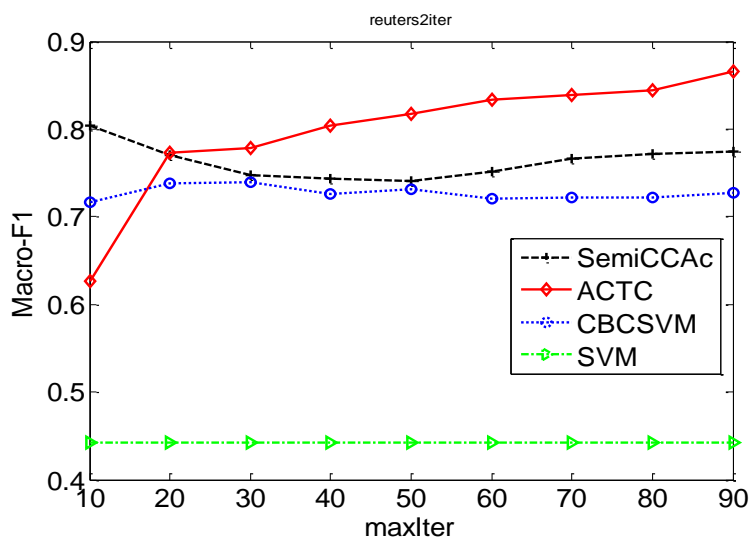


Fig.3. Performance with *maxIter* on Reuters

From figure 3, we can see that ACTC outperforms the other algorithms when *maxIter* > 20. This may lie in the fact that the self-labeled training data are unbalanced for each class in SemiCCAc, and that SemiCCAc may filter out useful examples when it copes with the cluster bias, which have relatively larger effect on the final SVM performance when the size of training data set is small. This is expected to be improved by exploring sampling technique on training data set, e.g. over-sampling. On Reuters data set, SemiCCAc

outperforms CBCSVM slightly and significantly outperforms SVM. This indicates that clustering gives better performance than that of SVM when the initial training data set is small.

To evaluate the performance of ACTC with a large range of labeled data, we run the algorithm together with CBCSVM, SVM and SemiCCAc on different percentage of the labeled data on the above two data sets. Figures 4 and 5 give the results. We set $p=0.5$ and $maxIter=60$. We conduct the experiments 30 runs and the average results are given. Training data are randomly sampled in each run.

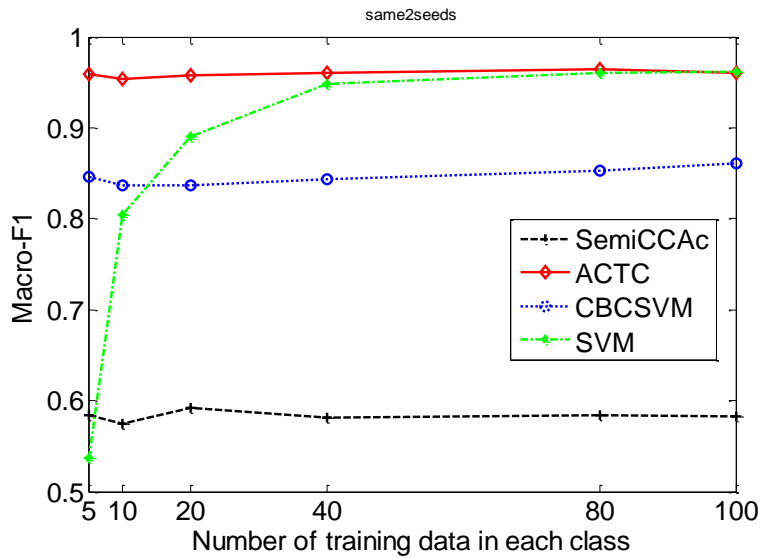


Fig.4. Performance with number of training data on same2

ACTC performs best on the two data sets with all size of training data set. SVM performs worst when the size of training data is 5 for each class. Then its performance improves fast with the increase of training data. SVM outperforms CBCSVM and SemiCCAc when the size of training data set is larger than 20 on same2 and larger than 10 on Reuters. With the increase of training data, the performance of ACTC, CBCSVM, and SemiCCAc grows very slowly. For ACTC and CBCSVM, the reason may be due to the effect of noise contained in the self-labeled training data. Therefore data editing or noise filtering techniques may be helpful to improve the performance. After noise filtering, feature selection and sampling may also be helpful to improve the overall performance. ACTC always significantly outperforms CBCSVM and SemiCCAc, which indicates that our two-stage classification framework is superior to that of CBC, and that the combination of generative model with discriminative model can overcome the shortcomings of both models.

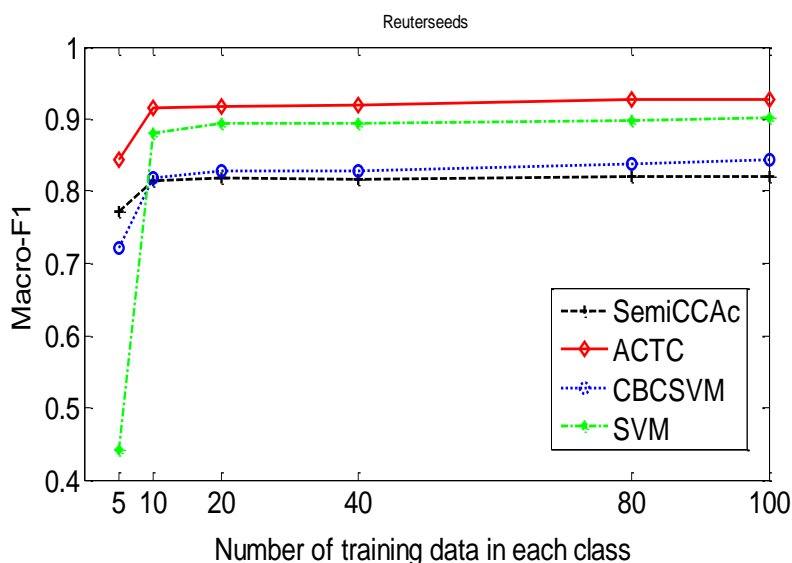


Fig.5. Performance with size of training data set on Reuters

In ACTC and CBCSVM, the parameter p determines the number of self-labeled examples in each selection process. Larger value of p indicates more examples are self-labeled in each selection, so fewer iterations are needed when the number of self-labeling examples are fixed. But more noise may be introduced into the training data set (please refer to figure 1).

6. Conclusion

This paper presents an active semi-supervised clustering based two-stage classification framework for sparsely labeled text classification. In order to address the cluster bias problem, an active semi-supervised clustering method is proposed. We use a self-training style clustering method to augment the training data set, so that we can convert the challenging problem of sparsely labeled text classification into a supervised one. Therefore supervised classification models can be used, e.g. SVM, and useful techniques for supervised learning can be employed to further improve the performance. The experiments show the superior performance of our method over SVM and CBC (SVM as base learner).

In the future, we plan to evaluate other clustering methods to address the cluster bias problem, e.g. affinity propagation clustering and density based clustering. In terms of noise control, data editing or noise filtering techniques will also be explored. Other directions include investigating the problems of example selection, confidence assessment, and resampling techniques.

Acknowledgment. The authors would like to thank the anonymous reviewers for their useful advice. This work is partially supported by the National Natural Science Foundation of China (No.61127005, No.61170054, No.50708085, and No.50978127), the special scientific research funding of Research Institute of Highway, Ministry of Transport (No.1206030211003), and the Project of Education Department of Jiangxi Province (No.GJJ08415).

References

1. Joachims, T.: Text categorization with support vector machines: Learning with Many Relevant Features. In Proceedings of the European Conference on Machine Learning. Chemnitz, Germany, April 21 – 24, 137--142 (1998).
2. Lewis, D. D.: Naïve Bayes at forty: The independence assumption in information retrieval. In Proceedings of the European Conference on Machine Learning. Chemnitz, Germany, April 21 – 24 (1998).
3. Masand, B., Linoff, G., Waltz, D.: Classifying news stories using memory based reasoning. In Proceedings of the 15th International ACM/SIGIR Conference on Research & Development in Information Retrieval. Copenhagen, Denmark, June 21 - 24, 59-64 (1992).
4. Ng, T. H., Goh, W. B., Low, K. L.: Feature selection, perception learning and a usability case study for text categorization. In Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Philadelphia, PA, USA, July 27-31, 1997.
5. Yang, Y. & Liu, X.: An re-examination of text categorization. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Berkeley, CA, USA, August 15-19, 1999.
6. Joachims, T.: Transductive inference for text classification using support vector machines. In Proceedings of the 16th international conference on machine learning (ICML1999). Bled, Slovenia, June 27-30, 200-209 (1999).
7. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with Co-Training. In Proceedings of the 11th Annual Conference on Computational Learning Theory. Madison, Wisconsin, July 24-26, 92-100 (1998).
8. Nigam, K., McCallum, A. K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103-134, 2000.
9. Seeger, M.: Learning with labeled and unlabeled data. Technical report, Edinburgh University, 2001.
10. Slonim, N., Tishby, N.: Document Clustering using Word Clusters via the Information Bottleneck Method. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Athens, Greece, July 24-28, 208--215 (2000).
11. Zeng, H. J., Wang, X. H., Chen, Z., Ma, W. Y.: CBC: Clustering based text classification requiring minimal labeled data. In Proceedings of the 3rd IEEE International Conference on Data Mining. Melbourne, Florida, USA, November 19 – 22, 2003.
12. Yu, H., Yang, J., Han, J.: Classifying large data sets using SVMs with hierarchical clusters. in Proceedings of the 9th ACM SIGKDD 2003, Washington, DC, USA, 2003.
13. Evans, R., Pfahringer, B., Holmes, G.: Clustering and Classification. 7th International conference on information technology in Asia (CITA 11). Sarawak, Malaysia, July 12-13, 1-8 (2011).

14. Kyriakopoulou, A.: (2008). Text classification aided by clustering: a literature review. *Tools in Artificial Intelligence*, 233-252, 2008.
15. Fung, G., Mangasarian, O.L.: (2001). Semi-supervised support vector machines for unlabeled data classification. *Optim. Methods Software*, 2001, v15 i1. 29-44.
16. A. Kyriakopoulou, T. Kalamboukis. (2008). Combining clustering with classification for spam detection in social bookmarking systems. in *Proceedings of ECML/PKDD Discovery Challenge 2008 (RSDC 2008)*, Antwerp, Belgium, 2008, pp. 47–54.
17. Kyriakopoulou, A.: Using Clustering and Co-Training to Boost Classification Performance. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence*. Patras, Greece, October 29-31, 325-330 (2007) .
18. Raskutti, B., Ferrá, H., Kowalczyk, A.: (2002). Combining clustering and co-training to enhance text classification using unlabelled data. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*. Edmonton, Alberta, Canada, July 23 - 26, 2002.
19. Cai, W., Chen, S., Zhang, D.: A multiobjective simultaneous learning framework for clustering and classification. *IEEE Transactions on Neural Networks*, 21(2): 185–200, 2010.
20. Qian, Q., Chen, S., Cai, W.: Simultaneous clustering and classification over cluster structure representation. *Pattern Recognition*, 2011, October 27.
21. Chapelle, O., Weston, J., Scholkopf, B.: Cluster kernels for semi-supervised learning. *Advances in Neural Information Processing Systems In NIPS 2002*, Vol. 15 (2003), 585-592.
22. Zhou, D., Bousquet, O., Lal, T. N., Weston, J., Scholkopf, B.: Learning with local and global consistency. *Advances in Neural Information Processing Systems 16*, 321-328, 2004.
23. Keswani, G., Hall, L.O.: Text classification with enhanced semi-supervised fuzzy clustering. *Handbook of Fuzzy Computation*, 1994, 511-515.
24. Ng, A. Y., Jordan, M. I.: On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. *Advances in Neural Information Processing Systems 14*, 2002.

Xue Zhang, received the BS degree in electronic engineering from XiDian University, Xian, China, in 1999. She received the MS degree in control theory and control engineering from Southwest University of Science and Technology, Mianyang, China, in 2003, and received the PhD degree in computer science from Southeast University, Nanjing, China, in 2007. From 2008 to the present, she is a postdoctoral fellow in Peking University. Her research interests include data mining and machine learning, with emphasis on the applications to text mining and bioinformatics.

Wangxin Xiao, received the PhD degree in traffic information and control engineering from Southeast University, Nanjing, China, in 2004. From 2005 to 2007, he engaged in postdoctoral research in Wuhan University of Technology. Since 2008 he has been an associate professor in Research Institute of Highway Ministry of Transport. From 2009 to 2011, he was also a postdoctoral fellow in Changsha University of Science and Technology. His research interests include pattern recognition, Intelligent Transport Systems (ITS) and data mining with applications to traffic data.

Received: January 30, 2012; Accepted: December 05, 2012.

A Novel Content Based Image Retrieval System using K-means/KNN with Feature Extraction

Ray-I Chang¹, Shu-Yu Lin^{1,2}, Jan-Ming Ho², Chi-Wen Fann², and Yu-Chun Wang²

¹ Dept. of Engineering Science and Ocean Engineering
National Taiwan University
Taipei, Taiwan (R.O.C)
{rayichang, d96525009}@ntu.edu.tw

² Research Center for Information Technology Innovation
Academia Sinica
Taipei, Taiwan (R.O.C)
{boymike, hoho, fann, zxaustin}@iis.sinica.edu.tw

Abstract. Image retrieval has been popular for several years. There are different system designs for content based image retrieval (CBIR) system. This paper propose a novel system architecture for CBIR system which combines techniques include content-based image and color analysis, as well as data mining techniques. To our best knowledge, this is the first time to propose segmentation and grid module, feature extraction module, K-means and k-nearest neighbor clustering algorithms and bring in the neighborhood module to build the CBIR system. Concept of neighborhood color analysis module which also recognizes the side of every grids of image is first contributed in this paper. The results show the CBIR systems performs well in the training and it also indicates there contains many interested issue to be optimized in the query stage of image retrieval.

Keywords: content based images retrieval; K-means clustering; feature extraction; image retrieval

1. Introduction

Image retrieval is the processing of searching and retrieving images from a huge dataset. As the images grow complex and diverse, retrieval the right images becomes a difficult challenge. For centuries, most of the images retrieval is text-based which means searching is based on those keyword and text generated by human's creation.[1] The text-based image retrieval systems only concern about the text described by humans, instead of looking into the content of images. Images become a mere replica of what human has seen since birth, and this limits the images retrieval. This may leads to many drawbacks which will be state in related works.

To overcome those drawbacks of text-based image retrieval, content-based images retrieval (CBIR) was introduced [2][3]. With extracting the images features, CBIR perform well than other methods in searching, browsing and content mining etc. The need to extract useful information from the raw data becomes important and widely discussed. Furthermore, clustering technique is usually introduced into CBIR to perform well and easy retrieval. Although many research improve and discuss about those issues, still many difficulties hasn't been solved. The rapid growing images information and complex diversity has build up the bottle neck.

To overcome this dilemma, in this paper, we propose a novel CBIR system with an optimized solution combined to K-means and k-nearest neighbor algorithm (KNN). A creative system flow model, image division and neighborhood color topology, is introduced and designed to increase the clustering accuracy. The rest of this paper is organized as follows. Section 2 briefly describes the concepts of images retrieval, feature extracting, K-means and the structure of CBIR system. In Section 3, we provide a description of the Optimized Content Based Image Retrieval model with K-means and KNN combing with those module proposed in this paper. Experimental results and discussions are presented in Section 4. Finally, Sections 5 discusses our conclusion and future works.

2. Related Work

This chapter introduces some important literatures review in this chapter. First come to content based image retrieval (CBIR). Before CBIR, the traditional image retrieval is usually based on text. Text based image retrieval has been discussed over those years. The text-based retrieval is easy to find out some disadvantages such as:

1. Manually annotation is always involved by human's feeling, situation, etc. which directly results in what is in the images and what is it about.
2. Annotation is never complete.
3. Language and culture difference always cause problems, the same image is usually text out by many different ways.
4. Mistakes such as spelling error or spell difference leads to totally different results.

In order to overcome these drawbacks, content based images retrieval (CBIR) was first introduced by Kato in 1992. The term, CBIR, is widely used for retrieving desired images from a large collection, which is based on extracting the features (such as color, texture and shapes) from images themselves. Content-Based Image Retrieval concerns about the visual properties of image objects rather than textual annotation. And the most popular and directly features is the color feature, which is also applied in this paper. In this work, color is selected as a primary feature in images clustering.

2.1. Data Clustering

Data Clustering is often taken as a step for speeding-up image retrieval and improving accuracy especially in large database. In general, data clustering algorithms can be divided into two types: Hierarchical Clustering Algorithms and Non-hierarchical Clustering Algorithms. However, Hierarchical Clustering is not suitable for clustering large quantities of data. Non-hierarchical is suggested to cluster large quantities of data. The most adapted method for non-hierarchical clustering is the K-Means clustering algorithm proposed by James MacQueen in 1967.

K-means clustering algorithm first defined the size of K clusters. Based on the features extracted from the images themselves, K-means allocates those into the nearest cluster. The algorithm calculates and allocates until there is little variation in the movement of feature points in each cluster. This paper applies k-means clustering algorithm for color clustering module based on this concept.

2.2. Feature Extraction

In the image retrieval, feature extraction is the most important issue of the first step. The features extracted from the images directly lead to the results. Some preprocessing is needed to avoid retrieval noise. Steps such as removing the background, highlight the objects. All the preprocessing steps help in feature extraction [18]. Different topic images usually contain different features [19]. Deciding the features needed to be extracted is always a popular issue, and then it still comes back to basic features of the images such as:

- Color histograms are basic and fundamental feature of the images. It is the most commonly used and usually gets obvious effect result. Many literatures apply color histograms as basic images comparison [20] [21].
- Tamura Features. Tamura proposed six image features based on human visualization [22], coarseness, contrast, directionality, line-likeness, regularity, and roughness. Those are usually considered in many related research [3] [2].
- Shape Feature. Besides color, shape is the most commonly used feature. Some image retrieval applications require the shape representation to be invariant to rotation, translation, and scaling. Shape representation can be divided into two categories, boundary-based and region-based [3].
- MPEG-7 Features. The Moving Picture Experts Group (MPEG) defines some description of visual named MPEG-7. It includes Color Layout, Color Structure, Dominant Color, Scalable Color, Edge Histogram, Homogeneous Texture, Texture Browsing, Region-based Shape, Contour-based Shape, Camera Motion, Parametric Motion and Motion Activity features [23].

2.3. Contrast Context Histogram

Contrast Context Histogram (CCH) is first proposed for image descriptor [15]. CCH first uses histograms to measure the differences in intensity between various salient points around an object.

Through comparing the positive and negative histograms of the same object, it derives efficient and discriminative descriptors. The CCH method effectively resists image variability between photos of the same object. The CCH extract features in our system as follow:

First, we assume that each photo has already been analyzed to derive several salient corners. For each center of a salient corner P_c , in the $N \times N$ pixel region R , we calculate the center-based contrast $C(p)$ of point p in the region R as follows:

$$C(p) = I(p) - I(p_c) \quad (1)$$

Then, for each central point p in every region R_i , the positive contrast histogram and negative contrast histogram as shown in (2) and (3) respectively:

$$H_{R_i+}(p_c) = \frac{\sum \{C(p) \mid p \in R_i \text{ and } C(p) \geq 0\}}{\#_{R_i+}} \quad (2)$$

$$H_{R_i-}(p_c) = \frac{\sum \{C(p) \mid p \in R_i \text{ and } C(p) < 0\}}{\#_{R_i-}} \quad (3)$$

To combine the values of the positive and negative contrast histograms of all regions in R_i , the CCH descriptor of salient corners is defined as:

$$CCH(p_c) = (H_{R_{i+}}, H_{R_{i-}}, H_{R_{j+}}, H_{R_{j-}}, \dots, H_{R_{i+}}, H_{R_{i-}}) \quad (4)$$

3. Proposed Method

In our CBIR system, it is divided into two parts: learning and querying. The learning step tells about the training process which a huge mount sample images are input in the first step, then the images' features are extracted for the clustering. K-means algorithm is selected to cluster the training data because of it is easy to implement, efficient and well developed in the recent 50 years. Finally the training output the clustering result as a learning code book. The query part describes the images searching process. Inputting the query images and matches to the training result. The output shows the most similar images for user's query. Figure 1 shows the overview of the CBIR system.

The main system architecture contains four modules both in learning stage and query stage: Segmentation and grid module, the K-Means clustering module, the feature extraction module and the neighborhood concept module. The flow chart of the system architecture is shown in Figure 2. Each module is described as follow.

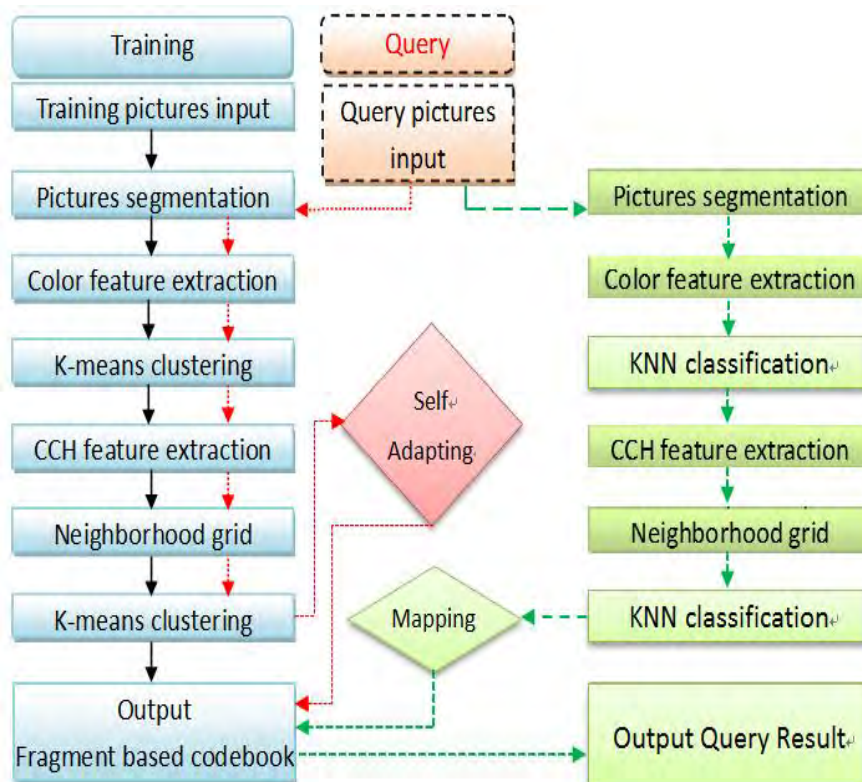


Fig. 1. Overview of the CBIR system

3.1. Image segmentation and grid module

The data images input into the system will be first processed in this module. In the images retrieval, larger images usually decrease the retrieval accuracy. Small images grids help in feature extraction and images processing. Therefore, this module first divides the images into $F \times F$ grid and every grid will be divided again into $S \times S$ sub-grids while during the feature extraction module. In this stage, an input image will finally be divided into $(F \times S)^2$ grids which the $F \times F$ grids are used both in the feature extraction module and the neighborhood module. The $S \times S$ grids are only used in the color feature extraction. Figure 3 shows the sample of image segmentation.

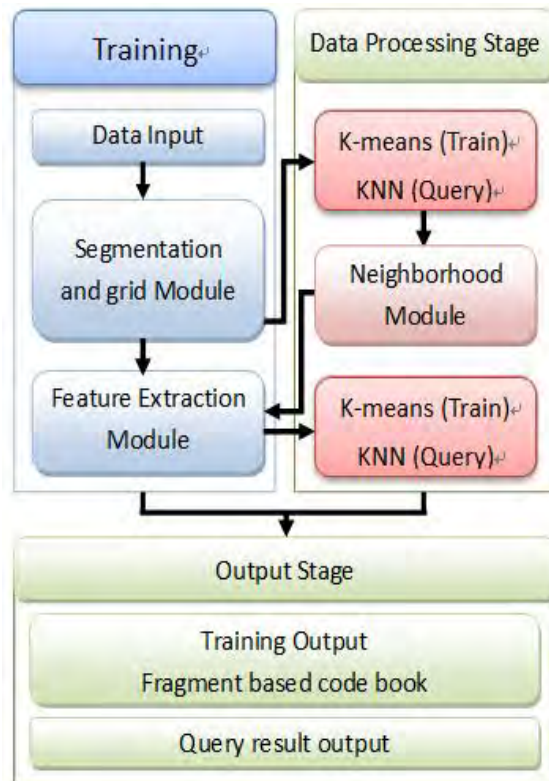


Fig. 2. Flow chart of the system architecture

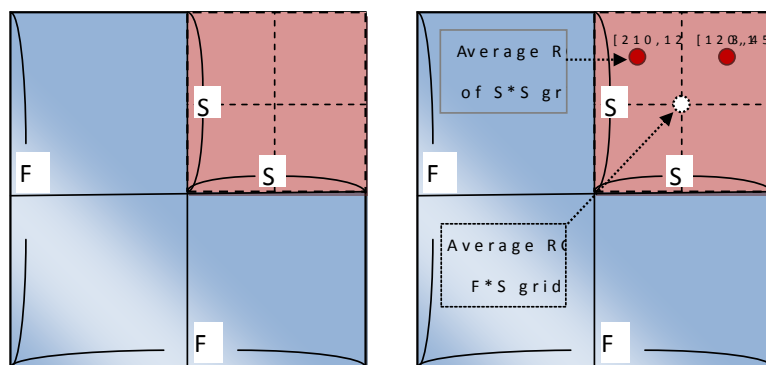


Fig. 3. Sample of Segmentation and grid module and Color feature extraction

3.2. Feature Extraction Module

The input images, including the training and query stage, are all processed in this module. It is also the most important in image retrieval. Since color is the most popular and intuitive feature based on human visualization, it is applied in the system. In order to get more powerful features, the CCH method is also applied for extracting the important feature point. The two feature extractions are described as below:

1. Color Feature Extraction

Input images will be divided into $F \times S$ grids before this stage. All grids are input to extract the color feature. First, the module compute the average RGB value of the $F \times F$ grids. Second, the inside $S \times S$ grids in every $F \times S$ grids will also be input to calculate the average RGB value. The $S \times S$ grids' detail RGB information is append after the $F \times S$ grids' color feature information. All those are prepared for first K-means clustering. Figure 3 illustrates the color feature extractions of this stage.

2. CCH feature extraction

The system utilizes CCH to find out the important feature points. All the points are detected for preparing the input data of the neighborhood module and K-means clustering or KNN classfying. The information of CCH feature points, including the 64 dimensions data, combines with the neighborhood module result. Taking it as the input for the second round K-means clustering, the K-means clustering results in a fragment-based database, call the Code book. As the same implementation in query step, K-means is replaced by KNN algorithm. Query data inputed will be classified to improve the training code book, also correct classified result helps for quicly retrieval. Figure 5 shows the sample for CCH feature extraction.



Fig. 4. Data presentation and processing of the neighborhood module

3.3. The neighborhood module

In this module, the input data is from the CCH feature points. Feature points of every image will consider as an index to build up the neighborhood table. Also the first K-means clustering result of every F*F grids are imported to denote the value of the neighborhood table. The steps of every detail are described as follow:

1. Input the CCH feature point Y of the X picture, represented as PICXY.
2. Get the first K-means clustering result based on the CCH feature point's coordinate.
3. Get the neighborhoods' first K-means clustering results.
4. Appending the results from step3 according with the order left to right then top to bottom. If there is no neighborhood, then the value will be "0", which stands for the side of the pictures.
5. Appending the CCH information into the neighborhood table.
6. K-means clustering based on the neighborhood table to generate the code book.

Figure 5 shows the data presentation and processing of the neighborhood module. Taking the same format, the query step apply KNN algorithm instead of K-means.

3.4. The K-means/KNN module

K-means clustering is applied twice in our system. The K-means clustering helps generate the code book. In order to keep those input (include the training and query stage) being clustered in the same standard, our architecture keeps the cluster central points in the training stage. The central points are imported into the K-means clustering in the query stage. Finally, the code book can be mapped in the same comparing standard.

K-nearest neighbor algorithm (KNN) is also involved into our CBIR system. Based on the training result, KNN is applied for the query data images. KNN helps to classify the input data; also it fixes the code book which means the training result can be self-adapted.

3.5. The image retrieval query module

This paper provides a query method based on modules mentioned before. Considering the grid fragment, color feature, and CCH feature points, all images input for query will be divided into pieces. Then KNN is applied to classify those images which maps to the training result (code book). The query grid images are compared with those picture grids in the same cluster, the system then calculates the difference based on the color feature which is introduced in 3.2. All fragments are tagged and linked with one grid in the

code book. By calculate the most amount of the grids, the CBIR finally output the query and retrieval result.

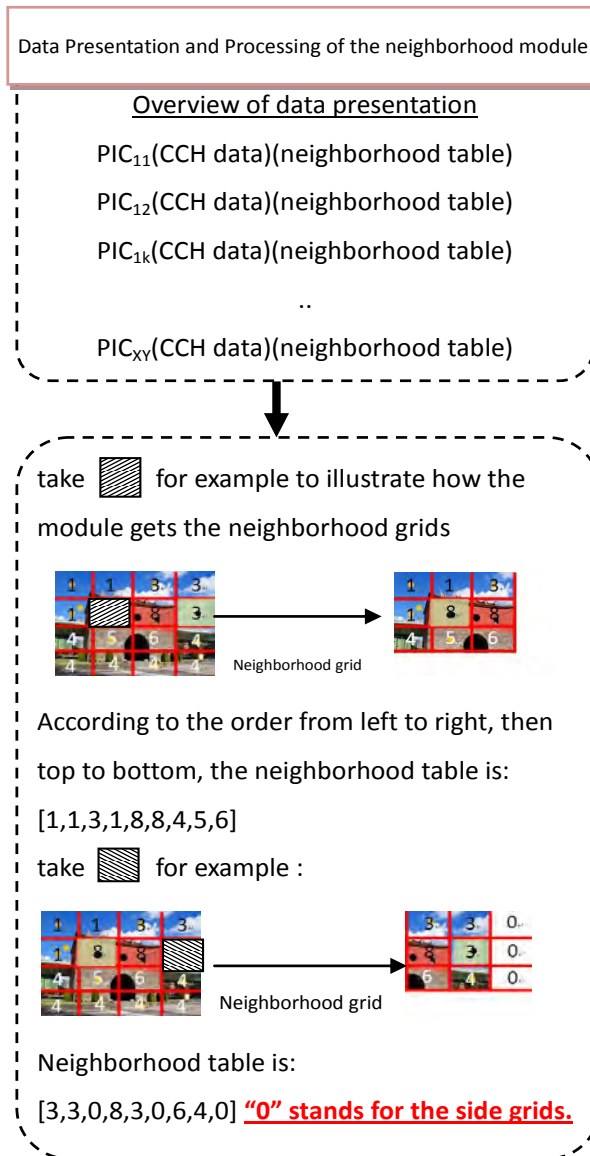


Fig. 5. Data presentation and processing of the neighborhood module

4. Implementation and Experimental Results

Based on the method provided in this paper, the experiments are designed to verify the architecture of the CBIR system. Also the experiments shows the modules proposed in this article perform good and well organized with the CBIR system architecture.

First the CBIR system splits and calculates the average RGB, we have implemented and figure 6 shows samples of the color feature and split result.



Fig. 6. Samples of the color feature result.

The color features extracted from X pictures divided into Y fragments is denoted as table 1 and pictures training clustered result is visualized as shown in figure 7 and figure 8.

The results show that the same color features of fragments are clustered together. Then the neighborhood module, CCH feature points are combined to be clustered to generate the training result, called the code book. Figure 9 and figure 10 show the visualization of the training results. The features extracted through CCH has been grouped together based on the color features.

In the code book, the color features and the CCH features are included. Based on the fragments, image can be retrieval in detail and the clustering helps decrease the computing cost.

Table 1. Representation of color feature

Data Representation			
Picture Number	Fragment Number	Average RGB	Fragment RGB
1	1	[123,213,21]	[12,23,23]...[13,43,234]
1	2	[23,3,23]	[1,43,123]...[5,10,87]
...
X	Y	[8,132,24]	[99,1,24]...[89,255,1]

A Novel Content Based Image Retrieval System using K-means/KNN with Feature Extraction



Fig. 7. Visualization of Cluster NO.2 derived by K-Means Color Clustering.

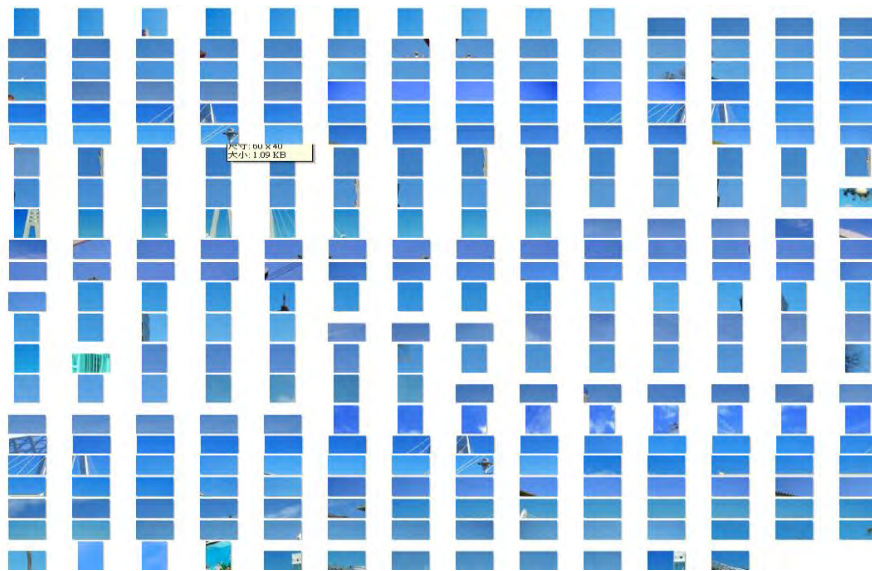


Fig. 8. Cluster NO.6 derived by K-Means Color Clustering.

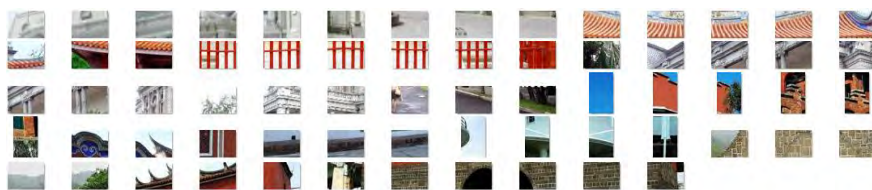


Fig. 9. Cluster NO.15 derived by K-Means in the code book.












Fig. 10. Cluster NO.13 derived by K-Means in the code book.

In the code book, the color features and the CCH features are included. Based on the fragments, image can be retrieval in detail and the clustering helps decrease the computing cost. This paper has verified several settings for proposed CBIR system architecture. The results indicates the system perform well for training and querying.
















In order to verify the CBIR system, the Wang’s dataset which includes 386*254 pixel images is applied as the testing and training data [25]. The 1,000 images from the Wang’s dataset are applied into the codebook. 50 images are randomly selected as query images. With the CCH feature points, the CBIR system proposed in this paper successfully retrievals the correct images for those query images. Furthermore, landscaped images are selected from 10,000 images as the training and querying images [25]. The results indicates the CBIR system proposed retrievals correctly when it is well trained. Table 2 shows the query sample of the Wang’s dataset.







Table 2. Retrieval results for the Wang’s dataset.

Images Input	Retrieval Results	Image with CCH points
		
		
		

Several images are input for images retrieval. To query for similar images, the system operates as section 3 described. Table 3 shows the image retrieval results. The CBIR system proposed in this paper successfully find out the pictures which are included in the code book. For those which are not in the code book, the similar pictures are also figured out as table 3 shown. Also the retrieval processes which are generated by the grid module is shown in table 3, the grid puzzle images. The experiment indicates that the CBIR system retrieves the easy-to-tell similar image, even though the image inputted for query is never trained in the codebook.

Table 3. Image retrieval results

Images Input	Retrieval Results	Grid puzzle images
Pictures included in the Code book		
		
		
		
Pictures not included in the Code book		
		
		

Images Input	Retrieval Results	Grid puzzle images
Pictures included in the Code book		
		
		

5. Conclusion and Future Work

The proposed system is designed to operate the content based image retrieval system. It has been verified with the photos of places of interest in Taiwan and the Wang's dataset [24][25]. Our experimental results demonstrate that our CBIR system architecture not only works well for image retrieval, but also improves its precision.

In our knowledge, this paper first combines segmentation and grid module, feature extraction module, K-means clustering and neighborhood module to build the CBIR system. Furthermore, the concept of neighborhood module which recognizes the side of every grids of image is first contributed in this paper. Applying the concept of fragment based code book into the content based image retrieval system also contributes in our system architecture. The experimental results confirm that the proposed CBIR system architecture attains better solution for image retrieval. Our model represents the first time in which combine new modules and techniques proposed in the paper have been integrated with CBIR system.

Images can be retrieval correctly through the proposed CBIR system. For those images which are contained in the code book, all of them can be searched as the most similar result. Also for general images selected randomly, the query results are similar to the input data. Since the CBIR system is based on the color feature, the retrieval results are directly and easy to tell the performances. In the future work, we hope to build a generalized query method which increase the system searching ability and provide more accurate content descriptions of places of interest places by performing color feature analysis and CCH image extraction simultaneously. As a result, the CBIR system will be able to suggest more relevant annotations and descriptions.

Furthermore, we hope to optimize the system architecture and modules proposed in this paper. There exists some detail setting can be discussed and optimized with the images retrieval issues.

Acknowledgment. This research was supported in part by Taiwan E-learning and Digital Archives Programs (TELDAP) sponsored by the National Science Council of Taiwan under NSC Grants: NSC101-2631-H-001-005, NSC101-2631-H-001-006.

References

1. Kato, T.: Database architecture for content-based image retrieval. *Image Storage and Retrieval Systems*, 112–123. (1999)
2. Datta, R., Joshi, D., Li, J., Wang, J. Z.: Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys*, Vol. 40, No. 2, Article 5, April. (2008)
3. Rui, Y., Huang, T.S., Chang, S. F.: Image Retrieval: Current Techniques, Promising Directions and Open Issues. *Journal of Visual Communication and Image Representation, Transaction on Systems, Man, and Cybernetics*, vol. 8, 460–472. (1999)
4. Rui, Y., She, A. C., Huang, T. S.: Modified Fourier Descriptors for Shape Representation - A Practical Approach. *Proc. of First International Workshop on Image Databases and Multi Media Search*. (1996)
5. Gupta, A., Jain, R.: Visual Information Retrieval. *Communications of the ACM*, vol. 40, 70–79. (1997)
6. Gross, M. H., Koch, R., Lippert, L.: A Dreger, Multiscale image texture analysis in wavelet spaces. *Proc. IEEE Int. Conf. on Image Proc.* (1994)
7. Chua, T. S., Tan, K.-L., Ooi, B. C.: Fast signature-based color-spatial image retrieval. *Proc. IEEE Conf. on Multimedia Computing and Systems*. (1997)
8. Chuang, G. C.-H., Kuo, C. -C. J.: Wavelet descriptor of planar curves: Theory and applications. *IEEE Trans. Image Proc.*, vol. 5, 56–70. (1996)
9. Faloutsos, C., Flickner, M., Niblack, W., Petkovic, D., Equitz, W., Barber, R.: Efficient and Effective Querying by Image Content. (1993)
10. Rui, Y., Huang, T.S., Mehrotra, S.: Relevance feedback techniques in interactive content-based image retrieval. In *Storage and Retrieval for Image and Video Databases VI*, 25–36. (1996)
11. Stricker, M., Orengo, M.: Similarity of color images. *Proc. SPIE Storage and Retrieval for Image and Video Databases*. (1995)
12. Smith, J. R., Chang, S.-F.: Automated binary texture feature sets for image retrieval. *Proc. IEEE Int. Conf. Acoust, Speech, and Signal Proc*, May. (1996)
13. Mirmehdi, M., Palmer, P. L., Josef K.: Optimising the Complete Image Feature Extraction Chain. *Proceedings of the Third Asian Conference on Computer Vision*, Vol. 2, 307–314, January. (1998)
14. Kwak, N., Choi, C.-H., Choi, C.-Y.: Feature extraction using ICA. *Proceedings of the International Conference on Artificial Neural Networks*, 568–573. (2001)
15. Huang, C. R., Chen, C. S., Chung, P. C.: Contrast context histogram - A discriminating local descriptor for image matching. *IEEE International Conference on Pattern Recognition*, 53–56. (2006)
16. Yu, H. H., Wolf, W.: Hierarchical, multi-resolution algorithms for dictionary-driven content-based image retrieval. *Proc. IEEE Int. Conf. on Image Proc.* (1997)
17. Ono, A., Amano, M., Hakaridani, M.: A flexible content-based image retrieval system with combined scene description keyword. *Proc. IEEE Conf. on Multimedia Computing and Systems*. (1996)

Ray-I Chang et al.

18. Prasad, B. G., Biswas, K. K., Gupta, S. K.: Region-based image retrieval using integrated color, shape, and location index. *Comput. Vis. Image Underst.* Vol. 94, 193–233. (2004)
19. Deslaers, T., Keysers, D., Ney, H.: Features for image retrieval: an experimental comparison. *Information Retrieval*, vol. 11, 77–107. (2008)
20. Puzicha, J., Rubner, Y., Tomasi, C., Buhmann, J.: Empirical Evaluation of Dissimilarity Measures for Color and Texture. *Proc. Int'l Conf. Computer Vision.* (1999)
21. 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*, 1349–1380. (2000)
22. Tamura, H., Mori, S., Yamawaki, T.: Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 460–473. (1978)
23. Eidenberger, H.: How good are the visual MPEG-7 features?. In *Proceedings SPIE Visual Communications and Image Processing Conference*, Vol. 5150, 476–488. (2003)
24. Jia, L., Wang, J. Z.: Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, 1075–1088. (2003)
25. Wang, J. Z., Jia, L., Gio, W.: SIMPLiCity: Semantics-sensitive Integrated Matching for Picture Libraries. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol 23, 947–963. (2001)

Ray-I Chang joined the Department of Engineering Science, National Taiwan University, in 2003. He has published over 150 original papers, including papers published in journals such as *IEEE Transactions on Multimedia*, *IEEE Transactions on Broadcasting*, and *IEEE Transactions Neural Networks*. His current research interests include multimedia networking and data mining. Dr. Chang is a member of IEEE.

Shu-Yu Lin is a Ph.D candidate in Department of Engineering Science, National Taiwan University, Taiwan, R.O.C. His research interests include artificial intelligence, machine learning, data mining and multimedia.

Jan-Ming Ho received the B.S. degree in Electrical Engineering from National Cheng Kung University, Taiwan, in 1978 and the M.S. degree from Institute of Electronics at National Chiao Tung University, Taiwan, in 1980. He received the Ph.D. degree in Electrical Engineering and Computer Science from Northwestern University, USA, in 1989. He joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan, as an associate research fellow in 1989 and was promoted to research fellow in 1994. He had served as Associate Editor of *IEEE Transaction on Multimedia*. He was Program Chair of Symposium on Real-time Media Systems, Taipei, 1994 - 1998, General Co-Chair of International Symposium on Multi-Technology Information

A Novel Content Based Image Retrieval System using K-means/KNN with Feature
Extraction

Processing, 1997 and will be General Co-Chair of IEEE RTAS 2001. He was also steering committee member of VLSI Design/CAD Symposium, and program committee member of several previous conferences including ICDCS 1999, and IEEE Workshop on Dependable and Real-Time E-Commerce Systems (DARE'98), etc.

Chi-Wen Fann is a research assistance in Institute of Information Science, Academia Sinica. He was an well-experienced project manager. His research interests includes multimemedia system design and data mining.

Yu-Chun Wang is a research assistance in Institute of Information Science, Academia Sinica. His research includes cloud computing, data mining and machine learning.

Received: January 22, 2012; Accepted: December 04, 2012.

Study On Prediction Models For Integrated Scheduling In Semiconductor Manufacturing Lines

Lu Guo, Jinghua Hao, and Min Liu*

Department of Automation, Tsinghua University,
Beijing , China,100084
guo-l08@mails.tsinghua.edu.cn, {haojinghua, lium}@tsinghua.edu.cn

Abstract. Quality prediction of lot operations is significant for integrated scheduling in semiconductor production line. The model-training algorithm needs to be fast and incremental to satisfy the online applications where data comes one by one or chunk by chunk. This paper presents a novel prediction model referred to as Incremental Extreme Least Square Support Vector Machine (IELSSVM), which transforms the data into ELM feature space and then minimizes the structural risk like LSSVM. The transformation into ELM feature space can be regarded as a good dimensionality reduction. The incremental formula is proposed for on-line industrial application to avoid retraining when data comes one by one or chunk by chunk. Detailed comparisons of the IELSSVM algorithm with other incremental algorithms are achieved by simulation on benchmark problems and real overlay prediction problem of lithography in semiconductor production line. The results show that IELSSVM has better performance than other incremental algorithms like OS-ELM.

Keywords: quality prediction model; ELM feature space; semiconductor production line; IELSSVM; overlay prediction; integrated scheduling

1. Introduction

With the advancement of technology and more fierce competition in the markets, scheduling has played an increasingly important role in maximizing the manufacturing capability and reducing the product makespan. Thus more and more enterprises and researchers have paid more attention to it [1]. Although many interesting research results have been attained, there are still a lot of difficulties in actual application because the lot state may have changed when a scheduling policy is carried out. For example, the lots may be held, reworked or prohibited from being processed by some machines because of quality problems. If the developed scheduling strategy does not

* To whom correspondence should be addressed.

consider these quality factors, the unexpected lot state change will severely deteriorate the scheduling performance. Integration of scheduling and quality prediction can effectively deal with the difficulties. The scheduling strategy decided on the basis of the accurate quality prediction will keep consistent with the actual environment, thus enhancing the validness of scheduling policy.

From above, we can see that quality prediction is very important in semiconductor manufacturing. Accurate prediction model is the key problem. There are three typical features for the datum attained from actual semiconductor production line. The first is high dimensional. There are hundreds of steps in the semiconductor line and hundreds of check points to measure some parameters of the lot. All these measure data is combined into one high dimensional sample point in the space. The second feature is that the data measured in actual production line is contaminated by stochastic noises. It is inevitable because of the limited precision of measuring equipments and the complexity of production environment. The third feature is originated from running time pressure in actual applications where data may come one by one or chunk by chunk. Retraining based on all the data when new data comes is very time consuming and can not meet the demand of online applications.

In this paper, a novel Incremental Extreme Least Square Support Vector Machine (IELSSVM) is presented to meet the demand of actual application and overcome the deficiencies of other intelligent modeling methods. The datum is at first transformed into ELM feature space and then the structural risk minimization principal is adopted as the objective to be optimized like LSSVM. The transformation of ELM can be regarded as a good dimensionality reduction. The incremental formula is proposed for online industrial applications to avoid retraining when data comes one by one or chunk by chunk, thus improving the training speed and efficiency.

2. Related Works

Over the past decades, batch learning algorithms have been discussed and investigated thoroughly. Many intelligent modeling methods have been proposed and implemented, such as Back-Propagation Neural Network (BP-NN)[2], Support Vector Machine (SVM) [3], Least Square Support Vector Machine (LSSVM) [4], Proximal Support Vector Machine (PSVM) [5], Extreme Support Vector Machine (ESVM) [6], Extreme Learning Machine (ELM) [7], and etc. BP-NN and ELM may be over-fitting because of the modeling criteria of Empirical Risk Minimization principal. SVM and LSSVM are constructed on the basis of Structural Risk Minimization principal, thus avoiding over-fitting and showing great prospects and having become very popular in recent years. But training of SVM may take a long time because of a quadratic programming problem to be solved. For ESVM, there is a time consuming matrix inversion calculation process, which restricts its online

application. In [8], Huang et al. developed a constrained optimization based Extreme Learning Machine as a unified learning framework for LS-SVM, PSVM and other regularization algorithms. The algorithm can perform much better than other algorithms and showing great prospects.

Originated from the batch learning methods that have been developed, many online sequential algorithms are presented to meet the actual application demand. In [9], a generalized growing and pruning RBF (GGAP-RBF) is presented. The authors first introduce the significance of the neurons and develop an online sequential algorithm by generalized growing and pruning methods. In [10], an online sequential extreme learning machine (OS-ELM) is introduced which is much faster and produces better generalization performance compared with other sequential learning algorithms, such as GGAP-RBF. Based on batching learning algorithm of SVM, an incremental support vector machine is presented in [11]. The basic idea is that the new SVM is built based on the new arrival data and the trained support vectors. The method is called SV-incremental algorithm in [12]. Fixed-size LSSVM (FS-LSSVM) is developed in [13] for large scale regression problems. Based on quadratic Renyi entropy criteria, an active support vector selection method is developed. Compared with LSSVM, FS-LSSVM needs much less support vectors and produces better performance.

GGAP-RBF and OS-ELM may be over-fitting because of the modeling basis of Empirical Risk Minimization principal. SV-incremental algorithm and FS-LSSVM are constructed on the basis of Structural Risk Minimization principal, thus avoiding over-fitting and having been very popular in recent years. However in the SV-incremental algorithm, the old support vectors may have only a little influence on the new SVM [12] and the performance can't be satisfied. Also, the computation cost of FS-LSSVM is very high because of the eigenvectors and eigenvalues computation.

3. Problem Description

ELM feature space is a special space into which data is mapped by a special single-hidden-layer feed-forward networks (SLFNs) with randomly chosen parameters for activation functions. It has been proved that with sufficient nodes, given any infinite differentiable activation functions, such as sigmoid function, RBF function and etc., the ELM network can approximate any continuous target functions [7,8,10]. In this paper, we take the RBF activation function for example, which has the form like $G(a_i, b_i, x_i) = e^{-b_i(\|x_i - a_i\|)}$. The parameters of $\{a_i, b_i\}_{i=1}^L$ are selected randomly and L is the number of hidden nodes.

Given the input training data set $X = \{x_i\}_{i=1}^N$ and the corresponding output training data set $Y = \{y_i\}_{i=1}^N$, where N is the total number of training data, all

the data is mapped into the ELM feature space by the mapping matrix $\Phi(X)$ with the form:

$$\Phi(X) = \begin{bmatrix} G(a_1, b_1, x_1) & G(a_2, b_2, x_1) & \cdots & G(a_L, b_L, x_1) \\ G(a_1, b_1, x_2) & G(a_2, b_2, x_2) & \cdots & G(a_L, b_L, x_2) \\ \cdots & \cdots & \cdots & \cdots \\ G(a_1, b_1, x_N) & G(a_2, b_2, x_N) & \cdots & G(a_L, b_L, x_N) \end{bmatrix} \quad (1)$$

According to the structural risk minimization principal, the risk minimization in ELM feature space is constructed as follows:

$$\min_{w, \xi, b} E = \frac{1}{2} \|w\|^2 + \frac{1}{2} \nu \|\xi\|^2 \quad (2)$$

$$s.t. \quad D[\Phi(X)w + Eb] = E - \xi \quad (3)$$

Where w 、 b are the weight vector and the constant offset respectively, and D is the diagonal matrix with plus one or minus one along its diagonal, which indicates the class of each sample point. Clearly, D^2 equals I . E is the vector with all elements are one and dimension is $N \times 1$. Parameter ν can be seen as the cost coefficient for misclassification. The specific formation of D and E is given as

$$D = \begin{bmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & y_N \end{bmatrix} \quad (4)$$

$$E = \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \end{bmatrix}_{N \times 1} \quad (5)$$

So, the classifier is to determine the parameters of w 、 b when training data set $\{X, Y\}$ is given.

4. IELSSVM Algorithm

First, we define the Lagrange relaxation function as follows

$$L(w, \xi, b, s) = \frac{1}{2} \|w\|^2 + \frac{1}{2} \nu \|\xi\|^2 - s^T \{D[\Phi(X)w + Eb] - E - \xi\} \quad (6)$$

where s is the Lagrange multipliers with $N \times 1$ dimensions, and s^T is the transpose of s . The optimized solution of (4) is the saddle point of

$L(w, \xi, b, s)$ according to KKT optimality conditions. So, we get the following equations

$$\frac{\partial L(w, \xi, b, s)}{\partial w} = w - \Phi(X)^T Ds = 0 \quad (7)$$

$$\frac{\partial L(w, \xi, b, s)}{\partial b} = s^T DE = 0 \quad (8)$$

$$\frac{\partial L(w, \xi, b, s)}{\partial \xi} = v\xi - s = 0 \quad (9)$$

$$D[\Phi(X)w + Eb] - E + \xi = 0 \quad (10)$$

Rewrite the above equations as

$$\begin{cases} w - \Phi(X)^T Ds = 0 \\ s^T DE = 0 \\ v\xi - s = 0 \\ D[\Phi(X)w + Eb] - E + \xi = 0 \end{cases} \quad (11)$$

The main parameters to be solved are w 、 b , so we substitute w 、 b for the parameters ξ 、 s . The result of substitution is given as

$$\begin{cases} v\Phi(X)^T \Phi(X)w + w + v\Phi(X)^T Eb = v\Phi(X)^T DE \\ E^T \Phi(X)w + E^T Eb = E^T DE \end{cases} \quad (12)$$

So we get the matrix expression of parameters w 、 b as

$$\begin{bmatrix} \frac{1}{v}I + \Phi(X)^T \Phi(X) & \Phi(X)^T E \\ E^T \Phi(X) & E^T E \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} \Phi(X)^T DE \\ E^T DE \end{bmatrix} \quad (13)$$

Clearly, parameters of $\begin{bmatrix} w \\ b \end{bmatrix}$ can be attained by a direct matrix inverse which is time consuming.

$$\begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} \frac{1}{v}I + \Phi(X)^T \Phi(X) & \Phi(X)^T E \\ E^T \Phi(X) & E^T E \end{bmatrix}^{-1} \begin{bmatrix} \Phi(X)^T DE \\ E^T DE \end{bmatrix} \quad (14)$$

In actual application, data may come chunk by chunk or one by one. When one new data comes, retraining all the data by a direct matrix inverse is time consuming and can not meet the requirements of online use. So we need to construct an incremental algorithm for online use in actual applications.

Suppose at time t , the datum that has been learned is denoted as $\{X_t, Y_t\}$, where $X_t = \{x_i\}_{i=1}^N$ is the input sample data set, and $Y_t = \{y_i\}_{i=1}^N$ is the output sample data set. At time $t + 1$, new data set $\{X_{IC}, Y_{IC}\}$ is received, where $X_{IC} = \{x_i\}_{i=N+1}^{N+k}$ and $Y_{IC} = \{y_i\}_{i=N+1}^{N+k}$. The ELM mapping matrix of the last received data set is given by

$$H(X_{IC}) = \begin{bmatrix} G(a_1, b_1, x_{N+1}) & G(a_2, b_2, x_{N+1}) & \cdots & G(a_L, b_L, x_{N+1}) \\ G(a_1, b_1, x_{N+2}) & G(a_2, b_2, x_{N+2}) & \cdots & G(a_L, b_L, x_{N+2}) \\ \cdots & \cdots & \cdots & \cdots \\ G(a_1, b_1, x_{N+k}) & G(a_2, b_2, x_{N+k}) & \cdots & G(a_L, b_L, x_{N+k}) \end{bmatrix} \quad (15)$$

So, the ELM mapping matrix of all the data that has been received up to time $t + 1$ can be written as

$$\Phi(X_{t+1}) = \begin{bmatrix} \Phi(X_t) \\ H(X_{IC}) \end{bmatrix} \quad (16)$$

For time $t + 1$, we have the following equations

$$\begin{bmatrix} \frac{1}{v} I + \Phi(X_{t+1})^T \Phi(X_{t+1}) & \Phi(X_{t+1})^T E_{t+1} \\ E_{t+1}^T \Phi(X_{t+1}) & E_{t+1}^T E_{t+1} \end{bmatrix} \begin{bmatrix} w_{t+1} \\ b_{t+1} \end{bmatrix} = \begin{bmatrix} \Phi(X_{t+1})^T D_{t+1} E_{t+1} \\ E_{t+1}^T D_{t+1} E_{t+1} \end{bmatrix} \quad (17)$$

For simplicity, we denote A_t, O_t, β_t as follows:

$$A_t = \begin{bmatrix} \frac{1}{v} I + \Phi(X_t)^T \Phi(X_t) & \Phi(X_t)^T E_t \\ E_t^T \Phi(X_t) & E_t^T E_t \end{bmatrix} \quad (18)$$

$$O_t = \begin{bmatrix} \Phi(X_t)^T D_t E_t \\ E_t^T D_t E_t \end{bmatrix} \quad (19)$$

$$\beta_t = \begin{bmatrix} w_t \\ b_t \end{bmatrix} \quad (20)$$

According to the relationship between time t and $t + 1$, $E_{t+1}, D_{t+1}, A_{t+1}, O_{t+1}$ can be written as

$$E_{t+1} = \begin{bmatrix} E_t \\ E_{k \times 1} \end{bmatrix} \quad (21)$$

$$D_{t+1} = \begin{bmatrix} D_t & 0 \\ 0 & D_{k \times k} \end{bmatrix} \quad (22)$$

$$A_{t+1} = A_t + \begin{bmatrix} H_{IC} \\ E_{k \times 1} \end{bmatrix} \begin{bmatrix} H_{IC} & E_{k \times 1} \end{bmatrix} \quad (23)$$

$$O_{t+1} = O_t + \begin{bmatrix} H_{IC} \\ E_{k \times 1} \end{bmatrix} \begin{bmatrix} y_{N+1} \\ y_{N+2} \\ \dots \\ y_{N+k} \end{bmatrix} \quad (24)$$

Also, we denote B_t and Y_{t+1} as

$$B_t = \begin{bmatrix} H_{IC} & E_{k \times 1} \end{bmatrix} \quad (25)$$

$$Y_{t+1} = \begin{bmatrix} y_{N+1} \\ y_{N+2} \\ \dots \\ y_{N+k} \end{bmatrix} \quad (26)$$

We have a simple expression of A_{t+1} and O_{t+1} as follows

$$A_{t+1} = A_t + B_t^T B_t \quad (27)$$

$$O_{t+1} = O_t + B_t^T y_{t+1} \quad (28)$$

The solution of β_{t+1} is written as

$$\beta_{t+1} = A_{t+1}^{-1} O_{t+1} = (A_t + B_t^T B_t)^{-1} (O_t + B_t^T Y_{t+1}) \quad (29)$$

The Sherman-Morrison-Woodbury (SMW) formula [14] can be used to deal with the matrix inverse problem

$$(A_t + B_t^T B_t)^{-1} = A_t^{-1} - A_t^{-1} B_t^T (B_t A_t^{-1} B_t^T + I_{k \times k})^{-1} B_t A_t^{-1} \quad (30)$$

So, we rewrite the formula of β_{t+1} as

$$\beta_{t+1} = (A_t^{-1} - A_t^{-1} B_t^T (B_t A_t^{-1} B_t^T + I_{k \times k})^{-1} B_t A_t^{-1}) (O_t + B_t^T Y_{t+1}) \quad (31)$$

We denote K_t as

$$K_t = I_{N \times 1} - A_t^{-1} B_t^T (B_t A_t^{-1} B_t^T + I_{k \times k})^{-1} B_t \quad (32)$$

So, the estimation of β_{t+1} can be given as

$$\beta_{t+1} = K_t \beta_t + K_t A_t^{-1} B_t^T Y_{t+1} \quad (33)$$

$$A_{t+1}^{-1} = K_t A_t^{-1} \quad (34)$$

According to above procedure, a summarization of the presented IELSSVM algorithm can be given as follows.

IELSSVM Algorithm:

Given the hidden node number L of ELM mapping feature space and a small number of training data to boost the algorithm, we present the algorithm as follows.

Step1: Initialization:

(1) Generate $\{a_i, b_i\}_{i=1}^L$ randomly to formulate the RBF activation function

$$G(a_i, b_i, x_i) = e^{-b_i(\|x_i - a_i\|)}$$

(2) Generate the mapping matrix as

$$\Phi(X) = \begin{bmatrix} G(a_1, b_1, x_1) & G(a_2, b_2, x_1) & \cdots & G(a_L, b_L, x_1) \\ G(a_1, b_1, x_2) & G(a_2, b_2, x_2) & \cdots & G(a_L, b_L, x_2) \\ \cdots & \cdots & \cdots & \cdots \\ G(a_1, b_1, x_N) & G(a_2, b_2, x_N) & \cdots & G(a_L, b_L, x_N) \end{bmatrix} \quad (35)$$

for the given small number of training data.

(3) Initialize the parameters of $\begin{bmatrix} w \\ b \end{bmatrix}$ for time t as

$$\begin{bmatrix} w_t \\ b_t \end{bmatrix} = \begin{bmatrix} \frac{1}{\nu} I + \Phi(X_t)^T \Phi(X_t) & \Phi(X_t)^T E_t \\ E_t^T \Phi(X_t) & E_t^T E_t \end{bmatrix}^{-1} \begin{bmatrix} \Phi(X_t)^T D_t E_t \\ E_t^T D_t E_t \end{bmatrix} \quad (36)$$

$$A_t^{-1} = \begin{bmatrix} \frac{1}{\nu} I + \Phi(X_t)^T \Phi(X_t) & \Phi(X_t)^T E_t \\ E_t^T \Phi(X_t) & E_t^T E_t \end{bmatrix}^{-1} \quad (37)$$

$$O_t = \begin{bmatrix} \Phi(X_t)^T D_t E_t \\ E_t^T D_t E_t \end{bmatrix} \quad (38)$$

Step 2: Online incremental learning procedure.

Repeat the following steps:

(1) Generate the matrix $H(X_{IC})$ for the last arrival data

$$H(X_{IC}) = \begin{bmatrix} G(a_1, b_1, x_{N+1}) & G(a_2, b_2, x_{N+1}) & \cdots & G(a_L, b_L, x_{N+1}) \\ G(a_1, b_1, x_{N+2}) & G(a_2, b_2, x_{N+2}) & \cdots & G(a_L, b_L, x_{N+2}) \\ \cdots & \cdots & \cdots & \cdots \\ G(a_1, b_1, x_{N+k}) & G(a_2, b_2, x_{N+k}) & \cdots & G(a_L, b_L, x_{N+k}) \end{bmatrix} \quad (39)$$

(2) Generate the other matrix such as

$$B_t = [H_{IC} \quad E_{k \times 1}] \quad (40)$$

$$Y_{t+1} = \begin{bmatrix} y_{N+1} \\ y_{N+2} \\ \dots \\ y_{N+k} \end{bmatrix} \quad (41)$$

(3) Estimate the parameters of $\beta = \begin{bmatrix} w \\ b \end{bmatrix}$ using the newest data

$$K_t = I_{N \times 1} - A_t^{-1} B_t^T (B_t A_t^{-1} B_t^T + I_{k \times k})^{-1} B_t \quad (42)$$

$$A_{t+1}^{-1} = K_t A_t^{-1} \quad (43)$$

$$\beta_{t+1} = K_t \beta_t + K_t A_t^{-1} B_t^T Y_{t+1} \quad (44)$$

Step 3: Online Use:

Use the learned parameters of $\beta = \begin{bmatrix} w \\ b \end{bmatrix}$ to evaluate the output \hat{Y}_{test} for the given n_{test} input vector X_{test} in actual application by

$$\hat{Y}_{test} = \Phi(X_{test})w + E_{n_{test} \times 1} b \quad (45)$$

5. Simulation Results

5.1. Simulation Results and Analysis of Benchmark Problems

In this section, the performance of the presented IELSSVM algorithm is shown by comparison with OS-ELM algorithm presented by [10] on a lot of Benchmark problems which are very famous and can be downloaded from UCI Repository of machine learning databases[15] conveniently. The specification of datasets of the benchmark problems is listed in table 1.

In table 1, the Secom dataset is a typical semiconductor manufacturing process dataset, in which all the data is collected from monitoring sensors or measurement points. Each data represents a lot with a number of measured feature values and a label for passing or failing in house line testing.

Simulation is carried out with the parameters selected as [10]. The parameters for initialization is that for regression problem $N_0 = L + 50$ and for classification problem $N_0 = L + 100$, where N_0 is the number of data randomly selected from the data set to initialize the algorithm and L is the number of

hidden nodes. For IELSSVM, the parameters of ν for regression problem is selected as e^{N_0-5} while for classification problem the parameter is selected as e^{N_0-20} . The detailed comparison is listed in table 2. We did not deliberately select the hidden nodes for Secom datasets because of the universal approximation capability of ELM based algorithm [7,8], for which good generalization performance can be attained as long as the hidden nodes is enough. Also, to predetermine the optimal hidden nodes is not very practical as the data comes chunk by chunk or one by one in actual application. Since [10] didn't give the standard deviation for the OS-ELM algorithm, we fill the corresponding items with "---" in the table 2 and table 3.

Table 1. Spcification of the dataSets

Dataset	Training Data	Testing Data	Attributes	Classes
Auto-MPG	320	72	7	---
Abalone	3000	1177	8	---
California Housing	8000	12640	8	---
Image Segment	1500	810	19	7
Satellite Image	4435	2000	36	6
Secom	1200	367	591	2

Table 2. Comparison for classification problems

Datasets	Algorithm	Nodes	Training accuracy(%)		Testing Accuracy (%)	
			Mean	Std	Mean	std
Image Segment	OS-ELM (RBF)[10]	180	96.65	---	94.53	---
	IELSSVM	180	97.19	0.28	95.14	0.45
Satellite Image	OS-ELM (RBF)[10]	400	93.18	---	89.01	---
	IELSSVM	400	94.73	0.28	91.42	0.36
Secom	OS-ELM (RBF)	200	94.57	0.12	90.28	0.33
	IELSSVM	200	94.59	0.12	90.31	0.40

From table 2 and table 3, we can see that IELSSVM attains better performance than OS-ELM for both classification problems and regression problems in most cases on benchmark problems.

Table 3. Comparison for regression problems

Datasets	Algorithms	Nodes	Training RMSE(10^{-3})		Testing RMSE(10^{-3})	
			Mean	Std	Mean	std
Auto-MPG	OS-ELM (RBF)[10]	25	69.6	---	75.9	---
	IELSSVM	25	68.4	1	64.4	1.6
Abalone	OS-ELM (RBF)[10]	25	75.9	---	78.3	---
	IELSSVM	25	73.8	0.3	73.2	0.3
California Housing	OS-ELM (RBF)[10]	50	132.1	---	134.1	---
	IELSSVM	50	129.9	0.7	134.4	1.9

5.2. Simulation results of actual overlay prediction in lithography of semiconductor production line for integrated scheduling

Lithography is the most important part in the semiconductor production line which transfers the patterns of a reticle onto the surface of a silicon wafer. Overlay is the key parameter for the lithography process to ensure that the relative position of current layer and its pre-layer is within the regulatory value. If the overlay value exceeds the regulatory value, the electrical properties will deteriorate and even become ineffective, thus resulting in lot scraps and reworks. In real semiconductor line, the offset value range of X-axis and the offset value range of Y-axis are strictly set. There are many factors to influence the overlay performance, such as the machine performance, the variety of the lot, the current layer of the lot, the machine used to process the pre-layer and etc. For different lithography machine, the machine performance may be different sharply. So the offset value may remarkably be different when processing the same lot. Prohibiting the lots from being processed by the machines with exceeding offset values will help to avoid unexpected quality events.

In order to accurately predict the overlay value of the lot, history data and theoretical offset compensation data of current lot are used together. The input attributes are selected as a 11-dimension vector, which includes the theoretical offset compensation data of current lot and the related data of past five lots that have been processed with the same product variety and the same layer. The theoretical offset compensation and the overlay

measurement attributes of the last five lots are selected. So there are total 11 attributes selected as the input attributes. The overlay measurement value of current lot is selected as the output attribute. Data of each input attribute is normalized into [-1,1]. All the data are collected from real semiconductor production line and the time is between 2010-8-2 and 2012-7-27. There are about 1192 records. 500 records selected randomly are used as the testing data while the other 692 records are used as the training data.

The parameters for initialization is that $N_0 = L + 200$, where N_0 is the number of data randomly selected from the data set to initialize the algorithm and L is the number of hidden nodes. For IELSSVM, the parameters of ν is selected by a wide range of trials from the 50 parameters $\{2^{-24}, 2^{-23}, \dots, 2^{24}, 2^{25}\}$. The best value $\nu = 2^{-6}$ is selected. The simulation results are given in table 4. Clearly, IELSSVM achieves better performance than OS-ELM on real overlay prediction problem.

Table 4. Comparison for real overlay prediction problem

Datasets	Attribute name	Algorithms	Nodes	Training RMSE(10^{-3})		Testing RMSE(10^{-3})	
				Mean	Std	Mean	std
Real overlay prediction dataset	X-offset	OS-ELM (RBF)	100	19.3	0.11	27.2	0.70
		IELSSVM	100	20.2	0.19	25.5	0.29
	Y-offset	OS-ELM (RBF)	100	17.4	0.98	27.9	1.5
		IELSSVM	100	18.3	0.10	22.0	0.27

In order to further evaluate the performance of the overlay prediction, the error curve of IELSSVM of all the data points is given in figure 1. Clearly, the prediction error of all the data is within the interval of [-0.1,0.1], which indicates that the prediction results can meet the accuracy requirements and can be used as the prediction model for integrated scheduling.

In actual integrated scheduling application, the overlay performance of each machine to process each lot is predicted by the IELSSVM algorithm. If the predicted value exceeds the permissible range predetermined, the lot should not be processed by the machine. The prediction results will be regarded as additional constraints when optimizing the scheduling strategy. The scheduling strategy developed on the basis of quality prediction will keep consistent with the actual environments, thus effectively reducing unexpected quality events and improving the scheduling performance.

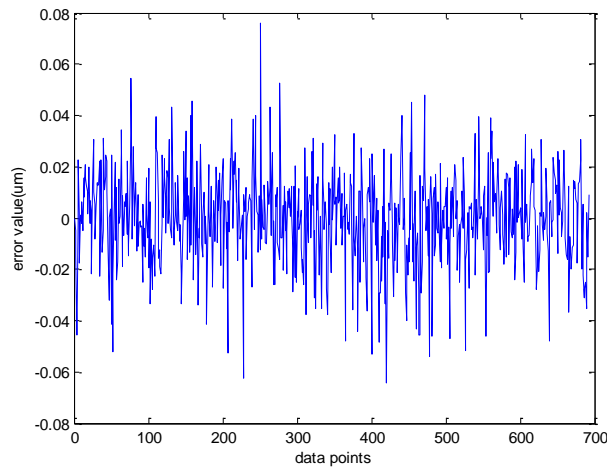


Fig. 1. The overlay error curve for all the real data points of lithography in semiconductor production line

6. Conclusion

In this paper, a novel Incremental Extreme Least Square Support Vector Machine for quality prediction and integrated scheduling of semiconductor production line is presented, in which data is first mapped into ELM feature space and then minimized under the structural risk principal like LSSVM. An incremental formula is given to accommodate the actual application where data may come one by one or chunk by chunk. Comparisons with the OS-ELM on classification and regression benchmark problems and real overlay prediction problems show that IELSSVM outperforms OS-ELM in most cases. That reason lies in that the IELSSVM is constructed on the basis of Structural Risk Minimization principal, thus avoiding over-fitting which OS-ELM constructed on the basis of Empirical Risk Minimization principal cannot avoid.

Acknowledgment. This work was supported by the National Key Basic Research and Development Program of China (2009CB320602), the National Natural Science Foundation of China (No. 60834004, 61025018, 61021063, 61104172) and the National Science and Technology Major Project of China (2011ZX02504-008).

References

1. Min Liu, Cheng Wu: Genetic algorithm using sequence rule chain for multi-objective optimization in re-entrant micro-electronic production line. *Robotics and Computer-Integrated Manufacturing*, 20: pp.225-236. (2004)
2. S. Haykin: *Neural Networks: A comprehensive Foundation*. Prentice Hall, New Jersey. (1999)
3. Corinna Cortes and Vladimir Vapnik: Support vector networks. *Machine Learning*, vol. 20, 3: pp. 273–297. (1995)
4. J. A. K. Suykens and J. Vandewalle: Least squares support vector machine classifiers. *Neural Processing Letters*, vol. 9, 3: pp. 293–300. (1999)
5. Glenn Fung and Olvi L Mangasarian: Proximal support vector machine classifiers. *The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, USA, pp.77-86. (2001)
6. Qiuge Liu, Qing He and Zhongzhi Shi: Extreme Support Vector Machine Classifier. *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, pp. 222-233. (2008)
7. G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew: Extreme learning machine: Theory and applications. *Neurocomputing*, vol. 70, 1–3: pp. 489–501 (2006)
8. Huang, G.-B. , Zhou, H. , Ding, X. , Zhang, R. , "Extreme Learning Machine for Regression and Multiclass Classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2011, Issue 99: 1-17
9. G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Trans. Neural Netw.*, 2005, vol. 16, 1: pp. 57-67
10. Nan-Ying Liang, Guang-Bin Huang, P. Saratchandran, N.Sundararajan: A Fast and Accurate On-line Sequential Learning Algorithm for Feedforward Networks. *IEEE Transactions on Neural Networks*, vol. 17, 6: pp. 1411-1423. (2006)
11. N. Syed, H. Liu, and K. Sung. "Incremental learning with support vector machines," In *Proceedings of the Workshop on Support Vector Machines at the International Joint Conference on Artificial Intelligence(IJCAI-99)*, Stockholm, Sweden, 1999
12. S. Ruping, "Incremental learning with support vector machines," *Proceedings of the IEEE International Conference on Data Mining*, 2001, pp:641-642
13. M. Espinoza, J. A. K. Suykens, and B. De Moor, "Fixed-size least squares support vector machines: A large scale application in electrical load forecasting," *Computational Management Science-Special Issue on Support Vector Machines*, 2006, vol. 3, 2: pp. 113-129
14. Gene H. Golub ,Charles F. van Van Loan: *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, Baltimore. (1996)
15. <http://archive.ics.uci.edu/ml/>

Lu Guo received his B.Eng. degree from Harbin Institute of Technology in 2001, Harbin, China and his M.Eng degree from the Second Academy of China Aerospace in 2004, Beijing, China. Now he is working toward his Ph.D. degree in the Department of Automation, Tsinghua University, Beijing, China. His research interests include scheduling optimization, extreme learning machines, machine learning and signal processing.

Jinghua Hao received Ph.D. degree of Control Science and Engineering in Tsinghua University. His research interests are modelling, simulation and optimization methods for complex manufacturing process. He has published more than 10 papers. He participated in several projects of the National program of China.

Min Liu received the Ph.D. degree from Tsinghua University, Beijing, China, in 1999. He is currently a Professor with the Department of Automation, Tsinghua University, Associate Director of Automation Science and Technology Research Department of Tsinghua National Laboratory for Information Science and Technology, Director of Control and Optimization of Complex Industrial Process, Tsinghua University, Director of China National Committee for Terms in Automation Science and Technology, Director of Intelligent Optimization Committee of China Artificial Intelligence Association. His main research interests are in optimization scheduling of complex manufacturing process and intelligent operational optimization of complex manufacturing process or equipment. He led more than 20 important research projects including the project of the National 973 Program of China, the project of the National Science and Technology Major Project of China, the project of the National Science Fund for Distinguished Young Scholars of China, the project of the National 863 High-Tech Program of China, and so on. He has published more than 100 papers and a monograph supported by the National Defense Science and Technology Book Publishing Fund. Dr. Liu won the National Science and Technology Progress Award.

Received: February 23, 2012; Accepted: November 20, 2012.

Application of Grid-based K-means Clustering Algorithm for Optimal Image Processing

Tingna Shi^a, Jeenshing Wang^b, Penglong Wang^a, and Shihong Yue^{a*}

^a School of Electrical Engineering and Automation, Tianjin University, Tianjin, China

^b Department of Electrical Engineering, National Cheng Kung University, Tainan 701, Taiwan

Email: Shyue1999@tju.edu.cn (* Correspondence Author)

Abstract. The effectiveness of K-means clustering algorithm for image segmentation has been proven in many studies, but is limited in the following problems: 1) the determination of a proper number of clusters. If the number of clusters is determined incorrectly, a good-quality segmented image cannot be guaranteed; 2) the poor typicality of clustering prototypes; and 3) the determination of an optimal number of pixels. The number of pixels plays an important role in any image processing, but so far there is no general and efficient method to determine the optimal number of pixels. In this paper, a grid-based K-means algorithm is proposed for image segmentation. The advantages of the proposed algorithm over the existing K-means algorithm have been validated by some benchmark datasets. In addition, we further analyze the basic characteristics of the algorithm and propose a general index based on maximizing grey differences between investigated objective grays and background grays. Without any additional condition, the proposed index is robust in identifying an optimal number of pixels. Our experiments have validated the effectiveness of the proposed index by the image results that are consistent with the visual perception of the datasets.

Keywords: electrical tomography; number of pixels; image reconstruction.

1. Introduction

Image processing plays an important role in a variety of applications such as robot vision, object recognition, and medical imaging [1][2][3]. Image segmentation is naturally a clustering course on the basis of pixels such as grey, veins, color and so on. As a result, the clustering analysis is widely applied in image segmentation.

With the rapid development of fundamental studies, more stringent demands are brought out for image segmentation. Under these circumstances, process tomography (PT) finds its unique role in studying the

multiphase flow phenomena encountered in chemical engineering fields, involving a number of tomographic techniques, such as X-ray, γ -ray, optical, ultrasonic, electrical, and nuclear magnetic resonance imaging. Among X-ray computed tomography (XCT) and electrical tomography (ET), an important issue of resolution in the XCT imaging process is to determine a proper number of pixels. Zhao *et al.* [4] proposed a multi-parameter method (includes area error and image centre position error) for the electrical capacitance tomography image evaluation. Graham and Adler [5], [6] defined the resolution in terms of fuzzy radius, and adopted the ratio between the radius of the interested region and that of the imaging region, the axial error and the image energy to evaluate the quality of reconstructed images. To date, however, none of the existing methods provides a universally acceptable solution to the number of pixels in any reconstructed x-ray images. In fact, these existing methods have many limitations. First, the methods to determine the number of pixels are often subjective with little satisfaction. Second, the objects under evaluation are binary images or gray-scale images. Third, some prior information such as image error resolution and duty ratio are considered in the evaluation process. The last but not the least, the evaluation process itself is not objective enough. For example, a known mesh of pixels is used during the whole image reconstruction process.

On the other hand, the K-means is one of the most frequently used methods for image segmentation [7] and its success chiefly attributes to the introduction of the belongingness of each image pixels. However, there are some key issues unsolved when using K-means clustering algorithm to deal with image segmentation, such as the determination of the number of clusters in a data set, the precise position of the cluster prototype, the real-time performance and initialization problem related to local optimization. Some research studies have attempted to overcome the above problems [8], [9]. Recently, an efficient grid-based k-means (G-K-means) algorithm for clustering has been proposed [10]. The G-K-means algorithm can efficiently overcome the above problems and take advantage over the existing K-means algorithms. In this paper, we further analyze the basic characteristics of the G-K-means algorithm, and demonstrate its advantages over the traditional K-means algorithm in image segmentation. Finally, we apply the G-K-means algorithm to image reconstruction in the x-ray field.

In this paper we propose a robust index to determine an optimal number of pixels based on maximizing grey differences between investigated grays and background objective grays. Results obtained by the finite element method (FEM) and Gent 4[®] software are also given. These experiments are conducted to validate the effectiveness and robustness of the proposed index.

2. Related work

In the following we present the main steps of both the K-means algorithm and

the G-K-means algorithm in terms of their iterative process and major problems in applications.

2.1. K-means algorithms

Let $X=\{x_i | i=1, 2, \dots, n\}$ be a dataset with n data distributed in c clusters. The objective function of the K-means algorithm can be stated as

$$\min \sum_{i=1}^K \sum_{j=1}^n u_{ij} d_{ij}, \quad s.t. \sum_{i=1}^K u_{ij} = 1, \quad (1)$$

where $d_{ij} = \|x_j - v_i\|$, v_i is the prototype of the i -th cluster, u_{ij} is the membership degree of the j -th data object to the i -th cluster, taken as either 0 or 1. In terms of the Lagrange optimization method, the optimal prototypes for Eq.(1) are

$$v_i = \sum_{j=1}^n u_{ij}^m x_j / \sum_{j=1}^n u_{ij}^m, \quad i=1, 2, \dots, c \quad (2)$$

Depending on the group of prototypes, the K-means algorithm assigns each object to the most similarity cluster based on the similar measure;

The above process is repeated and the algorithm stops if a convergence criterion is met. For the K-means algorithm, all membership degrees are expressed in a $n \times c$ matrix $U=[u_{ij}]$.

Although the K-means algorithm has been succeeded in many applications, it suffers largely from the following problems. First, the clustering results of the K-means algorithm greatly depend on its initialization, and are prone to fall into a local solution without a good initialization partition. Second, since the data assigning problem in the K-means algorithm is not optimal, and thus they are not suitable for applications where the membership degrees are assumed to represent typicality/compatibility with an elastic constraint, or are applied in noise circumstances. Third, the number of clusters must be known in advance when the FCM algorithm is used for clustering a dataset. In the past decades, many studies have been presented to overcome these problems, but these algorithms are little satisfied in image segmentation application except the G-K-MEANS algorithm as introduced below.

2.2. G-K-means algorithms

The G-K-means algorithm firstly divides the minimum closed set containing the data into optimal grid structures according to the optimal partition index [8], and then divides the grid structures into the online grid (to participate in the current iteration) and the offline grid (not to participate in the current iteration) as follows.

Definition 1. Mean center and geometric center. The mean center of a cluster is the arithmetic average of all object vectors in the cluster, while the geometric center of a cluster is the center of a minimal hypersphere that encloses all objects in this cluster.

Usually, the geometric center and the mean center of any cluster are different but are very close to each other if the cluster is symmetric (e.g. sphere(ellipsoid)-shaped).

Definition 2. On line grid and off line grid. Let J be an integer and S be a set of grids. A grid is called an online grid if the grid is one of top K high-density grids in S . Any grid that is not an online grid in S is called an offline grid.

All J grids in S are assigned into two sets $X_1 = \{G_1, G_2, \dots, G_K\}$ and $X_2 = \{G_{K+1}, G_{K+2}, \dots, G_J\}$ such that

$$|G_{K+1}| \geq |G_{K+2}| \geq \dots \geq |G_J|. \quad (3)$$

Thus X_1 is the set of online grids and X_2 is the set of offline grids.

Next, the steps which are similar to the K -means algorithm but have different mechanisms are carried out with the following objective function:

$$\text{Max} \sum_{i=1}^K |G_i - G_1 - G_2 - \dots - G_{i-1}| / V_i. \quad (4)$$

The G - K -means algorithm can work without the number of clusters K but with a database containing n objects based on the following six steps:

1. Choose K geometric centers of online grids as the initial mean centers, v_1, v_2, \dots, v_K ;
2. Repeat;
3. (Re)assign objects to G_i that is centralized on v_i , for $i = 1, 2, \dots, K$;
4. Update cluster centers by

$$v_i = \sum_{j=1}^{|G_i|} u_{ij} x_j / u_{ij}, i = 1, 2, \dots, K \quad (5)$$

5. Go to step 6 if a convergence criterion is met; Otherwise, go to Step 2.

Stop if there are no overlapping grids in X_1 ; Otherwise, add the offline grid with the highest density into X_1 ; go to Step 2.

The G - K -means algorithm can work under two states. If the number of clusters is known a priori, the value of K remains invariant in each iteration. Otherwise, Step 6 can automatically determine the final number of clusters. As opposed to the K -means algorithm, the G - K -means algorithm tends to find cluster centers in high-density areas while K -means cannot guarantee this point. Therefore, the cluster quality should be better from the viewpoint of density. Moreover, the algorithm assigns objects to grids, which needs much less computation than that of the Euclidean distance widely applied in many algorithms such as the K -means algorithm.

3. Basic characteristics of the G-K-means algorithm and optimal number of pixels

In the following, we first explore two key problems using the G-K-means algorithm in applications in terms of their basic characteristics and convergence domains. Then an index to determine an optimal number of pixels is proposed in image segmentation and X-ray image processing.

3.1. Basic characteristics and convergence of the G-K-means algorithm

◆ Objective function: The objective functions from the K-means to the G-K-means algorithm satisfy

$$\sum_{i=1}^K \sum_{j=1}^n u_{ij} d_{ij} \rightarrow \min \quad \mapsto \quad \sum_{i=1}^K |G_i - G_1 - G_2 - \dots - G_{i-1}| / V_i \quad (6)$$

In (6), the objective function of the K -means algorithm is reformulated by a grid-based form in the G-K-means algorithm. Consequently, the maximum of the objective function in the K -means algorithm is attained by maximizing the average density of a group of online grids, G_i , $i = 1, 2, \dots, K$. Thus the objective function in the K -means algorithm is data-object-based while the one in the G-K-MEANS algorithm is grid-based.

◆ Object assigning principle: Let each cluster be enclosed by a separate minimal hypersphere. When an object falls in a hypersphere, the object usually is closer to the geometric center of the hypersphere than any other geometric centers. In the K -means algorithm, the object is assigned to the cluster based on the hyperspheres. Consequently, the object assigning principle is to iteratively find K minimal hyperspheres that enclose K clusters respectively such that the geometric centers of these hyperspheres can be well-determined (see Fig.1).

Contrarily, the G-K-means algorithm iteratively applies K online grids that are centralized on v_1, v_2, \dots, v_K in place of K hyperspheres to enclose those objects in each clusters. Namely, an object is assigned to the i -th cluster if the object uniquely falls into the grid G_i that stands for the cluster, for $i = 1, \dots, K$. Finally, the G-K-means algorithm assigns all remaining objects that are not covered in any grid to their closest centers.

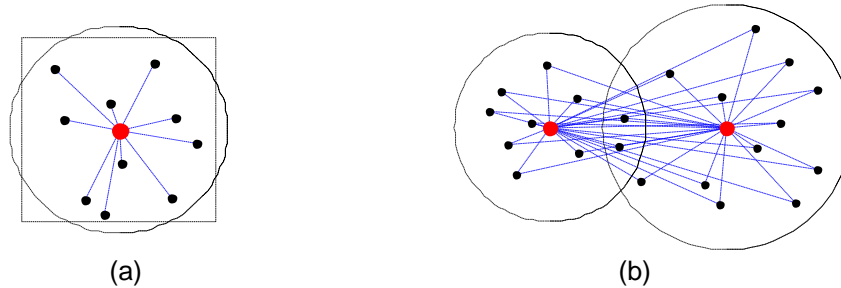


Fig. 1. Comparison of the assigning principle of the K -means and the G-K-MEANS algorithms. (a) Grid-shaped and sphere-shaped neighborhoods; (b) All Euclidean distances in two partially overlapped clusters.

◆ **Center update:** Any new mean center in the G-K-means algorithm is locally computed by these objects limited to the related grid. In contrast, any mean center in the K -means algorithm may be affected by any object no matter how far it is from its involved center. In the G-K-means algorithm, if the distance between two mean centers of any two online grids is less than a threshold ε , the two online grids is considered as the same cluster. The low-populated one in the two online grids is removed from X_1 and the most high-populated offline grid in X_2 is added into X_1 in the next iteration. All iterations are terminated if the difference of the objective function values between two consecutive iterations is smaller than the threshold ε and $|X_1|=K$. Fig. 2 shows the flowchart of the G-K-means algorithm.

Comparing the convergence of the G-K-means algorithm with that of the K -means algorithm, we explain the problem as follows. The local convergence is an indicator which measures the risk for algorithms falling into a local minimum but it is difficult to theoretically analyze its characteristics. For the G-K-means algorithm and the K -means algorithm, the definitions of convergence are as follows.

Definition 3. Convergence domain. Choose a group of points as the initial clustering prototypes of the G-K-means or the K -means algorithms. Starting from the group of points, if the G-K-means algorithm or the K -means algorithm can converge to a local optimum, the group of points is called the convergence points of the algorithm. The set of all convergence points is called a convergence domain for the corresponding algorithm.

For example, we introduce a set of two-dimensional data points distributed in six irregular clusters data set to illustrate the concept of convergence domain, and to directly analyze the convergence of the experimental results. All data points are enclosed by a minimal closed set in which any point serves as a candidate of a convergence point to examine. The two algorithms group all data points over the minimum closed set which contains the space that minimizes the objective function. Denote O_{\max} as a sufficiently small number (say 10^{-5}). For any group of initial points, if the difference between the final iteration is less than O_{\max} , we denote these data as convergence points. The set of all the convergence points consists of a convergence domain. Fig.

2 (b) and (c) show that the convergence of the G-K-means algorithm covers almost all the high-density areas (the yellow areas), only a small part of the high-density areas is uncovered, which is due to the impact of other clusters. It has no effect on the clustering results because the small part of the high-density online gray area is affected by adjacent clusters. Thus, they deviate from the high-density areas and become offline grids. Fig. 2 (a) clearly shows that the convergence domain of the K-means algorithm is less than that of G-K-means. This means that it is greater risk for the former than the latter to minimize the objective function. In particular, once the iteration point falling out of the domain of convergence there exists at least one cluster that is incorrectly distinguished because of lacking a flexible mediation mechanism between online and offline grids (pixels).

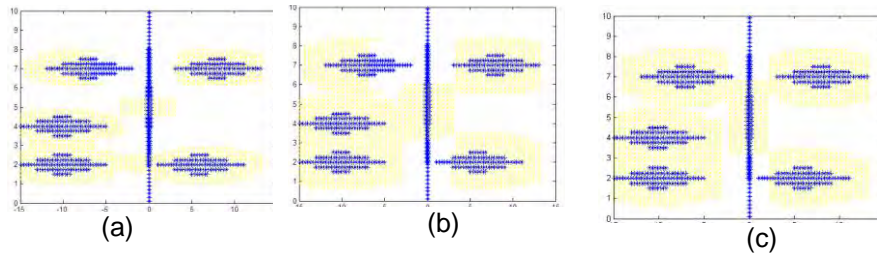


Fig.2. The convergence domain of different algorithms. (a) The convergence domain of K-means algorithm (b) The convergence domain of the G-K-MEANS algorithm with 16 initial grids. (c) The convergence domain of the G-K-MEANS algorithm with 32 initial grids.

3.2. X-ray imaging principle

The principle of the radiation attenuation is the first foundation of XCT. Assuming that the X-rays are monoenergetic from a parallel beam, the residual intensity of X-rays is attenuated by the materials that the X-rays pass through:

$$I = I_0 e^{-\int_L \mu_m dl} \quad (7)$$

where μ_m is the mass attenuation coefficient (m^2/kg) mainly due to the photoelectric, Compton and pair production effects; and dl represents infinitesimal thin layer of the object (m). The corresponding expression of the projection data is

$$b_L = \ln\left(\frac{I_0}{I}\right) = \int_L \mu_m \rho ds \quad (8)$$

where ρ is the density of the object (kg/m^3). The image reconstruction techniques use the measured projection data as the *input* to calculate the

density distribution of the desired cross-section of the investigated sample as the *output*. Accordingly, the 2D image of the desired cross-section can be obtained. The mostly applied reconstruction techniques are divided into two categories, the back-projection reconstruction and the iterative reconstruction. As an example of the back-projection reconstruction technique, the filtered back-projection (FBP) technique is used in most commercial medical scanners and has proved to be extremely accurate and amenable to fast implementation. This technique can give a rather straightforward intuitive rationale because each projection represents a nearly independent measurement of the object.

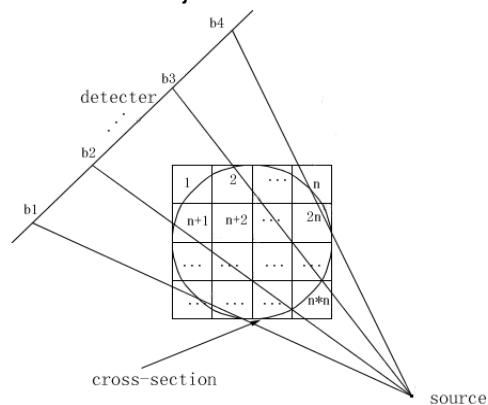


Fig.3. Illustration of the principle of XCT

As shown in Fig. 3, a 2D object is placed between the X-ray source and the detector. The investigated area is discretized by rectangular grids or triangular ones. The mathematical expression of the tomographic problem is given by the equations:

$$b = AX, \text{ i.e., } b_k = \sum_{j=1}^{M \times N} a_{kj}, k = 1, 2, \dots, K \quad (9)$$

where K is the total number of the projections, $M \times N$ is the total number of pixels across the area, a_{kj} means the weighting factor for the contribution of pixel j to the projection element b_k . The goal is to determine the unknown image X when the experimental projections b are available. The weighting factor, a_{kj} , is calculated based on the geometrical consideration as the intersection length/area of the k th projection ray with the image pixel j . In the discrete form, the aim of image reconstruction for the X-ray field is to find the unknown X from the known b by using Eq. (10), that is

$$X = A^{-1}b \quad (10)$$

However, the direct analytical solution for Eq. (9) does not exist since the inverse problem is both nonlinear and ill-posed, little noise in the measured data could cause large errors in the estimated conductivity. Consequently, it is necessary to use numerical techniques to approximate S^{-1} as accurate as

possible after applying some residual criterion. One efficient criteria is minimal least error defined as

$$\|b - AX\|_2^2 \rightarrow \min \quad (11)$$

Eq. (11) is used to identify the optimal values of X . Many variants of Eq. (11) have proposed to solve the ill-posed problem among which the most used ET image reconstruction algorithms are the linear back projection (LBP). In the LBP algorithm the conductivity distributions are assumed to comprise a number of discrete regions within the measurement space such that the conductivity within each region is constant. According to Eq. (10),

$$X = A^T B / A^T u_\lambda, \quad s.t. u_\lambda = [1, 1, \dots, 1] \quad (22)$$

Eq. (12) shows that the grey values of any pixels are calculated using a weighted form in the algorithm [13].

In this paper, we first use the LBP algorithm for X-ray imaging reconstruction to examine the optimality of space resolutions of X-ray imaging under various numbers of pixels. Different from the existing LBP algorithm, these calculated values by Eq. (12) are indirectly applied to reconstruct any images. Inversely, we denote \bar{X} as the set of all values of X from Eq. (12) and then the G-K-means algorithm is applied to cluster all members in \bar{X} . Finally, these members in different clusters are endowed to different gray while the members in the same cluster have the same gray. These gray degrees over all pixels thus reflect the information of the spatial distributions of the investigated materials.

When applying the existing K-means clustering algorithm, one needs to predefine the number of clusters K . Currently, there are two applicable ways to determine the number of clusters. First, in some cases, there is a priori knowledge of the actual number of clusters since the multiphase flows in a measured field consists of determined components such that these multiphase flows consist of as the gas, water, and oil in a crude oil transmission pipe. Thus we take the number of clusters that is larger than three since the X-ray imaging inevitably contains the trail traces that have to be represented by some additional clusters of more than three. Second, if there is no prior number of clusters available, the issue of determining the number of clusters falls into the category of clustering validity indices in a given dataset [14], i.e., the number of clusters can be determined by a proper function called the clustering validity index [15], [16]. There are some efficient validity indices in fuzzy clustering such as the partition entropy (PE) proposed by Bezdek [17], the Xie-He index [18], etc. As the running time increases, these indices work well for some datasets. However, these methods to determine the number of clusters have their limitations. Essentially, most of them are specifically designed to the K-means-like algorithm. As compared, the G-K-means algorithm can automatically suggest an optimal number of clusters and overcome most limitations of the existing K-means clustering algorithm in image segmentations.

An optimal index to determine the optimal number of pixels

The optimal number of pixels plays an important role in image processing. When applying the G-K-means algorithm, the number of pixels is determined by the grid size; that is, the grid size can affect the partitions of the G-K-means algorithm. An overlarge grid may contain two or more clusters while too small grid may lead to too many online grids to be assigned to the same cluster and a significant increase of CPU possessing runtime. Basically, a cluster consists of the centric high-density grids while the surrounding low-density grids and empty grids separate the given data structure into different groups (clusters) that aggregates similar data objects (see Fig. 4). All nonempty grids are ordered into three sets of grids: $D(t)$, $t = 1, 2, 3$, satisfying that the object number of any grid in $D(t)$ is larger than any other grid in $D(t+1, j)$, $t = 1, 2$. The two classes of high and low populated grids respectively cover the center and margin areas of all clusters. The larger differences between the two classes of grids are, the easier identification of different clusters by a clustering algorithm is. This is the core idea hidden in (10). It has been demonstrated that there must be an optimal value of the ration between $D(1)$ and $D(3)$ for any dataset with any cluster structure, and the optimal value indicates that most of the clusters in the dataset have been broken by the partitioned grids [9]. Consequently, the ration can act as the upper bound of initial number of grids and predict the optimal grid size.

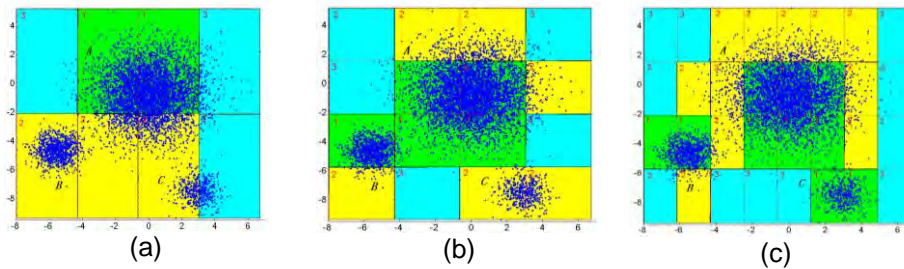


Fig.4. Three classes of grids under different grid sizes in $D(1)$, $D(2)$ and $D(3)$ from high to low density correspond to center, inner, and margin areas of most clusters in a dataset with three clusters. (a)–(c) corresponds to three classes of grids, respectively. The same color indicates the same class of grids

Accordingly, if the number of pixels can be incorrectly determined, a good-quality segmented image cannot be guaranteed. We thus propose a general index based on maximizing grey differences between investigated objectives and background as follows.

Consider a pixel dataset $M = [x_1, x_2, \dots, x_n] \in R^{d \times n}$, where each column of M , $x_i = (x_{i1}, x_{i2}, \dots, x_{id})^T \in R^d$, is a singleton data object in a d -dimensional data space. For example, when the number of ray resources is 16, $d=16$. Let ε_{\max} and ε_{\min} be the maximal and minimal distances between any pair of data objects in M respectively and $\varepsilon \in [\varepsilon_{\min}, \varepsilon_{\max}]$. We take a number of equidistant breaking points $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{s-1}, \varepsilon_s$ from ε_1 to ε_s to partition ε ,

$\varepsilon_1 = \varepsilon_{\min}$, $\varepsilon_s = \varepsilon_{\max}$, where s is the number of breaking points. A breaking point s_k can be characterized as

$$\varepsilon_k = \varepsilon_1 + (k-1)\Delta, k = 1, 2, \dots, s \quad (33)$$

where Δ is the distance between any two adjacent breaking points. Let $N_{\varepsilon_k}(x_j)$ be the subset of the data objects in the ε_k -neighborhood of x_j , $k = 1, 2, \dots, s$; $j = 1, 2, \dots, n$. Group all subsets of x_j into three sets of grids: $D(s, \varepsilon_k)$, $s = 1, 2, 3$, such that the density of any subset in $D(t, \varepsilon_k)$ is greater than any other subset in $D(t+1, \varepsilon_k)$, $t = 1, 2$. Hereafter, $|\bullet|$ denotes the number of data objects in the contained set. The optimal neighborhood size, say GPI, is determined by maximizing the ratio of $|D(1, \varepsilon_k)|$ and $|D(3, \varepsilon_k)|$; that is,

$$\text{GPI} = \arg \max_{\varepsilon_k} \{ |D(1, \varepsilon_k)| / |D(3, \varepsilon_k)| \}, \quad (44)$$

Hereafter GPI is called a general partitioning index (GPI) in this paper. The two classes of subsets in $D(1, \varepsilon_k)$ and $D(3, \varepsilon_k)$ correspond to the centric and margin locations of all clusters. If $\varepsilon \rightarrow \varepsilon_{\min}$ each data point will have no neighborhood and $\varepsilon_{\text{opt}} = 1$ while $\varepsilon \rightarrow \varepsilon_{\max}$, the space will degenerate to a subset. The differences between any pair of subsets will disappear and $\varepsilon_{\text{opt}} = 1$. Consequently, there must be at least one maximum on Eq. (8) without any additional conditions. In the experiments, we will graphically show these maximums.

GPI in (9) aims at maximizing grey differences between investigated objectives and background since the investigated objectives respond to these pixels whose components have larger values while the background responds to these pixels whose components have smaller values.

Fig.5 shows the flowchart of our proposed approach in image segmentation application.

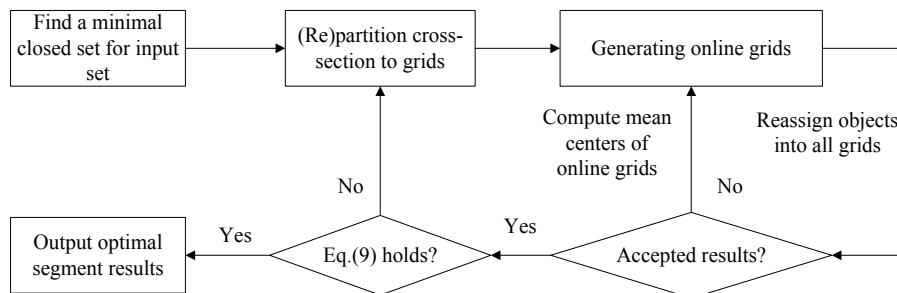


Fig.5. Flowchart of the proposed approach in image segmentation application.

4. Experiments

In this section we conduct two groups of experiments to validate of the effectiveness of the G-K-means algorithm and GPI respectively. In addition, we apply the K-means algorithm to the experiments under various numbers of pixels using the same datasets. These experiments are performed in the X-ray sensitiveness fields and analyzed by space resolution. Here the space resolution refers to the relative error of all n pixels in a partitioned image as

$$\zeta = \frac{1}{n} \sum_{j=1}^n \frac{(g_j - g_j^*)}{g_j^*} \quad (55)$$

where g_j is the reference grey degree of the j -th pixel, that is the real grey value; g_j^* is the grey value of the j -th pixel in the reconstructed image, $j=1, 2, \dots, n$; and ζ is the average of all errors of K pixels.

4.1. Image segmentation in a group of benchmark datasets

We apply the color and grey templates of the well-known Lena image as original images, and use the K-means and G-K-means algorithms for image segmentation. For the original Lena color images, the information is kept into $m \times n \times 3$ data matrix. We give each element in term of the array of pixel values as red(R) blue(B), green(G). Assuming the matrix I stored the information of Lena color pixel, we can get the color value of each pixel, denoted as $I_R = I(:, :, 1)$, $I_G = I(:, :, 2)$, $I_B = I(:, :, 3)$. A three-dimensional matrix are firstly constructed by I_R , I_G and I_B , and then we make use of the above two algorithm for image segmentation. The segmentation quality is measured by the following two indices:

1) The mis-segmented number of pixels. It is an important index to evaluate the segmented images in practice and is further formulated as Eq. (15).

2) Time cost. The runtime of an algorithm decides its applicable range. In case of most situations with the real-time demands, the K-means algorithm is too slow to be applied.

Fig. 5 and Fig. 6 visually show the segmented images by the above two algorithms. These results of the two algorithms are obtained under two conditions: (1) the number of clusters is selected from 2 to 12, and the best segmented image is shown in Figs. 5(b) and 5(c), and Figs. 6(b) and 6(c) respectively. (2) Four times of initializations are implemented in order to overcome the local minimum when we apply the K-means algorithm. In contrast to the two original images in Fig. 6 and Fig. 7, the segmented images from the G-K-means algorithm are clearer and tidier than those from the K-means algorithm, and more close to the real templates. Table 2 further verifies that the two statistical indices of the G-K-means algorithm are

superior to those of the K-means algorithm. In particular, the G-K-MEANS algorithm is rather stable and robust, and hardly is affected by the initialization problems that often are encountered in the applications of the K-means algorithm.



Fig.6. Segmented results of standard Lena image. (a) Original Lena image template. (b) K-means. (c) G-K-means.



Fig.7. Segmented results of standard Lena image by the two algorithms. (a) Lena image template. (b) K-means (c) G-K-means.

Table 1. Comparison of the image segmentation quality by two algorithms.

Algorithms	Mis-segmented pixel number	Relative error	Runtime (Seconds)
K-MEANS	1177	0.2619	0.2349
G-K-means	1026	0.1914	0.0143

4.2. Image segmentation in X-ray fields

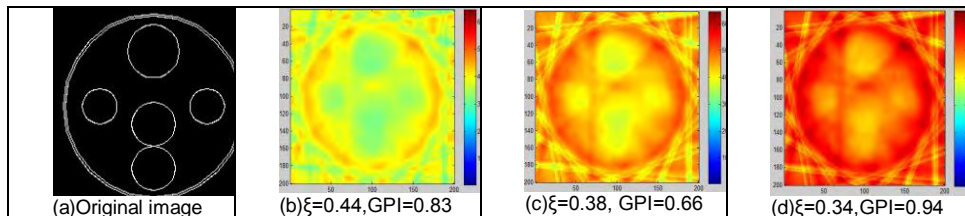
The datasets of this set of simulations are obtained from Geant4 [19]. The tested original images are five and twenty-five circles with continuously distributed materials, as shown in Table 3(a) and Table 4(a).

These circles have the same material component and thus should be shown as the same gray degree. The background must respond to two kinds of different grey degrees since there is a class of trail traces at least. We perform the LBP algorithm with various numbers of pixels (e.g., grids in the FEM) for image reconstruction, while the values of GPI are computed to find

its maximum for the optimal number of pixels. For comparison, the space resolutions of Eq. (10) are computed. As the number of pixels gradually decreases, these images of different space resolutions are obtained. It can be observed that GPI can clearly attain their maximums and, at the same time, the space resolution of the images that respond to the maximum by Eq. (10) attains its maximum. Please note that Eq. (10) is a subjective evaluation of space resolution due to dependence of necessary reference image. Inversely, GPI works well without any other information except the reconstructed image itself. Consequently, GPI perfectly is an objective evaluation index. After calculating these values of Eq. (7) for all pixels, we apply the G-K-MEANS algorithm to cluster all members in the set of the obtained values. When GPI attains its maximum, the corresponding number of pixels is optimal. In fact, the trail traces in a reconstructed image of the maximum of GPI is minimal, and the boundary of investigated objectives (see Table 3 (f) and Table 4 (d)). It can visually be observed that the trail traces in other images, whose values of GPI are far away from the maximum, are widely distributed, and even some circles are incorrectly connected to the same area or some circles cannot be found. Thus, these images show that GPI can better distinguish the optimal X-ray images by the determination of the optimal number of pixels. Table 3 gives the quantitative description of the imaging results of these two algorithms. Table 3 shows that all the space resolutions of the G-K-MEANS algorithm by Eq. (10) are superior to those of the K-means algorithm, while the time cost is far less than that of the K-means algorithm.

Fig. 8 summarizes the curve of GPI values in the above two groups of X-ray images. It can be seen from Fig. 3 that the two maximums are dominantly larger than other values. Thus GPI can clearly predict the optimal number of pixels and evaluate the X-ray image of the best space resolution. GPI can represent different physical meanings and can work objectively without any additional information. Specifically, to measure the robustness of GPI, the radiuses of these images in the two sets are taken by a group of very different values. Fig. 7 shows that the maximums of GPI are encountered in most radiuses. This demonstrates GPI is useful and efficient in most datasets. Indeed, in some datasets, GPI is inefficient and thus there still is a necessity to improve its performance.

Table 2. Original and reconstructed images that correspond to the original image of five circles.



Application of Grid-based K-means Clustering Algorithm for Optimal Image Processing

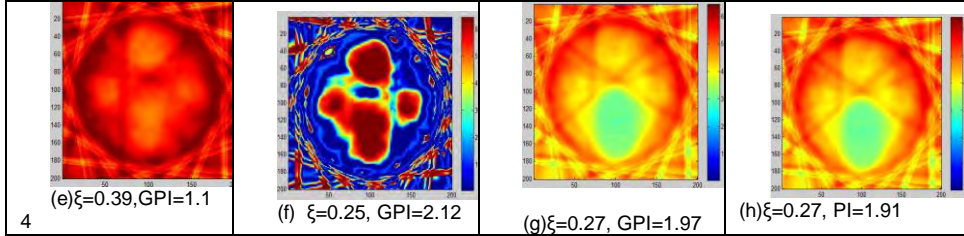


Table 3. Original and reconstructed images that correspond to the original image of twenty-five circles.

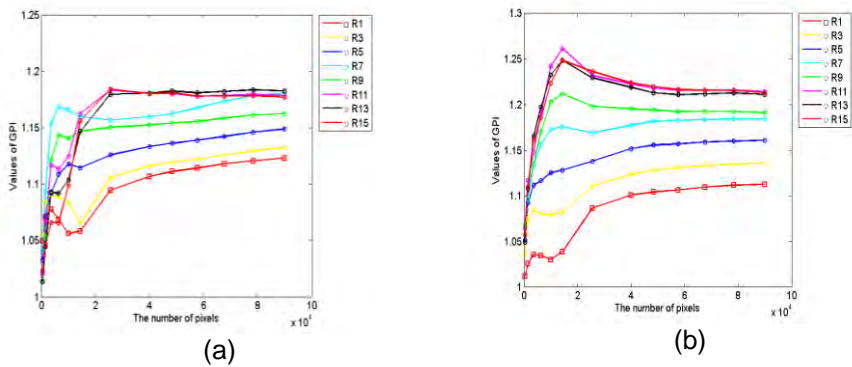
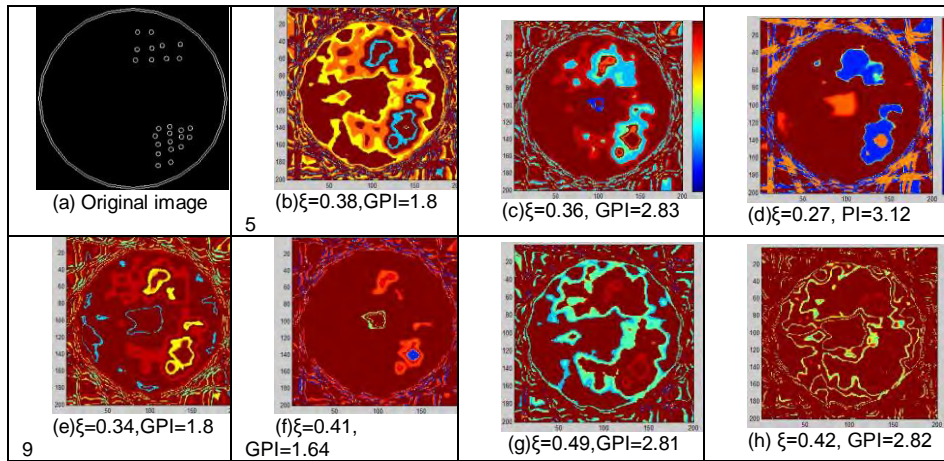


Fig. 8. Maximums of GPI in reconstructed images under different numbers of pixels from the LBP algorithms. (a) and (b) correspond to the two sets of three and twenty-five circles, respectively. The signs “R \times ” stand for a group of various radii.

Table.4. Comparison of the image segmentation quality by two algorithms

Algorithms	Mis-segmented pixel number	Relative error	Runtime (Seconds)
K-MEANS	2387	0.3697	0.2590
G-K-MEANS	2011	0.2273	0.0875

5. Conclusion

Image segmentation is the foundation of image analysis, image understanding and pattern recognition. However, due to the lack of fundamental measures to deal with the problem of image segmentation, the current common clustering algorithms cannot achieve unsupervised clustering. In this paper we apply the G-K-means algorithm to image segmentation applications. The G-K-means algorithm has not only linearly computational complexity but also simple and effective operation. The advantage of the G-K-means algorithm over the existing K-means algorithm has been demonstrated by its fast convergence and clustering performance. Our experimental results have validated that the proposed index for determining the optimal number of pixels is effective and comprehensible in its reconstructed images of high spatial resolution.

Acknowledgment. This work is supported by the National Science Foundation of China under Grant No. 61774014, 60572065, 60772080 and the National Science Foundation of Tianjin under Grant N0.08JCYBJC13800.

References

1. Zhang Y.: Image segmentation. Beijing: Science Press. (2001).
2. Olson C. F.: Maximum-Likelihood Image Matching, IEEE Trans. Patt. Anal. Mach. Intell., vol. 24, no. 6, pp. 853-891. (2002)
3. Pham D.L.: Partial models for fuzzy clustering, Computer Vision and Image Understanding, vol.84, pp.285-297. (2001)
4. Dudukovic, M.P.: Opaque multiphase flows: experiments and modeling. Experimental Thermal and Fluid Science, vol. 26, no.3, pp. 747-753. (2002)
5. Ahmed M. N., Yamany S M, and Mohamed N.: A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data, IEEE Trans. Medical Imaging, vol.21, no.3, pp.193~199. (2002)
6. Banholzer, W.F., Spiro, C.L., Kosky, P.G., Maylotte, D.H.: Direct imaging of time-averaged flow patterns in a fluidized reactor using X-ray computed tomography. Industrial & Engineering Chemistry Research 26, 763-770. (1987)
7. Bartholomew, R.N., Casagrande, R.M.: Measuring solids concentration in fluidized systems by Gamma-ray absorption. Industrial & Engineering Chemistry, vol.49, pp.428-436. (1957)
8. Zhao J, Fu W and Hu Q.: Investigation of evaluating method for reconstructed

- image quality of electrical capacitance tomography system. *Journal of Guangxi University (Science)*, Vol. 28, pp. 61-64 (2003)
9. Dyakowski T., Edwards, R.B., Xie, C. G. Williams, R.A.: Application of capacitance tomography to gas solid flows. *Chemical Engineering Science* , vol.52, pp.2099-2112. (1997)
 10. Grassler, T., Wirth, K.E.: X-ray computer tomography—potential and limitation for the measurement of local solids distribution in circulating fluidized beds. *Chemical Engineering Journal*, vol. 77, pp.65-73. (2000)
 11. Harvel G.D., Hori, K., Kawanishi, K., Chang, J. S.: Real-time cross-sectional averaged void fraction measurements in vertical annulus gas liquid two-phase flow by neutron radiography and X-ray tomography techniques. *Nuclear Instruments and Methods in Physics Research Section A*, vol.371, no.3, pp.544-554. (1996)
 12. Harvel, G.D., Hori, K., Kawanishi, K., Chang, J.S.: Cross-sectional void fraction distribution measurements in a vertical annulus two-phase flow by high speed X-ray computed tomography and real-time neutron radiography techniques. *Flow Measurement and Instrumentation*, vol. 10, no.4, pp.259-266. (1999)
 13. Xie X.L., Beni G.: A validity measure for fuzzy clustering, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.13, no.8, pp. 841–847, 1991.
 14. Yue S., Wang J., Gao T., Wang H.: An unsupervised grid-based approach for clustering analysis, *Science China-Information Science*. vol.53, no.6, pp. 1372-1384. (2010)
 15. Yue S., Wei M., Wang J., Wang H.: A general grid-clustering approach, *Patt. Recognition Letter*, vol.29, no.9, pp.1372-1384. (2008)
 16. Kai, T., Misawa, M., Takahashi, T., Tiseanu, I., Ichikawa, N., Takada, N.: Application of fast X-ray CT scanner to visualization of bubbles in fluidized bed. *Journal of Chemical Engineering of Japan*, vol.33, no.6, pp.906-911. (2000)
 17. Kantzas, A., Kalogeraki, N.: Monitoring the fluidization characteristics of polyolefin resins using X-ray computer assisted tomography scanning. *Chemical Engineering Science* 51 (10). (1979)
 18. Kantzas, A., Wright, I., Kalogerakis, N.: Quantification of channeling in polyethylene resin fluid beds using X-ray computer assisted tomography (CAT). *Chemical Engineering Science*, vol. 52, no.13, pp.2023-2033. (1997)
[Http://geant4.slac.stanford.edu/installation/](http://geant4.slac.stanford.edu/installation/)

Tingna Shi received the B.Sc. and M.S. degrees in the Electrical engineering department from Zhjiang University in China, respectively. In 2009 she received the Ph.D degree in the School of electrical engineering and automation from Tianjin University in China. And she works in the School of electrical engineering and automation from Tianjin University as an associate professor from 2003 to 2012 and a professor now. Her research interests include wind power, electrical technique, and intelligence control.

Penglong Wang received the B.Sc. degree in the Automation department from Tianjin University of Technology in China. From 2011 till now he studies as a student for the M.S. degree in the Tianjin University in China. His research interests include pattern recognition, information fusion and intelligence control.

Tingna Shi, Penglong Wang, Jeenshing Wang, and Shihong Yue

Jeenshing Wang received the B.S and M.S. degrees in electrical engineering from the University of Missouri, Columbia, in 1996 and 1997, respectively, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 2001. He is currently an Associate Professor with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. His research interests include computational intelligence, intelligent control, clustering analysis, and optimization.

Shihong Yue received the B.Sc. degree in the mathematics department from Yili Normal University, Xinjiang in China. And the M.S. and Ph.D degree in application mathematics from the Xi`An University of Technology and in the mathematics department from Xi`An Jiaotong University, in 1997 and 2000, respectively. From 2000 till 2004, he works in Institute of Industrial process control from Zhejiang University as a postdoctoral and received deputy professor in July 2001. As a visiting professor, he works in the department of computer science and Information engineering, Taiwan, in 2002. From 2004 till now, he works as a professor in the School of Electrical Engineering and Automation from Tinjin University. His research interests include pattern recognition, information fusion and intelligence control.

Received: January 26, 2012; Accepted: November 17, 2012.

Agent Negotiation on Resources with Nonlinear Utility Functions

Xiangrong Tong and Wei Zhang

School of Computer Science, Yantai University
Shandong 264005, China
txr@ytu.edu.cn

Abstract. To date, researches on agent multi-issue negotiation are mostly based on linear utility functions. However, the relationship between utilities and resources is usually saturated nonlinear. To this end, we expand linear utility functions to nonlinear cases according to the law of diminishing marginal utility. Furthermore, we propose a negotiation model on multiple divisible resources with two phases to realize Pareto optimal results. The computational complexity of the proposed algorithm is polynomial order. Experimental results show that the optimized efficiency of the proposed algorithm is distinctly higher than prior work.

Keywords: Nonlinear utility function, Multi-agent Systems, Multi-issue Negotiation, Resources Allocation, Incomplete Information.

1. Introduction

Allocation of multiple divisible resources under incomplete information is always an interesting problem where negotiation is one of essential approaches, such as the work of Luo et al [1], Saha et al [2], Chevaleyre et al [3], Fatima et al [5]. They proposed Pareto optimal negotiation, envy-free negotiation, and welfare optimal negotiation in their researches. In our past work [6–8], we also studied some problems of multi-issue negotiation.

However, prior researches mostly did not take explicit utility functions, or directly regarded resources as agents' utilities. Therefore, only linear utility functions were considered. However, there are much cases where utility functions are nonlinear. For example, \$100 will produce lots of utilities if an agent has not a bean, while \$100 will produce few utilities if that agent has a capital of a hundred million dollars. The utilities of autonomous agents are some kinds of subjective belief, and resources are objective things, i.e. resources generally have firing effect on utilities.

So the relationship between utilities and resources is usually saturated nonlinear just as indicated by Wooldridge [9]. Furthermore, the relationship should follow the law of diminishing marginal utility according to microeconomics, i.e. the increments of utilities decrease along with the increments of resources. Recently, some researches gradually paid more attention to nonlinear utility functions such as [11, 12, 21–24].

We try to expand linear utility functions to nonlinear cases in this paper. This work is motivated by the sigmoid function of artificial neuron of artificial neural network. In particular, we propose an improved nonlinear utility function to describe the relationship between utilities and resources. The proposed function not only follows the law of diminishing marginal utility, but also is consistent with the assumption of Wooldridge et al.

Sequentially, we bring forward a negotiation model on multiple divisible resources. In particular, we firstly take resources as indivisible units and reach a preliminary agreement using strict alternation in the first phase. Information about the preferences and firing rate of the opponent is also obtained at the same time. Subsequently, we divide resources using a greedy algorithm in the second phase and realize Pareto optimal results.

The first contribution of this paper is the proposal of a new utility function which follows the law of diminishing marginal utility while it is mostly ignored by previous work. Moreover, in the first phase of negotiation, strict alternation provides not only a preliminary agreement but also some useful information for agents. Therefore, history information is not necessarily required. Secondly, this model can generate Pareto optimal results for all negotiation agents. The computational complexity of the proposed algorithm is polynomial order and it is usually lower than that of Fatima et al. Experimental results show that the optimized efficiency of the proposed algorithm has an advantage over the work of Fatima et al. The application of this work includes many similar domains such as allocation of band width, allocation of computational resources, and allocation of business etc.

The rest of this paper is organized as follows. Section 2 describes the basic notions about negotiation scenes and utility functions. We propose a negotiation model and a feasible optimal algorithm in section 3. Section 4 discusses negotiation with complete information, and section 5 discusses negotiation under incomplete information. Experimental settings and results are provided in section 6. Background and related work is provided in section 7. Conclusions are drawn in the last section.

2. Basic Notions

In this section, we describe some basic notions about negotiation such as negotiation scenes, information states of agents, a nonlinear utility function, marginal utility function, and the ratio of marginal utility function. We also present some useful properties of the proposed nonlinear utility function.

2.1. Negotiation Scenes

We improve the negotiation scenes of Fatima et al [5] where agent a and b negotiate over m ($m \geq 2$) divisible resources (issues). We assume that the m resources are independent each other. We let agent a firstly make a bid for instance, and agent b is taken as the opponent.

Definition 1. Negotiation scenes

$$N = \langle a, b, R, m, n, \rho, t \rangle \quad (1)$$

where, N denotes a negotiation scene. a, b denote agents who are participating the negotiation. $R = \{r_1, r_2, \dots, r_m\}$ denotes divisible resources to be allocated. they also can be regarded as issues of negotiation. m denotes the number of resources to be allocated. n denotes the deadline of negotiation, namely the maximum turn of negotiation is n . $\rho = \{\rho_1, \rho_2, \dots, \rho_m\}$ denotes discount factors of resources ($0 \leq \rho_i \leq 1$). $t \in \mathcal{N}$ denotes the current turn of negotiation. \mathcal{N} denotes natural number.

Definition 2. Information state of agents

$$I_a = \langle k_a, x_a, C_b, P_b \rangle \quad (2)$$

Where, I_a is the information state of agent a . $k_a = \{k_a^1, k_a^2, \dots, k_a^m\}$ denotes resources/issues preferences of agent a . In which, k_a^i denotes the preference of agent a for r_i . We assume that $k_a^i \in [0, 1]$, $\sum_{i=1}^m k_a^i = 1$. In this paper, we assume that the preference of agent is invariable along with the quantity of resources allocated. It is a fixed factor for the agent initially. $x_a = \{x_a^1, x_a^2, \dots, x_a^m\}$ denotes the proportion of resources allocated to agent a . In which, $x_a^i \in [0, 1]$ denotes the proportion of resource r_i allocated to agent a . $C_b = \{C_b^1, C_b^2, \dots, C_b^l\}$ denotes l possible types of agent b known by agent a in advance. The type of agent b is known by agent a if negotiation is in complete information environment. $P_b = \{P_b^1, P_b^2, \dots, P_b^l\}$, in which, P_b^i denotes the probability of case that agent b is type C_b^i .

2.2. A Nonlinear Utility Function

There are various researches on nonlinear utility functions of negotiation recently. Some of researches focused on general nonlinear utility function and did not give practical nonlinear utility function, some of researches took Bell function such as [12]. Most of researches did not provide any rules to select nonlinear utility functions synthetically. To this end, we present these rules illustrated as follows.

Most models in economics and finances assume a non-decreasing utility function with diminishing marginal utility. While Arrow-Pratt's definition of absolute risk aversion is related to concave utility functions which implies decreasing marginal utility.

Rule 1. *The utility function should be separable, cumulative, and continuous. Such a function should satisfy the demand of negotiation on multiple divisible resources.*

Rule 2. *The utility function should be bounded. That is to say that the utility of agent would not increase infinitely.*

Rule 3. *The utility function should be monotonously nondecreasing. It means that agent hope to get more resources, no matter how much he has obtained.*

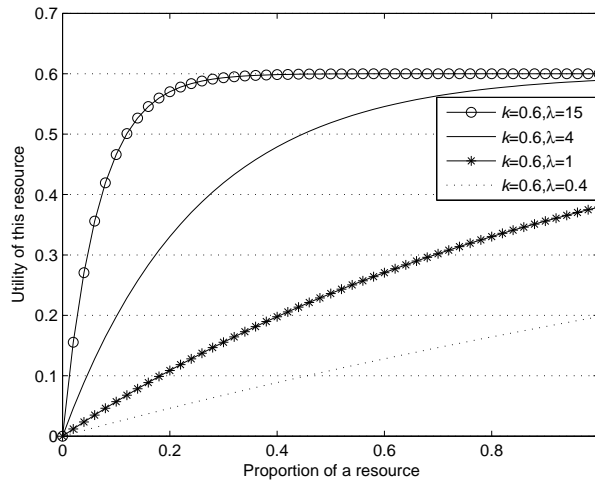


Fig. 1. Relationship between utilities and resources

Rule 4. *The utility function should be a concave function. It means that the increment of utility is monotonously non-increasing.*

Rule 5. *The utility function should follow the property of Arrow-Pratt absolute risk aversion. When an agent has more of one resource, he would confront more risk about this resource.*

Due to the above five rules of utility function, we are motivated by the sigmoid function of artificial neuron. The relationship between utilities and resources(money) indicated by Woodridge is absolutely suitable for sigmoid function. So we propose an improved nonlinear utility function through revising sigmoid function illustrated in Fig. 1.

Definition 3. *An improved nonlinear utility function*

$$u_a^i = k_a^i (1 - e^{-\lambda_a x_a^i}) \tag{3}$$

where, u_a^i denotes the nonlinear utility of agent a for resource r_i . $\lambda_a \in (0, \inf)$ denotes the firing rate of resources to the utilities of agent a . It shows the impact factor of resource to utility. For simplicity, we assume that λ_a is same to all resources of agent a .

The more λ_a is, the faster u_a^i increases. Especially, u_a^i is a step function when $\lambda_a \rightarrow \inf$. The utility of agent a can not obtain k_a^i even if all of r_i are allocated to it when $\lambda_a < 4$ because agent a can not be satisfied by the existing resource.

The proposed utility function is different from the prior functions. In particular, when $\lambda_a \leq 1$, the utility function is just approximately linear. Therefore, the prior work of Fatima et al is a particular one of our nonlinear utility function when λ is

small. So our proposed utility function is more general and common, including linear and nonlinear cases concurrently.

Definition 4. *The total utilities of agents*

$$U_a = \sum_{i=1}^m u_a^i (\rho_i)^{t-1} \quad (4)$$

U_a denotes the total utilities of agent a for all resources. t denotes the current turn of negotiation. u_a^i denotes the utilities of agent a for resource r_i which is defined in definition 3.

Definition 5. *Marginal utility function of agents in nonlinear domain*

$$\frac{du_a^i}{dx_a^i} = \frac{d(k_a^i(1 - e^{-\lambda_a x_a^i}))}{dx_a^i} = k_a^i \lambda_a e^{-\lambda_a x_a^i} \quad (5)$$

It is shown that the marginal utility in nonlinear domain is related to not only the preferences of agents, i.e. k_a^i , but also the firing rate of resources to the utilities of agent a , i.e. λ_a , as well as the quantity of resources allocated to agent a , i.e. x_a^i .

If we take linear utility function, $u_a^i = k_a^i x_a^i$, then the marginal utility is as follows.

Definition 6. *Marginal utility function of agents in linear domain*

$$\frac{du_a^i}{dx_a^i} = \frac{d(k_a^i x_a^i)}{dx_a^i} = k_a^i \quad (6)$$

So the marginal utility of linear function is k_a^i .

Definition 7. *The ratio of marginal utility function*

$$\theta_i = \frac{(u_b^i)'}{(u_a^i)'} \quad (7)$$

where, θ_i denotes the ratio of the marginal utility function of agent b to that of agent a .

If we use a greedy algorithm to generate counter-offer, agent a wishes to maximize its own utilities after satisfying the requirement of agent b 's utilities. It divides the m pies such that it reserves the maximum possible shares for those resources where θ_i is low, and gives to agent b the maximum possible shares for those resources where θ_i is high. Thus, agent a would begin by giving agent b the maximum possible shares for the resource with the highest θ_i . It then does the same for the resource with the next highest θ_i and repeat this process until utility requirement of agent b is satisfied. Therefore, the remainders of resources are allocated to agent a to maximize agent a 's utilities. In this way, not only the requirement of agent b 's utilities is quickly realized, but also the maximum utilities of agent a is obtained.

Assumption 1. *If the proportion of r_i allocated to agent a is x_a^i , then the proportion of r_i allocated to agent b is $(1 - x_a^i)$. The higher the value of θ_i is, the earlier the resource r_i is allocated to agent b .*

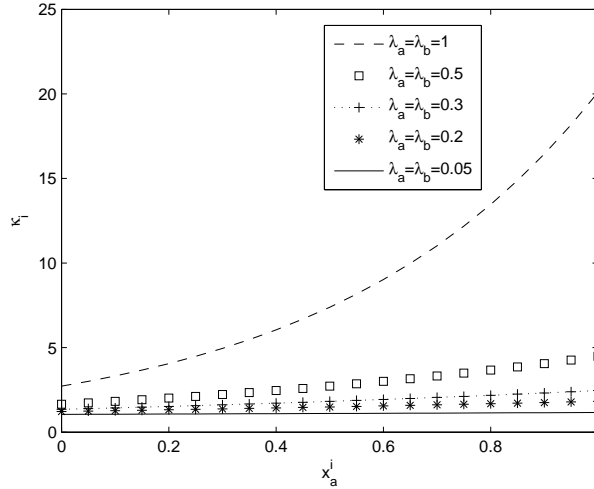


Fig. 2. The relationship among θ_i and x_a^i , λ_a , λ_b

$$\theta_i = \frac{k_b^i \lambda_b e^{-\lambda_b x_b^i}}{k_a^i \lambda_a e^{-\lambda_a x_a^i}} = \frac{k_b^i \lambda_b e^{-\lambda_b (1-x_a^i)}}{k_a^i \lambda_a e^{-\lambda_a x_a^i}} = \frac{k_b^i \lambda_b e^{-\lambda_b}}{k_a^i \lambda_a} e^{(\lambda_a + \lambda_b)x_a^i} \quad (8)$$

According to greedy algorithm, agent a should always choose the highest θ_i at point x_a^i , whenever k_b^i/k_a^i is. Particularly, the choice of resource is depended on x_a^i , as well as k_a^i , k_b^i , λ_a , λ_b . In a word, the only criteria is θ_i . **End.**

Theorem 1. The value of θ_i of nonlinear utility function is related to not only k_a^i , k_b^i , but also λ_a , λ_b and the quantity of resources allocated to agent a , i.e. x_a^i .

Proof: see formula (8). **End.**

The relationship among the value of θ_i and x_a^i , λ_a , λ_b is illustrated in Fig. 2. When $\lambda_a = \lambda_b \leq 0.3$, the value of θ_i is approximately invariable, no matter how much x_a^i is.

Theorem 2. The value of θ_i of linear utility function is $\frac{k_b^i}{k_a^i}$.

Proof: Due to $\frac{du_a^i}{dx_a^i} = \frac{d(k_a^i x_a^i)}{dx_a^i} = k_a^i$, $\theta_i = \frac{(u_b^i)'}{(u_a^i)'} = \frac{k_b^i}{k_a^i}$. **End.**

We can find that the work of Fatima where $\frac{k_b^i}{k_a^i}$ is adopted is just the rank of θ_i of linear utility function.

2.3. Properties of the Proposed Nonlinear Utility Function

The proposed utility function follows the following properties which are important for micro-economics. The utility function is bounded, is a concave function,

follows the law of diminishing marginal utility, and displays increasing absolute risk aversion according to Arrow-Pratt [10].

Property 1. $u_a^i \in [0, k_a^i)$ is bounded.

Proof: Known $\lambda_a > 0, x_a^i \geq 0$, u_a^i is minimal when $x_a^i = 0$, $u_a^i = k_a^i(1 - e^0) = 0$, while u_a^i is maximal when $x_a^i = 1$, $u_a^i = k_a^i(1 - e^{-\text{inf}}) \rightarrow k_a^i$ if $\lambda_a \rightarrow \text{inf}$. **End.**

Property 2. u_a^i follows the law of diminishing marginal utility.

Proof: Known $\lambda_a > 0, x_a^i \geq 0$, $(u_a^i)' = k_a^i \lambda_a e^{-\lambda_a x_a^i} \geq 0$. It shows that u_a^i is monotonously nondecreasing. Particularly, u_a^i is monotonously increasing if $k_a^i > 0 \wedge \lambda_a > 0$.

$(u_a^i)'' = (k_a^i \lambda_a e^{-\lambda_a x_a^i})' = -k_a^i (\lambda_a)^2 (1 - e^{-\lambda_a x_a^i}) \leq 0$ is monotonously non-increasing. Particularly, u_a^i is monotonously decreasing if $k_a^i > 0 \wedge \lambda_a > 0$. **End.**

Property 3. u_a^i is a concave function.

Proof: Known $\lambda_a > 0, x_a^i \geq 0$, $(u_a^i)' = k_a^i \lambda_a e^{-\lambda_a x_a^i} \geq 0$. $(u_a^i)'' = (k_a^i \lambda_a e^{-\lambda_a x_a^i})' = -k_a^i (\lambda_a)^2 (1 - e^{-\lambda_a x_a^i}) \leq 0$, so u_a^i is a concave function. **End.**

Property 4. u_a^i displays increasing absolute risk aversion.

Proof: The Arrow-Pratt measure is an attribute of a utility function. If we denote a utility function by $u(x)$. The Arrow-Pratt measure of absolute risk aversion is defined by:

$$\gamma(x) = -u''(x)/u'(x).$$

$$\text{So } \gamma(x_a^i) = \frac{k_a^i (\lambda_a)^2 (1 - e^{-\lambda_a x_a^i})}{k_a^i \lambda_a e^{-\lambda_a x_a^i}} = \lambda_a (e^{\lambda_a x_a^i} - 1). \text{ We know that } \gamma(x_a^i) \text{ also in-}$$

creases when x_a^i increases. It means that the risk aversion of agent a increases when x_a^i increases. Therefore, u_a^i displays increasing absolute risk aversion. Therefore, when agent a possesses lots of this resource, he prefers to reject rather than accept it. **End.**

3. Negotiation Model

In this section, we first discuss the proposed greedy algorithm for generating optimal proposals. Sequentially, we discuss the conditions of acceptance for offers.

3.1. Generating Optimal Proposals

Fatima et al [5] used a greedy algorithm to realize a Pareto optimal result. Just like knapsack problem, the greedy algorithm ranks k_b^i/k_a^i . Actually, agent a indeed ranks θ_i which is linear because $(u_a^i)' = (k_a^i x_a^i)' = k_a^i$. However, the utility function in this paper is nonlinear so that it may be difficult to choose resources.

We propose an improved greedy algorithm to obtain Pareto optimal solutions with nonlinear utility functions.

$$\begin{aligned} & \max \sum_{i=1}^m u_a^i \\ & \text{s.t. } \sum_{i=1}^m u_b^i = U_b^I \end{aligned} \quad (9)$$

where, U_b^I denotes the sum of agent b 's utilities that must be satisfied when agent a uses a greedy algorithm.

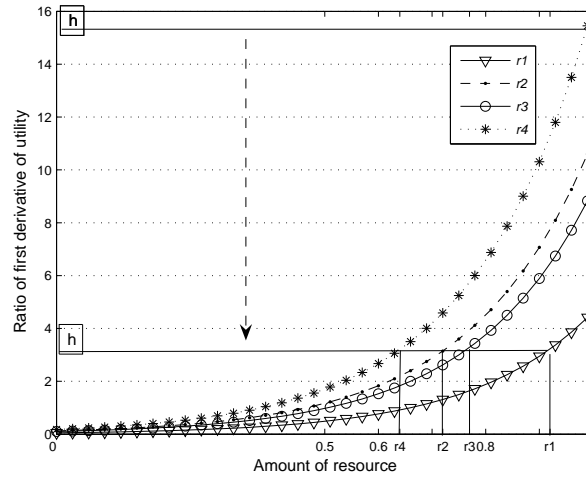


Fig. 3. The relationship between θ_i and x_a^i

According to the above explanation, the choice of resource is depended on θ_i . So we should always choose the biggest θ_i and try to satisfy the requirement of agent b 's utilities. Because the value of θ_i is depended on x_a^i , so the biggest θ_i will change among different resources.

Example 1. For example, if $m = 2, n = 2, k_a^1 = 0.4, k_a^2 = 0.6, k_b^1 = 0.5, k_b^2 = 0.5, \lambda_a = 2, \lambda_b = 3$, then when $x_a^1 = 0.7, x_a^2 = 0.4, \theta_1 = 3.09 > \theta_2 = 0.46$, when $x_a^1 = 0.4, x_a^2 = 0.8, \theta_1 = 0.69 < \theta_2 = 3.40$. So we find that the biggest θ_i changes from resource 1 to resource 2 when the quantity of resources changes. More examples are illustrated in table 1.

Table 1. The change of the biggest θ_i along with x_a^i

x_a^1	x_a^2	θ_1	θ_2	The biggest θ_i
0.7	0.4	3.09	0.46	θ_1
0.4	0.8	0.69	3.4	θ_2
0.4	0.2	0.69	0.17	θ_1

The proposed approach draws horizontal line in Fig. 3 from the top to the bottom. It means that the value of θ_i decreases from the biggest one until agent b is satisfied. The value of abscissa of the point of intersection with the line of θ_i is x_a^i , and $(1 - x_a^i)$ of the resource is allocated to agent b . Agent a occupies all the remainders of this resource, as long as agent b is satisfied. According to the greedy algorithm, agent a can obtain its Pareto optimal results.

The process of optimization is illustrated in algorithm 1, in which, ω is the amount of pieces of each resource divided by two agents, χ is the initial value of θ_i , ι is the step size of θ_i decreasing, u_b^I is the initial utility of agent b , δ is a toleration of θ_i 's error. $u_b^p(i)$ is the optimal utilities of agent b , while $u_a^p(i)$ is the optimal utilities of agent a , $r_a(i)$ is the the share of resource i allocated to agent a .

Algorithm 1

```

Set  $x = 0 : 0.01 : 1; s = (\lambda^b / \lambda^a)(k^b / k^a)e^{-b};$ 
1.  $u_b^p(i) := 0, (i = 1, \dots, m);$ 
2. while  $\sum_{i=1}^m u_b^p(i) < u_b^I$  //satisfying agent  $b$ 's utility
3.   for  $i = 1 : m$  //m resources
4.      $j := \omega + 1$  //each resource is divided into  $\omega$  pieces
5.     while  $abs(\theta_i(j) - \chi) > \delta$ 
6.        $j := j - 1;$ 
7.     end while
8.      $u_a^p(i) := u_a^i(j); u_b^p(i) := u_b^i(j); r_a(i) := (j - 1) / \omega;$ 
9.   end for
10.   $\chi := \chi - \iota$ 
11. end while
12.  $U_a^p := \sum_{i=1}^m u_a^p(i); U_b^p := \sum_{i=1}^m u_b^p(i);$ 

```

Theorem 3. *Pareto optimal results can be obtained by algorithm 1.*

Proof: Because algorithm 1 is based on the greedy algorithm, the utility of agent a is optimal without decreasing the utilities of agent b . According to the definition, algorithm 1 can realize Pareto optimal of agents. **End.**

Theorem 4. *The computational complexity of algorithm 1 is $O(\omega m \eta)$, $\eta = \chi / \iota$.*

Proof: Agent b will obtain all of resources in the worst case. Thus the inner nesting of 'While' cycle should be done ω times, 'For' cycle should be done m times, and the outer 'While' cycle should be done η times. Therefore, the computational complexity of algorithm 1 is $O(\omega m \eta)$. **End.**

Property 5. *The computational complexity of algorithm 1 is usually lower than that of Fatima et al.*

Proof: The computational complexity of the work of Fatima et al is $O(m\hat{\pi}l^3t(h-t/2))$, where $t = \min(2l - 1, h)$, h is the maximum turn of negotiation. Thus it is $O(m\hat{\pi}l^3(2l - 1)(h - l + 0.5))$ i.e. $O(m\hat{\pi}l^4h)$ when $h \geq (2l - 1) \rightarrow t = 2l - 1$. It is $O(m\hat{\pi}l^3h^2)$ when $h < (2l - 1) \rightarrow t = h$. The computational complexity of algorithm 1 is $O(\omega m \eta)$. If we control the size of ω and η , the computational complexity of algorithm 1 is usually lower than that of Fatima et al. **End.**

3.2. Conditions of Acceptance for Offers

There are two conditions of acceptance for offers. One is that the utility of the current offer that the offering agent received is not smaller than that of the next his own counter-offer. Another is that the offering agent can not optimize the

utility of own without decreasing the utility of the opponent, i.e. the current offer is a Pareto optimal offer.

Assumption 2. *Conditions of acceptance for offers*

Rule 6. *The utility of the current offer that the offering agent received is not smaller than that of the next his counter-offer.*

Rule 7. *The offering agent can not optimize the utility of own without decreasing the utility of the opponent.*

4. Negotiation with Complete Information

We assume that agent a and b know each other and the public information such as R, m, n, ρ is known by the two agents. We take the package deal procedure for this negotiation, then we can derive from the work of Fatima et al [5] that an agreement takes place at $t = 1$ for the package deal procedure, and the time taken to determine an equilibrium offer for $t = 1$ is $O(mn)$, where m is the number of resources and n is the deadline. We also know that the package deal procedure generates a Pareto optimal outcome.

We discuss the process of negotiation with complete information as follows. If negotiation reaches the deadline, then the agent whose turn it is takes everything and leaves nothing for its opponent. We then consider the preceding time periods ($t < n$), the offering agent will propose a counter-offer using greedy algorithm. Due to the complete information, each agent should consider the possibility of acceptance of the opponent. Furthermore, because of the discount of resources, each agent will not waste any time in order to avoid the loss of utility. Everyone knows that any more turn of negotiation means a loss of utility for both agents. So an agreement takes place at $t = 1$ for the package deal procedure.

So if agent b is the offering agent at $t = n$ turn, then agent b will take everything and leave nothing for agent a . It means that the utilities of agent b is $\rho^{n-1} \sum u_{b0}^i$, where $u_{b0}^i = k_b^i(1 - e^{-\lambda_b})$. At $t = n - 2$ turn, agent a uses greedy algorithm to generate optimal offer. This offer should maximize the utilities of agent a , as long as satisfying the requirement of agent b ' utility, i.e. $\rho^{n-1} \sum u_{b0}^i$. So we can obtain the followings.

$$\begin{aligned} \rho^{n-1} \sum u_{b0}^i &= \rho^{n-2} \sum u_b^i, \\ \Rightarrow \rho \sum u_{b0}^i &= \sum u_b^i, \\ \Rightarrow \rho \sum k_b^i(1 - e^{-\lambda_b}) &= \sum k_b^i(1 - e^{-\lambda_b x_b^i}), \\ \Rightarrow \rho(1 - e^{-\lambda_b}) &= \sum k_b^i(1 - e^{-\lambda_b x_b^i}), \end{aligned}$$

Now we can use a greedy algorithm to generate x_b^i, x_a^i . Because of complete information, agent b can not find a better outcome than this offer. So this offer is the offer at $t = 1$ turn for agent a .

If agent a is the offering agent at $t = n$ turn, then we can use a similar way to generate optimal offer at $t = 1$ turn.

Example 2. if $m = 2, n = 2, k_a^1 = 0.4, k_a^2 = 0.6, k_b^1 = 0.5, k_b^2 = 0.5, \lambda_a = 2, \lambda_b = 3, \rho_1 = \rho_2 = 0.5$, then

At $t = 2$ turn, agent b will take everything, then the utility of agent b is

$$\rho(k_b^1(1 - e^{-\lambda_b}) + k_b^2(1 - e^{-\lambda_b})) = 0.5 * (0.5 * (1 - e^{-3}) + 0.5 * (1 - e^{-3})) = 0.5 * (1 - e^{-3}) = 0.475.$$

Now we take greedy algorithm to compute x_b^i , where

$$U_b^I = \rho(\sum k_b^i(1 - e^{-\lambda_b})) = \rho(k_b^1(1 - e^{-\lambda_b}) + k_b^2(1 - e^{-\lambda_b})).$$

We find that $\rho(1 - e^{-\lambda_b}) = \sum k_b^i(1 - e^{-\lambda_b x_b^i})$,

$$\Rightarrow 0.5(1 - e^{-3}) = 0.5(1 - e^{-3x_b^1}) + 0.5(1 - e^{-3x_b^2}),$$

$$\Rightarrow e^{-3x_b^1} + e^{-3x_b^2} = 1 + e^{-3}, \text{ where } \theta_1 = \theta_2.$$

So we can find that $x_b^1 = 0.258, x_b^2 = 0.177$ using the proposed greedy algorithm. Here $\theta_1 = 3.81 = \theta_2, u_b^1 = 0.2694, u_b^2 = 0.206, u_a^1 = 0.31, u_a^2 = 0.48$. So $u_b^1 + u_b^2 = 0.475 = \rho(k_b^1(1 - e^{-\lambda_b}) + k_b^2(1 - e^{-\lambda_b}))$.

Table 2. The offer of negotiation with complete information

Turn	x_a^1	x_a^2	x_b^1	x_b^2	u_a^1	u_a^2	u_b^1	u_b^2	U_a	U_b	θ_1	θ_2
2	0	0	1	1	0	0	0.475	0.475	0	0.475	0.093	0.062
1	0.742	0.823	0.258	0.177	0.309	0.484	0.269	0.206	0.79	0.475	3.81	3.81

5. Negotiation under Incomplete Information

We propose a negotiation model with two phases. There are two aims in the first phase. We use the strict alteration [2] where resources are taken as indivisible units and two agents alter to choose one resource to reach a preliminary agreement. At the same time, we can obtain some useful information about the opponent's preferences reasoned by two rules derived from the strict alteration. Sequentially, we take resources as divisible units and make tradeoff between two agents to get a Pareto optimal result in the second phase.

5.1. The First Phase

We cite the approach of strict alteration [2] in the first phase. Each agent will always choose the resource with the biggest preference in the remainder resources.

First phase:

Step 1: A random device chooses one of the two agents and marks this agent as s . Denote the set of resources yet to be negotiated by G . Initially, $G = R$.

Step 2: Now, s will choose one of the remaining resources $r \in G$, r is allocated to s .

Step 3: Mark the other agent as s and update G to $G - \{r\}$. If $|G| \geq 1$ return to Step 2, otherwise stop.

If agent a is the first agent to choose resources and m is assumed to be even, then the following sequences are obtained. $x_a^1, x_a^2, \dots, x_a^{m/2}$ & $x_b^1, x_b^2, \dots, x_b^{m/2}$ (if m is odd, $x_a^{(m+1)/2}$ is added).

We can obtain a preliminary agreement in the first phase. Two agents allocate all resources where each resource is regarded as indivisible one. At the same time, this agreement is the base of the second phase where a greedy algorithm runs.

Through analyzing the above two sequences, agent a can obtain some useful information of agent b . Particularly, two rules of preferences of agent b can be obtained.

Agent a and b rank the resources with the preference. The resource with biggest preference of agent a is denoted by r_a^1 , the resource with the i th big preference of agent a is denoted by r_a^i . The same is done for agent b . So agent a and b can obtain the following sequence.

$$r_a^1, r_a^2, \dots, r_a^m \text{ and } x_b^1, x_b^2, \dots, x_b^m$$

Example 3. For example, we assume that agent a and b alter to choose 6 resources $\{r_1, r_2, r_3, r_4, r_5, r_6\}$. We assume the following 4 cases illustrated in table 3. Furthermore, the results of choices of agent a and b are illustrated in table 4.

Table 3. The rank of the resources

Cases	r_a^1	r_a^2	r_a^3	r_a^4	r_a^5	r_a^6	r_b^1	r_b^2	r_b^3	r_b^4	r_b^5	r_b^6
Case 1	r_1	r_2	r_3	r_4	r_5	r_6	r_6	r_5	r_4	r_3	r_2	r_1
Case 2	r_1	r_2	r_3	r_4	r_5	r_6	r_1	r_2	r_3	r_4	r_5	r_6
Case 3	r_1	r_2	r_3	r_4	r_5	r_6	r_2	r_1	r_3	r_4	r_5	r_6
Case 4	r_1	r_2	r_3	r_6	r_5	r_4	r_1	r_3	r_5	r_4	r_2	r_6

Table 4. The results of choices of agent a and b

Cases	x_a^1	x_a^2	x_a^3	x_b^1	x_b^2	x_b^3
Case 1	r_b^6	r_b^5	r_b^4	r_b^1	r_b^2	r_b^3
Case 2	r_b^1	r_b^3	r_b^5	r_b^2	r_b^4	r_b^6
Case 3	r_b^2	r_b^3	r_b^5	r_b^1	r_b^4	r_b^6
Case 4	r_b^1	r_b^5	r_b^6	r_b^2	r_b^3	r_b^4

Rule 8. x_b^i may be r_b^i to r_b^{2i} .

Rule 9. x_a^i may be r_b^i to x_b^m .

Agent a should always choose the resource remained with the biggest r_a^i , no matter whether it is that of agent b . whether agent a has chosen agent b 's expected resource is the key problem because agent a always be ahead of agent b and can affect agent b 's selection. Three cases exist according to the relationship between agent a and agent b .

If two agents do not conflict with each other anytime, i.e. agent a and b can always choose the expected resources. So x_b^i should always be r_b^i , and x_a^i may

be $r_b^{m/2}$ to x_b^m . Let's see case 1, x_b^1 is r_b^1 , x_b^2 is r_b^2 , and x_b^3 is r_b^3 . While x_a^1 is r_b^6 , x_a^2 is r_b^5 , x_a^3 is r_b^4 .

If two agents conflict with each other all along, i.e. agent b can not choose the expected resources all the while. What agent a chooses is the expected resource of agent b . It means that x_a^i should be r_b^i to r_b^{2i-1} . What agent b can choose is chosen by agent a . So agent b can only choose r_b^{2i} . So x_b^i should always be r_b^{2i} . Let's see case 2, x_a^1 is r_b^1 , x_a^2 is r_b^3 , x_a^3 is r_b^5 . So agent b can only choose the next resource, then x_b^1 is r_b^2 , x_b^2 is r_b^4 , and x_b^3 is r_b^6 .

If two agents partly conflict with each other, then x_b^i may be in $\{r_b^i \dots r_b^{2i}\}$, and x_a^i may be in $\{r_b^i \dots x_b^m\}$. Let's see case 4, x_a^1 is r_b^1 , x_a^2 is r_b^5 , x_a^3 is r_b^6 . So x_b^1 is r_b^2 , x_b^2 is r_b^3 , and x_b^3 is r_b^4 .

Example 4. Let's consider such a setting, agent a and b negotiate on four divisible resources $R = \{r_1, r_2, r_3, r_4\}$. Agent b is well known to be one of three possible classes of the opponents whose preferences and λ^b are shown in the followings.

Table 5. Information state of agents($P^b = \langle 0.5, 0.3, 0.2 \rangle$)

Items	r_1	r_2	r_3	r_4	λ	Rank of resources
k_a	0.4	0.25	0.15	0.2	2	$r_1 \succeq r_2 \succeq r_4 \succeq r_3$
k_b^{C1}	0.2	0.3	0.15	0.35	2.5	$r_4 \succeq r_2 \succeq r_1 \succeq r_3$
k_b^{C2}	0.4	0.25	0.15	0.2	2	$r_1 \succeq r_2 \succeq r_4 \succeq r_3$
k_b^{C3}	0.1	0.4	0.3	0.2	3	$r_2 \succeq r_3 \succeq r_4 \succeq r_1$

$k_a = \langle 0.4, 0.25, 0.15, 0.2 \rangle$, $\lambda_a = 2$; $k_b^{C1} = \langle 0.2, 0.3, 0.15, 0.35 \rangle$, $\lambda_b^{C1} = 2.5$;
 $k_b^{C2} = \langle 0.4, 0.25, 0.15, 0.2 \rangle$, $\lambda_b^{C2} = 2$; $k_b^{C3} = \langle 0.1, 0.4, 0.3, 0.2 \rangle$, $\lambda_b^{C3} = 3$;
 $P^b = \langle 0.5, 0.3, 0.2 \rangle$.

Where, $k_b^{C_i}$ denotes the preference vector when agent b is C_i class, $\lambda_b^{C_i}$ denotes the firing rate when agent b is C_i class.

If $\langle x_a^1 = r_1, x_b^1 = r_4, x_a^2 = r_2, x_b^2 = r_3 \rangle$ is obtained through the approach of strict alternation in the first phase. We can reason that only k_b^{C1} satisfies the fact that r_4 is ranked as the first or second order of agent b . Therefore agent b must be C_1 class. The approach is generally feasible when there are a few classes of agents.

Therefore, agent b obtains all of r_3 & r_4 in the first phase through the approach of strict alternation. The initial utilities of agent b are 0.45896 while those of agent a are 0.56203. **End.**

Alternatively, if agent b firstly chooses resources and $\langle x_b^1 = r_4, x_a^1 = r_1, x_b^2 = r_2, x_a^2 = r_3 \rangle$ is obtained. We also obtain that only k_b^{C1} satisfies the fact r_4 is ranked as the first order. Therefore agent b must be C_1 class. Agent b obtains all of r_4 & r_2 in the first phase of negotiation through the approach of strict alternation.

Based on the above rules, we can obtain some useful information which will help us in the next phase. If there is only one possible class, then agent b must

Table 6. Strict alteration of the first phase

The first agent to offer	x_b^1	x_a^1	x_b^2	x_a^2	$k_b^{C_i}$	r_a	r_b	U_a^I	U_b^I
<i>a</i>	r_1	r_4	r_2	r_3	$k_b^{C_1}$	$r_1 \& r_2$	$r_3 \& r_4$	0.56203	0.45896
<i>b</i>	r_4	r_1	r_2	r_3	$k_b^{C_1}$	$r_1 \& r_3$	$r_2 \& r_4$	0.47557	0.59664

be it; Otherwise, we can take advantage of probability of opponents to select the type with the maximal probability. So not only preliminary agreement but also preferences and firing rate of opponents can be obtained in the first phase.

5.2. The Second Phase

After phase 1, we can obtain a preliminary agreement and some useful information. Sequentially, we enter the second phase to obtain a Pareto optimal result.

Based on the agreement of phase 1, if agent *b*'s utility requirement is satisfied, then agent *b* should give this offer to agent *a*. Otherwise agent *b* should firstly satisfy its own requirement and give its new offer to agent *a*.

When agent *a* receives an offer from agent *b*, it should compute its own utility using algorithm 1. If agent *a*'s utility requirement can be satisfied by this offer, then agent *a* should send agent *b* the new offer. Because the new offer would satisfy agent *b*'s requirement, agent *b* should accept this new offer and negotiation succeeds. If agent *a*'s utility requirement can not be satisfied by this offer, then agent *a* would not consider the requirement of agent *b* and only satisfies its own utility requirement, then it provides agent *b* the new offer based on its own preferences and utility function.

When agent *b* receives an offer from agent *a*, it should compute its own utility. If agent *b*'s utility requirement can be satisfied by the counter-offer, agent *b* will accept the counter-offer. Negotiation succeeds. If it is not, then return to the beginning step of "Otherwise".

We give the following flow to understand the above operations.

Second phase:

Step 1: Set the result of the first phase as the initial allocation of resources. Choose one of the two agents and mark it as *a*, the other agent is marked as *b*.

Step 2: If agent *b*'s utility requirement can be satisfied in the first phase, then agent *b* gives the allocation of the first phase to agent *a*.

Step 3: Else agent *b* provides the offer based on its own preferences and utility function, where its utility requirement can be satisfied. end if

Step 4: agent *a* receives the offer from agent *b*, then to compute its utility using algorithm 1.

Step 5: If agent *a*'s utility requirement can be satisfied by this offer of algorithm 1, then agent *a* gives the optimal

counter-offer to agent b . Agent b should accept the counter-offer because its own utility requirement can be satisfied. Negotiation succeeds.

Step 6: Else agent a provides the offer based on its own preferences and utility function, where its utility requirement can be satisfied.

Step 7: If agent b 's utility requirement can be satisfied by the counter-offer, agent b will accept the counter-offer. Negotiation succeeds.

Step 8: Else go to step 3.

6. Experimental Evaluation

6.1. Optimal Performance of Algorithm 1

Recall example 4, agent a and b negotiate on 4 resources using this model. After the first phase, agent a obtained the following optimal results by algorithm 1.

Table 7. The results of experiments where agent a firstly chooses resources

Utility	agent a	agent b	x
r_1	0.33647	0.036254	0.92
r_2	0.19077	0.15102	0.72
r_3	0.11784	0.065594	0.77
r_4	0.14439	0.2077	0.64
Initial sum	0.56203	0.45896	1,1,0,0
Optimal sum	0.78948	0.46057	0.92,0.72,0.77,0.64

After optimizing in the second phase of negotiation, agent b obtains 0.92 proportion of r_1 , 0.72 proportion of r_2 , 0.77 proportion of r_3 , 0.64 proportion of r_4 . Sequentially, the optimal utility of agent b is 0.46057 and that of agent a is 0.78948 through the second phase. Experimental results show that the utilities of agent a significantly excel the preliminary agreement without decreasing the utilities of agent b . We practically improve the utilities of agent a by 40.5% (0.78948:0.56203) and that of agent b by 0.35% (0.46057:0.45896). The results of experiments are shown in Table 7.

Alternatively, if agent b firstly chooses resources. We improve the utilities of agent a by 53.8% (0.73155:0.47557) and that of agent b by 0.9% (0.60193:0.59664). Experimental results are shown in Table 8.

6.2. Influence of λ_a and λ_b

If we change the value of λ_a and λ_b from 0.2 to 20, we get the results illustrated in Fig. 5, where U_a^o denotes the optimized utilities of agent a . U_b^o denotes the

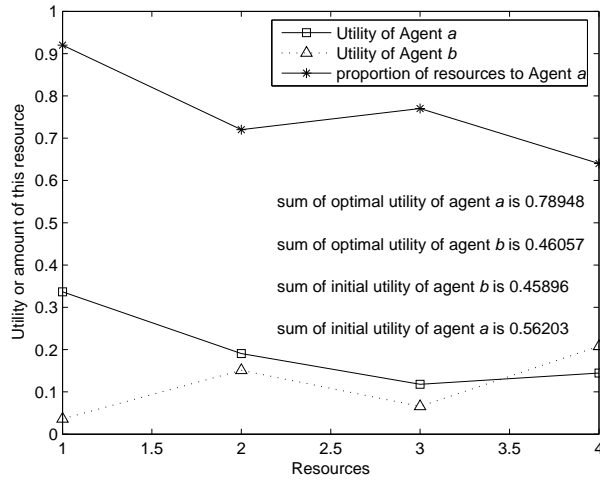


Fig. 4. Optimized results of algorithm 1

Table 8. The results of experiments where agent b firstly chooses resources

Utility	agent a	agent b	x
r_1	0.31924	0.078694	0.8
r_2	0.1747	0.18964	0.6
r_3	0.10829	0.089015	0.64
r_4	0.12931	0.24458	0.52
Initial sum	0.47557	0.59664	1,0,1,0
Optimal sum	0.73155	0.60193	0.8,0.6,0.64,0.52

optimized utilities of agent b. $U_{a\&b}^o$ denotes the optimized utilities of agent a and agent b.

We can find that when the value of $\lambda_a = \lambda_b$ is small, the effect of optimization is mainly on agent a. If $\lambda_a = \lambda_b < 4$, few effect is added on agent b. If $\lambda_a = \lambda_b > 5$, the utilities of agent a are nearly saturated. At the same time, the utility of agent b is optimized greatly. So the effect of optimization is perfect. However, if $\lambda_a = \lambda_b > 10$, the utilities of agent a and b are both nearly saturated. So the effect of optimization is not well.

6.3. Influence of Different δ_1 and δ_2

If we change δ_1 from 0.0001 to 0.05, the results of experiments are shown in Fig. 6. In which, a_i denotes the initial utilities (not optimized) of agent a, a_o denotes the optimized utilities of agent a, b_i denotes the initial utilities of agent a, b_o denotes the optimized utilities of agent b.

Agent Negotiation on Resources with Nonlinear Utility Functions

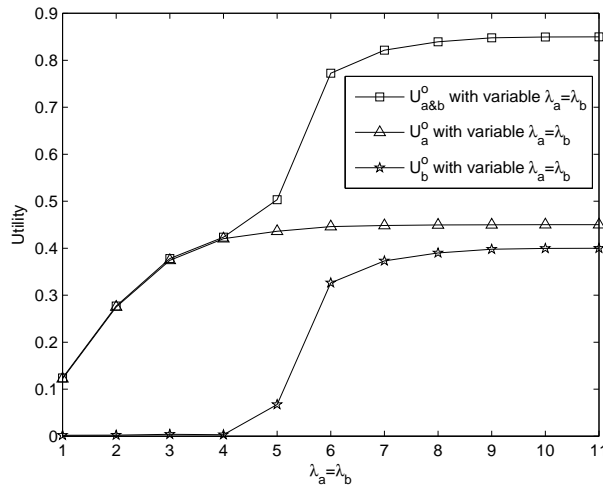


Fig. 5. Influence of different $\lambda_a = \lambda_b$ to the utilities of agent a and b

We can find that the optimal utilities of agent b becomes bigger, that of agent a becomes smaller, and the social welfare of optimal utility of agent a and b becomes also bigger.

6.4. Influence of Various Kinds of Agents

If various kinds of agents participate the negotiation, we can get the average optimized efficiency in a small range. We take 18 various kinds of agents and these results are illustrated in Fig. 7, where U_a^I denotes the initial utilities of agent a , U_a^o denotes the optimized utilities of agent a , U_b^I denotes the initial utilities of agent b , U_b^o denotes the optimized utilities of agent b .

We can find that the average utilities of agent a is improved by 59.56%. The average utilities of agent b is improved by 0.85%. The average utilities of agent a and b is improved by 28.95%.

The utilities of agent a and b is usually bigger than 1 because the parameter λ is enough big and the firing rate is enough high and it realizes the result of win-win.

6.5. Comparison With Linear Utility Functions

If we compare the optimized results of nonlinear utility functions with those of linear utility functions, we can find that the former is higher than the latter. The reason may be that the former considers not only the preferences of agents, but also the quantity of resources. Thus the former is more possible to get

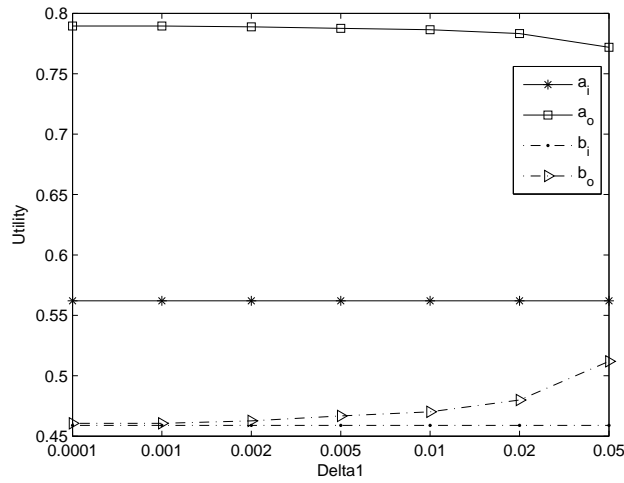


Fig. 6. Influence of different δ_1 to the utilities of agent a and b

more utilities than the latter. Table 8 and 9 show the comparison of optimized efficiency among the work of Fatima, Faratin and ours. We can discover the above conclusion from these tables. More results of comparison are illustrated in section 6.6.

Table 9. Comparison of the results of optimization between linear utility function and nonlinear utility function where agent a firstly chooses resources

Utility	agent a	agent b	Sum of agent a & b	Percent of optimization
Initial	0.56203	0.45896	1.02099	-
Optimization of Fatima	0.7092	0.45896	1.16816	14.41444
Optimization of Faratin	0.56203	0.635564	1.197594	17.29733
Optimization of ours	0.78948	0.46057	1.25005	22.43509

6.6. Comparison With the Work of Fatima

We compare experimental results of algorithm of Fatima et al and that of algorithm 1. The result is illustrated in Fig. 8, where U_a^I denotes the initial utilities of agent a , U_a^o denotes the optimized utilities of agent a using the algorithm of this paper, U_{aF}^I denotes the initial utilities of agent b using the algorithm of Fatima et al, U_{aF}^o denotes the optimized utilities of agent a using the algorithm of Fatima et al.

Agent Negotiation on Resources with Nonlinear Utility Functions

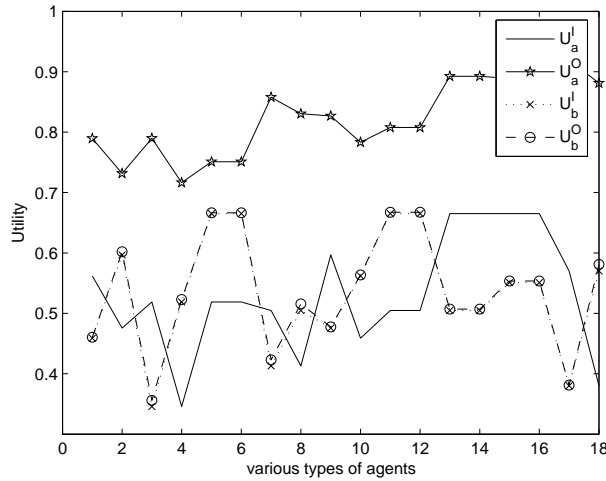


Fig. 7. Influence of various kinds of agents to the utilities of agent a and b

Table 10. Comparison of the result of optimization between linear utility function and nonlinear utility function where agent b firstly chooses resources

Utility	agent a	agent b	Sum of agent a & b	Percent of optimization
Initial	0.47557	0.59664	1.07221	-
Optimization of Fatima	0.594467	0.59664	1.191107	11.08893
Optimization of Faratin	0.47557	0.72443	1.2	11.91837
Optimization of ours	0.73155	0.60193	1.33348	24.36743

There are 18 cases if agent a and b are different kinds and alternatively firstly choose resources in the first phase. The average initial utilities of agent a in all 18 cases using algorithm 1 is 0.529661 and that of optimized utilities is 0.822555, therefore algorithm 1 improves utilities of agent a by 59.56%. While using methods of Fatima et al, these data are 0.580556, 0.583611, 0.55% respectively. Experimental results show algorithm 1 takes an advantage over the work of Fatima et al.

7. Background and Related Work

In recent years, researches on agent multi-issue negotiation with incomplete information attract more and more attention. Generally speaking, there are simultaneous procedure, sequential procedure, and package deal procedure approaches. However, only package deal procedure can ensure Pareto optimal solution indicated by Fatima et al [5]. Therefore, we only introduce the approach of package deal procedure to ensure Pareto optimal results in this paper.

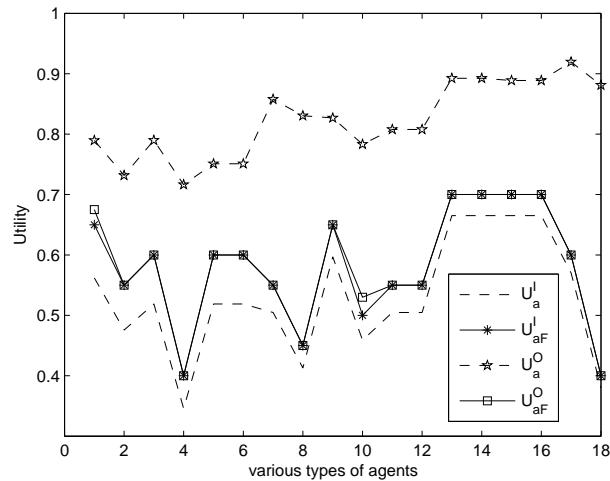


Fig. 8. Comparison with the work of Fatima in the utilities of agent a

Negotiation on multiple resources is always a challenging task. Dunne et al [15] study automatic contract negotiation in e-commerce and e-trading environments, they consider the computational complexity of a number of natural decision problems of one-resource-at-a-time to construct a mutually beneficial optimal reallocation by trading resources. An et al [13] present a proportional resource allocation mechanism and give a game theoretical analysis of the optimal strategies with incomplete information. Saha et al [2] present a protocol for negotiation on multiple indivisible resources which can be used by rational agents to reach efficient outcomes through searching negotiation tree. Chevaleyre et al [3] try to balance efficiency and fairness requirements to set up a distributed negotiation framework which will allow a group of agents to reach an allocation of goods that is both efficient and envy-free. Lin et al [18] conduct experiments on the allocation of circumstance of work and the World Health Organization Framework Convention on Tobacco Control. However, all of these existed researches have not given clear utility function, instead, utility value is given in advance or resources are directly taken as utilities. Unfortunately, there are many cases where the utility function is nonlinear. As well as Wooldridge [9] points out utility is not money but it is a useful analogy and he gives a typical relationship between utility and money like saturated nonlinear function.

A number of researches are dedicated to coping with incomplete information by learning preferences of opponents. Luo et al [1, 19] learns preferences of opponents based on knowledge models and through "default then adjust" approach. Lin et al [18] determine kinds of agents through offers in the process of negotiation based on the approach of naive Bayes class. The principle of

our work is similar with the approach of Lin et al that we determine types of opponents by reasoning from the information obtained in the first phase.

A Pareto optimal result is another important aim in multi-issue negotiation and some effective researches are done recently. Faratin et al [16] propose to maximize the utilities of opponents based on the same utilities of oneself to improve acceptable possibility of opponents. While Fatima et al [5] propose a similar approach to maximize the utilities of oneself based on the same utilities of opponent. The approach of our work is similar with that of Fatima et al but different from the definition and the treatment of utility functions.

Nonlinear utility functions were focused in the past years. Fatima et al [22] analyze bilateral multi-issue negotiation involving nonlinear utility functions. They show that it is possible to reach Pareto-optimal agreements by negotiating all the issues together, and that finding an equilibrium is not computationally easy if the agents' utility functions are nonlinear. They investigate two solutions: approximating nonlinear utility spaces with linear functions, and using a simultaneous procedure where the issues are discussed in parallel but independently of each other. They show that the equilibrium solution can be computed in polynomial time. However, their work is focused on symmetric negotiations where the agent's preferences are identically distributed, and the utility functions are separable in nonlinear polynomials of a single variable.

Vazirani et al [11] investigated some issues of computational complexity on market equilibrium under separable, piecewise-linear, concave utilities. They consider Fisher and Arrow-Debreu markets under additively separable, piecewise-linear, concave utility functions and obtain the following results. For both market models, if an equilibrium exists, there is one that is rational and can be written using polynomially many bits. There is no simple necessary and sufficient condition for the existence of an equilibrium. Under standard (mild) sufficient conditions, the problem of finding an exact equilibrium is in PPAD for both market models. Finally, they prove that under these sufficient conditions, finding an equilibrium for Fisher markets is PPAD-hard.

Complex nonmonotonic preference spaces were investigated by various researchers such as Ito et al. [21], Marsa-Maestre et al. [23,24], Lopez-Carmona et al. [12], where Lopez-Carmona et al. proposed a region-based automated multi-issue negotiation protocol (RBNP), which is built upon a recursive non-mediated bargaining mechanism. The non-monotonic negotiation scenarios are created using an aggregation of Bell functions. In contrast to prior researches, which usually assume that agents have relatively simple preferences on the issues (e.g., can be characterized by strictly concave utility functions), they make a more general assumption that the preference of each agent can be non-monotonic and non-differentiable.

8. Conclusions

This paper proposed an improved nonlinear utility function motivated by firing function of artificial neuron of artificial neural network. The proposed utility func-

tion follows the law of diminishing marginal utility. The negotiation model presented in this paper can deal with the instances of incomplete information, and Pareto optimal solutions will be obtained as a result. Experimental results show that the efficiency of optimal approach has an advantage over prior work.

The future work may improve the first phase of negotiation to reach a more rational preliminary agreement. A new optimal approach is also required to maximize the utilities of two agents, as well as maximizing the utilities of agent a .

Acknowledgments. The authors would like to thank School of Computer Science of Yantai University in China. In the laboratory of Intelligence Information Processing, the authors have received lots of enthusiasm and help without any personal motivations.

This work is partly supported by the Major State Basic Research Development Program of China under Grant (2007CB307100), the National Science Foundation of China (61170224, 60973075), the Shandong Natural Science Foundation (ZR2011FL018, 2012GGB01017, ZR2012FL07), and the Technology Research Program of Education Commission of Shandong Province (J11LG35, J10LG27).

References

1. X. Luo, N. R. Jennings, N. Shadbolt, *Acquiring user tradeoffs and preferences for negotiating agents: a default then adjust method*. International Journal of Human Computer Studies, (2006).
2. Sabyasachi Saha, Sandip Sen, *An Efficient Protocol for Negotiation over Multiple Indivisible Resources*. IJCAI'07, India, 1494-1499, (2007).
3. Yann Chevaleyre, Ulle Endriss, Sylvia Estivie, Nicolas Maudet, *Reaching Envy-free States in Distributed Negotiation Settings*. IJCAI'07, India, 1239-1244, (2007).
4. S. S. Fatima, M. Wooldridge, N. R. Jennings, *An agenda-based framework for multi-issue negotiation*. Artificial Intelligence, 152(1), pp. 1-45, (2004).
5. S. S. Fatima, M. Wooldridge, N. R. Jennings, *Multi-Issue Negotiation with Deadlines*. Journal of Artificial Intelligence Research (JAIR), 27, pp. 381-417, (2006).
6. Tong Xiangrong, Zhang wei. *Agent multi-issue negotiation with cases*. The 5th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'08), Jinan, 96-100, (2008).
7. Tong Xiangrong, Houkuan Huang, Zhang wei. *A Case based Agent Multi-issue Negotiation Model*. Journal of Researches and Development of Computer, 46(9): 1508-1514(in Chinese), (2009).
8. Tong Xiangrong, Guo Liang. *Learning opponent's indifference curve in multi-issue negotiation combining rule based reasoning with bayesian classifier*. The proceedings of 2010 sixth international conference on natural computation(ICNC'10), Yantai, 2916-2920, (2010).
9. Michael Wooldridge, *An introduction to multiAgent systems*. England: John Wiley Sons, Inc, (2002).
10. J.W. Pratt, *Risk Aversion in the Small and in the Large*, Econometrica 32:122-136, (1964).
11. Vijay V.Vazirani, Mihalis Yannakakis. *Market Equilibrium under Separable, Piecewise-linear, Concave Utilities*. Journal of ACM, 58(3):10, (2011).

12. Miguel A. Lopez-Carmona, Ivan Marsa-Maestre, Enrique De La Hoz, and Juan R. Velasco. *A Region-based Multi-issue Negotiation Protocol for Nonmonotonic Utility Spaces*. Computational Intelligence, 27(2): 166-217, (2011).
13. Bo An, Chunyan Miao, Zhiqi Shen, *Market Based Resource Allocation with Incomplete Information*. IJCAI'07, India, 1193-1198, (2007).
14. R. M. Coehoorn, N. R. Jennings, *Learning an opponent's preferences to make effective multi-issue negotiation tradeoffs*. Proc. 6th Int Conf. on E-Commerce, Delft, The Netherlands, 59-68, (2004).
15. Paul E. Dunne, Michael Wooldridge, Michael Laurence, *The complexity of contract negotiation*. Artificial Intelligence 164, pp. 23-46, (2005).
16. P. Faratin, C. Sierra, N. R. Jennings, *Using similarity criteria to make trade-offs in automated negotiations*. Artificial Intelligence 142 (2), pp. 205-237, (2002).
17. Sarit Kraus, Penina Hoz-Weiss, Jonathan Wilkenfeld, David R. Andersen Amy Pate, *Resolving crises through automated bilateral negotiations*. Artificial Intelligence, 172(1): 1-18, (2008).
18. R. Lin, S. Kraus, J. Wilkenfeld, J. Barry, *Negotiating with Bounded Rational Agents in Environments with Incomplete Information Using an Automated Agent*. Artificial Intelligence 172(6-7), pp. 823-851, (2008).
19. X. Luo, N. R. Jennings, N. Shadbolt, H. Leung, J. H. Lee, *A fuzzy constraint based model for bilateral multi-issue negotiations in semi-competitive environments*. Artificial Intelligence 148 (1-2), pp. 53-102, (2003).
20. Sarvapali D. Ramchurn, Carles Sierra, Lluis Godo, Nicholas R. Jennings, *Negotiating using rewards*. Artificial Intelligence 171, 805-837, (2007).
21. Takayuki Ito, Hiromitsu Hattori and Mark Klein, *Multi-issue Negotiation Protocol for Agents: Exploring Nonlinear Utility Spaces*. IJCAI'07, pp. 1347-1352, (2007).
22. Fatima, S., M. Wooldridge, and N. Jennings. *An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions*. In AAMAS'09: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1041-1048, (2009).
23. Marsa-Maestre, I., M.A. Lopez-Carmona, J.R. Velasco, and E. De La Hoz. *Effective bidding and deal identification for negotiations in highly nonlinear scenarios*. In Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS'09), Budapest, Hungary, 1057-1064, (2009).
24. Marsa-Maestre, I., M.A. Lopez-Carmona, J.R. Velasco, T. Ito, M. Klein, and K. Fujita. *Balancing utility and deal probability for auction-based negotiations in highly nonlinear utility spaces*. In 21st International Joint Conference on Artificial Intelligence (IJCAI'09), Pasadena, California, 214-219, (2009).

Xiangrong Tong, Ph. D. Associate Professor, Member of China Computer Federation, School of Computer Science, Yantai University, Yantai, China. His interests of researches include distribute artificial intelligence, multi-agent systems and data mining.

Wei Zhang, Professor, Senior Member of China Computer Federation, School of Computer Science, Yantai University, Yantai, China. His interests of researches include distribute artificial intelligence and multi-agent systems.

Xiangrong Tong and Wei Zhang

Houkuan Huang, Professor and Ph. D. supervisor, Senior member of China Computer Federation, Beijing Jiaotong University, Beijing, China. His interests of researches include artificial intelligence, data mining, and machine learning.

Received: March 6, 2012; Accepted: December 3, 2012.

University Campus Social Network System for Knowledge Sharing

Zhao Du¹, Xiaolong Fu¹, Can Zhao¹, Ting Liu¹, and Qifeng Liu¹

¹ Information Technology Center, Tsinghua University,
100084 Beijing, China
{dz, fxl, zc, liut, lqf}@cic.tsinghua.edu.cn

Abstract. Public online social network services have achieved dazzling success in recent years. As a result, vertical social network services for universities are expected warmly by campus users. As the majority of activities in university campus are knowledge and social interaction intensive, one of the core functions of campus social network system is to facilitate knowledge sharing on campus. In the cyberspace of universities, knowledge is stored in various kinds of digital resources such as documents, photos, videos etc. In this paper, we discuss the design and implementation of our campus social network system, concentrating on knowledge sharing mechanism in the system. The knowledge sharing mechanism has five features including the utilization of users' personal social network to facilitate the dissemination of digital resources, the use of a six-tuple model based tagging to realize the unified labeling for digital resources, the fine-grained access control based on friend lists for safer knowledge sharing, the adoption of a multi-scale evaluation method for digital resources and personalized recommendation for digital resource with social graph based collaborative filtering as its core idea. With all these considerations, we expect to improve the efficiency and effectiveness of knowledge while enlarging the dissemination scope of digital resources carrying it in cyberspace of universities.

Keywords: Vertical Social Network Services, Knowledge Sharing, Digital Resource, Social Tagging.

1. Introduction

Online social network services are considered one of the most popular network services in recent years, with billions of users spending huge amount of time every day on social network sites including Facebook, Twitter, Flickr, YouTube, LinkedIn in U.S. and Sina Weibo, RenRen, QZone in China etc. [4, 16, 20, 25, 28, 36, 38]. People use online social network services to share information and knowledge with their friends in real life, build online connections with new friends on network, follow celebrities and other network friends to get more and fresher news, and release their own information to

their followers or the public. These behaviors can help users to strengthen their weak ties effectively with only low cost [14], and weak ties are generally thought to be beneficial for creativity [29]. As the tremendous success that social network services have gained during the past few years, it's also regarded as the third revolutionary application of the Internet after search engine and Web2.0 applications.

Campus users of universities including students, faculty members and staff members are among the most active users of Internet services, especially emerging services including social networking services. As we all know, Facebook and RenRen, which are the biggest social network systems in the world and in China respectively, are both originated from prestigious American and Chinese universities. Although public social network services like Facebook, Twitter, Sina Weibo and RenRen provide good individual and group communication services to their users, they are isolated from universities' cyberspace of campus users. For this reason, public social network services have two disadvantages. On one hand, because it cannot be connected with the real environment and activities of universities' campus, it cannot support the learning, teaching, research and cultural activities on universities' campus closely and timely; on the other hand, public social network systems cannot provide strict information and privacy protection to their users and organizations that users are belonged to because of their consideration on commercial interest.

As it's known to us all, the fundamental responsibilities and core competitiveness of universities are the creation, dissemination and utilization of knowledge of all disciplines; and the majority of activities in universities are learning, teaching and research. While learning and teaching can be regarded as activities for knowledge sharing and collaborative knowledge building [19], research can be regarded as activities of knowledge creation. In a word, all these activities are knowledge and social interaction intensive. Research has shown that social network and social interaction within it are important to both knowledge creation and knowledge sharing. On one hand, social interaction positively influences the quality of the knowledge created [7]; on other hand, social network can help students to share experiences and collaborate on relevant topics [21].

Based on the above considerations, it's both necessary and useful to build campus social network system in the cyberspace of universities. The core function of campus social network system is to facilitate and encourage knowledge sharing and knowledge creation on campus. Specifically, the core functions of campus social network system are to support online communication, sharing and collaboration in learning, teaching and research activities of campus users. Knowledge sharing and knowledge creation are closely related with and continuously influenced by each other. Because campus social network system provides direct support to knowledge sharing and indirect support to knowledge creation, we put our emphasis on knowledge sharing in the following part of this paper.

In the following sections, we begin by the discussion on the definition and classification of social network system. Then we make detailed analysis of

the changes that campus social network system will bring to the knowledge sharing in the cyberspace of universities' through the comparison of digital resources dissemination without and with campus social network system. After that, we examine the design and implementation of our campus social network system which consists of fundamental components, core components, application services and open platform interfaces. Finally, we discuss the five features that are expected to stimulate and facilitate knowledge sharing in campus social network system.

2. Related Works

Social network system for organizations is a special category of social network system which accepts users within a single organization or multiple organizations. Different from public social network system which may use nickname or real name, social network system for organizations use real names and to provide higher level of trust [26]. Furthermore, social network system for organizations is more business-focused, the core responsibility of it is to help members of the organizations to achieve higher efficiency and better performance. Therefore, social network systems for different type of organizations may differ in content and styles of discussion [23, 30]. Generally speaking, large-scale organizations tend to set up independent social network systems which are usually customized to the organization's special requirement and integrated with other enterprise applications, small and medium-sized organizations tend to use shared social network systems which are similar in design and independent of other enterprise applications.

Enterprise social network system and campus social network system are two typical type of social network system for organizations. Because enterprises and universities have different mission, functions and business; enterprise social network system and campus social network system have different design concerns. For enterprise social network system, its content may be more work-related and cannot be open to non-employee of its belonging enterprise. For campus social network system its content may be more diverse and a considerable part of its content can be open to users who are outside of the belonging university. Furthermore, the access time of enterprise social network system may be more intensive in working hours, while the access time of campus social network system may be more scattered.

Many world-class companies have built up their own enterprise social network systems. Beehive is the enterprise social network system for IBM employees since 2006 [8, 9]. IBM Connections is the commercial software suite of new generation enterprise social network platform; it serves more than 400,000 employee of IBM in 175 countries and many employees in IBM's enterprise customers at present [26]. InnovationCafe is the enterprise social network system for NEC which mainly provides blog based social network services to employees. It's reported that InnovationCafe has

changed employees working styles and accelerated their communications beyond the barriers of different departments and organizations [12]. Yammer is one of the most popular corporate Twitter clones which is restricted to employees of the enterprise and served as online services [37]. It provides basic services for free and charges for additional services.

There are also many campus social network systems for universities. PhoenixConnect is an academic social network for the University of Phoenix in U.S. [18]. It serves more than 1,100,000 students, faculty member, and alumni of the university. Ewhaian is a social network system for Ewha Womans University in South Korea. It was launched since 2001 and has about 50% users who will log into the site on a daily basis. It's quiet successfully in terms of steadily increasing active users and engaging participation of current students [35]. Hotseat is a social networking tool in Purdue University that aims to connect students to each other and their instructors in these large lecture conditions. It also creates a backchannel of collaborative discussion both in and out of the classroom [1].

The university campus social network system we proposed in this paper is similar to the enterprise social network systems and campus social network systems we mentioned above in terms of the common features of social network systems. It's different from them in terms of target users, user relationships, digital resource sharing models, and focus. Firstly, it targets students, teachers, and staff members of our university. Secondly, because almost all students and many teachers and staff members live in the same campus, there are much more abundant relationships between them. Thirdly, we provide fine-grained access control for digital resources which allows users to share digital resources with more careful access control. By this way, we are expected to be able to decrease the possible conflicts in different social spheres of users. Finally, we put our focus on facilitating and encouraging knowledge sharing and knowledge creation on campus. We provide campus users with a simple and useful method to label digital resources, adopts multi-scale evaluation for digital resource and personalized recommendation for digital resources, and helps users to find more useful digital resources through personalized recommendation for digital resources.

3. Definition and Classification of Social Network Systems

Social network systems, also called social network sites or social network services, are defined as web-based services that allow users to construct a public or semi-public profile within a bounded system, articulate a list of other users whom they share a connection, and view and traverse their list of connections and those made by others within the system [6]. From the definition, we can conclude that the three core elements of social network systems are (1) user profile, (2) user relationship, and (3) the visibility of user profile, user relationship, user created information, user behavior etc. under

the privacy rules of the system and the privacy setting of users. User created information may include tweets, blogs, photos, videos etc. that users released and their value-add information such as comments, collections, recommendations etc. in social network system. User behavior is manifested by dynamic messages describing it in social network systems. The visibility of users profile, user relationship, user-created information, and user behavior can help to accelerate the socialization of individuals and enhance the interactions between them.

There are different ways to classify social network systems. According to the focus of social network systems, they can be classified as organic social network systems and hybrid social network systems. Organic social network systems are people-focused and embed social network features within. Hybrid social network systems are content-focused; they combine traditional Internet services and social network by integrating social features. For example, Twitter is an organic social network system and Flickr is a hybrid social network system [31]. According to users' features and purposes of using social network systems, we can classify social network systems as public social network systems and vertical social network systems. Public social network systems usually accept all Internet users as system's user, and allow them to do things like sharing, commenting, forwarding, keeping and recommending on tweets, photos, blogs etc. Most of the well-known social network systems including Facebook, Twitter, Weibo and RenRen are public social network systems. Vertical social network systems usually accept users with some common features or within the same organization. Enterprise social network system, campus social network system and other social network system for organizations all belong to vertical social network. Except for the functions that public social network systems provide, vertical social network systems also add some specific features for their special purposes. Because we have listed many examples of enterprise social network system and campus social network system in the previous section, next we will only introduce several examples of vertical social network system which accept users with some common features. For example, LinkedIn is a vertical social network system for professionals [28], Sermo is a vertical social network system for licensed physicians, INmobile is a vertical social network system the wireless industry [34], Instant Knowledge is an enterprise based social network that aims to introduce employees of the enterprise to contacts within the organization who may have skills relevant to particular tasks [33].

4. Knowledge Sharing in Campus Social Network System

As we have mentioned above, learning, teaching and research are the three basic categories of activities of universities' students, faculty members and staff members. One of the most important features of these activities is knowledge and social interaction intensive. Learning and teaching are mainly organized by means of courses, which can be regarded as activities for

knowledge dissemination and collaborative knowledge building; research is mainly organized by means of research projects, which can be regarded as activities of knowledge creation. In the cyberspace of universities, knowledge is stored in various kinds of digital resources such as documents, photos, videos etc. In order to illustrate the changes that campus social network system will bring to the knowledge sharing in the cyberspace of universities', we are going to make detailed analysis of the dissemination of digital resources without and with campus social network system using the dissemination of courseware as our example scenario.

4.1. Digital Resources Dissemination without Campus Social Network System

Digital resources dissemination without campus social network system is usually done through email or web sites for courses and projects. When digital resources are disseminated through email, only course members or project members who are in the mailing list of specific courses or projects can get related digital resources. When digital resources are disseminated through web sites, although everyone who has access to the web sites can get the digital resources theoretically, only few people except for course members or project members will know the existence of these web sites. In the latter case, as there is no notification for the changes of digital resources, users should check also the web sites from time to time to get the up-to-date digital resources. This is both time consuming and boring. As a result, the dissemination scope of most knowledge will usually be in very limited scope. Furthermore, because courses and most projects have a relatively short lifecycle, such as one semester or one year, the effectiveness of knowledge will be influenced by its limited time of existence.

Figure 1 depicts the scenario of courseware dissemination through learning management system and email. In learning management system, the courseware created by teacher A can only be accessed by the students in his/her classes. Students can download the courseware from the course space in learning management system after the teacher release it. Besides, if his/her colleges know about his/her course and want to get its courseware, they may ask him/her for the courseware, and the teacher can send it as attachments of the email. After the end of the course, course members will seldom access the courseware, leaving it in a state of existence with rare attention. In other words, the lifecycle of courseware is usually the same as the course. The courseware in the mailbox of his/her colleges will always be in similar status in most cases. Maybe the receiver will delete it or let it stay in his mailbox and forget its existence after short scan. In both situations, the courseware will be disseminated in a limited scope and likely to be useless in the long term.

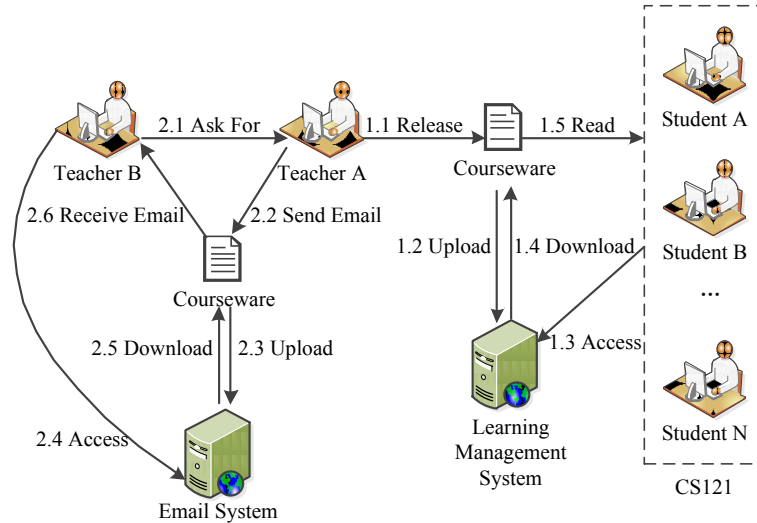


Fig. 1. Digital Resource Dissemination without Campus Social Network System

4.2. Digital Resources Dissemination with Campus Social Network System

Every user in campus social network system has many contacts. These contacts may be his/her friends in Facebook and RenRen, mutual followed user in Twitter and Weibo, or group members in all of the above. Collections of the contacts form the personal social network of a user. With campus social network system, dissemination of digital resources is greatly different. Firstly, the dissemination of knowledge will be in a larger scope and much more timely. All users in the personal social network of the user who released the digital resources will receive them immediately. This can not only reduce the cost of getting digital resources and save time for receivers, but also achieve better user experience. Secondly, the effectiveness of knowledge will be enhanced because the digital resources carrying knowledge can exist actively for much longer time than the lifecycle of a course or a project.

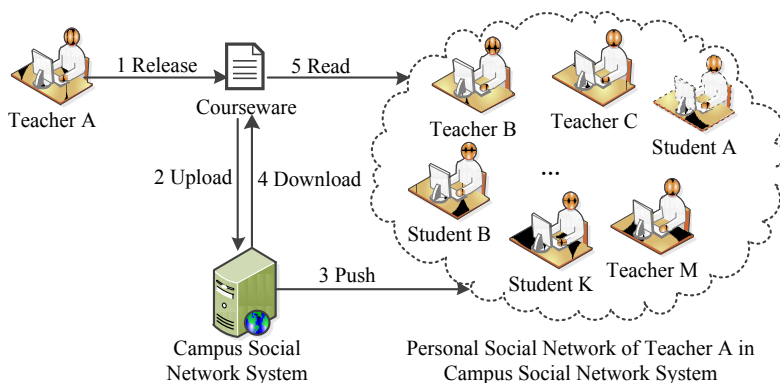


Fig. 2. Digital Resource Dissemination with Campus Social Network System

Figure 2 depicts the scenario of courseware dissemination with campus social network system. After teacher A release a courseware in campus social network system, every user in his/her personal social network will receive a short message from the system immediately saying that he/she has just released a courseware by a special mechanism named news feed which will reduce users' cost of accessing information from their contacts greatly [17, 32]. If someone is interested in the courseware, he/she can download it to his/her computer or just read it online. After that, he/she may write some comments on the courseware and forward it to the users in his/her own personal social network. This process can be repeated many times within only a few minutes, making the courseware be read by hundreds and thousands of users.

The dissemination scope of digital resources in campus social network system is decided by several factors including the influence of the user, the number and influence of his/her contacts, the usefulness of the digital resources, and even the time of release etc. By this way, digital resources may exist actively for longer time than they are in learning management system or users' mailbox which will make it possible to be much more useful.

5. Design of Campus Social Network System

Campus social network system is a typical vertical social network system. Firstly, it only provides services to campus users of one or more universities. Secondly, its key value is to facilitate knowledge sharing on campus and its core functions are to support the daily communication and campus activities including learning, teaching and research of campus users. Thirdly, it's an important and indispensable part of universities' cyberspace which has close connection with other parts of it. Finally, it adopts a well-designed information and privacy protection mechanism for campus users and universities.

University Campus Social Network System for Knowledge Sharing

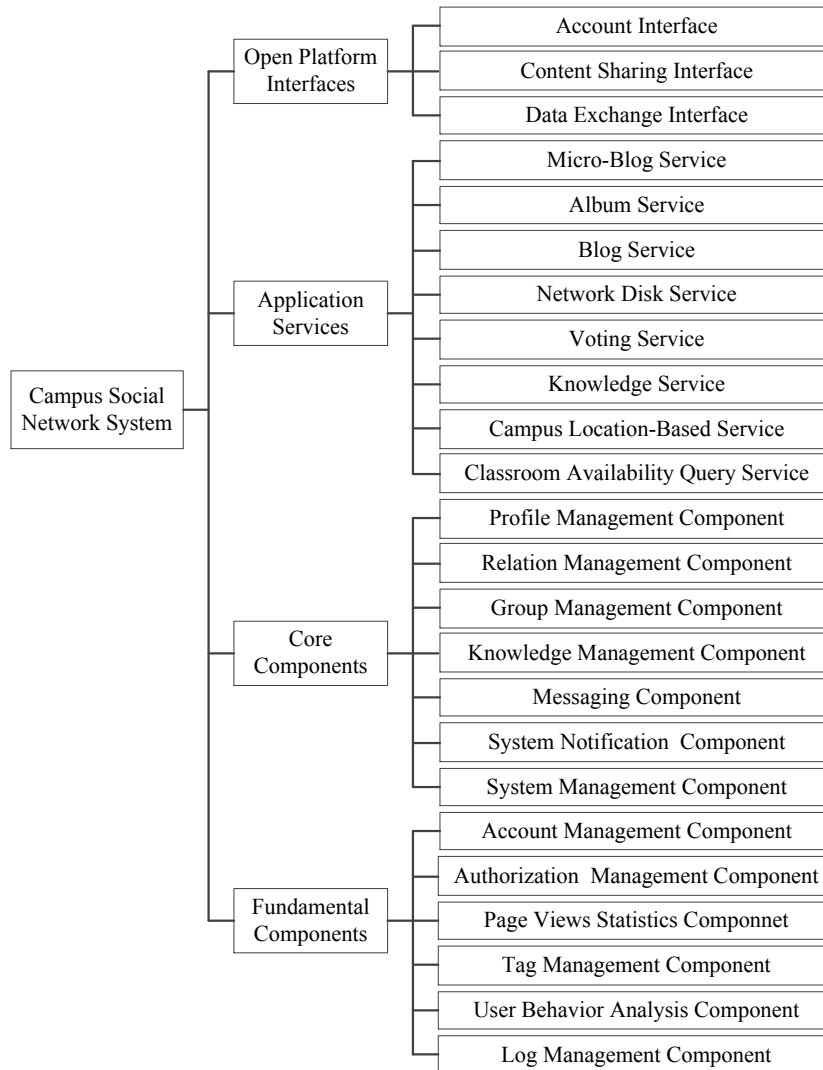


Fig. 3. Architecture of Campus Social Network System

Campus social network system consists of four parts: fundamental components, core components, application services and open platform services, as shown in figure 3.

5.1. Fundamental Components

Fundamental components are common components of information systems which are similar to campus social network system. These components not

only deal with the system's accounts, authorization, tags and logs; but also keep tracking of users' page views and fulfill basic user behavior analysis in system.

5.2. Core Components

Core components contain the three components that manage core elements of social network system: user profiles, user relationship and groups; the key component of campus social network system: knowledge management; and other three components which are closely related to them: messaging component, system notification management and system management component.

5.3. Application Services

Because users usually spend most of their time on application services when they are active in campus social network system, they are the key to achieve good user experience of the system. In campus social network system, application services can be divided into two categories. The first category of application services contain common services including micro-blog service, album service, blog service, network disk service and voting service that most of the current social network systems have. The second category of application services contains specialized services including knowledge management service, campus location-based service and classroom availability query service that are designed specifically for campus users.

5.4. Open Platform Interfaces

There are two purposes to implement open platform interfaces for campus social network system. The first purpose is to connect campus social network system with other parts of universities' cyberspace. As we mentioned above, campus social network system is an important part of universities' cyberspace. Therefore, it must exchange information and coordinate with other parts of universities' cyberspace. The second purpose is to allow people who are outside of R&D team of campus social network system to develop application services independently and freely. These interfaces are designed to address the problems of account sharing, content sharing and data exchange.

6. Knowledge Sharing Mechanism in Campus Social Network System

Knowledge sharing in computer systems is not a new research topic in computer science field. The main purpose of knowledge sharing is to improve knowledge reusability to make them more valuable. A lot of research and engineering efforts have been spent on the design and implementation of knowledge-based systems using ontologies [15]. By this way, knowledge can be understood and used by computer systems. With the popular of web services, the application of ontologies is extended to the research of semantic web [24].

We use a social network based method to facilitate the sharing of knowledge in campus social network system. The knowledge sharing mechanism in campus social network system has the following five key points. Firstly, personal social network of users provide the basic channel for knowledge sharing in campus social network system. Secondly, unified labeling for digital resources not only provides users with a simple and useful method to label digital resources, but also makes it possible to build another channel for knowledge sharing. Thirdly, fine-grained access control for digital resources allows users to share the digital resources with more careful access control. Fourthly, multi-scale evaluation for digital resource and personalized recommendation for digital resources make users can find more useful and high quality digital resources with less effort. Finally, personalized recommendation for digital resources is expected to help users to find more useful digital resources.

6.1. Personal Social Network of Users

The personal social networks of users are the most special and valuable data of campus social network system. The collection of these personal social networks can be described using a huge graph that contains countless informal social networks. These informal social networks are recognized as important and unique relationships among scientific and technical personnel [3]. As a result, we regard personal social networks of users as the basic channel for knowledge sharing. With network relationships and news feed, digital resources released by campus users can not only be able to be disseminated rapidly or even explosively in a short time, but also be able to be disseminated in a wider scope than it will do in tradition ways.

6.2. Unified Labeling for Digital Resources

Tagging is a simple and useful method to label digital resources. People use tags to describe digital resources for sharing and future use with minor efforts using keywords [22]. When used in social applications like social network

system, tagging is known as social tagging. We take unified tagging strategy for digital resources in campus social network systems. Different from the traditional triple model, we use a six-tuple to describe each tag:

Tagging (Object, Tag, Tagger, Source, Type, Time)

In the six-tuple, "Object" is the digital resource to be tagged; "Tag" is the keyword used to describe to the digital resource; "Tagger" is the user who tagged the digital resource; "Source" is the domain that the digital resource belongs to; "Type" refers to the type of digital resource, such as document, photo, music, video, web address, user, group, activity etc.; and "Time" is the time of tagging [11].

When a considerable part of digital resources in social network system have been tagged by large amounts of users, we can build another channel for knowledge sharing through campus users' collaborative tagging for digital resources [10]. The channel provides a new chance for users to discover and access digital resources according to their content features.

6.3. Fine-Grained Access Control for Digital Resources

In most social network systems, access control of contents and digital resources is coarse-grained. Specifically, when a user wants to publish a message such as a micro-blog, he will usually be allowed to choose whether to publish it publicly or publish it to all of his friends. Some social network systems also allow users to publish it privately. People need to maintain independent social spheres in real life. That is to say, they usually play different role, build connections and communicate with different people, and join different groups in their daily life. As a result, the technological features of news feed based on the entire friend list in most current social network services may make their social spheres in conflict and bring online tension to them [5].

In order to achieve better and more natural user experience, we adopt a fine-grained access control design for digital resources in our campus social network system. The fine-grained access control is based on friend categories of users. With this design, users can choose to publish digital resources to specified categories of friends. Furthermore, digital resources for specified categories of friends cannot be shared again by users who received it. By this way, we make online knowledge sharing more safely and comfortably, and this are expected to stimulate users' behavior of knowledge sharing in campus social network system.

6.4. Multi-Scale Evaluation for Digital Resources

We adopt a multi-scale evaluation method for digital resources in campus social network system. This method combines objective evaluation criteria with subjective evaluation criteria to get comprehensive evaluation

information for digital resources. The objective evaluation criteria include number of visits, comments and collections, reflecting the popularity of digital resources; the subjective evaluation criteria include 10-point scale score and number of sharing, reflecting the quality of the digital resources.

Objective evaluation information and subject evaluation information for digital resources from campus users form collective knowledge of the digital resources which will be helpful to users in making their decision [27]. For example, when a user wants to find lecture notes about data mining, he/she can just browse lecture notes that have the highest score or the maximum clicks. In a word, multi-scale evaluation for digital resource helps users to find high quality digital resources with just minor effort.

6.5. Personalized Recommendation for Digital Resources

Personalized recommendation is an important feature for most e-commerce web sites such as Amazon, TaoBao etc. and many other web based services including Google, YouTube, Flickr etc. The objects of recommendation may be books and any other commodities in e-commerce web sites; or virtual objects such as web sites, videos, photos, and blogs etc. in many web based services. Collaborative filtering is recognized as one of the most useful and widely used recommendation algorithms [13, 2].

To help users to find more useful digital resources, we incorporate a personalized recommendation module for digital resources in our campus social network system. The module uses collaborative filtering as the core idea of recommendation. Moreover, it takes advantage of the social graph in the system. Instead of generalize and rank recommendation result totally based on the history of users' behavior, we add the strength of users' relationship into consideration. The digital resources accessed by closest friends with the most similar interest will get the highest ranking.

7. Summary

As a typical vertical social network system for campus users of universities, campus social network system is a newcomer to the cyberspace of universities. We implemented a campus social network system for our university. The system consists of four parts: fundamental components, core components, application services and open platform interfaces. Firstly, like most social networking systems, campus social network system maintains personal social networks of campus users which can be described using a huge social graph that contains countless informal social networks. Utilizing the informal social networks by news feed in campus social network system will enlarge the dissemination scope of digital resources where knowledge is stored in. Secondly, through the use of unified labeling and the adoption of multi-scale evaluation method for digital resources, users can evaluate the

Zhao Du, Xiaolong Fu, Can Zhao, Ting Liu, and Qifeng Liu

digital resources and add new value to them. The collection of tags and evaluations of all users will not only produce the collaborative tagging, classification, and ranking etc. of digital resources; but also provide a new chance for users to discover and access digital resources according to their content features and quality. Furthermore, fine-grained access control for digital resources using friend lists are expect to achieve better and more natural user experience, which will stimulate users' behavior of knowledge sharing. Finally, personalized recommendation using collaborative filtering as its core idea can help users to find more useful digital resources. Our future work concerns empirical study of user behaviors for knowledge sharing, improvement to the performance of the system and design of more suitable evaluation method for digital resources.

Acknowledgment. We thank Zhenyu Zhou for his efforts in implementing many parts of the system; Dongxing Jiang for his important suggestions at the milestones of the system; Yueyue Zhu and Juan Du for their continuous support from the very early stage of this research work; Yu Zhang, Qixin Liu and Chun Yu in the system integration work of campus social network system with existing applications in digital campus of our university. Finally, we would like to thank all the anonymous reviewers for all their comments and suggestions. This work is supported by the National Basic Research Program of China (No. 2012CB316000) and the Beijing Education and Science "Twelfth Five-Year Plan" (No. CJA12134).

References

1. Aagard, H. P., Bowen, K., Olesova L. A.: Hotseat: Opening the Backchannel in Large Lectures. *EDUCAUSE Quarterly*, vol. 33, no. 3 (2010). [Online]. Available: <http://www.educause.edu/ero/article/hotseat-opening-backchannel-large-lectures> (Current July 2012)
2. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, 734-749. (2005)
3. Allen, J., James A. D., Gamlen, P.: Formal versus Informal Knowledge Networks in R&D: a Case Study Using Social Network Analysis. *R&D Management*, Vol. 37, No. 3, 179-196. (2007)
4. Benevenuto, F., Duarte, F., Rodrigues T., Almeida, V. A. F., Almeida, J. M., Ross, K. W.: Understanding Video Interactions in YouTube. *Proceedings of the 16th ACM International Conference on Multimedia*. ACM, Vancouver, Canada, 2008: 761-764.
5. Binder, J., Howes, A., Sutcliffe, A.: The Problem of Conflicting Social Spheres: Effects the Network Structure on Experienced Tension in Social Network. *Proceedings of the 27th international conference on Human factors in computing systems*. ACM, Boston, USA, 965-974. (2009)
6. Boyd, D. M., Ellison, N. B.: Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication*, Vol. 13, No. 1, 210-230. (1997)

7. Chua, A.: The Influence of Social Interaction on Knowledge Creation. *Journal of Intellectual Capital*, Vol. 3, No. 4, 375-392. (2002)
8. DiMicco, J., Millen, D. R., Geyer W., Dugan, C., Brownholtz, B., Muller, M.: Motivations for Social Networking at Work. *Proceedings of the 2008 ACM conference on Computer Supported Cooperative Work*, ACM, Savannah, Georgia, USA, 712-720. (2008)
9. DiMicco, J. M., Geyer, W., Millen, D. R., Dugan, C., Brownholtz, B.: People Sensemaking and Relationship Building on an Enterprise Social Network Site. *Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS'09)*, IEEE, Big Island, HI, USA, 1-10. (2009)
10. Ding, P.Y., Huang, T.Q., Liu, W.: The State of the Art in Tag Ontologies: A Semantic Model for Tagging and Folksonomies. *New Technology of Library and Information Service*, No. 3, 30-37. (2009)
11. DU, Z., ZHAO, C., FU X.: Research and Implementation of Social Tagging System for Campus Social Network Services. To appear in *Computer Engineering and Design*, Vol. 33, No. 8. (2012)
12. FUKUOKA, H.: "InnovationCafe", an In-house Social Network Service (SNS) Used in NEC. *NEC Technical Journal*, Vol.2, No.2, 63-66. (2007)
13. Goldberg, D., Nichols, D., Oki, B. M., Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry. *Communications of ACM*, Vol. 35, No. 12, 61-70. (1992)
14. Granovetter, M. S.: The Strength of Weak Ties. *American Journal of Psychology*, Vol. 78, No. 6, 1360-1380. (1973)
15. Gruber, T. R.: Toward Principles for the Design of ontologies used for Knowledge Sharing. *International Journal of Human Computer Studies*, Vol. 43, 907-928. (1995)
16. Java, A., Song, X., Finin, T., Tseng, B.: Why we Twitter: Understanding Microblogging Usage and Communities. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web Mining and Social Network analysis*, ACM, San Jose, CA, USA, 56-65. (2007)
17. Jun, S., Ahamad, M.: FeedEx: Collaborative Exchange of News Feeds. *Proceedings of the 15th International Conference on World Wide Web*, ACM, Edinburgh, Scotland, UK, 113-122. (2006)
18. King, G., Sen, M.: The Troubled Future of Colleges and Universities. [Online]. Available: https://gking.harvard.edu/files/distruptu_0.pdf (Current July 2012)
19. Lee, S. W. Y.: The Interplay between Self-Directed Learning and Social Interactions: Collaborative Knowledge Building in Online Problem-Based Discussions. *Proceedings of the 7th International Conference on Learning Sciences*, International Society of the Learning Sciences, Bloomington, IN, USA, 390-396. (2006)
20. Li, L.: Social Network Sites Comparison between the United States and China: Case Study on Facebook and Renren Network. *Proceedings of 2011 International Conference on Business Management and Electronic Information (BEMI)*, IEEE, Guangzhou, China, 825-827. (2011)
21. Lincardi, I., Ounnas, A., Pau R., Massey, E., Kinnunen, P., Lewthwaite, S., Midy, M. A., Sarkar, C.: The Role of Social Networks in Students' Learning Experience. *ACM SIGCSE Bulletin*, Vol. 39, No. 4, 224-237. (2007)
22. Marlow, C., Naaman, M., Boyd, D., Davis, M.: HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, to Read. *Proceedings of the 17th Conference on Hypertext and Hypermedia*, ACM, Odense, Denmark, 31-39. (2006)

23. McDermott, R.: Learning across teams: The role of communities of practice in team organizations. *Knowledge Management Review*, Vol.3, No. 8, 32-36. (1999)
24. Meadche, A., Staab, S.: Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, Vol. 16, No. 2, 73-79. (2001)
25. Mislove, A., Koppula, H. S., Gummadi, K. P., Druschel, P., Bhattacharjee, B.: Growth of the Flickr Social Network. *Proceedings of the First Workshop on Online Social Networks*. ACM, Seattle, WA, USA, 25-30. (2008)
26. Muller, M., Ehrlich, K., Matthews, T., Perer, A., Ronen†, I., Guy, I.: Diversity among Enterprise Online Communities: Collaborating, Teaming, and Innovating through Social Media. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Seattle, WA, USA, 2815-2824. (2012)
27. Murakata, K., Matsuo, T.: Computational Collective Intelligence in Electronic Commerce: Incentive Design in Evaluation System. *Proceedings of 2011 International Conference on Systems, Man, and Cybernetics*, IEEE, Anchorage, Alaska, USA, 948-953. (2011)
28. Papacharissi, Z.: The Virtual Geographies of Social Networks: A Comparative analysis of Facebook, LinkedIn and ASmallWorld. *New Media & Society*, Vol. 11, No. 1&2, 199-220. (2009)
29. Perry-Smith, J. E., Shalley, C. E.: The Social Side of Creativity: A Static and Dynamic Social Network Perspective. *Academy of Management Review*, Vol. 28, No. 6, 89-106. (2003)
30. Powell, A., Piccoli, G., Ives, B.: Virtual teams: A review of current literature and directions for future research, *DATABASE Adv. Info. Sys.* Vol. 35, No. 1, 6-36. (2004)
31. Quan, H.: Online Social Networks & Social Network Services: A Technical Survey. *Pervasive Communication Handbook*. CRC, Chapter III.4. (2011)
32. Sun, E., Rosenn, I., Marlow, C. A., Lento, T. M.: Gesundheit! Modeling Contagion through Facebook News Feed. *Proceedings of the Third International ICWSM Conference*, AAAI, San Jose, California, USA, 146-153. (2009)
33. Tomlinson, A., Yau, P. W., MacDonald J. A.: Privacy Threats in a Mobile Enterprise Social Network. *Information Security Technical Report*, Vol. 15, No. 2, 57-66. (2010)
34. Vascellaro, J. E.: Social Networking Goes Professional: Doctors, Salesmen, Executives Turn to New Sites to Consult, Commiserate With Peers ; Weeding Out Impostors. *Wall Street Journal*, August 28. (2007)
35. Wohm, D. Y.: Sustainability of a College Social Network Site: Role of Autonomy, Engagement, and Relatedness. *Proceedings of the 2012 ACM annual conference on Human Factors in Computer Systems*. ACM, Austin, Texas, USA, 737-740. (2012)
36. Yu, L., Asur, S., Huberman, B.A.: What Trends in Chinese Social Media. *Proceedings of the 5th SNA-KDD Workshop'11 (SNA-KDD'11)*, ACM, San Diego, CA, USA. (2011)
37. Zhang, J., Qu, Y., Code J., Wu, Y.: A case study of micro-blogging in the enterprise: use, value, and related issues. *Proceedings of the 28th international conference on Human factors in computing systems*, ACM, Atlanta, GA, USA,123-132. (2010)
38. Zhong, E., Fan, W., Wang, J., Xiao, L., Li, Y.: ComSoc: Adaptive Transfer of User Behaviors over Composite Social Network. To appear in *Proceedings of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Minings*, Beijing, China. (2012)

Zhao Du is the vice director of R&D Division at Information Technology Center, Tsinghua University. She received her B.Sc. degree in Computer Science & Technology from Northern JiaoTong University in 2002 and Master Degree in Computer Science & Technology from Tsinghua University in 2005. Her research interests include social network analysis, online learning, and software usability.

Xiaolong Fu is the assistant director of Information Technology Center, Tsinghua University. He received his B.Sc. degree in Computer Science & Technology from Tsinghua University in 2000 and Master Degree in Management Science and Engineer from Tsinghua in 2003. His research interests include information system and software architecture.

Can Zhao is an engineer at Information Technology Center, Tsinghua University. She receives her B.Sc. degree in Electrical Engineering and Automation from Nanjing University of Aeronautics and Astronautics in 2004 and Master Degree in Traffic Information Engineering and Control from Nanjing University of Aeronautics and Astronautics in 2007. Her research interests include information system and software testing.

Ting Liu is an engineer at Information Technology Center, Tsinghua University. She receives her B.Sc. degree in Information Security from Beijing Information Science and Technology University in 2010. Her research interests include information system.

Qifeng Liu is an engineer at Information Technology Center, Tsinghua University. He receives her B.Sc. degree in Computer Science and Technology from Beijing Institute of Technology in 2007. His research interests include information system.

Received: February 15, 2012; Accepted: November 22, 2012.

CIP – Каталогizacija u publikaciji
Народна библиотека Србије, Београд

004

COMPUTER Science and Information
Systems : the International journal /
Editor-in-Chief **Mirjana Ivanović**. – Vol. 9,
No 4 (2012) - . – Novi Sad (**Trg D. Obradovića**
3): ComSIS Consortium, 2012 - (Belgrade
: Sgra star). –30 cm

Polugodišnje. – Tekst na engleskom jeziku

ISSN 1820-0214 = Computer Science and
Information Systems
COBISS.SR-ID 112261644

Cover design: **V. Štavljanin**

Printed by: Sgra star, Belgrade

ComSIS Vol. 9, No. 4, Special Issue, December 2012



Contents

Editorial

Papers

- 1361 SMP-SIM: an SMP-based Discrete-event Execution-driven Performance Simulator
Yufei Lin, Xinhai Xu, Yuhua Tang, Xin Zhang, Xiaowei Guo
- 1385 Orthogonal Sequences Based Multi-CFO Estimation and Semi-Blind ICA Based Equalization for Multiuser CoMP Systems
Yufei Jiang, Xu Zhu, Enggee Lim, Yi Huang
- 1407 Speech Unit Category based Short Utterance Speaker Recognition
Nakhat Fatima, Xiaojun Wu, Thomas Fang Zheng
- 1431 Formal Verification of Signature-monitoring Mechanisms by Model Checking
Lanfeng Tan, Qingping Tan, Jianjun Xu, Huiping Zhou
- 1453 An Uncertain Optimal Control Model with n Jumps and Application
Liubao Deng, Yuanguo Zhu
- 1469 A Formal Approach to Testing Programs in Practice
Shaoying Liu, Wuwei Shen, Shin Nakajima
- 1493 Image Denoising Using Anisotropic Second and Fourth Order Diffusions Based on Gradient Vector Convolution
Huaibin Wang, Yuanquan Wang, Wengi Ren
- 1513 Active Semi-supervised Framework with Data Editing
Wangxin Xiao, Xue Zhang
- 1533 An Ant System based on Moderate Search for TSP
Ping Guo, and Zhujin Liu
- 1553 Optimal Node Placement of Industrial Wireless Sensor Networks Based on Adaptive Mutation Probability Binary Particle Swarm Optimization Algorithm
Ling Wang, Wei Ye, Haikuan Wang, Xiping Fu, Minrui Fei, Muhammad Ilyas Menhas
- 1577 Automatic T-S fuzzy Model with Application to Designing Predictive Controller
Zhi-gang Su, Pei-hong Wang, Yu-fei Zhang
- 1603 Superior Performance of Using Hyperbolic Sine Activation Functions in ZNN Illustrated via Time-Varying Matrix Square Roots Finding
Yunong Zhang, Long Jin, Zhende Ke
- 1627 Clustering based Two-Stage Text Classification Requiring Minimal Training Data
Xue Zhang, Wangxin Xiao
- 1645 A Novel Content Based Image Retrieval System using K-means/KNN with Feature Extraction
Ray-I Chang, Shu-Yu Lin, Jan-Ming Ho, Chi-Wen Fann, Yu-Chun Wang
- 1663 Study on Prediction Models for Integrated Scheduling in Semiconductor Manufacturing Lines
Lu Guo, Jinghua Hao, Min Liu
- 1679 Application of Grid-based K-means Clustering Algorithm for Optimal Image Processing
Tingna Shi, Jeenshing Wang, Penglong Wang, Shihong Yue
- 1697 Agent Negotiation on Resources with Nonlinear Utility Functions
Xiangrong Tong, Wei Zhang
- 1721 University Campus Social Network System for Knowledge Sharing
Zhao Du, Xiaolong Fu, Can Zhao, Ting Liu, Qifeng Liu
-