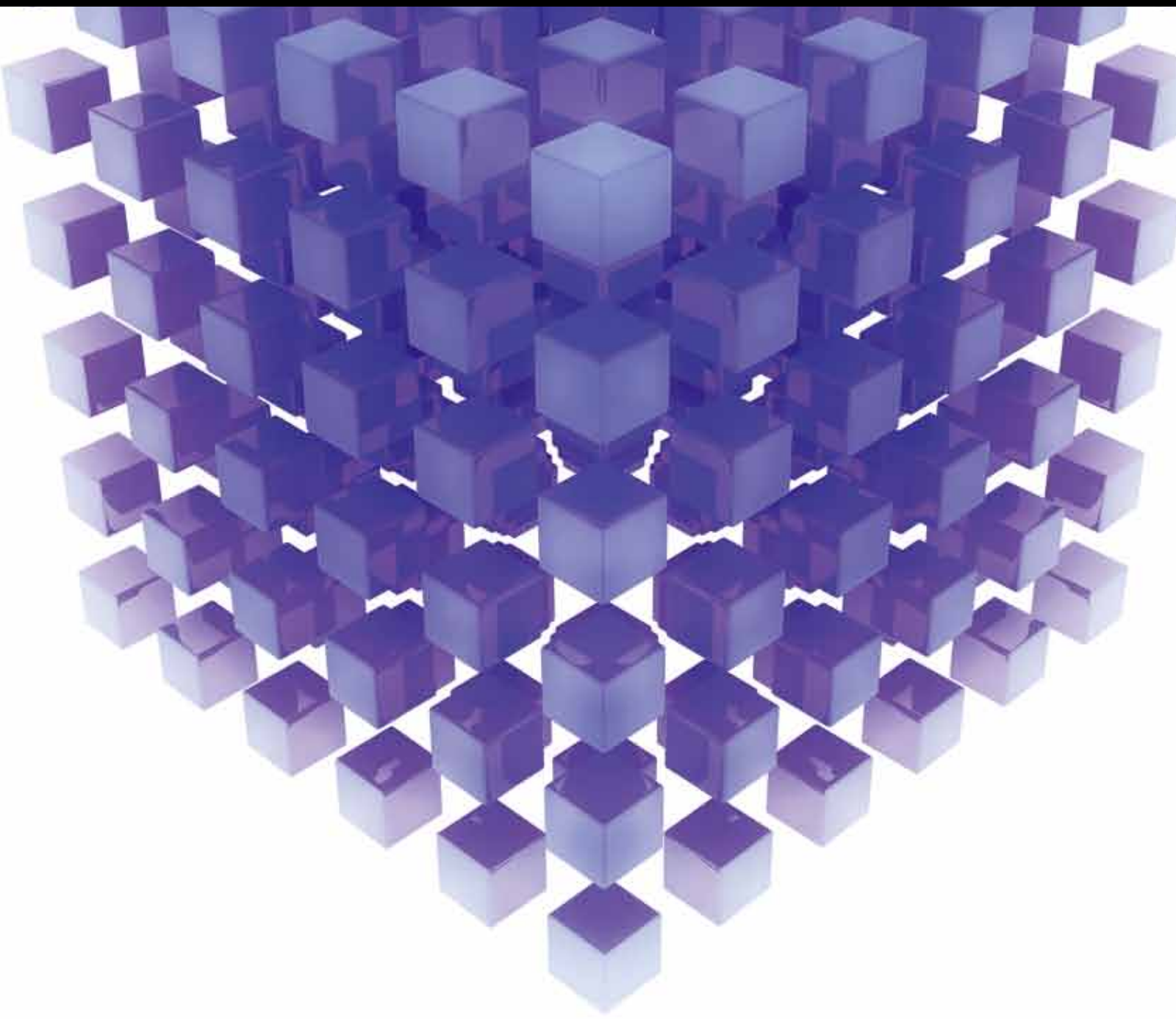
Mathematical Problems in Engineering

Theory and Applications of Complex Networks

Guest Editors: Tingwen Huang, Zhichun Yang, and Chuandong Li



Theory and Applications of Complex Networks

Mathematical Problems in Engineering

Theory and Applications of Complex Networks

Guest Editors: Tingwen Huang, Zhichun Yang,
and Chuandong Li

Copyright © 2014 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in "Mathematical Problems in Engineering." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

M. Abd El Aziz, Egypt
E. M. Abdel-Rahman, Canada
R. K. Abu Al-Rub, USA
Sarp Adali, South Africa
Salvatore Alfonzetti, Italy
Igor Andrianov, Germany
Sebastian Anita, Romania
W. Assawinchaichote, Thailand
Erwei Bai, USA
Ezzat G. Bakhoum, USA
José M. Balthazar, Brazil
R. K. Bera, India
C. Bérenguer, France
Jonathan N. Blakely, USA
Stefano Boccaletti, Spain
Stephane P.A. Bordas, USA
Daniela Boso, Italy
M. Boutayeb, France
Michael J. Brennan, UK
Salvatore Caddemi, Italy
Piermarco Cannarsa, Italy
Jose E. Capilla, Spain
Carlo Cattani, Italy
Marcelo M. Cavalcanti, Brazil
Diego J. Celentano, Chile
Mohammed Chadli, France
Arindam Chakraborty, USA
Yong-Kui Chang, China
Michael J. Chappell, UK
Kui Fu Chen, China
Xinkai Chen, Japan
Kue-Hong Chen, Taiwan
Jyh-Horng Chou, Taiwan
Slim Choura, Tunisia
Cesar Cruz-Hernandez, Mexico
Swagatam Das, India
Filippo de Monte, Italy
Antonio Desimone, Italy
Yannis Dimakopoulos, Greece
Baocang Ding, China
Joao B. R. Do Val, Brazil
Daoyi Dong, Australia
B. Dubey, India
Horst Ecker, Austria
M. Onder Efe, Turkey
Elmetwally Elabbasy, Egypt
A. Elías-Zúñiga, Mexico
Anders Eriksson, Sweden
Vedat S. Erturk, Turkey
Moez Feki, Tunisia
Ricardo Femat, Mexico
Robertt A. Valente, Portugal
C. Fuerte-Esquivel, Mexico
Zoran Gajic, USA
Ugo Galvanetto, Italy
Furong Gao, Hong Kong
Xin-Lin Gao, USA
Behrouz Gatmiri, Iran
Oleg V. Gendelman, Israel
Didier Georges, France
Paulo B. Gonçalves, Brazil
Oded Gottlieb, Israel
Fabrizio Greco, Italy
Quang Phuc Ha, Australia
M. R. Hajj, USA
Tony S. W. Hann, Taiwan
Thomas Hanne, Switzerland
K. R. (Stevanovic) Hedrih, Serbia
M.I. Herreros, Spain
Wei-Chiang Hong, Taiwan
Jaromir Horacek, Czech Republic
Huabing Huang, China
Chuangxia Huang, China
Gordon Huang, Canada
Yi Feng Hung, Taiwan
Hai-Feng Huo, China
Asier Ibeas, Spain
Anuar Ishak, Malaysia
Reza Jazar, Australia
Zhijian Ji, China
Jun Jiang, China
J. J. Judice, Portugal
Tadeusz Kaczorek, Poland
Tamas Kalmar-Nagy, USA
Tomasz Kapitaniak, Poland
Hamid Reza Karimi, Norway
Metin O. Kaya, Turkey
Nikolaos Kazantzis, USA
Farzad Khani, Iran
K. Krabbenhoft, Australia
Ren-Jieh Kuo, Taiwan
Jurgen Kurths, Germany
Claude Lamarque, France
Usik Lee, Korea
Marek Lefik, Poland
Stefano Lenci, Italy
Roman Lewandowski, Poland
Shanling Li, Canada
Ming Li, China
Jian Li, China
Shihua Li, China
Teh-Lu Liao, Taiwan
Panos Liatsis, UK
Shueei M. Lin, Taiwan
Yi-Kuei Lin, Taiwan
Jui-Sheng Lin, Taiwan
Yuji Liu, China
Wanquan Liu, Australia
Bin Liu, Australia
Paolo Lonetti, Italy
V. C. Loukopoulos, Greece
Junguo Lu, China
Chien-Yu Lu, Taiwan
Alexei Mailybaev, Brazil
Manoranjan K. Maiti, India
O. D. Makinde, South Africa
R. Martinez-Guerra, Mexico
Driss Mehdi, France
Roderick Melnik, Canada
Xinzhu Meng, China
Yuri V. Mikhlin, Ukraine
G. Milovanovic, Serbia
Ebrahim Momoniat, South Africa
Trung Nguyen Thoi, Vietnam
Hung Nguyen-Xuan, Vietnam
Ben T. Nohara, Japan
Sotiris K. Ntouyas, Greece
Gerard Olivar, Colombia
Claudio Padra, Argentina
Bijaya Ketan Panigrahi, India
Francesco Pellicano, Italy
Matjaz Perc, Slovenia
Vu Ngoc Phat, Vietnam
M. do Rosário Pinho, Portugal
A. Pogromsky, The Netherlands

Seppo Pohjolainen, Finland
Stanislav Potapenko, Canada
Sergio Preidikman, USA
Carsten Proppe, Germany
Hector Puebla, Mexico
Justo Puerto, Spain
Dane Quinn, USA
K. R. Rajagopal, USA
Gianluca Ranzi, Australia
Sivaguru Ravindran, USA
G. Rega, Italy
Pedro Ribeiro, Portugal
J. Rodellar, Spain
R. Rodriguez-Lopez, Spain
A. J. Rodriguez-Luis, Spain
Ignacio Romero, Spain
Hamid Ronagh, Australia
Carla Roque, Portugal
Rubén R. García, Spain
Manouchehr Salehi, Iran
Miguel A. Sanjuán, Spain
Ilmar F. Santos, Denmark
Nickolas S. Sapidis, Greece
E. J. Sapountzakis, Greece
Bozidar Sarler, Slovenia
Andrey V. Savkin, Australia
Massimo Scalia, Italy
Mohamed A. Seddeek, Egypt
A. P. Seyranian, Russia
Leonid Shaikhet, Ukraine
Cheng Shao, China
Bo Shen, Germany
Jian-Jun Shu, Singapore
Zhan Shu, UK
Dan Simon, USA
Luciano Simoni, Italy
Grigori M. Sisoev, UK
Christos H. Skiadas, Greece
Davide Spinello, Canada
Sri Sridharan, USA
Rolf Stenberg, Finland
Changyin Sun, China
Jitao Sun, China
Xi-Ming Sun, China
Andrzej Swierniak, Poland
Yang Tang, Germany
Allen Tannenbaum, USA
Cristian Toma, Romania
Irina N. Trendafilova, UK
Alberto Trevisani, Italy
Jung-Fa Tsai, Taiwan
K. Vajravelu, USA
Victoria Vampa, Argentina
Josep Vehi, Spain
Stefano Vidoli, Italy
Xiaojun Wang, China
Dan Wang, China
Youqing Wang, China
Yongqi Wang, Germany
Cheng C. Wang, Taiwan
Moran Wang, China
Yijing Wang, China
Gerhard-Wilhelm Weber, Turkey
J. A. S. Witteveen, The Netherlands
Kwok-Wo Wong, Hong Kong
Ligang Wu, China
Zhengguang Wu, China
Gongnan Xie, China
Wang Xing-yuan, China
Xi Frank Xu, USA
Xuping Xu, USA
Jun-Juh Yan, Taiwan
Xing-Gang Yan, UK
Suh-Yuh Yang, Taiwan
Mahmoud T. Yassen, Egypt
Mohammad I. Younis, USA
Bo Yu, China
Huang Yuan, Germany
S.P. Yung, Hong Kong
Ion Zaballa, Spain
Ashraf M. Zenkour, Saudi Arabia
Jianming Zhan, China
Xu Zhang, China
Yingwei Zhang, China
Lu Zhen, China
Liancun Zheng, China
Jian Guo Zhou, UK
Zexuan Zhu, China
Mustapha Zidi, France

Contents

Theory and Applications of Complex Networks, Tingwen Huang, Zhichun Yang, and Chuandong Li
Volume 2014, Article ID 315059, 2 pages

Complex Dynamical Network Control for Trajectory Tracking Using Delayed Recurrent Neural Networks, Jose P. Perez, Joel Perez Padron, Angel Flores Hemandez, and Santiago Arroyo
Volume 2014, Article ID 162610, 7 pages

Reliability Analysis-Based Numerical Calculation of Metal Structure of Bridge Crane, Wenjun Meng, Zhengmao Yang, Xiaolong Qi, and Jianghui Cai
Volume 2013, Article ID 260976, 5 pages

Multimodal Processes Rescheduling: Cyclic Steady States Space Approach, Grzegorz Bocewicz, Robert Wójcik, Zbigniew Antoni Banaszak, and Paweł Pawlewski
Volume 2013, Article ID 407096, 24 pages

Photovoltaic Power Prediction Based on Scene Simulation Knowledge Mining and Adaptive Neural Network, Dongxiao Niu, Yanan Wei, and Yanchao Chen
Volume 2013, Article ID 260351, 6 pages

Using Software Dependency to Bug Prediction, Peng He, Bing Li, Yutao Ma, and Lulu He
Volume 2013, Article ID 869356, 12 pages

ACO-Based Routing Algorithm for Cognitive Radio Networks, Peng Huang, Feng Lin, and Jiliu Zhou
Volume 2013, Article ID 264504, 8 pages

Modeling Computer Virus and Its Dynamics, Mei Peng, Xing He, Junjian Huang, and Tao Dong
Volume 2013, Article ID 842614, 5 pages

An Evaluative Study on the Choice of Regional Strategic Emerging Industry Based on DEMATEL: Heilongjiang Province as an Example, Xiuying Guo, Xiaofeng Hui, and Junxiang Wang
Volume 2013, Article ID 430945, 9 pages

Impulsive Consensus for Leader-Following Multiagent Systems with Fixed and Switching Topology, Zhi-Wei Liu, Zhi-Hong Guan, and Hong Zhou
Volume 2013, Article ID 762861, 10 pages

User Demand Aware Grid Scheduling Model with Hierarchical Load Balancing, P. Suresh and P. Balasubramanie
Volume 2013, Article ID 439362, 8 pages

A Resilience Approach to Symbiosis Networks of Ecoindustrial Parks Based on Cascading Failure Model, Yu Zeng, Renbin Xiao, and Xiangmei Li
Volume 2013, Article ID 372368, 11 pages

Persistent Homology of Collaboration Networks, C. J. Carstens and K. J. Horadam
Volume 2013, Article ID 815035, 7 pages

Memristive Chebyshev Neural Network and Its Applications in Function Approximation, Lidan Wang, Meitao Duan, and Shukai Duan
Volume 2013, Article ID 429402, 7 pages

A New Multiobjective Evolutionary Algorithm for Community Detection in Dynamic Complex Networks, Guoqiang Chen, Yuping Wang, and Jingxuan Wei
Volume 2013, Article ID 161670, 7 pages

Memristive Perceptron for Combinational Logic Classification, Lidan Wang, Meitao Duan, and Shukai Duan
Volume 2013, Article ID 625790, 7 pages

Resonant Tunneling Diodes-Based Cellular Nonlinear Networks with Fault Tolerance Analysis, Shukai Duan, Xiaofang Hu, Lidan Wang, and Shiyong Gao
Volume 2013, Article ID 170202, 8 pages

Observer-Based Feedback Stabilization of Networked Control Systems with Random Packet Dropouts, Pengpeng Chen and Shouwan Gao
Volume 2013, Article ID 218682, 7 pages

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 315059, 2 pages
http://dx.doi.org/10.1155/2014/315059

Editorial

Theory and Applications of Complex Networks

Tingwen Huang,[1] Zhichun Yang,[2] and Chuandong Li[3]

[1] *Texas A&M University at Qatar, P.O. Box 23874, Doha, Qatar*
[2] *School of Mathematics and Statistics, Chongqing Normal University, Chongqing 400047, China*
[3] *School of Information, Southwest University, Chongqing 400715, China*

Correspondence should be addressed to Tingwen Huang; tingwen.huang@qatar.tamu.edu

Received 30 October 2013; Accepted 30 October 2013; Published 20 January 2014

Copyright © 2014 Tingwen Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Between two randomly selected persons in the world, roughly how many friends are there connecting them together? How are people infected by epidemics such as AIDS and bird flu all over the world? All of these are typical examples of complex networks. The impact of complex and dynamical networks pervades the physical, biological, and social sciences. Its influence on modern engineering and technology is notable and will be far-reaching. Since the discoveries of Watts and Strogatz on small-world networks, and those of Barabási and Albert on scale-free networks, many studies on various real-world networks have been carried out and reported from different points of view. The recent study of complex networks has attracted interest to the modeling and understanding of complex networks. In the past few decades, we have witnessed great success of the study in the coordination dynamics or cooperative collective behaviors of autonomous agents in complex networks.

This special issue focuses on modeling, analysis, control, and applications of real-world complex systems.

The paper entitled "*Photovoltaic power prediction based on scene simulation knowledge mining and adaptive neural network*" analyzes influence of light intensity, day type, temperature, and season on photovoltaic power.

The paper entitled "*Using software dependency to bug prediction*" uses several quantitative network metrics to explore their relationships with bug prediction in terms of software dependency.

The paper entitled "*ACO-based routing algorithm for cognitive radio networks*" proposes an ACO-based on-demand routing algorithm to tackle the routing problem.

The paper entitled "*Modeling computer virus and its dynamics*" presents a novel computer virus model.

The paper entitled "*An evaluative study on the choice of regional strategic emerging industry based on dematel: heilongjiang province as an example*" expands the DEMATEL method to the field of the complex weighted networks.

The paper entitled "*Impulsive consensus for leader-following multiagent systems with fixed and switching topology*" establishes the consensus problem of the leader-following multiagent system.

The paper entitled "*User demand aware grid scheduling model with hierarchical load balancing*" proposes the algorithm of hierarchical load balancing approach with user demand aware grid scheduling.

The paper entitled "*A resilience approach to symbiosis networks of ecoindustrial parks based on cascading failure model*" constructs a cascading model with tunable parameters for symbiosis networks of EIPs.

The paper entitled "*Persistent homology of collaboration networks*" analyzes the structure of weighted networks using persistent homology.

The paper entitled "*Memristive chebyshev neural network and its applications in function approximation*" presents a novel Chebyshev neural network combined with memristors to perform the function approximation.

The paper entitled "*A new multi-objective evolutionary algorithm for community detection in dynamic complex networks*" develops a novel multiobjective algorithm for detecting communities in dynamic networks.

The paper entitled "*Memristive perceptron for combinational logic classification*" proposes a new synaptic weight update rule.

The paper entitled "*Resonant tunneling diodes-based cellular nonlinear networks with fault tolerance analysis*" presents a

resonant tunneling diodes cellular neural/nonlinear network (RTD-CNN) model.

The paper entitled "*Observer-based feedback stabilization of networked control systems with random packet dropouts*" designs an adaptive controller of Networked Control Systems with Random Packet Dropouts.

Acknowledgments

We would like to express our sincere gratitude to the reviewers whose professional comments and suggestions guaranteed the high quality of these selected papers. Also, we are grateful to the authors for their contributions. This work is partially supported by the National Natural Science Foundation of China under Grant no. 10971240 and Grant no. 61374078. This paper was also made possible by NPRP Grant no. NPRP 4-1162-1-181 from the Qatar National Research Fund (a Member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Tingwen Huang
Zhichun Yang
Chuandong Li

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 162610, 7 pages
http://dx.doi.org/10.1155/2014/162610

Research Article

Complex Dynamical Network Control for Trajectory Tracking Using Delayed Recurrent Neural Networks

Jose P. Perez, Joel Perez Padron, Angel Flores Hemandez, and Santiago Arroyo

Facultad de Ciencias Fisico Matematicas, Universidad Autonoma de Nuevo Leon (UANL), 66451 San Nicolás de los Garza, NL, Mexico

Correspondence should be addressed to Jose P. Perez; josepazp@gmail.com

Received 21 March 2013; Accepted 12 September 2013; Published 2 January 2014

Academic Editor: Chuandong Li

Copyright © 2014 Jose P. Perez et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, the problem of trajectory tracking is studied. Based on the V-stability and Lyapunov theory, a control law that achieves the global asymptotic stability of the tracking error between a delayed recurrent neural network and a complex dynamical network is obtained. To illustrate the analytic results, we present a tracking simulation of a dynamical network with each node being just one Lorenz's dynamical system and three identical Chen's dynamical systems.

1. Introduction

The analysis and control of complex behavior in complex networks, which consist of dynamical nodes, have become a point of great interest in the recent studies, [1–3]. The complexity in networks comes not only from their structure and dynamics but also from their topology, which often affects their function.

Recurrent neural networks have been widely used in the fields of optimization, pattern recognition, signal processing and control systems, among others. They have to be designed in such a way that there is one equilibrium point that is globally asymptotically stable. In biological and artificial neural networks, time delays arise in the processing of information storage and transmission. Also, it is known that these delays can create oscillatory or even unstable trajectories, [4]. Trajectory tracking is a very interesting problem in the field of theory of systems control; it allows the implementation of important tasks for automatic control such as: high speed target recognition and tracking, real-time visual inspection, and recognition of context sensitive and moving scenes, among others. We present the results of the design of a control law that guarantee the tracking of general complex dynamical networks.

2. Mathematical Models

2.1. General Complex Dynamical Networks. Consider a network consisting of N linearly and diffusively coupled nodes, with each node being an n-dimensional dynamical system, described by

$$\dot{x}_i = f_i\left(x_i\right) + \sum_{\substack{j=1 \\ j\neq i}}^{N} c_{ij} a_{ij} \Gamma\left(x_j - x_i\right), \quad i = 1, 2, \ldots, N, \tag{1}$$

where $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T \in \mathbb{R}^n$ are the state vectors of the node i, $f_i : \mathbb{R}^n \mapsto \mathbb{R}^n$ represents the self-dynamics of the node i, and the constants $c_{ij} > 0$ are the coupling strengths between node i and node j, with $i, j = 1, 2, \ldots, N$. $\Gamma = (\tau_{ij}) \in \mathbb{R}^{n\times n}$ is a constant internal matrix that describes the way of linking the components in each pair of connected node vectors $(x_j - x_i)$: this means that for some pairs (i, j) with $1 \leq i, j \leq n$ and $\tau_{ij} \neq 0$, the two coupled nodes are linked through their ith and jth substate variables, respectively, while the coupling matrix $A = (a_{ij}) \in \mathbb{R}^{N\times N}$ denotes the coupling configuration of the entire network: this means that if there is a connection between node i and node j $(i \neq j)$, then $a_{ij} = a_{ji} = 1$; otherwise, $a_{ij} = a_{ji} = 0$.

2.2. Delayed Recurrent Neural Networks. Consider a delayed recurrent neural network in the following form:

$$\dot{x}_{n_i} = A_{n_i} x_{n_i} + W_{n_i} \sigma\left(x_{n_i}(t-\tau)\right) + u_{n_i} + \sum_{\substack{j=1\\ j\neq i}}^{N} c_{n_i n_j} a_{n_i n_j} \Gamma\left(x_{n_j} - x_{n_i}\right), \quad i = 1, 2, \ldots, N, \tag{2}$$

where τ is the fixed known time delay [5, 6], $x_{n_i} = (x_{n_i 1}, x_{n_i 2}, \ldots, x_{n_i n})^T \in \mathbb{R}^n$ is the state vector of the neural network i, $u_{n_i} \in \mathbb{R}^n$ is the input of the neural network i, $A_{n_i} = -\lambda_{n_i} I_{n\times n}$, $i = 1, 2, \ldots, N$, is the state feedback matrix, with λ_{n_i} being a positive constant, $W_{n_i} \in \mathbb{R}^{n\times n}$ is the connection weight matrix with $i = 1, 2, \ldots, N$, and $\sigma(\cdot) \in \mathbb{R}^n$ is a Lipschitz sigmoid vector function [7, 8], such that $\sigma(x_{n_i}) = 0$ only at $x_{n_i} = 0$, with Lipschitz constant L_{σ_i}, $i = 1, 2, \ldots, N$, and neuron activation functions $\sigma_i(\cdot) = \tanh(\cdot)$, $i = 1, 2, \ldots, n$.

3. Trajectory Tracking

The objective is to develop a control law such that the ith neural network (2) tracks the trajectory of the ith dynamical system (1). We define the tracking error as $e_i = x_{n_i} - x_i$, $i = 1, 2, \ldots, N$, whose derivative with respect to time is

$$\dot{e}_i = \dot{x}_{n_i} - \dot{x}_i, \quad i = 1, 2, \ldots, N. \tag{3}$$

Substituting (1) and (2) in (3), we obtain

$$\dot{e}_i = A_{n_i} x_{n_i} + W_{n_i} \sigma\left(x_{n_i}(t-\tau)\right) + u_{n_i} - f_i(x_i) + \sum_{\substack{j=1\\ j\neq i}}^{N} c_{n_i n_j} a_{n_i n_j} \Gamma\left(x_{n_j} - x_{n_i}\right) - \sum_{\substack{j=1\\ j\neq i}}^{N} c_{ij} a_{ij} \Gamma\left(x_j - x_i\right), \quad i = 1, 2, \ldots, N. \tag{4}$$

Adding and subtracting $W_{n_i}\sigma(x_i(t-\tau))$, $\alpha_i(t)$, $i = 1, 2, \ldots, N$, to (4), where α_i will be determined below, and considering that $x_{n_i} = e_i + x_i$, $i = 1, 2, \ldots, N$, then

$$\begin{aligned}\dot{e}_i = {} & W_{n_i}\left(\sigma\left(x_{n_i}(t-\tau)\right) - \sigma\left(x_i(t-\tau)\right)\right) + \left(u_{n_i} - \alpha_i(t)\right) + A_{n_i} e_i \\ & + \left(A_{n_i} x_i + W_{n_i}\sigma\left(x_i(t-\tau)\right) + \alpha_i(t)\right) - f_i(x_i) \\ & + \sum_{\substack{j=1\\ j\neq i}}^{N} c_{n_i n_j} a_{n_i n_j} \Gamma\left(x_{n_j} - x_{n_i}\right) \\ & - \sum_{\substack{j=1\\ j\neq i}}^{N} c_{ij} a_{ij} \Gamma\left(x_j - x_i\right), \quad i = 1, 2, \ldots, N.\end{aligned} \tag{5}$$

In order to guarantee that the ith neural network (2) tracks the ith reference trajectory (1), the following assumption has to be satisfied:

Assumption 1. There exist functions $\rho_i(t)$ and $\alpha_i(t)$, $i = 1, 2, \ldots, N$, such that

$$\frac{d\rho_i(t)}{dt} = A_{n_i}\rho_i(t) + W_{n_i}\sigma\left(\rho_i(t)\right) + \alpha_i(t), \quad \rho_i(t) = x_i(t), \quad i = 1, 2, \ldots, N. \tag{6}$$

Let us define

$$\tilde{u}_{n_i} = \left(u_{n_i} - \alpha_i(t)\right), \quad \phi_\sigma(t-\tau) = \sigma\left(x_{n_i}(t-\tau)\right) - \sigma\left(x_i(t-\tau)\right), \quad i = 1, 2, \ldots, N. \tag{7}$$

Considering (6) and (7), equation (5) is reduced to

$$\dot{e}_i = A_{n_i} e_i + W_{n_i}\phi_\sigma(t-\tau) + \tilde{u}_{n_i} + \sum_{\substack{j=1\\ j\neq i}}^{N} c_{n_i n_j} a_{n_i n_j} \Gamma\left(x_{n_j} - x_{n_i}\right) - \sum_{\substack{j=1\\ j\neq i}}^{N} c_{ij} a_{ij} \Gamma\left(x_j - x_i\right), \quad i = 1, 2, \ldots, N. \tag{8}$$

Writing the summations as

$$\begin{aligned}\sum_{\substack{j=1\\ j\neq i}}^{N} c_{n_i n_j} a_{n_i n_j} \Gamma\left(x_{n_j} - x_{n_i}\right) &= \Gamma\left(\sum_{\substack{j=1\\ j\neq i}}^{N} c_{n_i n_j} a_{n_i n_j} x_{n_j} - x_{n_i} \sum_{\substack{j=1\\ j\neq i}}^{N} c_{n_i n_j} a_{n_i n_j}\right) \times \sum_{\substack{j=1\\ j\neq i}}^{N} c_{ij} a_{ij} \Gamma\left(x_j - x_i\right) \\ &= \Gamma\left(\sum_{\substack{j=1\\ j\neq i}}^{N} c_{ij} a_{ij} x_j - x_i \sum_{\substack{j=1\\ j\neq i}}^{N} c_{ij} a_{ij}\right), \quad i = 1, 2, \ldots, N\end{aligned} \tag{9}$$

and using that $c_{n_i n_j} = c_{ij}$ and $a_{n_i n_j} = a_{ij}$, then, using the equations above, (8) becomes

$$\dot{e}_i = A_{n_i} e_i + W_{n_i}\phi_\sigma(t-\tau) + \tilde{u}_{n_i} + \Gamma\left(\sum_{\substack{j=1\\ j\neq i}}^{N} c_{ij} a_{ij} e_j - e_i \sum_{\substack{j=1\\ j\neq i}}^{N} c_{ij} a_{ij}\right)$$

$$= A_{n_i} e_i + W_{n_i} \phi_\sigma (t-\tau) + \tilde{u}_{n_i} + \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} a_{ij} \Gamma \left(e_j - e_i\right), \quad i = 1, 2, \ldots, N. \tag{10}$$

It is clear that $e_i = 0$, $i = 1, 2, \ldots, N$, is an equilibrium point of (10), when $\tilde{u}_{n_i} = 0$, $i = 1, 2, \ldots, N$. In this way, the tracking problem can be restated as a global asymptotic stabilization problem for the system (10).

4. Tracking Error Stabilization and Control Design

In order to establish the convergence of (10) to $e_i = 0$, $i = 1, 2, \ldots, N$, which ensures the desired tracking, first, we propose the following Lyapunov function:

$$\begin{aligned} V_N(e) &= \sum_{i=1}^{N} V(e_i) \\ &= \sum_{i=1}^{N} \left(\frac{1}{2} \|e_i\|^2 + \int_{t-\tau}^{t} \left(\phi_\sigma^T(s) W_{n_i}^T W_{n_i} \phi_\sigma(s) \right) ds \right), \\ & \qquad e = \left(e_1^T, \ldots, e_N^T\right)^T. \end{aligned} \tag{11}$$

The time derivative of (11), along the trajectories of (10), is

$$\begin{aligned} \dot{V}_N(e) &= \frac{\partial V_N(e)}{\partial e} \dot{e} = \sum_{i=1}^{N} \frac{\partial V_N(e)}{\partial e_i} \dot{e}_i \\ &= \sum_{i=1}^{N} \Bigg(\phi_\sigma^T(t) W_{n_i}^T W_{n_i} \phi_\sigma(t) \\ &\quad - \phi_\sigma^T(t-\tau) W_{n_i}^T W_{n_i} \phi_\sigma(t-\tau) \\ &\quad + e_i^T \Bigg(A_{n_i} e_i + W_{n_i} \phi_\sigma(t-\tau) + \tilde{u}_{n_i} \\ &\quad + \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} a_{ij} \Gamma \left(e_j - e_i\right) \Bigg) \Bigg). \end{aligned} \tag{12}$$

Reformulating (12), we get

$$\begin{aligned} \dot{V}_N(e) = \sum_{i=1}^{N} \Bigg(& - \lambda_{n_i} \|e_i\|^2 + e_i^T W_{n_i} \phi_\sigma(t-\tau) \\ & + e_i^T \tilde{u}_{n_i} + \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} a_{ij} e_i^T \Gamma \left(e_j - e_i\right) \\ & + \phi_\sigma^T(t) W_{n_i}^T W_{n_i} \phi_\sigma(t) \\ & - \phi_\sigma^T(t-\tau) W_{n_i}^T W_{n_i} \phi_\sigma(t-\tau) \Bigg). \end{aligned} \tag{13}$$

Next, let us consider the following inequality, proved in [9, 10]:

$$X^T Y + Y^T X \leq X^T \Lambda X + Y^T \Lambda^{-1} Y, \tag{14}$$

which holds for all matrices $X, Y \in \mathbb{R}^{n \times k}$ and $\Lambda \in \mathbb{R}^{n \times n}$ with $\Lambda = \Lambda^T > 0$. Applying (14) with $\Lambda = I_{n \times n}$ to the term $e_i^T W_{n_i} \phi_\sigma(t-\tau)$, $i = 1, 2, \ldots, N$, we get

$$\begin{aligned} & e_i^T W_{n_i} \phi_\sigma(t-\tau) \\ & \quad \leq \frac{1}{2} e_i^T e_i + \frac{1}{2} \phi_\sigma^T(t-\tau) W_{n_i}^T W_{n_i} \phi_\sigma(t-\tau) \\ & \quad \times \frac{1}{2} \|e_i\|^2 + \frac{1}{2} \phi_\sigma^T(t-\tau) W_{n_i}^T W_{n_i} \phi_\sigma(t-\tau), \\ & \qquad i = 1, 2, \ldots, N. \end{aligned} \tag{15}$$

Then, we have that

$$\begin{aligned} \dot{V}_N(e) \leq \sum_{i=1}^{N} \Bigg(& -\lambda_{n_i} \|e_i\|^2 + \frac{1}{2} \|e_i\|^2 \\ & + \frac{1}{2} \phi_\sigma^T(t-\tau) W_{n_i}^T W_{n_i} \phi_\sigma(t-\tau) \\ & + e_i^T \tilde{u}_{n_i} + \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} a_{ij} e_i^T \Gamma \left(e_j - e_i\right) \\ & + \phi_\sigma^T(t) W_{n_i}^T W_{n_i} \phi_\sigma(t) \\ & - \phi_\sigma^T(t-\tau) W_{n_i}^T W_{n_i} \phi_\sigma(t-\tau) \Bigg). \end{aligned} \tag{16}$$

By simplifying (16), we obtain

$$\begin{aligned} \dot{V}_N(e) &\leq \sum_{i=1}^{N} \Bigg(- \lambda_{n_i} \|e_i\|^2 + \frac{1}{2} \|e_i\|^2 \\ &\quad - \frac{1}{2} \phi_\sigma^T(t-\tau) W_{n_i}^T W_{n_i} \phi_\sigma(t-\tau) + e_i^T \tilde{u}_{n_i} \\ &\quad + \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} a_{ij} e_i^T \Gamma \left(e_j - e_i\right) \\ &\quad + \phi_\sigma^T(t) W_{n_i}^T W_{n_i} \phi_\sigma(t) \Bigg) \\ &\leq \sum_{i=1}^{N} \Bigg(- \lambda_{n_i} \|e_i\|^2 + \frac{1}{2} \|e_i\|^2 \\ &\quad + \phi_\sigma^T(t) W_{n_i}^T W_{n_i} \phi_\sigma(t) + e_i^T \tilde{u}_{n_i} \\ &\quad + \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} a_{ij} e_i^T \Gamma \left(e_j - e_i\right) \Bigg). \end{aligned} \tag{17}$$

Since ϕ_σ is Lipschitz with Lipschitz constant $L_{\phi_{\sigma_i}}$ [7], then

$$\begin{aligned}\|\phi_\sigma(t)\| &= \left\|\sigma\left(x_{n_i}(t)\right)-\sigma\left(x_i(t)\right)\right\| \\ &\le L_{\phi_{\sigma_i}}\left\|x_{n_i}(t)-x_i(t)\right\| \\ &= L_{\phi_{\sigma_i}}\|e_i(t)\|, \quad i=1,2,\ldots,N.\end{aligned} \tag{18}$$

Applying (18) to $\phi_\sigma^T(t)W_{n_i}^TW_{n_i}\phi_\sigma(t)$, we obtain

$$\begin{aligned}\phi_\sigma^T(t)W_{n_i}^TW_{n_i}\phi_\sigma(t) &\le \left\|\phi_\sigma^T(t)W_{n_i}^TW_{n_i}\phi_\sigma(t)\right\| \\ &\le \left(L_{\phi_{\sigma_i}}\right)^2\|W_{n_i}\|^2\|e_i\|^2, \quad i=1,2,\ldots,N.\end{aligned} \tag{19}$$

Now, (17) is reduced to

$$\begin{aligned}\dot V_N(e) &\le \sum_{i=1}^N\Bigg(-\lambda_{n_i}\|e_i\|^2+\frac{1}{2}\|e_i\|^2 \\ &\quad+\left(L_{\phi_{\sigma_i}}\right)^2\|W_{n_i}\|^2\|e_i\|^2+e_i^T\tilde u_{n_i} \\ &\quad+\sum_{\substack{j=1\\j\ne i}}^N c_{ij}a_{ij}e_i^T\Gamma\left(e_j-e_i\right)\Bigg) \\ &= \sum_{i=1}^N e_i^T\Bigg(-\lambda_{n_i}e_i-\sum_{\substack{j=1\\j\ne i}}^N c_{ij}a_{ij}\Gamma e_i \\ &\quad+\left(\frac{1}{2}+L_{\phi_{\sigma_i}}^2\|W_{n_i}\|^2\right)e_i \\ &\quad+\sum_{\substack{j=1\\j\ne i}}^N c_{ij}a_{ij}\Gamma e_j+\tilde u_{n_i}\Bigg).\end{aligned} \tag{20}$$

We define $\tilde u_{n_i}=\tilde u_{n_i}^{(1)}+\tilde u_{n_i}^{(2)}$, $i=1,2,\ldots,N$, and then (20) becomes

$$\begin{aligned}\dot V_N(e) &\le \sum_{i=1}^N e_i^T\Bigg(-\lambda_{n_i}e_i-\sum_{\substack{j=1\\j\ne i}}^N c_{ij}a_{ij}\Gamma e_i \\ &\quad+\left(\frac{1}{2}+L_{\phi_{\sigma_i}}^2\|W_{n_i}\|^2\right)e_i+\tilde u_{n_i}^{(1)} \\ &\quad+\sum_{\substack{j=1\\j\ne i}}^N c_{ij}a_{ij}\Gamma e_j+\tilde u_{n_i}^{(2)}\Bigg).\end{aligned} \tag{21}$$

Now, we propose the use of the following control law:

$$\begin{aligned}\tilde u_{n_i} &= -\left(\frac{1}{2}+L_{\phi_{\sigma_i}}^2\|W_{n_i}\|^2\right)e_i \\ &\quad-\sum_{\substack{j=1\\j\ne i}}^N c_{ij}a_{ij}\Gamma e_j, \quad i=1,2,\ldots,N.\end{aligned} \tag{22}$$

Then, $\dot V_N(e)<0$ for all $e\ne 0$. This means that the proposed control law (22) can globally and asymptotically stabilize the ith error system (10), therefore ensuring the tracking of (1) by (2). Finally, the control action driving the recurrent neural networks is given by

$$\begin{aligned}u_{n_i} &= f_i(x_i)+\lambda_{n_i}x_i-W_{n_i}\sigma\left(x_i(t-\tau)\right) \\ &\quad-\left(\frac{1}{2}+L_{\phi_{\sigma_i}}^2\|W_{n_i}\|^2\right)e_i \\ &\quad-\sum_{\substack{j=1\\j\ne i}}^N c_{ij}a_{ij}\Gamma e_j, \quad i=1,2,\ldots,N.\end{aligned} \tag{23}$$

5. Simulations

In order to illustrate the applicability of the discussed results, we consider a dynamical network with just one Lorenz's node and three identical Chen's nodes. The single Lorenz system is described by

$$\begin{gathered}\begin{pmatrix}\dot x_1\\ \dot x_2\\ \dot x_3\end{pmatrix}=\begin{pmatrix}10x_2-10x_1\\ -x_2-x_1x_2+28x_1\\ x_1x_2-\left(\frac{8}{3}\right)x_3\end{pmatrix} \\ x_i(0)=(10,0,10)^T, \quad i=1.\end{gathered} \tag{24}$$

And the Chen's oscillator is described by

$$\begin{gathered}\begin{pmatrix}\dot x_{i1}\\ \dot x_{i2}\\ \dot x_{i3}\end{pmatrix} \\ =\begin{pmatrix}p_1\left(x_{i2}-x_{i1}\right)+\sum_{\substack{j=1\\j\ne i}}^N c_{ij}a_{ij}\left(x_{j1}-x_{i1}\right) \\ \left(p_3-p_2\right)x_{i1}-x_{i1}x_{i3}+p_3x_{i2}+\sum_{\substack{j=1\\j\ne i}}^N c_{ij}a_{ij}\left(x_{j2}-x_{i2}\right) \\ x_{i1}x_{i2}-p_2x_{i3}+\sum_{\substack{j=1\\j\ne i}}^N c_{ij}a_{ij}\left(x_{j3}-x_{i3}\right)\end{pmatrix} \\ x_i(0)=(-10,0,37)^T, \quad i=1,2,3,4.\end{gathered} \tag{25}$$

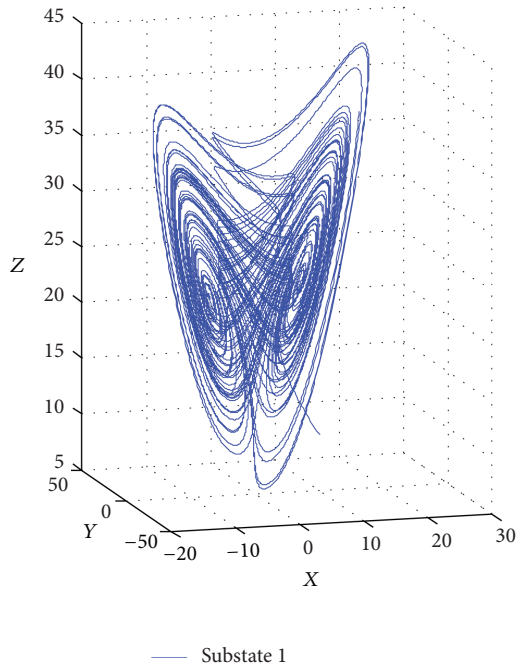


Figure 1: Substate of Lorentz's attractor with initial condition $X_1(0) = (10; 0; 10)^T$.

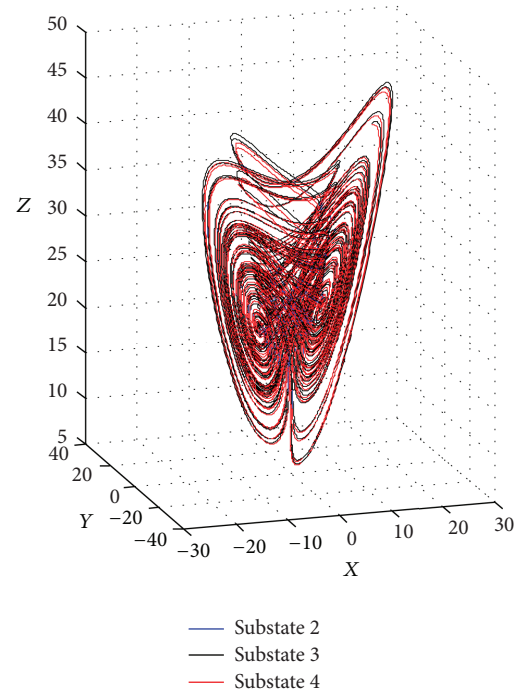


Figure 2: Sub-States of Chen's attractor with initial condition $X_{2,3,4}(0) = (-10; 0; 37)^T$.

If the system parameters are selected as $p_1 = 35$, $p_2 = 3$, and $p_3 = 28$, then the Lorenz's system and Chen's system are shown in Figures 1 and 2, respectively. In this set of system parameters, one unstable equilibrium point of the oscillator (25) is $x = (7.9373, 7.9373, 21)^T$ [11].

Suppose that each pair of two connected Lorenz and Chen's oscillators are linked together through their identical substate variables; that is, $\Gamma = \mathrm{diag}(1, 1, 1)$, and the coupling strengths are $c_{12} = c_{21} = \pi$, $c_{13} = c_{31} = \pi$, $c_{23} = c_{32} = \pi$, $c_{14} = c_{41} = 2\pi$, $c_{24} = c_{42} = 2\pi$, and $c_{34} = c_{43} = 2\pi$. Figure 3 visualizes this entire dynamical network.

The neural network is selected as

$$A_{n_i} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \qquad W_{n_i} = \begin{pmatrix} 1 & 2 & 0 \\ -3 & 4 & 0 \\ 0 & 2 & 3 \end{pmatrix},$$

$$\sigma\left(x_{n_i}(t-\tau)\right) = \begin{pmatrix} \tanh\left(x_{n_i 1}(t-\tau)\right) \\ \tanh\left(x_{n_i 2}(t-\tau)\right) \\ \tanh\left(x_{n_i 3}(t-\tau)\right) \end{pmatrix},$$

$$\tau = 10 \text{ seconds},$$

$$L_{\phi_{\sigma_i}} \triangleq n = 3,$$

$$x_{n_i}(0) = (20, 20, -10)^T, \quad i = 1, 2, 3, 4. \tag{26}$$

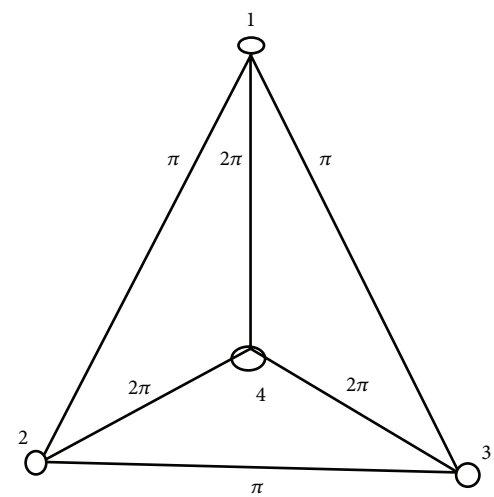


Figure 3: Structure of the network with each node being a Lorentz and Chen's system.

Theorem 2. *For the unknown nonlinear system modeled by* (1), *the on-line learning law* $\mathrm{tr}\{W^T W\} = -e^T W\sigma(x)$ *and the control law* (23) *ensure the tracking of nonlinear reference model* (4), *[12].*

Remark 3. From (21), we have $\dot{V}_N(e) \le \sum_{i=1}^{N} e_i^T(-\lambda_{n_i} e_i - \sum_{\substack{j=1 \\ j\ne i}}^{N} c_{ij} a_{ij} \Gamma e_i + ((1/2) + L_{\phi_{\sigma_i}}^2 \|W_{n_i}\|^2) e_i + \tilde{u}_{n_i}^{(1)} + \sum_{\substack{j=1 \\ j\ne i}}^{N} c_{ij} a_{ij} \Gamma e_j + \tilde{u}_{n_i}^{(2)}) < 0$, $\forall\, e \ne 0$, $\forall W$, and therefore V is decreasing and

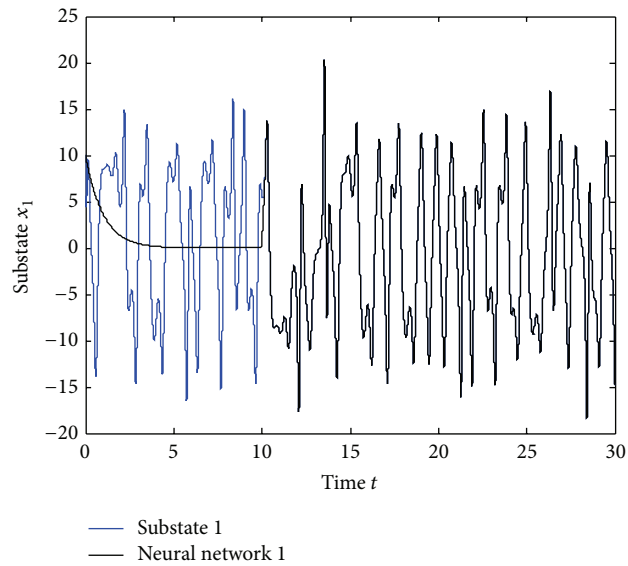


FIGURE 4: Time evolution for sub-state 1 with initial state $X_{n1}(0) = (10; 0; 10)^T$.

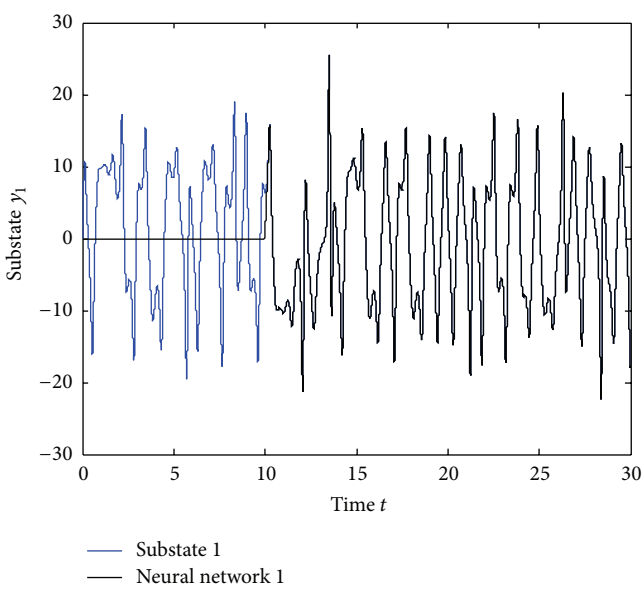


FIGURE 5: Time evolution for sub-state 1 with initial state $X_{n1}(0) = (10; 0; 10)^T$.

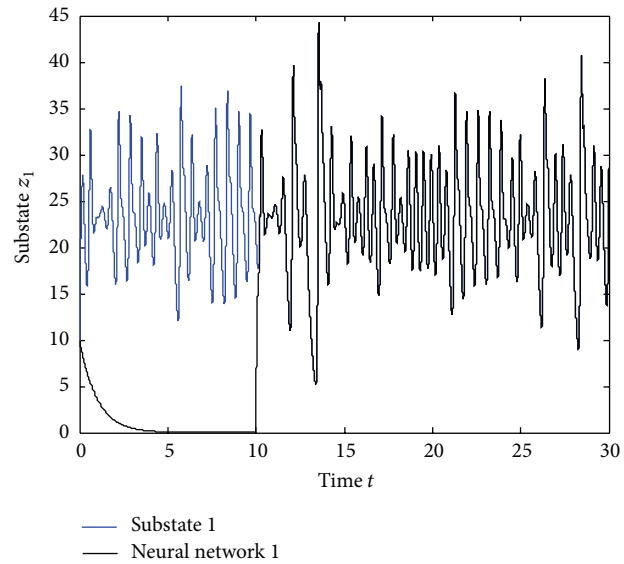


FIGURE 6: Time evolution for sub-state 2 with initial state $X_{n1}(0) = (10; 0; 10)^T$.

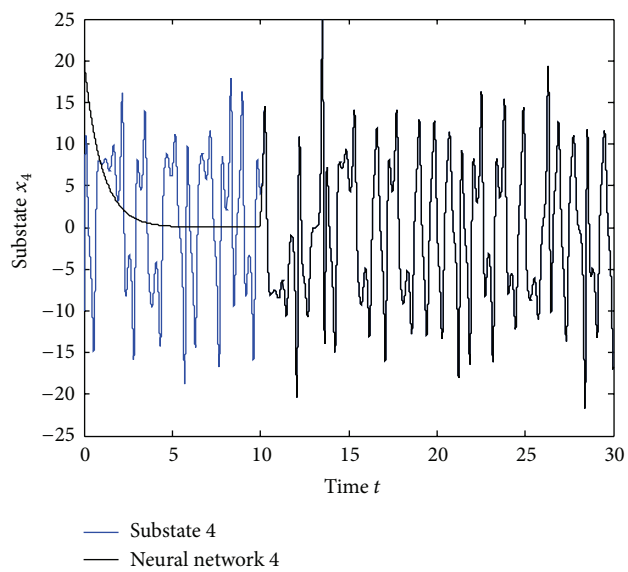


FIGURE 7: Time evolution for sub-state 4 with initial state $X_{n4}(0) = (20, 20, -10)^T$.

bounded from below by $V(0)$. Since $V_N(e) = \sum_{i=1}^{N}((1/2)\|e_i\|^2 + \int_{t-\tau}^{t}(\phi_\sigma^T(s)W_{n_i}^T W_{n_i}\phi_\sigma(s))\, ds)$, then we conclude that $e, W \in L_1$; this means that the weights remain bounded.

The experiment is performed as follows. Both systems, the delayed neural network (2) and the dynamical networks (24) and (25), evolve independently until $t = 10$ seconds; at that time, the proposed control law (23) is incepted. Simulation results are presented in Figures 4, 5, and 6 for sub-sates of node 1. As can be seen, tracking is successfully achieved and error is asymptotically stable, as it is shown in Figures 7, 8, and 9 for sub-states of node 4.

6. Conclusions

We have presented the controller design for trajectory tracking determined by a general complex dynamical network. This framework is based on dynamic delayed neural networks and the methodology is based on V-stability and Lyapunov theory. The proposed control is applied to a dynamical network with each node being a Lorenz and Chen's dynamical systems, respectively, being able to also stabilize in asymptotic

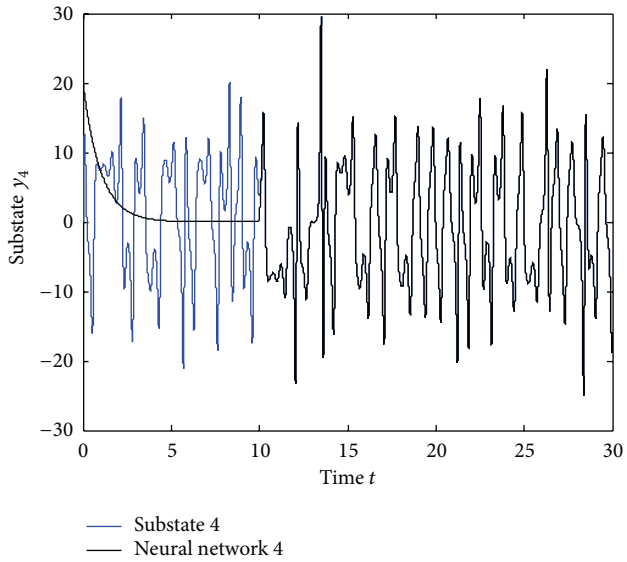


Figure 8: Time evolution for sub-state 4 with initial state $X_{n4}(0) = (20, 20, -10)^T$.

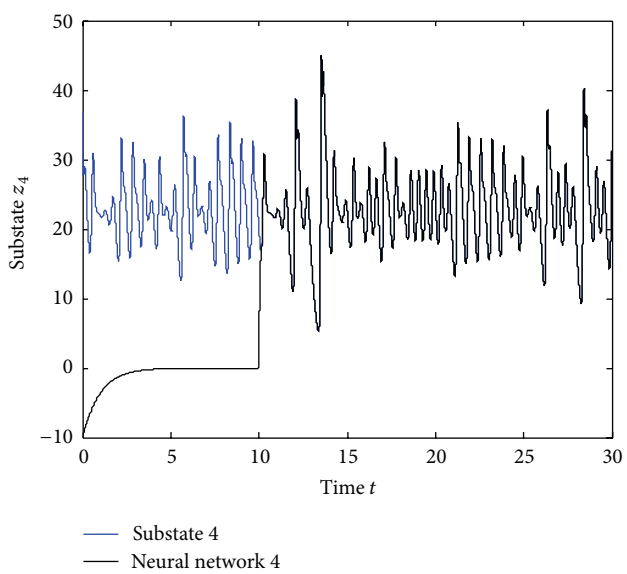


Figure 9: Time evolution for sub-state 4 with initial state $X_{n4}(0) = (20, 20, -10)^T$.

form the tracking error between the two systems. The results of the simulation clearly show clearly the desired tracking. In future work, we will consider the stochastic case for the complex dynamical network.

Conflict of Interests

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

Acknowledgments

The first and second authors thank the support of CONACYT and the Matematicas Aplicadas group of the Facultad de Ciencias Fisico-Matematicas, Universidad Autonoma de Nuevo Leon, Mexico.

References

[1] X. F. Wang, "Complex networks: topology, dynamics and synchronization," *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 12, no. 5, pp. 885–916, 2002.

[2] C. Wu, *Synchronization in Coupled Chaotic Circuits and Systems*, World Scientific, Singapore, 2002.

[3] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[4] P. Baldi and A. F. Atiya, "How delays affect neural dynamics and learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 612–621, 1994.

[5] X. Liao, G. Chen, and E. N. Sanchez, "LMI-based approach for asymptotically stability analysis of delayed neural networks," *IEEE Transactions on Circuits and Systems I*, vol. 49, no. 7, pp. 1033–1039, 2002.

[6] X. Liao, G. Chen, and E. N. Sanchez, "Delay-dependent exponential stability analysis of delayed neural networks: an LMI approach," *Neural Networks*, vol. 15, no. 7, pp. 855–866, 2002.

[7] H. Khalil, *Nonlinear System Analysis*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 1996.

[8] M. Krstic and H. Deng, *Stabilization of Nonlinear Uncertain Systems*, Springer, New York, NY, USA, 1998.

[9] E. N. Sanchez and J. P. Perez, "Input-to-state stability (ISS) analysis for dynamic neural networks," *IEEE Transactions on Circuits and Systems I*, vol. 46, no. 11, pp. 1395–1398, 1999.

[10] E. N. Sanchez, J. P. Perez, and J. Perez, "Trajectory tracking for delayed recurrent neural networks," in *Proceedings of the American Control Conference*, pp. 2280–2284, Minneapolis, Minn, USA, June 2006.

[11] X. Li, X. Wang, and G. Chen, "Pinning a complex dynamical network to its equilibrium," *IEEE Transactions on Circuits and Systems I*, vol. 51, no. 10, pp. 2074–2087, 2004.

[12] P. A. Ioannou and J. Sun, *Robust Adaptive Control*, Prentice-Hall, Upper Saddle River, NJ, USA, 1995.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 260976, 5 pages
http://dx.doi.org/10.1155/2013/260976

Research Article

Reliability Analysis-Based Numerical Calculation of Metal Structure of Bridge Crane

Wenjun Meng,[1] Zhengmao Yang,[1] Xiaolong Qi,[2] and Jianghui Cai[1]

[1] *Taiyuan University of Science and Technology, Taiyuan 030024, China*
[2] *Tongji University, Shanghai 200092, China*

Correspondence should be addressed to Wenjun Meng; tyustmwj@tyust.edu.cn

Received 17 March 2013; Accepted 21 September 2013

Academic Editor: Zhichun Yang

Copyright © 2013 Wenjun Meng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The study introduced a finite element model of DQ75t-28m bridge crane metal structure and made finite element static analysis to obtain the stress response of the dangerous point of metal structure in the most extreme condition. The simulated samples of the random variable and the stress of the dangerous point were successfully obtained through the orthogonal design. Then, we utilized BP neural network nonlinear mapping function trains to get the explicit expression of stress in response to the random variable. Combined with random perturbation theory and first-order second-moment (FOSM) method, the study analyzed the reliability and its sensitivity of metal structure. In conclusion, we established a novel method for accurately quantitative analysis and design of bridge crane metal structure.

1. Introduction

Bridge crane metal structure is significant in the bridge crane to host, tract, walk, and brake, which is the most critical part of the performance to determine its safety. As the stability and reliability of metal structure is the guarantee of the entire system, its corresponding analysis, therefore, is crucial before the application [1, 2]. Currently, there are various uncertainties in the metal structure of bridge crane, such as size parameters, material properties, and external load. These uncertainties have directly influenced the safety and reliability of the metal structure. Therefore, studies using the reliability analysis theory to ensure the stability are encouraging. In addition, due to the different influence degrees of these random parameters, the reliability sensitivity analysis is also essential to obtain dependence of metal structure reliability to each random parameter, which will instruct the optimized design and applicable conditions of the metal structure.

The study introduced a finite element model of bridge crane metal structure and then applied finite element simulation and orthogonal experimental design to obtain the dangerous parts and stress response in the most extreme condition. We utilized BP neural network fitting technology to get an explicit expression of the stress on the design variables. Besides, random perturbation theory and FOSM method were used for the reliability analysis. The matrix differential technology further contributed to deduce the influence degree of various random parameters on the reliability of the bridge crane metal structure.

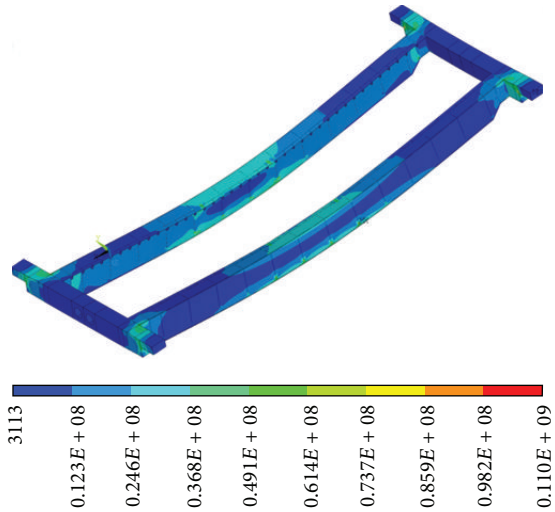
2. Finite Element Analysis of Bridge Crane Metal Structure

2.1. Finite Element Analysis. ANSYS parametric design language (APDL) was used to establish the finite element model of bridge crane metal structure (Figure 1). Box-type bridge structure model selected shell elements SHELL63 (4-Node Elastic Shell, UX, UY, UZ, ROTX, ROTY, ROTZ) because the unit had capacity to deal with stress stiffening and large deflection [3]. Car track model selected beam element BEAM 189 (3 Node 3-D Quadratic Finite Strain Beam, UX, UY, UZ, ROTX, ROTY, ROTZ), which is based on the Timoshenko beam analysis theory, with default of shear effects and large deformation effects [3, 4].

2.2. Stress Response Analysis. We applied finite element analysis software ANSYS to simulate and analyze the metal structure of bridge crane. The stress response in the most severe

Finite element analysis of structure of bridge crane

Figure 1: Finite element model of bridge crane metal structure.



Finite element analysis of structure of bridge crane

Figure 2: Stress nephogram of metal structure under full-load condition.

operating conditions was obtained (Figure 2). With full load, the maximum stress (110 MPa) was observed at the midspan of metal structure and the maximum deformation occurred at the midspan of top flange plate. Meanwhile, the biggest stress existed at the intersection of the lower flange plate, the main web and the diaphragms of box structure (Figure 2). Although the intersection presented the potential fracture and failure, the value did not exceed the material yield strength and conformed to the requirements of static strength and static stiffness. Therefore, it can be concluded that the metal structure will not permanently deform and can meet the needs of safe carrying and running. The intermetal structure von-Mises stress nephogram in the condition of full load was shown in Figure 2.

3. BP Neural Network Fitting Stress Response

Since the metal structure of bridge machine is an extremely complex box structure, the function of stress and random design variables is a highly nonlinear and implicit. Therefore, we utilized the orthogonal experimental design method and finite element simulation to get sample data. Then, we combined the BP neural network technology to map the relationship between structure design variables and stress values in dangerous parts, fitting to acquire their explicit expression [5–7].

3.1. Neural Network Training Samples. The study applied orthogonal experimental design method to establish the reasonable quantity and distribution of neural network training samples to accurately express the mapping relationship of neural network model. According to the reliability theory, we selected the simulated full-load condition as the object and obtained the data of random variable as well as the stress response (Table 1).

The output samples of stress response of dangerous parts could be gained through the experimental design. It would be further used as the training samples of BP neural network to eventually fit required explicit expressions: $S = (W, H, B, E, e_YYB, e_FB)$.

Training samples of the neural network model was obtained by using the orthogonal table L_{25} (5^6); that is, the total number of samples was 25, the factors were 6, each factor had 5 levels [4] (Table 2). The stress response of its corresponding 25 samples, as output variables of neural network model, was calculated by the finite element model.

Table 1: Random variable and its statistical properties of bridge crane metal structure.

Random variable (factors)	Meaning	Average	Standard deviation
$W/10^6$ N	Lifting load	250	20
H/mm	Main beam height	8155	366
B/mm	Main beam width	4108	207
E/MPa	Elastic modulus	206	10
e_YYB/mm	Upper and lower flange plate thickness	14	4.2
e_FB/mm	Main, vice web thickness	12	2.5

Table 2: Orthogonal factors and the level.

Level	W	H	B	E	e_YYB	e_FB
1	230	8000	3800	196	11	6
2	240	8100	3900	201	12	7
3	250	8200	4000	206	13	8
4	260	8300	4100	211	14	9
5	270	8400	4200	216	15	10

3.2. Neural Network Model. For the 6 random variables of bridge crane metal structure system, BP neural network input layer had 6 neurons and the hidden layer had 15 neurons. Each neuron of the output layer was described as the stress response of the bridge crane metal structure.

To get the stress response neural network model of the global significance, each random variable in its feasible region took 5 discrete values (horizontal) and generated 25 design samples as the training set by the combination of

orthogonal test. Then, finite element simulation experiments were carried out on all samples by ANSYS software.

3.3. Neural Network Learning. After the model structure of BP neural network was determined, the toolbox (NNET Toolbox) of MATLB was used in training the network using the input sample set and the output sample set to achieve a given input-output mapping relationship and further correct the thresholds and weights of the network [8].

Some neurons reached saturation due to the large difference in the magnitude of each variable in the original sample. As a result, the input samples should be normalized first and then selected the neurons pass for hidden layer and output layer as tansig() and purelin(), respectively [9]. Meanwhile, the training function trainlm in LM (Levenberg-Marquardt) algorithm was selected for training network when convergence speed and accuracy were taken into consideration.

3.4. Neural Network Generalization Test. Typically, the training patterns included in the training set are only part of the source data set. Even if the network was trained by all the patterns within the training set, it also could not guarantee that the test by another mode could give satisfactory results. Therefore, the generalization ability of fitting functions of 25 new samples, as a test set, is still needed to be examined after the network learning. From the result of test error (Figure 3), the fitted model met the requirement of accuracy and could replace the finite element simulation for reliability analysis.

In addition, the convergence of the neural network training was demonstrated (Figure 4).

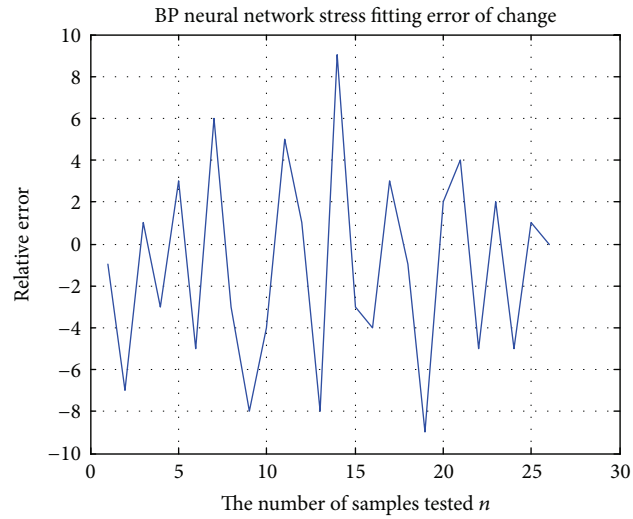


Figure 3: Relative error of BP neural network fitted model.

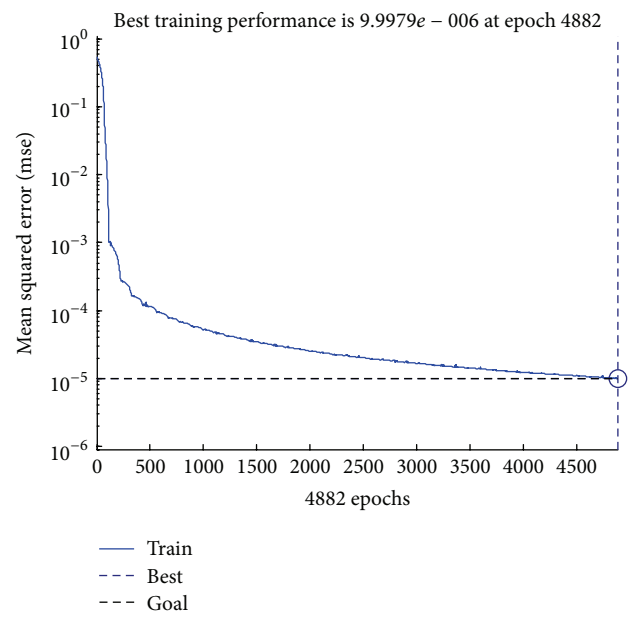


Figure 4: Grid training convergence curve.

4. Reliability Analysis of Bridge Crane Metal Structure

A set of basic random variables $X = (X_1, X_2, \ldots, X_n)^T$ features the design dimensions, the material properties, the load that are normally distributed. Its joint probability density function is $f_X(X)$. The limit state function of the bridge crane metal structure is expressed as

$$g(X) = g(X_1, X_2, \ldots, X_n). \tag{1}$$

$g(X) < 0$ represents the failure state, $g(X) = 0$ is limit state, and $g(X) > 0$ is the security state. Based on the probability theory, the reliability is calculated using multiple integrals in reliable domain when $g(X) > 0$; namely,

$$R = \int_{g(X)>0} \cdots \int I[g(x)] f(x_1, x_2, \ldots, x_n) dx_1 dx_2 \cdots dx_n, \tag{2}$$

when $g(X) > 0$, $I[g(X) > 0] = 1$; when $g(X) < 0$, $I[g(X) < 0] = 0$.

Based on the perturbation theory, X and $g(X)$ can be expanded as follows:

$$\begin{aligned} X &= X_d + \Delta X_p, \\ g(X) &= g_d(X) + \Delta g_p(X), \end{aligned} \tag{3}$$

where subscript d and p separately denote the determining part and random part of random variable and have zero mean; Δ denotes higher-order infinitesimal. Based on the random analysis theory, the mean and variance of formula (3) are as follows:

$$\begin{aligned} \mu_g &= E[g(X)] = E[g_d(X)] + \Delta E[g_p(X)] = g_d(X), \\ \sigma_g^2 &= \mathrm{Var}[g(X)] = \Delta^2 E\left[\left(\frac{\partial g_d(X)}{\partial X^T}\right)\Delta X_p\right], \\ &= \left(\frac{\partial g_d(X)}{\partial X^T}\right)\mathrm{Var}(X). \end{aligned} \tag{4}$$

Based on the FOSM method, take a linear term of the Taylor expansion of limit state equation and calculate the probability of the function greater than zero by basic random variables of

first moment (mean) and second moment (variance). Reliability is expressed as R:

$$R = \Phi(\beta) = \Phi\left(\frac{\mu_g}{\sigma_g}\right) = \Phi\left(\frac{E[g(X)]}{\sqrt{\mathrm{Var}[g(X)]}}\right). \quad (5)$$

Set σ for the corresponding stress intensity of the dangerous point of bridge machine metal structure. Then, the limit state function is written as

$$g(X) = \sigma - S(W, H, B, E, e_YYB, e_FB). \quad (6)$$

5. Reliability Sensitivity Analysis of Bridge Crane Metal Structure

Based on the FOSM method and matrix differential, the mean and variance sensitivity of the metal structure reliability to each random variable are expressed as follows:

$$\begin{aligned} \frac{dR_i}{d\overline{X}^T} &= \frac{\partial R_i}{\partial \beta_i}\frac{\partial \beta_i}{\partial \mu_{g_i}}\frac{\partial \mu_{g_i}}{\partial \overline{X}^T}, \quad (i = 1, 2, \ldots, n), \\ \frac{dR_i}{d\,\mathrm{Var}\,X} &= \frac{\partial R_i}{\partial \beta_i}\frac{\partial \beta_i}{\partial \sigma_{g_i}}\frac{\partial \sigma_{g_i}}{\partial\,\mathrm{Var}\,X}, \quad (i = 1, 2, \ldots, n), \end{aligned} \quad (7)$$

where

$$\begin{aligned} \frac{\partial R_i}{\partial \beta_i} &= \varphi(\beta_i), \qquad \frac{\partial \beta_i}{\partial \mu_{g_i}} = \frac{1}{\sigma_{g_i}}, \\ \frac{\partial \mu_{g_i}}{\partial \overline{X}^T} &= \left[\frac{\partial \overline{g}_i}{\partial X_1}\frac{\partial \overline{g}_i}{\partial X_2}\cdots\frac{\partial \overline{g}_i}{\partial X_n}\right], \\ \frac{\partial \beta_i}{\partial \sigma_{g_i}} &= -\frac{\mu_{g_i}}{\sigma_{g_i}^2}, \\ \frac{\partial \sigma_{g_i}}{\partial\,\mathrm{Var}\,X} &= \frac{1}{2\sigma_{g_i}}\left[\frac{\partial \overline{g}_i}{\partial X} \otimes \frac{\partial \overline{g}_i}{\partial X}\right]. \end{aligned} \quad (8)$$

6. Results and Analysis

According to formula (5), the reliability of the metal structure is $R = 0.999866$. Use Monte Carlo simulation for test to calculate 106 times. The reliability is $R_{\mathrm{MCS}} = 0.999764$. Relative error is as follows:

$$\varepsilon_R = \left|\frac{R - R_{\mathrm{MSC}}}{R_{\mathrm{MCS}}}\right| = 1.306 \times 10^{-4}. \quad (9)$$

Based on the known conditions and reliability calculation results, the reliability sensitivity of random variables including dR_i/dX^T and $dR_i/d\,\mathrm{Var}\,X$ can be calculated according to formula (7) [10]. The mean sensitivity of the bridge machine metal structure is as follows:

$$\frac{dR}{dX^T} = \begin{bmatrix} R(W) \\ R(H) \\ R(B) \\ R(E) \\ R(e_YYB) \\ R(e_FB) \end{bmatrix} = \begin{bmatrix} 4.043 \times 10^{-3} \\ 1.692 \times 10^{-4} \\ 1.432 \times 10^{-4} \\ 3.453 \times 10^{-5} \\ 3.308 \times 10^{-7} \\ 4.682 \times 10^{-8} \end{bmatrix}. \quad (10)$$

Variance sensitivity is as follows:

$$\frac{dR}{d\,\mathrm{Var}\,X} = \begin{bmatrix} R_{\mathrm{Var}}(W) \\ R_{\mathrm{Var}}(H) \\ R_{\mathrm{Var}}(B) \\ R_{\mathrm{Var}}(E) \\ R_{\mathrm{Var}}(e_YYB) \\ R_{\mathrm{Var}}(e_FB) \end{bmatrix} = \begin{bmatrix} -6.385 \times 10^{-10} \\ -4.902 \times 10^{-11} \\ -7.746 \times 10^{-13} \\ -8.658 \times 10^{-11} \\ -3.537 \times 10^{-7} \\ -6.135 \times 10^{-9} \end{bmatrix}. \quad (11)$$

It can be drawn from the sensitivity matrix of dR_i/dX^T that the reliability sensitivity of bridge crane metal structure can be influenced by lifting load W, main beam height H, main beam width B, and elastic modulus E [10]. Moreover, the influence degree of 6 variables on structural reliability in decreasing order are lifting load W, main beam height H, main beam width B, elastic modulus E, flange plate thickness e_YYB, and web thickness e_FB.

7. Conclusions

(i) The study applied the neural network technology and orthogonal experimental design method to solve the reliability calculation with the implicit limit state function and to obtain the explicit expression of the random variable and the stress response of bridge crane metal structure.

(ii) The integrated use of orthogonal experimental design method and finite element simulation test in the reliability analysis engineering can significantly reduce the design costs and markedly shorten the cycle.

(iii) Based on the theory of reliability design and sensitivity, we made a derivation analysis of the reliability and its sensitivity and then established a basis for accurately quantitative analysis of the reliability of bridge crane metal structure.

Acknowledgments

The project is supported by the Natural Science Foundation Committee (51075289); National Natural Science Foundation International Cooperation and Exchanges Project (51110105011); 2012 Higher School Specialized Research Fund for the Doctoral Program Joint Funding Issues (20121415110004); Research Foundation for Returning Scholar in 2009 of Shanxi province (20091074); International Scientific and Technological Cooperation Projects in 2010 of Shanxi province (2010081039); Shanxi Natural Science Foundation (2011011019-3); 2010 Scientific Star Project in Taiyuan

city (2010011605); 2009 Shanxi province Graduates' Excellent and Innovative Project (20093099); Shanxi province 2011 UIT Item, TYUST 2010 UIT Item (201008X); Doctor Start up Item and Characteristics & Leading Academic Discipline Project of Universities of Shanxi Province.

References

[1] D. N. Thatoi, H. C. Das, and D. R. Parhi, "Review of techniques for fault diagnosis in damaged structure and engineering system," *Advances in Mechanical Engineering*, vol. 2012, Article ID 327569, 11 pages, 2012.

[2] N. Saravanan, V. N. S. K. Siddabattuni, and K. I. Ramachandran, "Fault diagnosis of spur bevel gear box using artificial neural network (ANN), and proximal support vector machine (PSVM)," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 344–360, 2010.

[3] P. Zeng, L. Lei, and G. Fang, *Finite Element Analysis Guide: Modeling and Analysis of Structure*, China Machine Press, Beijing, China, 2013 (Chinese).

[4] X. He and Y. Zhang, "Reliability sensitivity design of mechanical systems," *Machinery Manufacturing*, vol. 5, pp. 26–28, 2009 (Chinese).

[5] M. Hohenbichler and R. Rackwitz, "Sensitivity and importance measures in structural reliability," *Civil Engineering Systems*, vol. 3, no. 4, pp. 203–209, 1986.

[6] K. Xu, M. Xie, L. C. Tang, and S. L. Ho, "Application of neural networks in forecasting engine systems reliability," *Applied Soft Computing Journal*, vol. 2, no. 4, pp. 255–268, 2003.

[7] T. Chen, H. Chen, and R.-W. Liu, "Approximation capability in C(Rn) by multilayer feedforward networks and related problems," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 25–30, 1995 (Chinese).

[8] K. Zhou and Y. Kang, *The Neural Network Model and Its MATLAB Simulation Program Designed*, Tsinghua University Press, Beijing, China, 2005.

[9] G. I. Schuëller, "On the treatment of uncertainties in structural mechanics and analysis," *Computational Stochastic Mechanics*, vol. 85, no. 5-6, pp. 235–243, 2007.

[10] H. Xiangdong, Z. Yimin, and W. Bangchun, "Reliability sensitivity design of mechanical systems," *Machinery Manufacturing*, vol. 47, no. 5, pp. 26–28, 2009.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 407096, 24 pages
http://dx.doi.org/10.1155/2013/407096

Research Article

Multimodal Processes Rescheduling: Cyclic Steady States Space Approach

Grzegorz Bocewicz,[1] Robert Wójcik,[2] Zbigniew Antoni Banaszak,[3] and Paweł Pawlewski[4]

[1] *Department of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland*
[2] *Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Wrocław, Poland*
[3] *Department of Business Informatics, Warsaw University of Technology, Warsaw, Poland*
[4] *Faculty of Engineering Management, Poznań University of Technology, Poznań, Poland*

Correspondence should be addressed to Grzegorz Bocewicz; bocewicz@ie.tu.koszalin.pl

Received 29 March 2013; Accepted 22 September 2013

Academic Editor: Zhichun Yang

Copyright © 2013 Grzegorz Bocewicz et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The paper concerns cyclic scheduling problems arising in a Multimodal Transportation Network (MTN) in which several unimodal networks (AGVs, hoists, lifts, etc.) interact with each other via common shared workstations as to provide a variety of demand-responsive material handling operations. The material handling transport modes provide movement of work-pieces between workstations along their manufacturing routes in the MTN. The goal is to provide a declarative framework enabling to state a constraint satisfaction problem aimed at AGVs fleet match-up scheduling taking into consideration assumed itineraries of concurrently manufactured product types. In that context, treating the different product types as a set of cyclic multimodal processes is the main objective to discuss the conditions sufficient for FMS rescheduling imposed by production orders changes. To conclude, the conditions sufficient for an FMS rescheduling imposed by changes of production orders treated as cyclic multimodal processes are stated as the paper's main contribution.

1. Introduction

The design and operational issues arising in an Automated Guided Vehicle (AGV) served Flexible Manufacturing System (FMS) when deciding on how to achieve a desired system performance are usually hard to determine and evaluate. This is because productivity of an AGV-served flow shop producing a set of different kinds of products, repetitively, depends on both the job flow sequencing (aimed at maximizing throughput or minimizing cycle time) and the material handling system required to achieve a prespecified throughput, that is, AGVs fleet sizing, assignment and scheduling [1–5]. So, assuming a deterministic system where demand is known in advance and the processing times of each job on each machine is a known constant, in order to improve its productivity, the AGV fleet size and the scheduling/dispatching influence on the system throughput are usually examined [1, 6–9].

In opposite to conventionally applied approach assuming AGVs fleet sizing, routing, and timetabling while matching up prescheduled multiproduct manufacturing flow (i.e., providing detailed transportation route and materials load/unload schedule between workstations in FMS), we propose AGVs following a Multimodal Transportation Network (MTN) concept. MTN can be seen as composition of several unimodal networks (AGVs, hoists, lifts, etc.) interacting with each other via common shared workstations as to provide a variety of demand-responsive material handling operations. The material handling transport modes provide movement of work-pieces between workstations along their manufacturing routes in the MTN. The MTN encompasses a well-known multimodal approach to freight transportation supply chains and passenger transportation systems. So, similarly to such systems having ability to use the appropriate mode of transport for movement of goods and people, the assumed material

handling transport modes provide movement of work-pieces between workstations along their manufacturing routes in the MTN. In that context, assuming that the modes can be treated as cyclic local transportation processes, and material (e.g., work-pieces/tools) flows supported by them can be treated as multimodal processes, the problems considered can be seen either as aimed at designing of a multimodal network infrastructure guaranteeing assumed materials flow or focused on work-pieces routing ensuring lag-free workstations service.

The problems arising concerning material transportation routing and scheduling belong to NP-hard ones [2]. Since the steady state of production flows has a cyclic character, hence servicing them AGV-served transportation processes (usually executed along loop-like routes) encompasses also cyclic behavior. That means that the periodicity of FMS [10] depends on both the periodicity of production flow cyclic schedule and following this schedule the AGVs periodicity. Of course, the Multimodal Transportation Network (MTN) throughput is maximized by the minimization of its cycle time.

It seems to be obvious that not all the behaviors (including cyclic ones) are reachable under constraints imposed by the system's structure. The similar observation concerns the system's behavior that can be achieved in systems possessing specific structural constraints. Since system constraints determine its behavior, both the system structure and the desired cyclic schedule have to be considered simultaneously. So, the problem solution requires that the system structure must be determined for the purpose of processes scheduling, yet scheduling must be done to devise the system configuration. In that context, our contribution provides a discussion of some solvability issues concerning cyclic processes dispatching problems, especially the conditions guaranteeing solvability of the cyclic processes scheduling as well as direct and/or indirect transitions between assumed cyclic steady states. Their examination may replace exhaustive searches for solution satisfying required system capabilities.

Many models and methods have been considered to date [11]. Among them, the mathematical programming approach [1, 12], max-plus algebra [13], constraint logic programming [14–16] evolutionary algorithms, Petri nets [17, 18], and heuristic frameworks [19] belong to the more frequently used. Most of these are oriented at finding a minimal cycle or maximal throughput while assuming deadlock-free processes flow. Note that processes' operations are blocking if they must stay on a resource (e.g., the station, the machine) after finishing when the next resource is occupied by a job from another process. During this stay the resource is blocked for other processes. The approaches trying to estimate the cycle time from cyclic processes structure and the synchronization mechanism employed (i.e., mutual exclusion instances) while taking into account deadlock phenomena are quite unique.

In our approach a declarative framework aimed at refinement and prototyping of the cyclic steady states for concurrently executed cyclic processes modelling material handling systems is employed. These systems are of the type of AGVs fleets in the flexible manufacturing systems (FMS) and are frequently encountered in industry. The following questions are the main focus of the research. Can the assumed material handling system, for example, AGVs, meet load/unload deadlines imposed by flow of scheduled work-pieces processing? Does there exist sufficient AGVS enabling to schedule the AGVs fleet as to ensure lag-free service of scheduled work pieces processing? So, the main question is Can the assumed AGVs fleet assignment reach its goal subject to constraints assumed on concurrent multiproduct manufacturing at hand?

Moreover the declarative framework enables to state a *problem of multimodal processes rescheduling* which boils down to searching for transient periods allowing the mutual reachability of MTN cyclic behaviors. In the case of MTNs, distinguishing many different cyclic steady states (cyclic behaviors), the following questions play a pivotal role. Does there exist the smooth (direct) transition between two assumed cyclic behaviors? What conditions guarantee the reachability of a given cyclic behavior from any other ones? Is it possible to come back to a given cyclic behavior from any disturbance-born one, for instance, caused by operation time delays, damage of devices, route modifications, and so forth?

In other words, the paper's objective concerns the MTN infrastructure assessment from the perspective of possible FMS oriented requirements imposed by AGVs fleet assignment, sizing, and scheduling (problems of cyclic processes scheduling and rescheduling). Due to the complexity implied in answering the above questions, the combined problem remains unsolved for all practical purposes. This is especially problematic as many manufacturing companies would stand to reap considerable rewards through better fleet assignment and scheduling. The presented approach addresses this through solving the combined rather than the separate problems individually.

The rest of the paper is organized as follows. Section 2 introduces the Automated Guided Vehicles System (AGVS) modelled in terms of concurrently flowing cyclic processes (SCCPs) and a cyclic steady state space concept. Section 3 regarding multimodal processes rescheduling provides a problem statement concerning AGVs fleet match-up scheduling with an assumed multiproduct manufacturing flow schedule. Sufficient conditions allowing one to search for states with mutual allocation as well as illustrative example of implementing them algorithm usage are discussed. The related work and concluding remarks are presented in Sections 4 and 5, respectively.

2. Multimodal Network Modeling

Considered case of multimodal processes rescheduling problem is presented on the example of Automated Guided Vehicle Systems (AGVS) modeled by Systems of Concurrently Flowing Cyclic Processes (SCCPs). To solve this kind of problems a nonempty state space, that is, containing nonempty set of possible cyclic behaviors, is required. For that reason the conditions guaranteeing cyclic behavior are presented, primarily. They are specified in terms of a declarative model formalism distinguishing the structure and behavior as the

main components of an SCCP. Relationship linking the structure and reachable behaviors is the main subject of this section.

2.1. Model of AGVS. Automated Guided Vehicle Systems are used for material handling within a Flexible Manufacturing System (FMS) and provide asynchronous movement pallets of products through a network of guide paths between the workstations by the AGVs. Each workstation is connected to the guide path network by a pick-up/delivery station where pallets are transferred from/to the AGVs.

In AGVS literature, most are related to AGVS design issues, which include determination of the number of vehicles required, flow path design, and route planning as well as vehicle dispatching and traffic management. Recently, an integrated problem of dispatching and conflict free routing of AGVs, that is, integrating the simultaneous assignment, scheduling, and conflict free routing of the vehicles, is receiving increasing attention. The above mentioned problems, in the general case, belong to the class of NP-hard problems. Since most processes observed in steady state manufacturing are periodic, cyclic schedules and following them, cyclic scheduling methods can be considered. Since cyclic schedules encompass repetitive character of manufacturing processes, the cyclic processes modeling approach seems to be a reasonable perspective [13–15].

To present some of the design and operational issues that arise in repetitive manufacturing systems served by AGVs in network of loops layout with unidirectional material flow, a case example of a simple FMS is shown in Figure 1. The concurrent multiproduct flows are depicted by bold green, blue, and orange color lines, while the transportation loop-like networks are distinguished by double solid line. Both kinds of material (jobs) and transportation (AGVs) flows shown in Figure 1 can be modeled in terms of Systems of Concurrently Flowing Cyclic Processes [14, 15] as shown in Figure 2. In turn, the SCCP framework provides a formal model enabling to state and resolve problems of AGV fleet size minimization as well as steady state cycle time minimization. The cycle time minimization is required to obtain maximum throughput rate.

Eight *local cyclic processes* are considered, namely, P_1, P_2, P_3, P_4, P_5, P_6, P_7, and P_8. The processes follow the *routes* are composed of transportation sectors and machines (distinguished in Figure 2 by the *set of resources* $R = \{R_1, \ldots, R_c, \ldots, R_{33}\}$, R_c—the cth resource). Some of the local cyclic processes are pipeline flow processes; that is, they contain *streams* (representing vehicles from Figure 1) of the processes following the same route while occupying different resources (sectors). For instance, processes P_4, P_5, and P_6 contain two streams: $P_4 = \{P_4^1, P_4^2\}$, $P_5 = \{P_5^1, P_5^2\}$, and $P_6 = \{P_6^1, P_6^2\}$, respectively, and P_2 contains three streams: $P_2 = \{P_2^1, P_2^2, P_2^3\}$, that is, the processes (vehicles) moving along the same route. The remaining local processes contain unique streams: $P_1 = \{P_1^1\}$, $P_3 = \{P_3^1\}$, $P_7 = \{P_7^1\}$, and $P_8 = \{P_8^1\}$. In other words, the streams P_1^1, P_2^1, P_2^2, P_2^3, P_3^1, P_4^1, P_4^2, P_5^1, P_5^2, P_6^1, P_6^2, P_7^1, and P_8^1 represent the 13 vehicles from Figure 1. The kth stream of the ith local process P_i is denoted by P_i^k.

Apart from local processes, we consider three *multimodal processes* (i.e., processes executed along the routes consisting of parts of the routes of local processes): mP_1, mP_2, and mP_3.

For example, the production route depicted by the orange line corresponds to the multimodal process mP_1 supported by AGVs (arbitrarily given), which in turn encompass local transportation streams P_1^1, P_2^1, P_2^3, and P_4^2. This means that the production route specifying how a multimodal process is executed can be considered to be composed of parts of the routes of local cyclic processes.

In the system considered each multimodal process consists of four streams: $mP_i = \{mP_i^1, mP_i^2, mP_i^3, mP_i^4\}$, $i = 1, 2, 3$, which means that along each production route four work- pieces are processed serially (four balls on the one production lines in Figure 1).

Processes can interact with each other through shared resources, that is, the transportation sectors. The routes of the considered local processes (streams) are as follows (see Figure 2):

$$\begin{aligned}
p_1^1 &= (R_{16}, R_3, R_{13}, R_6), \\
p_2^1 &= (R_{21}, R_8, R_{18}, R_9, R_{14}, R_6, R_{17}, R_5), \\
p_2^2 &= (R_{17}, R_5, R_{21}, R_8, R_{18}, R_9, R_{14}, R_6), \\
p_2^3 &= (R_{14}, R_6, R_{17}, R_5, R_{21}, R_8, R_{18}, R_9), \\
p_3^1 &= (R_{12}, R_{19}, R_9, R_5), \\
p_4^1 &= (R_{29}, R_7, R_{25}, R_8, R_{22}, R_{11}, R_{26}, R_{10}), \\
p_4^2 &= (R_{26}, R_{10}, R_{29}, R_7, R_{25}, R_8, R_{22}, R_{11}), \\
p_5^1 &= (R_2, R_{20}, R_5, R_{24}, R_4, R_{27}, R_1, R_{23}), \\
p_5^2 &= (R_{27}, R_1, R_{23}, R_2, R_{20}, R_5, R_{24}, R_4), \\
p_6^1 &= (R_{32}, R_4, R_{28}, R_7, R_{31}), \\
p_6^2 &= (R_{31}, R_{32}, R_4, R_{28}, R_7), \\
p_7^1 &= (R_{33}, R_7, R_{29}, R_{10}), \\
p_8^1 &= (R_1, R_{27}, R_4, R_{30}),
\end{aligned} \tag{1}$$

where $R_1 - R_2$, $R_4 - R_{11}$, R_{14}, $R_{17} - R_{18}$, $R_{20} - R_{29}$, R_{31}, and R_{32} (due to two streams of process P_6, resources R_{31}, R_{32} represent one sector) are resources shared by local processes, that is, by at least two streams, and R_3, R_{12}, R_{13}, R_{15}, R_{16}, R_{19}, R_{30}, and R_{33} are the nonshared resources, that is, exclusively used by one stream only.

In the general case, the route p_i^k is the sequence of resources used in order to execute the operations of the stream P_i^k. Note that the streams P_2^1, P_2^2, and P_2^3 which belong to P_2 and the streams of processes P_4, P_5, and P_6 follow the same route but start from different resources (these streams correspond to vehicles moving along the same route).

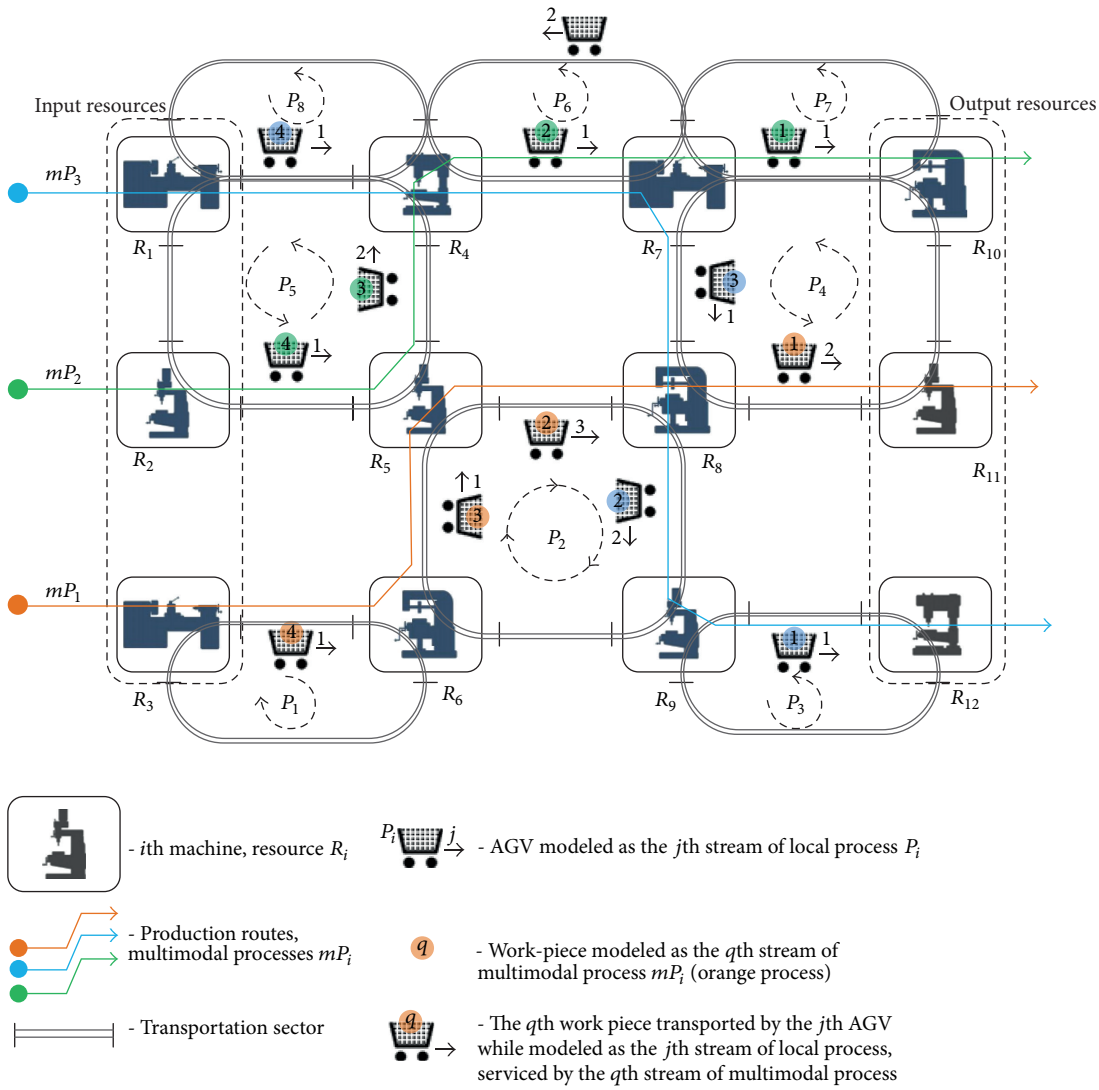


Figure 1: Example of an AGVS.

Similarly the cyclic multimodal processes mP_1, mP_2, and mP_3 follow the routes (see Figure 2) as follows:

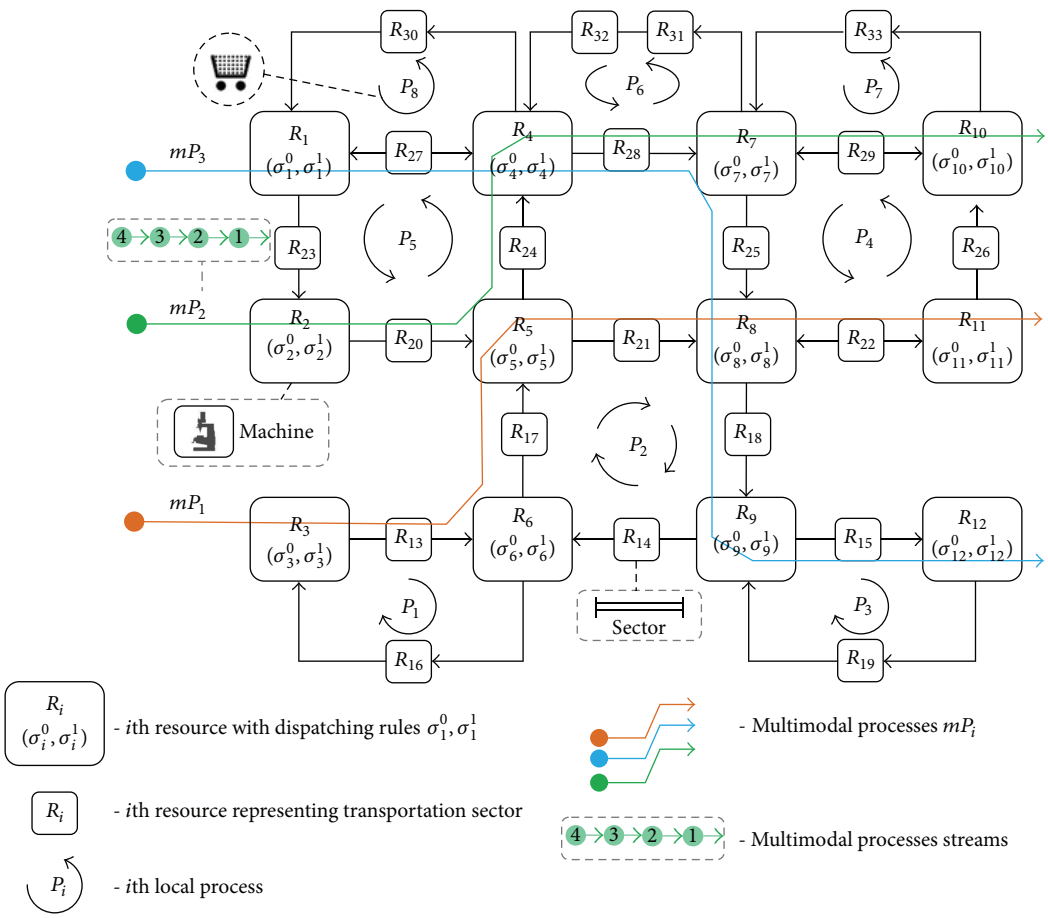
$$\begin{aligned} mp_1^k &= (R_3, R_{13}, R_6, R_{17}, R_5, R_{21}, R_8, R_{22}, R_{11}), \quad k = 1, \ldots, 4, \\ mp_2^k &= (R_2, R_{20}, R_5, R_{24}, R_4, R_{28}, R_7, R_{29}, R_{10}), \quad k = 1, \ldots, 4, \\ mp_3^k &= (R_1, R_{27}, R_4, R_{28}, R_7, R_{25}, R_8, R_{18}, R_9, R_{15}, R_{12}), \quad k = 1, \ldots, 4. \end{aligned} \tag{2}$$

Let us assume that mp_1^k, mp_2^k, and mp_3^k can be seen as follows (see Figure 2):

$$\begin{aligned} mp_1^k &= ((R_3, R_{13}, R_6), (R_{17}, R_5), (R_{21}, R_8), (R_{22}, R_{11})), \quad k = 1, \ldots, 4, \\ mp_2^k &= ((R_2, R_{20}, R_5), (R_{24}, R_4), (R_{28}, R_7), (R_{29}, R_{10})), \quad k = 1, \ldots, 4, \\ mp_3^k &= ((R_1, R_{27}, R_4), (R_{28}, R_7), (R_{25}, R_8), (R_{18}, R_9), (R_{15}, R_{12})), \quad k = 1, \ldots, 4, \end{aligned} \tag{3}$$

where one has the following: (R_3, R_{13}, R_6), (R_{17}, R_5), (R_{21}, R_8), and (R_{22}, R_{11})—subsequences of routes p_1^1, p_2^1, p_2^3, and p_4^2 defining the transportation sections of mp_1^k; (R_2, R_{20}, R_5), (R_{24}, R_4), (R_{28}, R_7), and (R_{29}, R_{10})—subsequences of routes p_5^1, p_5^2, p_6^1, and p_7^1 defining the transportation sections of mp_2^k; and (R_1, R_{27}, R_4), (R_{28}, R_7), (R_{25}, R_8), (R_{18}, R_9), and (R_{15}, R_{12})—subsequences of routes p_8^1, p_6^2, p_4^1, p_2^2, and p_3^1 defining the transportation sections of mp_3^k.

Let us assume also that the considered SCCPs follow the constraints stated below [15]:



Dispatching rules Θ^0					
σ_1^0	(P_5^2, P_8^1, P_5^1)	σ_{12}^0	(P_3^1)	σ_{23}^0	(P_5^2, P_5^1)
σ_2^0	(P_5^1, P_5^2)	σ_{13}^0	(P_1^1)	σ_{24}^0	(P_5^1, P_5^2)
σ_3^0	(P_1^1)	σ_{14}^0	(P_2^3, P_2^1, P_2^2)	σ_{25}^0	(P_4^1, P_4^2)
σ_4^0	$(P_6^1, P_8^1, P_6^2, P_5^1, P_5^2)$	σ_{15}^0	(P_3^1)	σ_{26}^0	(P_4^2, P_4^1)
σ_5^0	$(P_2^2, P_2^3, P_5^1, P_5^2, P_2^1)$	σ_{16}^0	(P_1^1)	σ_{27}^0	(P_5^2, P_8^1, P_5^1)
σ_6^0	$(P_2^3, P_2^1, P_1^1, P_2^2)$	σ_{17}^0	(P_2^2, P_2^3, P_2^1)	σ_{28}^0	(P_6^1, P_6^2)
σ_7^0	$(P_4^1, P_4^2, P_6^1, P_7^1, P_6^2)$	σ_{18}^0	(P_2^1, P_2^2, P_2^3)	σ_{29}^0	(P_4^1, P_4^2, P_7^1)
σ_8^0	$(P_2^1, P_4^1, P_2^2, P_2^3, P_4^2)$	σ_{19}^0	(P_3^1)	σ_{30}^0	(P_8^1)
σ_9^0	$(P_2^1, P_2^2, P_3^1, P_2^3)$	σ_{20}^0	(P_5^1, P_5^2)	σ_{31}^0	(P_6^2, P_6^1)
σ_{10}^0	(P_4^2, P_7^1, P_4^1)	σ_{21}^0	(P_2^1, P_2^2, P_2^3)	σ_{32}^0	(P_6^1, P_6^2)
σ_{11}^0	(P_4^1, P_4^2)	σ_{22}^0	(P_4^1, P_4^2)	σ_{33}^0	(P_7^1)

Dispatching rules Θ^1					
σ_1^1	$(mP_3^1, mP_3^2, mP_3^3, mP_3^4)$	σ_{12}^1	$(mP_3^3, mP_3^4, mP_3^1, mP_3^2)$	σ_{23}^1	$\varnothing$
σ_2^1	$(mP_2^1, mP_2^2, mP_2^3, mP_2^4)$	σ_{13}^1	$(mP_1^1, mP_1^2, mP_1^3, mP_1^4)$	σ_{24}^1	$(mP_2^1, mP_2^2, mP_2^3, mP_2^4)$
σ_3^1	$(mP_1^1, mP_1^2, mP_1^3, mP_1^4)$	σ_{14}^1	$\varnothing$	σ_{25}^1	$(mP_3^3, mP_3^4, mP_3^1, mP_3^2)$
σ_4^1	$(mP_3^4, mP_2^4, mP_3^1, mP_2^1$ $mP_3^2, mP_2^2, mP_3^1, mP_2^1)$	σ_{15}^1	$(mP_3^3, mP_3^4, mP_3^1, mP_3^2)$	σ_{26}^1	$\varnothing$
σ_5^1	$(mP_1^3, mP_2^4, mP_1^4, mP_2^1$ $mP_1^1, mP_2^2, mP_1^2, mP_2^3)$	σ_{16}^1	$\varnothing$	σ_{27}^1	$(mP_3^1, mP_3^2, mP_3^3, mP_3^4)$
σ_6^1	$(mP_1^1, mP_1^2, mP_1^3, mP_1^4)$	σ_{17}^1	$(mP_1^1, mP_1^2, mP_1^3, mP_1^4)$	σ_{28}^1	$(mP_3^4, mP_2^4, mP_3^1, mP_2^1$ $mP_3^2, mP_2^2, mP_3^1, mP_2^1)$
σ_7^1	$(mP_2^3, mP_3^4, mP_2^4, mP_3^1$ $mP_2^1, mP_3^2, mP_2^2, mP_3^3)$	σ_{18}^1	$(mP_3^3, mP_3^4, mP_3^1, mP_3^2)$	σ_{29}^1	$(mP_2^3, mP_2^4, mP_2^1, mP_2^2)$
σ_8^1	$(mP_3^3, mP_1^3, mP_3^4, mP_1^4$ $mP_3^1, mP_1^1, mP_3^2, mP_1^2)$	σ_{19}^1	$\varnothing$	σ_{30}^1	$\varnothing$
σ_9^1	$(mP_3^3, mP_3^4, mP_3^1, mP_3^2)$	σ_{20}^1	$(mP_2^1, mP_2^2, mP_2^3, mP_2^4)$	σ_{31}^1	$\varnothing$
σ_{10}^1	$(mP_2^3, mP_2^4, mP_2^1, mP_2^2)$	σ_{21}^1	$(mP_1^3, mP_1^4, mP_1^1, mP_1^2)$	σ_{32}^1	$\varnothing$
σ_{11}^1	$(mP_1^3, mP_1^4, mP_1^1, mP_1^2)$	σ_{22}^1	$(mP_1^3, mP_1^4, mP_1^1, mP_1^2)$	σ_{33}^1	$\varnothing$

Figure 2: Example of FMS-SCCP model of an AGVS.

(i) a new subsequent operation of the local process may start on a required resource only if the current operation has been completed and the resource has been released;

(ii) each new consecutive operation in the multimodal process may start its execution on an assigned resource only if the current operation has been completed and the resource has been released and an

appropriate local process begins its consecutive operation on this resource;

(iii) local/multimodal processes share common resources in the mutual exclusion mode, the operation of a local/multimodal process can only be suspended if the necessary resource is occupied, suspended processes cannot be released, and processes are nonperceptible; that is, a resource may not be taken from a process as long as it is used by it;

(iv) the SCCP resources can be shared by local and multimodal processes as well as by both of them;

(v) multimodal processes encompassing production flow conveyed by AGVs follow local transportation routes;

(vi) different multimodal processes can be executed simultaneously along the same local process;

(vii) local and multimodal processes execute cyclically with periods α and $m\alpha$, respectively; resources occur uniquely in each transportation route;

(viii) in a cyclic steady state, each ith stream must cover its local route the same number of times.

A resource conflict (caused by the application of the mutual exclusion protocol) is resolved with the aid of a priority dispatching rule [14–16], which determines the order in which streams access shared resources. For instance, in the case of the resource R_5, the priority dispatching rule $\sigma_5^0 = (P_2^2, P_2^3, P_5^1, P_5^2, P_2^1)$ determines the order in which streams of local processes can access the shared resource R_5; in the case considered the stream P_2^2 is allowed to access as first, then the stream P_2^3 as next, then streams P_5^1, P_5^2, and P_2^1, and then once again P_2^2, and so on. The stream P_i^k occurs the same number of times (in the considered system once) in each dispatching rule associated with the resources featuring in its route (in the case considered each stream P_i^k occurs uniquely). The SCCP shown in Figure 2 is specified by the following set of dispatching rules: $\Theta = \{\Theta^0, \Theta^1\}$, where $\Theta^0 = \{\sigma_1^0, \ldots, \sigma_{33}^0\}$ is set of rules determining the orders of local processes and $\Theta^1 = \{\sigma_1^1, \ldots, \sigma_{33}^1\}$ is set of rules determining the orders of multimodal processes.

In general, the following notation is used.

(i) A sequence $p_i^k = (p_{i,1}^k, p_{i,2}^k, \ldots, p_{i,j}^k, \ldots, p_{i,lr(i)}^k)$ specifies *the route of the stream of a local process* P_i^k (the kth stream of the ith local process P_i). Its components define the resources used in the execution of operations, where $p_{i,j}^k \in R$ (the set of resources $R = \{R_1, R_2, \ldots, R_c, \ldots, R_m\}$) denotes the resources used by the kth stream of the ith local process in the jth operation; in the rest of the paper, *the jth operation executed on the resource* $p_{i,j}^k$ *in the stream* P_i^k will be denoted by $o_{i,j}^k$; $lr(i)$ is the length of the cyclic process route (all streams of P_i are of the same length). For example, the route $p_1^1 = (R_{16}, R_3, R_{13}, R_6)$ of the stream of process P_1 (Figure 2) is the sequence $p_1^1 = (p_{1,1}^1, p_{1,2}^1, p_{1,3}^1, p_{1,4}^1)$, where the first element $p_{1,1}^1$ is equal to R_{16}, whereas the second, $p_{1,2}^1 = R_3$, and so on, $p_{1,3}^1 = R_{13}$, $p_{1,4}^1 = R_6$.

(ii) $x_{i,j}^k(l) \in \mathbb{N}$ is the timing of commencement of operation $o_{i,j}^k$ in the lth cycle.

(iii) $t_i^k = (t_{i,1}^k, t_{i,2}^k, \ldots, t_{i,j}^k, \ldots, t_{i,lr(i)}^k)$ specifies *the operation times of local processes*, where $t_{i,j}^k$ denotes the time of execution of operation $o_{i,j}^k$.

(iv) $mp_i^k = (mpr_{i_1}^{q_1}(a_{i_1}, b_{i_1}), mpr_{i_2}^{q_2}(a_{i_2}, b_{i_2}), \ldots, mpr_{i_y}^{q_y}(a_{i_y}, b_{i_y}))$ specifies *the route of the stream* mP_i^k *from the multimodal process* mP_i (the kth stream of the ith multimodal process mP_i), where

$$mpr_i^q(a,b) = \begin{cases} \left(p_{i,a}^q, p_{i,a+1}^q, \ldots, p_{i,b}^q\right), & a \le b, \\ \left(p_{i,a}^q, p_{i,a+1}^q, \ldots, p_{i,lr(i)}^q, p_{i,1}^q, \ldots, p_{i,b-1}^q, p_{i,b}^q\right) & a > b, \end{cases} \quad a, b \in \{1, \ldots, lr(i)\}, \tag{4}$$

is the subsequence of the route $p_i^q = (p_{i,1}^q, p_{i,2}^q, \ldots, p_{i,j}^q, \ldots, p_{i,lr(i)}^q)$ containing elements from $p_{i,a}^q$ to $p_{i,b}^q$. In other words, the transportation route mp_i is a sequence of parts of routes of local processes. For instance, the route followed by process mP_1 (see Figure 2) is as follows: $mp_1^k = ((R_3, R_{13}, R_6), (R_{17}, R_5), (R_{21}, R_8), (R_{22}, R_{11}))$, where $mpr_1^1(2,4) = (R_3, R_{13}, R_6)$; $mpr_2^1(7,8) = (R_{17}, R_5)$; $mpr_2^3(1,2) = (R_{21}, R_8)$, $mpr_4^2(5,6) = (R_{22}, R_{11})$.

(v) $mx_{i,j}^k(l) \in \mathbb{N}$ is the timing of commencement of operation $mo_{i,j}^k$ in the lth cycle.

(vi) $mt_i^k = (mt_{i,1}^k, mt_{i,2}^k, \ldots, mt_{i,j}^k, \ldots, mt_{i,lm(i)}^k)$ specifies *the operation times of multimodal processes*, where $mt_{i,j}^k$ denotes the time of execution of operation $mo_{i,j}^k$ and $lm(i)$ is the length of the cyclic process route mp_i^k.

(vii) $\Theta = \{\Theta^0, \Theta^1\}$ is the set of *priority dispatching rules*, $\Theta^i = \{\sigma_1^i, \sigma_2^i, \ldots, \sigma_c^i, \ldots, \sigma_m^i\}$ is the set of priority dispatching rules for local ($i = 0$)/multimodal ($i = 1$) processes where $\sigma_c^i = (s_{c,1}^i, \ldots, s_{c,d}^i, \ldots, s_{c,lp(c)}^i)$ are sequence components which determine the order in which the processes can be executed on the resource R_c, $s_{c,d}^0 \in H$ (where H is the set of local streams, e.g., in the case presented in Figure 2, $H = \{P_1^1, P_2^1, P_2^2, P_2^3, P_3^1, P_4^1, P_4^2, P_5^1, P_5^2, P_6^1, P_6^2, P_7^1\}$), and $s_{c,d}^1 \in mH$ (where mH is the set of multimodal streams, e.g., in case from Figure 2, $mH = \{mP_1^k, mP_2^k, mP_3^k \mid k = 1 \cdots 4\}$).

Using the above notation, a SCCP can be defined as a tuple:

$$\mathrm{SC} = ((R, \mathrm{SL}), \mathrm{SM}), \tag{5}$$

where one has the following: $R = \{R_1, R_2, \ldots, R_c, \ldots, R_m\}$—the set of resources, m—the number of resources, $\mathrm{SL} = (U, T, \Theta^0)$—the structure of local processes, that is, $U = \{p_1^1, \ldots, p_1^{ls(1)}, \ldots, p_n^1, \ldots, p_n^{ls(n)}\}$—the set of routes of local process, $ls(i)$—the number of streams belonging to the process P_i, n—the number of local processes, $T = \{t_1^1, \ldots, t_1^{ls(1)}, \ldots, t_n^1, \ldots, t_n^{ls(n)}\}$—the set of sequences of operation times in local processes, $\Theta^0 = \{\sigma_1^0, \sigma_2^0, \ldots, \sigma_c^0, \ldots, \sigma_m^0\}$—the set of priority dispatching rules for local processes, $\mathrm{SM} = (M, mT, \Theta^1)$—the structure of multimodal processes, that is, $M = \{mp_1^1, \ldots, mp_1^{lsm(1)}, \ldots, mp_w^1, \ldots, mp_w^{lsm(n)}\}$—the set of routes of a multimodal process, $lsm(i)$—the number of streams belonging to the process mP_iw—the number of multimodal processes, $mT = \{mt_1^1, \ldots, mt_w^{lsm(1)}, \ldots, mt_n^1, \ldots, mt_w^{lsm(n)}\}$—the set of sequences of operation times in multimodal processes, and $\Theta^1 = \{\sigma_1^1, \sigma_2^1, \ldots, \sigma_c^1, \ldots, \sigma_m^1\}$—the set of priority dispatching rules for multimodal processes.

The SCCP model (5) can be seen as a multilevel model, cf. Figure 3, that is, a model composed of an "R level" (resources), an "SL level" (local cyclic processes), and an "SM^1 level" (multimodal cyclic processes), as well as an "SM^q level" (the qth metamultimodal process). The SL level is defined by the transportation routes structure following the set U of local processes and the set of parameters Θ determining the required system behavior. In turn, the SM^1 level takes into account multimodal processes, as well as metamultimodal processes (SM^2 level) composed of multimodal processes from the SM^1 level. In other words, it is assumed that the variables describing SM^q are the same as in the case of SM, whereas the routes of the multimodal process of the qth level remain composed of the processes from the $(q-1)$th level. The presented model is an extended version of a simplified model limited to R and SL levels, which is introduced in [15].

Therefore, in general, the $\mathrm{SC} = ((R, \mathrm{SL}), \mathrm{SM})$ model can be seen as composed of lp levels as follows:

$$\mathrm{SC}^{lp} = \left(\left(\left(\left((R, \mathrm{SL}), \mathrm{SM}^1\right), \mathrm{SM}^2\right), \ldots, \mathrm{SM}^q\right), \ldots, \mathrm{SM}^{lp}\right). \tag{6}$$

The SC^{lp} model emphasizes structural characteristics of the SCCP modeled. In turn, the behavioral characteristics can be specified in terms of the admissible states reachability concept, that is, either taking into account the state space concept including both cyclic steady states and leading to them transient states [16] or the cyclic steady state space only [14, 15].

The second way, which does not take into account initial states leading through the transient states to the cyclic steady states, seems to be quite close to the cyclic scheduling methods widely used in many real-life cases [9, 11].

In that context the relevant multilevel cyclic schedule X^{lp} encompassing the SCCP behavior on each of its processes level, that is, local SL and multimodal SM^{lp} processes, can be defined as follows:

$$X^{lp} = \left(\left(\left(\left((X, \alpha), \left(m^1 X, m^1 \alpha\right)\right), \ldots, \right), \ldots, \left(m^q X, m^q \alpha\right)\right), \ldots, \left(m^{lp} X, m^{lp} \alpha\right)\right), \tag{7}$$

where one has the following: $X, \alpha/(m^q X, m^q \alpha)$—sequence of commencement of operations and periodicity of local/multimodal (ith level) processes executions.

It should also be noted that the schedule X^{lp} can be defined as a sequence of ordered pairs describing behaviors of local (X, α) and multimodal $(m^i X, m^i \alpha)$ processes, where X is a sequence of timings $x_{i,j}^k$ (for $l = 0$th cycle) of commencement of operations $o_{i,j}^k$ from streams P_i^k executed along local processes $X = (x_{1,1}^1, x_{1,2}^1, \ldots, x_{n,lr(n)}^{ls(n)})$, and by analogy, the sequence $m^q X = (m^q x_{1,1}^1, m^q x_{1,2}^1, \ldots, m^q x_{w(q),lm(w,q)}^{lsm(w,q)})$ consists of the timing of commencement of multimodal processes operations (from the qth level, where one has the following: $w(q)$—the number of multimodal processes at the qth level, $lsm(i, q)$—the number of streams of the ith process at the qth level, and $lm(i, q)$—the number of operation of the ith process at the qth level).

Variables $x_{i,j}^k / m^q x_{i,j}^k \in \mathbb{Z}$ describe the timing of commencement of operations in the qth cycle of the SCCP cyclic steady state behavior: $x_{i,j}^k(l) = x_{i,j}^k + l \cdot \alpha / m^q x_{i,j}^k(l) = m^q x_{i,j}^k + l \cdot m^q \alpha$.

Since values of $x_{i,j}^k / m^q x_{i,j}^k$ follow from system structure, hence the cyclic behavior X^{lp} of SCCP is determined by its structure SM^{lp}. Moreover, the multimodal processes behavior $(m^i X, m^i \alpha)$ also depends on the local cyclic processes behavior (X, α).

In the general case, besides depending on the structural characteristics of SCCP the values of the considered variables $x_{i,j}^k / m^q x_{i,j}^k$ depend on constraints from the mutual exclusion protocol, that is, the set of priority dispatching rules Θ, operation times, and so forth [14, 15] as well as on the way the local and multimodal processes interacting with each other. For example, in case of the two levels structure model, that is, including levels SL and SM as shown in Figure 2, the constraints determining $x_{i,j}^k / m^q x_{i,j}^k$ can be expressed by the following rules [15]:

(i) *for local processes*: the timing of commencement of operation $o_{i,j}^k$ beginning states for a maximum of the completion time of operation $o_{i,j-1}^k$ preceding $o_{i,j}^k$ and the release time (with delay Δt) of the resource $p_{i,j}^k$ awaiting for $o_{i,j}$ execution is as follows:

$$\begin{aligned} &\text{moment of operation } o_{i,j}^k \text{ beginning} \\ &\quad = \max\left\{\left(\text{moment of } p_{i,j}^k \text{ release} + \text{lag time } \Delta t\right), \right. \\ &\qquad \left. \left(\text{moment of operation } o_{i,j-1}^k \text{ completion}\right)\right\} \\ &\quad i = 1, \ldots, n; \quad j = 1, \ldots, lr(i); \quad k = 1, \ldots, ls(i); \end{aligned} \tag{8}$$

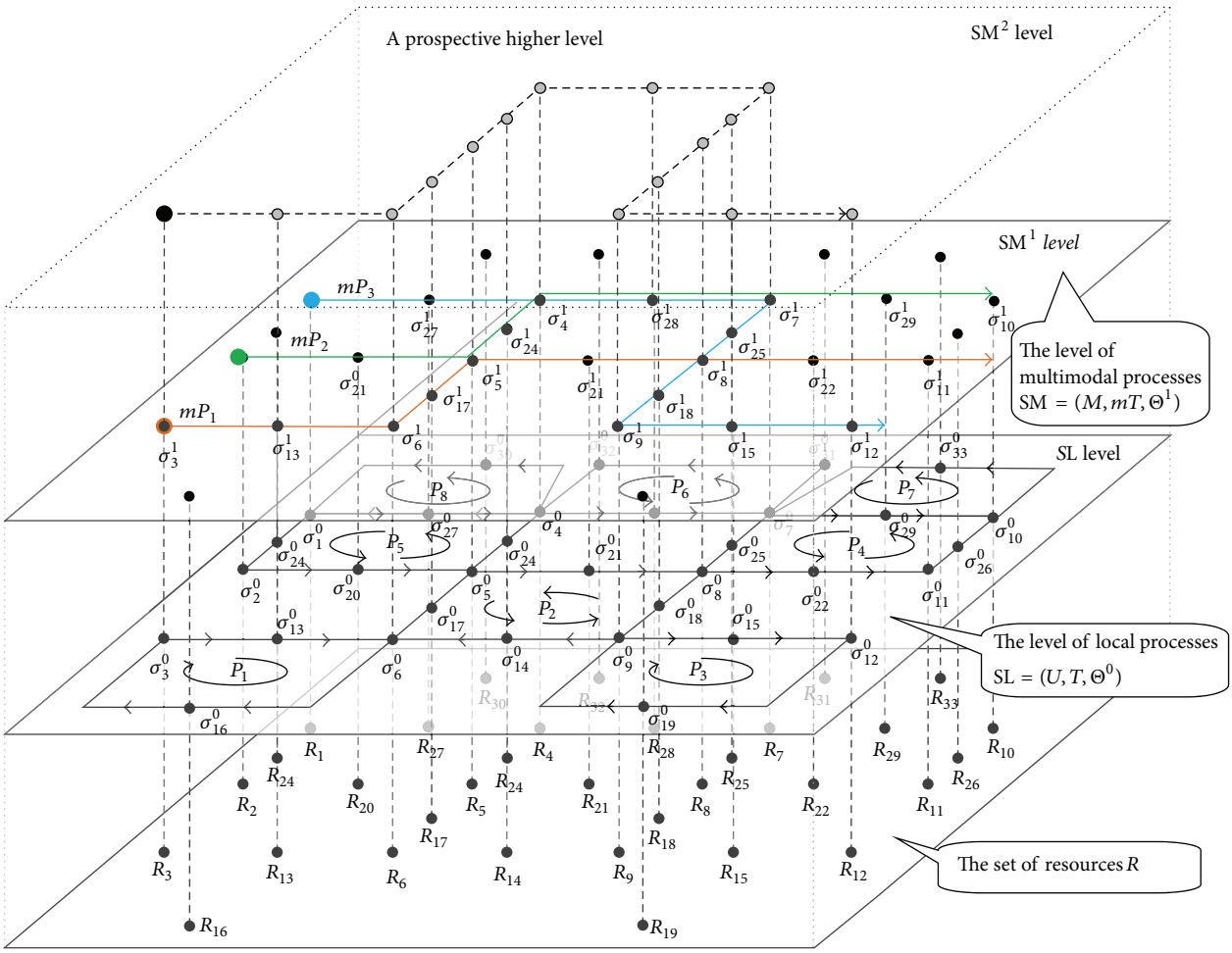


Figure 3: Multilayered model of the SCCP from Figure 2.

(ii) *for multimodal processes*: the timing of commencement of operation $mo_{i,j}^k$ beginning is equal to the nearest admissible value (determined by set $\mathcal{X}_{i,j}^k$ of values $mx_{i,j}^k$) being a maximum of both the completion time of operation $mo_{i,j-1}^k$ preceding $mo_{i,j}$ and the release time (lag time Δtm) of the resource $mp_{i,j}^k$ awaiting for $mo_{i,j}$ execution. Consider

$$\begin{aligned}&\text{moment of operation } mo_{i,j}^k \text{ beginning}\\&= \left\lceil \max\left\{\left(\text{moment of } mp_{i,j}^k \text{ release} + \text{lag time } \Delta tm\right),\right.\right.\\&\qquad\left.\left.\text{moment of operation } mo_{i,j-1}^k \text{ completion}\right\}\right\rceil_{\mathcal{X}_{i,j}^k}\\&i = 1,\ldots,n;\quad j = 1,\ldots,lm(i);\quad k = 1,\ldots,lsm(i),\end{aligned} \tag{9}$$

where one has the following: $\mathcal{X}_{i,j}^k$—set of values $mx_{i,j}^k$ following the set of local processes X, ($\mathcal{X}_{i,j}^k$—the set of moments when operation $mo_{i,j}^k$ of multimodal process may use required local process) and $\lceil a \rceil_B$—the smallest integer greater than or equal to a in terms of the set B: $\lceil a \rceil_B = \min\{k \in B : k \geq a\}$.

In other words, the timing of commencement of operation $mo_{i,j}$ belongs to the set $\mathcal{X}_{i,j}^k$ and being coincident with operation processes by relevant local transportation process P_i.

The constraints determining (due to introduced rules (8) and (9)) the timing of commencement of operations for the SCCP from Figure 2 are gathered in Tables 1 and 2.

2.2. Cyclic Steady State Space. According to the previous section, a set of cyclic process achieved in the structure SC^{lp} (6) can be represented as a cyclic schedule X^{lp} (7) that meets the constraints (8) and (9). Figure 4 presents an example of a cyclic schedule which can be achieved in the system illustrated in Figure 1. It shows that for a set of production plan (represented by multimodal processes mP_1, mP_2, and mP_3) it is possible to organize the work of AGVs responsible for transportation of elements between workstations (local processes P_1–P_8) in such a way that no deadlocks appear in the system or there is no process waiting at the resources. An

Table 1: Constraints determining the local processes behavior of the SCCP from Figure 2.

Constraints of the local processes		
$x^1_{1,1} = x^1_{1,4} + t^1_{1,4} - \alpha$	$x^2_{4,1} = \max\{(x^1_{4,8} + \Delta t - \alpha), (x^2_{4,8} + t^2_{4,8} - \alpha)\}$	—
$x^1_{1,2} = x^1_{1,1} + t^1_{1,1}$	$x^2_{4,2} = \max\{(x^1_{4,1} + \Delta t), (x^2_{4,1} + t^2_{4,1})\}$	$x^2_{5,1} = \max\{(x^1_{5,7} + \Delta t - \alpha), (x^2_{5,8} + t^2_{5,8} - \alpha)\}$
$x^1_{1,3} = x^1_{1,2} + t^1_{1,2}$	$x^2_{4,3} = \max\{(x^1_{4,2} + \Delta t), (x^2_{4,2} + t^2_{4,2})\}$	$x^2_{5,2} = \max\{(x^1_{5,8} + \Delta t - \alpha), (x^2_{5,1} + t^2_{5,1})\}$
$x^1_{1,4} = \max\{(x^1_{2,7} + \Delta t), (x^1_{1,3} + t^1_{1,3})\}$	$x^2_{4,4} = \max\{(x^1_{4,3} + \Delta t), (x^2_{4,3} + t^2_{4,3})\}$	$x^2_{5,3} = \max\{(x^1_{5,1} + \Delta t), (x^2_{5,2} + t^2_{5,2})\}$
—	$x^2_{4,5} = \max\{(x^1_{4,4} + \Delta t), (x^2_{4,4} + t^2_{4,4})\}$	$x^2_{5,4} = \max\{(x^1_{5,2} + \Delta t), (x^2_{5,3} + t^2_{5,3})\}$
$x^1_{2,1} = \max\{(x^3_{2,6} + \Delta t - \alpha), (x^1_{2,8} + t^1_{2,8} - \alpha)\}$	$x^2_{4,6} = \max\{(x^3_{2,7} + \Delta t), (x^2_{4,5} + t^2_{4,5})\}$	$x^2_{5,5} = \max\{(x^1_{5,3} + \Delta t), (x^2_{5,4} + t^2_{5,4})\}$
$x^1_{2,2} = \max\{(x^2_{4,7} + \Delta t - \alpha), (x^1_{2,1} + t^1_{2,1})\}$	$x^2_{4,7} = \max\{(x^1_{4,6} + \Delta t), (x^2_{4,6} + t^2_{4,6})\}$	$x^2_{5,6} = \max\{(x^1_{5,4} + \Delta t), (x^2_{5,5} + t^2_{5,5})\}$
$x^1_{2,3} = \max\{(x^3_{2,8} + \Delta t - \alpha), (x^1_{2,2} + t^1_{2,2})\}$	$x^2_{4,8} = \max\{(x^1_{4,7} + \Delta t), (x^2_{4,7} + t^2_{4,7})\}$	$x^2_{5,7} = \max\{(x^1_{5,5} + \Delta t), (x^2_{5,6} + t^2_{5,6})\}$
$x^1_{2,4} = \max\{(x^3_{2,1} + \Delta t), (x^1_{2,3} + t^1_{2,3})\}$	—	$x^2_{5,8} = \max\{(x^1_{5,6} + \Delta t), (x^2_{5,7} + t^2_{5,7})\}$
$x^1_{2,5} = \max\{(x^3_{2,2} + \Delta t), (x^1_{2,4} + t^1_{2,4})\}$	$x^1_{5,1} = \max\{(x^2_{5,5} + \Delta t - \alpha), (x^1_{5,8} + t^1_{5,8} - \alpha)\}$	—
$x^1_{2,6} = \max\{(x^3_{2,3} + \Delta t), (x^1_{2,5} + t^1_{2,5})\}$	$x^1_{5,2} = \max\{(x^2_{5,6} + \Delta t - \alpha), (x^1_{5,1} + t^1_{5,1})\}$	$x^1_{6,1} = \max\{(x^2_{6,3} + \Delta t - \alpha), (x^1_{6,5} + t^1_{6,5} - \alpha)\}$
$x^1_{2,7} = \max\{(x^3_{2,4} + \Delta t), (x^1_{2,6} + t^1_{2,6})\}$	$x^1_{5,3} = \max\{(x^3_{2,5} + \Delta t), (x^1_{5,2} + t^1_{5,2})\}$	$x^1_{6,2} = \max\{(x^2_{5,1} + \Delta t), (x^1_{6,1} + t^1_{6,1})\}$
$x^1_{2,8} = \max\{(x^2_{5,7} + \Delta t), (x^1_{2,7} + t^1_{2,7})\}$	$x^1_{5,4} = \max\{(x^2_{5,8} + \Delta t - \alpha), (x^1_{5,3} + t^1_{5,3})\}$	$x^1_{6,3} = \max\{(x^2_{6,5} + \Delta t - \alpha), (x^1_{6,2} + t^1_{6,2})\}$
—	$x^1_{5,5} = \max\{(x^2_{6,4} + \Delta t), (x^1_{5,4} + t^1_{5,4})\}$	$x^1_{6,4} = \max\{(x^2_{4,5} + \Delta t), (x^1_{6,3} + t^1_{6,3})\}$
$x^2_{2,1} = \max\{(x^1_{2,8} + \Delta t - \alpha), (x^2_{2,8} + t^2_{2,8} - \alpha)\}$	$x^1_{5,6} = \max\{(x^1_{8,3} + \Delta t), (x^1_{5,5} + t^1_{5,5})\}$	$x^1_{6,5} = \max\{(x^2_{6,2} + \Delta t), (x^1_{6,4} + t^1_{6,4})\}$
$x^2_{2,2} = \max\{(x^1_{2,1} + \Delta t), (x^2_{2,1} + t^2_{2,1})\}$	$x^1_{5,7} = \max\{(x^1_{8,2} + \Delta t), (x^1_{5,6} + t^1_{5,6})\}$	—
$x^2_{2,3} = \max\{(x^1_{2,2} + \Delta t), (x^2_{2,2} + t^2_{2,2})\}$	$x^1_{5,8} = \max\{(x^2_{5,4} + \Delta t), (x^1_{5,7} + t^1_{5,7})\}$	$x^2_{6,1} = \max\{(x^2_{6,3} + \Delta t), (x^2_{6,5} + t^2_{6,5} - \alpha)\}$
$x^2_{2,4} = \max\{(x^1_{4,5} + \Delta t), (x^2_{2,3} + t^2_{2,3})\}$	—	$x^2_{6,2} = \max\{(x^1_{6,2} + \Delta t), (x^2_{6,1} + t^2_{6,1})\}$
$x^2_{2,5} = \max\{(x^1_{2,4} + \Delta t), (x^2_{2,4} + t^2_{2,4})\}$	$x^1_{3,1} = x^1_{3,4} + t^1_{3,4} - \alpha$	$x^2_{6,3} = \max\{(x^1_{8,4} + \Delta t), (x^2_{6,2} + t^2_{6,2})\}$
$x^2_{2,6} = \max\{(x^1_{2,5} + \Delta t), (x^2_{2,5} + t^2_{2,5})\}$	$x^1_{3,2} = x^1_{3,1} + t^1_{3,1}$	$x^2_{6,4} = \max\{(x^1_{6,4} + \Delta t), (x^2_{6,3} + t^2_{6,3})\}$
$x^2_{2,7} = \max\{(x^1_{2,1} + \Delta t), (x^2_{2,6} + t^2_{2,6})\}$	$x^1_{3,3} = \max\{(x^2_{2,7} + \Delta t), (x^1_{3,2} + t^1_{3,2})\}$	$x^2_{6,5} = \max\{(x^1_{7,3} + \Delta t), (x^2_{6,4} + t^2_{6,4})\}$
$x^2_{2,8} = \max\{(x^1_{1,1} + \Delta t - \alpha), (x^2_{2,7} + t^2_{2,7})\}$	$x^1_{3,4} = x^1_{3,3} + t^1_{3,3}$	—
—	—	$x^1_{7,1} = x^1_{7,4} + t^1_{7,4} - \alpha$
$x^3_{2,1} = \max\{(x^2_{2,8} + \Delta t - \alpha), (x^3_{2,8} + t^3_{2,8} - \alpha)\}$	$x^1_{4,1} = \max\{(x^1_{7,4} + \Delta t - \alpha), (x^1_{4,8} + t^1_{4,8} - \alpha)\}$	$x^1_{7,2} = \max\{(x^1_{6,5} + \Delta t), (x^1_{7,1} + t^1_{7,1})\}$
$x^3_{2,2} = \max\{(x^2_{2,1} + \Delta t), (x^3_{2,1} + t^3_{2,1})\}$	$x^1_{4,2} = \max\{(x^2_{6,1} + \Delta t), (x^1_{4,1} + t^1_{4,1})\}$	$x^1_{7,3} = \max\{(x^2_{4,4} + \Delta t), (x^1_{7,2} + t^1_{7,2})\}$
$x^3_{2,3} = \max\{(x^2_{2,2} + \Delta t), (x^3_{2,2} + t^3_{2,2})\}$	$x^1_{4,3} = \max\{(x^2_{4,6} + \Delta t - \alpha), (x^1_{4,2} + t^1_{4,2})\}$	$x^1_{7,4} = \max\{(x^2_{4,3} + \Delta t), (x^1_{7,3} + t^1_{7,3})\}$
$x^3_{2,4} = \max\{(x^2_{2,3} + \Delta t), (x^3_{2,3} + t^3_{2,3})\}$	$x^1_{4,4} = \max\{(x^1_{2,3} + \Delta t), (x^1_{4,3} + t^1_{4,3})\}$	—
$x^3_{2,5} = \max\{(x^2_{2,4} + \Delta t), (x^3_{2,4} + t^3_{2,4})\}$	$x^1_{4,5} = \max\{(x^2_{4,8} + \Delta t - \alpha), (x^1_{4,4} + t^1_{4,4})\}$	$x^1_{8,1} = \max\{(x^2_{5,3} + \Delta t), (x^1_{8,4} + t^1_{8,4} - \alpha)\}$
$x^3_{2,6} = \max\{(x^2_{2,5} + \Delta t), (x^3_{2,5} + t^3_{2,5})\}$	$x^1_{4,6} = \max\{(x^2_{4,1} + \Delta t), (x^1_{4,5} + t^1_{4,5})\}$	$x^1_{8,2} = \max\{(x^2_{5,2} + \Delta t), (x^1_{8,1} + t^1_{8,1})\}$
$x^3_{2,7} = \max\{(x^2_{2,6} + \Delta t), (x^3_{2,6} + t^3_{2,6})\}$	$x^1_{4,7} = \max\{(x^2_{4,2} + \Delta t), (x^1_{4,6} + t^1_{4,6})\}$	$x^1_{8,3} = \max\{(x^1_{6,3} + \Delta t), (x^1_{8,2} + t^1_{8,2})\}$
$x^3_{2,8} = \max\{(x^2_{2,7} + \Delta t), (x^3_{2,7} + t^3_{2,7})\}$	$x^1_{4,8} = \max\{(x^1_{7,1} + \Delta t - \alpha), (x^1_{4,7} + t^1_{4,7})\}$	$x^1_{8,4} = x^1_{8,3} + t^1_{8,3}$

assumption was made that the transportation of work-pieces (streams of multimodal processes) takes one unit of time and their loading to/unloading from AGVs (local processes) takes place in the first and the last unit of each operation (see Figure 4).

The work [20] shows that schedules of this kind (ensuring such an organization of AGVs that guarantees the accomplishment of the set of production routes without any stoppages) can be obtained by means of solving the following constraint satisfaction problem (10):

$$\mathrm{CS}_{XT} = \left(\left(\left\{X^{lp}, T^{lp}\right\}, \{D_X, D_T\}\right), \{C_L, C_M\}\right), \tag{10}$$

where one has the following: X^{lp}, T^{lp}—decision variables, X^{lp}—cyclic schedule (7), T^{lp}—sequence of operation times $T^{lp} = (((T, mT^1), \ldots, mT^q), \ldots, mT^{lp})$, D_X, D_T, $D_\alpha = \mathbb{N}$—domains determining admissible value of decision variables $D_X : m^q x^k_{i,j}$, $x^k_{i,j} \in \mathbb{Z}, m^q\alpha$, $\alpha \in \mathbb{N}$; $D_T : m^q t^k_{i,j}$, $t^k_{i,j} \in \mathbb{N}$, $\{C_L, C_M\}$—the set of constraints C_L and C_M describing SCCP behavior, C_L—constraints determining cyclic steady state of local processes, that is, their cyclic schedule, and C_M—constraints determining multimodal processes behavior.

The sequence of operations times T^{lp} and the sequence of their beginning moments X^{lp} both of them follow constraints C_L, C_M which are the sufficient conditions for the SCCP cyclic steady state behavior, that is, the state following the problem (10) solution. In case of two level systems, that is, SL and SM (see Figure 3), the constraints C_L, C_M can be specified in terms of constraints (8) and their extension (9) [15].

Presenting cyclic processes as X^{lp} schedules is commonly used both in SCCP [11, 12, 14, 15] and in various problems of cyclic scheduling. Instead of cyclic schedules, the so-called behavior digraphs [16], that create the state space $\mathscr{P}$, can also be applied.

Table 2: Constraints (9) determining the multimodal processes behavior of the SCCP from Figure 2.

Constraints for stream mP_1^1	Constraints for stream mP_2^1	Constraints for stream mP_3^1
		$mx_{3,1}^1 = \lceil \max\{(mx_{3,2}^4 + \Delta tm - m\alpha), (mx_{3,11}^1 + mt_{3,11}^1 - m\alpha)\} \rceil_{\mathcal{X}_{3,1}^1}$
$mx_{1,1}^1 = \lceil \max\{(mx_{1,2}^4 + \Delta tm - m\alpha), (mx_{1,9}^1 + mt_{1,9}^1 - m\alpha)\} \rceil_{\mathcal{X}_{1,1}^1}$	$mx_{2,1}^1 = \lceil \max\{(mx_{2,2}^4 + \Delta tm - m\alpha), (mx_{2,9}^1 + mt_{2,9}^1 - m\alpha)\} \rceil_{\mathcal{X}_{2,1}^1}$	$mx_{3,2}^1 = \lceil \max\{(mx_{3,3}^4 + \Delta tm - m\alpha), (mx_{3,1}^1 + mt_{3,1}^1)\} \rceil_{\mathcal{X}_{3,1}^1}$
$mx_{1,2}^1 = \lceil \max\{(mx_{1,3}^4 + \Delta tm - m\alpha), (mx_{1,1}^1 + mt_{1,1}^1)\} \rceil_{\mathcal{X}_{1,1}^1}$	$mx_{2,2}^1 = \lceil \max\{(mx_{2,3}^4 + \Delta tm - m\alpha), (mx_{2,1}^1 + mt_{2,1}^1)\} \rceil_{\mathcal{X}_{2,1}^1}$	$mx_{3,3}^1 = \lceil \max\{(mx_{2,6}^4 + \Delta tm), (mx_{3,2}^1 + mt_{3,2}^1)\} \rceil_{\mathcal{X}_{3,3}^1}$
$mx_{1,3}^1 = \lceil \max\{(mx_{1,4}^4 + \Delta tm - m\alpha), (mx_{1,2}^1 + mt_{1,2}^1)\} \rceil_{\mathcal{X}_{1,3}^1}$	$mx_{2,3}^1 = \lceil \max\{(mx_{1,6}^4 + \Delta tm), (mx_{2,2}^1 + mt_{2,2}^1)\} \rceil_{\mathcal{X}_{2,3}^1}$	$mx_{3,4}^1 = \lceil \max\{(mx_{2,7}^4 + \Delta tm - m\alpha), (mx_{3,3}^1 + mt_{3,3}^1)\} \rceil_{\mathcal{X}_{3,4}^1}$
$mx_{1,4}^1 = \lceil \max\{(mx_{1,5}^4 + \Delta tm - m\alpha), (mx_{1,3}^1 + mt_{1,3}^1)\} \rceil_{\mathcal{X}_{1,4}^1}$	$mx_{2,4}^1 = \lceil \max\{(mx_{2,5}^4 + \Delta tm - m\alpha), (mx_{2,3}^1 + mt_{2,3}^1)\} \rceil_{\mathcal{X}_{2,4}^1}$	$mx_{3,5}^1 = \lceil \max\{(mx_{2,8}^4 + \Delta tm), (mx_{3,4}^1 + mt_{3,4}^1)\} \rceil_{\mathcal{X}_{3,5}^1}$
$mx_{1,5}^1 = \lceil \max\{(mx_{2,4}^1 + \Delta tm), (mx_{1,4}^1 + mt_{1,4}^1)\} \rceil_{\mathcal{X}_{1,5}^1}$	$mx_{2,5}^1 = \lceil \max\{(mx_{3,4}^1 + \Delta tm), (mx_{2,4}^1 + mt_{2,4}^1)\} \rceil_{\mathcal{X}_{2,5}^1}$	$mx_{3,6}^1 = \lceil \max\{(mx_{3,7}^4 + \Delta tm), (mx_{3,5}^1 + mt_{3,5}^1)\} \rceil_{\mathcal{X}_{3,6}^1}$
$mx_{1,6}^1 = \lceil \max\{(mx_{1,7}^4 + \Delta tm), (mx_{1,5}^1 + mt_{1,5}^1)\} \rceil_{\mathcal{X}_{1,6}^1}$	$mx_{2,6}^1 = \lceil \max\{(mx_{3,5}^1 + \Delta tm), (mx_{2,5}^1 + mt_{2,5}^1)\} \rceil_{\mathcal{X}_{2,6}^1}$	$mx_{3,7}^1 = \lceil \max\{(mx_{1,8}^4 + \Delta tm), (mx_{3,6}^1 + mt_{3,6}^1)\} \rceil_{\mathcal{X}_{3,7}^1}$
$mx_{1,7}^1 = \lceil \max\{(mx_{3,8}^1 + \Delta tm), (mx_{1,6}^1 + mt_{1,6}^1)\} \rceil_{\mathcal{X}_{1,7}^1}$	$mx_{2,7}^1 = \lceil \max\{(mx_{3,6}^1 + \Delta tm), (mx_{2,6}^1 + mt_{2,6}^1)\} \rceil_{\mathcal{X}_{2,7}^1}$	$mx_{3,8}^1 = \lceil \max\{(mx_{3,9}^4 + \Delta tm), (mx_{3,7}^1 + mt_{3,7}^1)\} \rceil_{\mathcal{X}_{3,8}^1}$
$mx_{1,8}^1 = \lceil \max\{(mx_{1,9}^4 + \Delta tm), (mx_{1,7}^1 + mt_{1,7}^1)\} \rceil_{\mathcal{X}_{1,8}^1}$	$mx_{2,8}^1 = \lceil \max\{(mx_{2,9}^4 + \Delta tm), (mx_{2,7}^1 + mt_{2,7}^1)\} \rceil_{\mathcal{X}_{2,8}^1}$	$mx_{3,9}^1 = \lceil \max\{(mx_{3,10}^4 + \Delta tm), (mx_{3,8}^1 + mt_{3,8}^1)\} \rceil_{\mathcal{X}_{3,9}^1}$
$mx_{1,9}^1 = \lceil \max\{(mx_{1,9}^4 + mt_{1,9}^4 + \Delta tm), (mx_{1,8}^1 + mt_{1,8}^1)\} \rceil_{\mathcal{X}_{1,9}^1}$	$mx_{2,9}^1 = \lceil \max\{(mx_{2,9}^4 + mt_{2,9}^4 + \Delta tm), (mx_{2,8}^1 + mt_{2,8}^1)\} \rceil_{\mathcal{X}_{2,9}^1}$	$mx_{3,10}^1 = \lceil \max\{(mx_{3,11}^4 + \Delta tm), (mx_{3,9}^1 + mt_{3,9}^1)\} \rceil_{\mathcal{X}_{3,10}^1}$
		$mx_{3,11}^1 = \lceil \max\{(mx_{1,11}^4 + mt_{1,11}^4 + \Delta tm), (mx_{3,10}^1 + mt_{3,10}^1)\} \rceil_{\mathcal{X}_{3,11}^1}$
		$mx_{3,1}^1 \in \mathcal{X}_{3,1}^1 = \{x_{8,1}^1(l) \mid x_{8,1}^1(l) = x_{8,1}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$
$mx_{1,1}^1 \in \mathcal{X}_{1,1}^1 = \{x_{1,2}^1(l) \mid x_{1,2}^1(l) = x_{1,2}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{2,1}^1 \in \mathcal{X}_{2,1}^1 = \{x_{5,1}^1(l) \mid x_{5,1}^1(l) = x_{5,1}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{3,2}^1 \in \mathcal{X}_{3,2}^1 = \{x_{8,2}^1(l) \mid x_{8,2}^1(l) = x_{8,2}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$
$mx_{1,2}^1 \in \mathcal{X}_{1,2}^1 = \{x_{1,3}^1(l) \mid x_{1,3}^1(l) = x_{1,3}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{2,2}^1 \in \mathcal{X}_{2,2}^1 = \{x_{5,2}^1(l) \mid x_{5,2}^1(l) = x_{5,2}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{3,3}^1 \in \mathcal{X}_{3,3}^1 = \{x_{8,3}^1(l) \mid x_{8,3}^1(l) = x_{8,3}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$
$mx_{1,3}^1 \in \mathcal{X}_{1,3}^1 = \{x_{1,4}^1(l) \mid x_{1,4}^1(l) = x_{1,4}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{2,3}^1 \in \mathcal{X}_{2,3}^1 = \{x_{5,3}^1(l) \mid x_{5,3}^1(l) = x_{5,3}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{3,4}^1 \in \mathcal{X}_{3,4}^1 = \{x_{6,4}^2(l) \mid x_{6,2}^2(l) = x_{6,4}^2 + l \cdot \alpha, l \in \mathbb{Z}\}$
$mx_{1,4}^1 \in \mathcal{X}_{1,4}^1 = \{x_{2,7}^1(l) \mid x_{2,7}^1(l) = x_{2,7}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{2,4}^1 \in \mathcal{X}_{2,4}^1 = \{x_{5,7}^2(l) \mid x_{5,7}^2(l) = x_{5,7}^2 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{3,5}^1 \in \mathcal{X}_{3,5}^1 = \{x_{6,4}^2(l) \mid x_{6,2}^2(l) = x_{6,4}^2 + l \cdot \alpha, l \in \mathbb{Z}\}$
$mx_{1,5}^1 \in \mathcal{X}_{1,5}^1 = \{x_{2,8}^1(l) \mid x_{2,8}^1(l) = x_{2,8}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{2,5}^1 \in \mathcal{X}_{2,5}^1 = \{x_{5,8}^2(l) \mid x_{5,8}^2(l) = x_{5,8}^2 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{3,6}^1 \in \mathcal{X}_{3,6}^1 = \{x_{4,3}^1(l) \mid x_{4,3}^1(l) = x_{4,3}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$
$mx_{1,6}^1 \in \mathcal{X}_{1,6}^1 = \{x_{2,5}^3(l) \mid x_{2,5}^3(l) = x_{2,5}^3 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{2,6}^1 \in \mathcal{X}_{2,6}^1 = \{x_{6,3}^1(l) \mid x_{6,3}^1(l) = x_{6,3}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{3,7}^1 \in \mathcal{X}_{3,7}^1 = \{x_{4,4}^1(l) \mid x_{4,3}^1(l) = x_{4,4}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$
$mx_{1,7}^1 \in \mathcal{X}_{1,7}^1 = \{x_{2,6}^3(l) \mid x_{2,6}^3(l) = x_{2,6}^3 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{2,7}^1 \in \mathcal{X}_{2,7}^1 = \{x_{6,4}^1(l) \mid x_{6,3}^1(l) = x_{6,4}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{3,8}^1 \in \mathcal{X}_{3,8}^1 = \{x_{2,5}^2(l) \mid x_{2,5}^2(l) = x_{2,5}^2 + l \cdot \alpha, l \in \mathbb{Z}\}$
$mx_{1,8}^1 \in \mathcal{X}_{1,8}^1 = \{x_{4,7}^2(l) \mid x_{4,7}^2(l) = x_{4,7}^2 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{2,8}^1 \in \mathcal{X}_{2,8}^1 = \{x_{7,3}^1(l) \mid x_{7,3}^1(l) = x_{7,3}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{3,9}^1 \in \mathcal{X}_{3,9}^1 = \{x_{2,6}^2(l) \mid x_{2,6}^2(l) = x_{2,6}^2 + l \cdot \alpha, l \in \mathbb{Z}\}$
$mx_{1,9}^1 \in \mathcal{X}_{1,9}^1 = \{x_{4,8}^2(l) \mid x_{4,8}^2(l) = x_{4,8}^2 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{2,9}^1 \in \mathcal{X}_{2,9}^1 = \{x_{7,4}^1(l) \mid x_{7,4}^1(l) = x_{7,4}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$	$mx_{3,10}^1 \in \mathcal{X}_{3,10}^1 = \{x_{3,3}^1(l) \mid x_{3,3}^1(l) = x_{3,3}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$
		$mx_{3,11}^1 \in \mathcal{X}_{3,11}^1 = \{x_{3,4}^1(l) \mid x_{3,4}^1(l) = x_{3,4}^1 + l \cdot \alpha, l \in \mathbb{Z}\}$
Constraints of the rest of streams mP_1^2, mP_1^3, mP_1^4, mP_2^2, mP_2^3, mP_2^4, mP_3^2, mP_3^3, and mP_3^4 are defined in the same way.		

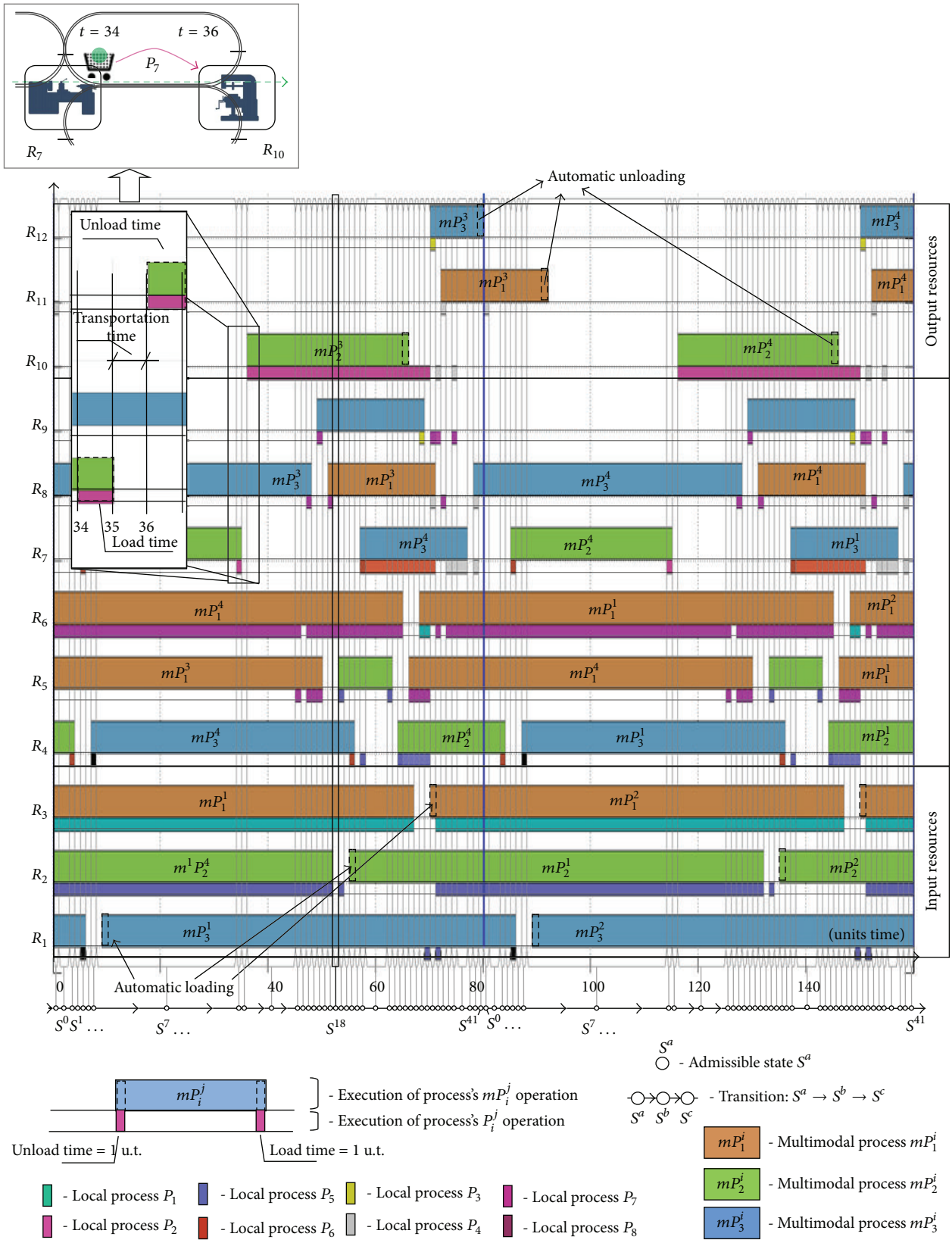


Figure 4: AGVs fleet cyclic schedule matching the multiproduct manufacturing.

The digraph G_W (denoted by W index, in case of cyclic processes representation) is a digraph whose vertexes represent the admissible states S^r of the system and arcs represent transitions between the states (while depicting a given transition function δ [16]).

In the considered case, the state of SCCP describes the present (valid in a given t period) allocation on the resources of local (AGVs) and multimodal (technological routes) processes and describes the current access rights to the resources (determined by dispatching priority rules Θ). Formally, in the state definition, three elements are distinguished that characterize processes at the particular behavior levels (see Figure 3).

In this approach, the state of SCCP takes the following form:

$$S^r = \left(\left(\left(\left(Sl^r, m^1 S^r\right), \ldots, m^l S^r\right), \ldots\right), m^{lp} S^r\right), \quad (11)$$

where one has the following: (i) lp—the number of levels of structure SC^{lp} (6) and (ii) Sl^r—the rth state of local processes executed on the SL level (5) as follows:

$$Sl^r = \left(A^r, Z^r, Q^r\right), \quad (12)$$

where one has the following: $A^r = (a_1{}^r, a_2{}^r, \ldots, a_c{}^r, \ldots, a_m{}^r)$—allocation of processes in the rth state, $a_c{}^r \in P \cup \{\Delta\}$, $a_c{}^r = P_i^k$—the cth resource R_c is occupied by the local stream P_i^k, and $a_c{}^r = \Delta$—the cth resource R_c is unoccupied.

$Z^r = (z_1{}^r, z_2{}^r, \ldots, z_c{}^r, \ldots z_m{}^r)$ is the sequence of semaphores corresponding to the rth state and $z_c{}^r \in P$ is name of the stream (specified in the cth dispatching rule σ_c, allocated to the cth resource) which was allowed to occupy the cth resource; for example, $z_c{}^r = P_i^k$ means that stream P_i^k is currently allowed to occupy the cth resource.

$Q^r = (q_1{}^r, q_2{}^r, \ldots, q_c{}^r, \ldots, q_m{}^r)$ is the sequence of semaphore indices, corresponding to the rth state and $q_c{}^r$ determines the position of the semaphore $z_c{}^r$ in the priority dispatching rule σ_c, $z_c{}^r = s_{c,(q_c{}^r)}$, $q_c{}^r \in \mathbb{N}$. For instance, $q_2{}^r = 2$ and $z_2{}^r = P_1^2$ correspond to the semaphore $z_2{}^r = P_1^2$ taking the 2nd position in the priority dispatching rule σ_2.

(iii) $m^l S^r$ is the rth state of multimodal processes executed on the SM^l level (6) as follows,

$$m^l S^r = \left(m^l A^r, m^l Z^r, m^l Q^r\right), \quad (13)$$

where one has the following: $m^l A^r = (m^l A_1^{r,1}, \ldots, m^l A_i^{r,h}, \ldots, m^l A_{lw(l)}^{r,lsm(lw(l),l)})$—allocation of the process streams $m^l P_i^h$; that is,

$$m^l A_i^{r,h} = \left(m^l a_{i,1}^{r,h}, m^l a_{i,2}^{r,h}, \ldots, m^l a_{i,c}^{r,h}, \ldots, m^l a_{i,m}^{r,h}\right), \quad (14)$$

where one has the following: $m^l P_i^h$—the hth stream of the ith multimodal process from l-level of SC^{lp} (6), m—the number of resources R, $m^l a_{i,c}^{r,h} \in \{m^l P_i^h, \Delta\}$, $m^l a_{i,c}^{r,h} = m^l P_i^h$ means that the cth resource R_c is occupied by the hth stream of multimodal process $m^l P_i$, and $m^l a_{i,c}^{r,h} = \Delta$—the cth resource R_c which is released by the hth stream of multimodal process $m^l P_i$.

$m^l Z^r = (m^l z_1{}^r, \ldots, m^l z_c{}^r, \ldots, m^l z_m{}^r)$ and $m^l Q^r = (m^l q_1{}^r, \ldots, m^l q_k{}^r, \ldots, m^l q_m{}^r)$ are the sequences of semaphores and indices defined similarly as Z^r and Q^r.

In the context of the introduced notions (allocation, semaphore, indexes) the X^{lp} schedule presented in Figure 4 illustrates only the allocation of processes in time. The particular allocations are denoted by a frame and symbol "◯" of the state S^r connected with the given allocation. Therefore, the cyclic process represented by X^{lp} schedule can be described with use of 42 states (S^0–S^{41}).

The digraph $G_W = (V_W, E_W)$ corresponding to X^{lp} schedule from Figure 4 was illustrated in Figure 5. The set of vertexes $V_W = \{S^0, \ldots, S^{41}\}$ includes all the states that can be achieved by the system in the course of implementing processes according to X^{lp} schedule. The set of arcs $E_W \subseteq V_W \times V_W$ defines the transitions between the states. Transition of the system from the state S^0 to the state S^1 (denoted by $S^0 \rightarrow S^1$, and in Figure 5 as ◯ → ◯) occurs according to the transition function δ, which was described in [16].

According to this approach, the digraph $G_W = (V_W, E_W)$ depicting the behavior from Figure 4 is a cycle including 42 vertexes; see Figure 5. Apart from S^r states, Figure 5 illustrates also local states Sl^r (12) which only represent the behavior of local processes. In the discussed case, the number of local states is the same as the number of S^r states. Generally, it does not have to be so, as there are situations when one local state is an element of numerous S^r states [14, 15]. The local states Sl^r as well as the digraph related to them should be perceived as a projection of S^r states and G_W digraph onto the level of local processes.

It should be noted that in the discussed example only one cyclic steady state is reachable, that is, as a result of the solution (10); that is, one schedule X^{lp} was obtained along with one digraph G_W corresponding to it. Generally, in the given structure SC^{lp} a lot of cyclic steady states can be reachable (which depends on the initial phases of dispatching rules) and therefore, we can consider the cyclic steady state space $\mathscr{P}$ as a set of digraphs that can be reachable in a given structure of G_W digraphs.

The cyclic steady state space can be illustrated as a graph being the sum of all reachable digraphs: $\mathscr{P} = (V_{\mathscr{P}}, E_{\mathscr{P}}) = G_{W,1} \cup G_{W,2} \cup \cdots \cup G_{W,lg}$ (where $G_{W,a} \cup G_{W,b} = (V_a \cup V_b, E_a \cup E_b)$) are reachable in a given structure. There are also situations where in a given structure no cyclic steady states are reachable; then $\mathscr{P} = (\emptyset, \emptyset)$. The main reason for the lack of cyclic behaviors is the possible deadlocks.

The form of the space of states $\mathscr{P}$ is strictly dependent upon the form of SC^{lp} structure. In other words, the structure determines behaviors reachable in the system. Another example of a cyclic behavior is shown in

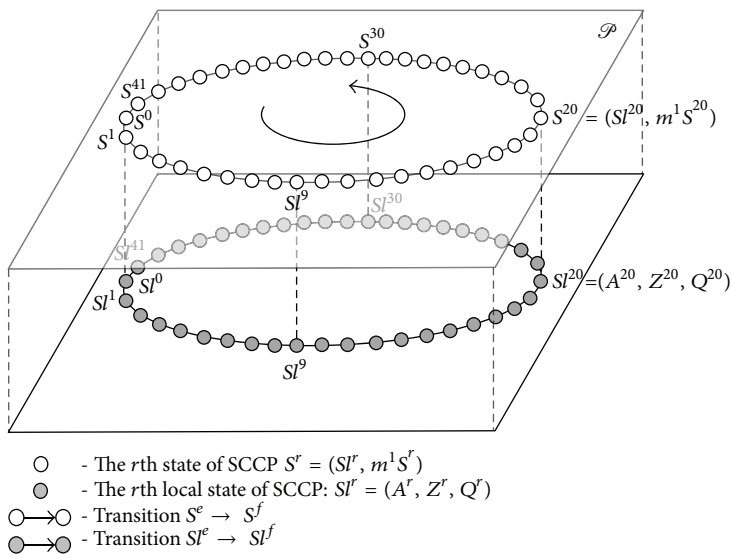


Figure 5: The digraph G_W following Gantt's chart from Figure 4.

Figure 6. It is a cyclic schedule X^{lp} reachable in SC structure (Figure 2) in which the manner of implementing production routes was changed (routes $mp_2^k = ((R_2, R_{20}, R_5), (R_{24}, R_4), (R_{28}, R_7), (R_{25}, R_8, R_{22}, R_{11}))$ and $mp_3^k = ((R_1, R_{27}, R_4), (R_{28}, R_7), (R_{29}, R_{10}))$. There are still seven types of AGVs used for transport (local processes P_1–P_7) with routes remaining unchanged yet different than previously, when they performed their operations.

This situation can be interpreted as introducing a new production order into the system, which requires a new method of processing elements. The presence of cyclic schedule X^{lp} (providing a solution to the problem (10)) and G_W digraph (Figure 7) related to it mean that such an order can be implemented.

The presence of two different cyclic steady states leads to the question of their mutual reachability: Is it possible to smoothly change the system behavior from $G_{W,1}$ to $G_{W,2}$ (Figure 7)? In other words, is it possible to smoothly (with no need to stop the system) proceed from implementing the first production order $G_{W,1}$ to the other $G_{W,2}$?

3. Multimodal Processes Rescheduling

3.1. Problem Formulation. The question of the possibility of smoothly changing the cyclic behaviors of the system is related to the problem of mutual reachability of the two behavior digraphs $G_{W,1}$, $G_{W,2}$. The digraphs presented so far (Figure 7) are cycles. In general, a digraph may take the form of the so-called vortex G_S, that is, a digraph including a cycle representing a cyclic steady states G_W and a tree $G_{T,i}$ representing transient states leading to cyclic steady states. An example of two vortexes $G_{S,1}$, $G_{S,2}$ is shown in Figure 8. Formally, the digraph of the vortex type is defined as follows:

$$G_S = (V_S, E_S) = G_W \cup \left(\bigcup_{i=1}^{lt} G_{T,i}\right), \tag{15}$$

where $\bigcup_{i=1}^{lt} G_{T,i} = G_{T,1} \cup G_{T,2} \cup \cdots \cup G_{T,lt}$, lt is the number of trees $G_{T,i}$ leading to G_W cycle.

In this approach, the problem of mutual reachability of the behavior digraphs of mutual space of states is defined as follows: there is a given structure SC^{lp} and the cyclic steady state space $\mathscr{P} = (V_{\mathscr{P}}, E_{\mathscr{P}})$ resulting from it, which includes two vortexes $G_{S,1}$, $G_{S,2}$ that represent two cyclic steady processes. It is therefore necessary to find answer to the following question: Are the digraphs $G_{W,1}$, $G_{W,2}$ (being subdigraphs of $G_{S,1}$, $G_{S,2}$) mutually reachable (Figure 8)? In the case they are mutually reachable the next question is how to make a transition between the digraphs?

In other words, the question of mutual reachability of cyclic behaviors can be treated as the question of the possibility of direct or indirect transition between the digraphs $G_{W,1}$, $G_{W,2}$. An example of such transitions is shown in Figure 8. An indirect transition should be understood as changing $G_{W,1}$ into $G_{W,2}$, resulting from the transition of the state $S^{x\prime}$ into transitory path (a path of digraph $G_{T,i}$) leading to $G_{W,2}$. The direct transition means a transition from the state S^x right into the cycle $G_{W,1}$.

3.2. Searching for States with Mutual Allocation

3.2.1. Sufficient Conditions. The possibility of transitions between cyclic steady states depends on the state $S^x(S^{x\prime})$, which is an element of two digraphs: $G_{S,1}$ and $G_{S,2}$. In other words, it is necessary that in case of direct transition the following transitions between states occur:

(i) for all $S^k \in V_{W,1}$ ($S^k \rightarrow \cdots \rightarrow S^x \rightarrow \cdots \rightarrow S^k$) for the digraph $G_{W,1}$,

(ii) for all $S^l \in V_{W,2}$ ($S^l \rightarrow \cdots \rightarrow S^x \rightarrow \cdots \rightarrow S^l$) for the digraph $G_{W,2}$.

In SCCP, transitions of this kind are not acceptable. It results from the property saying that each state has at least

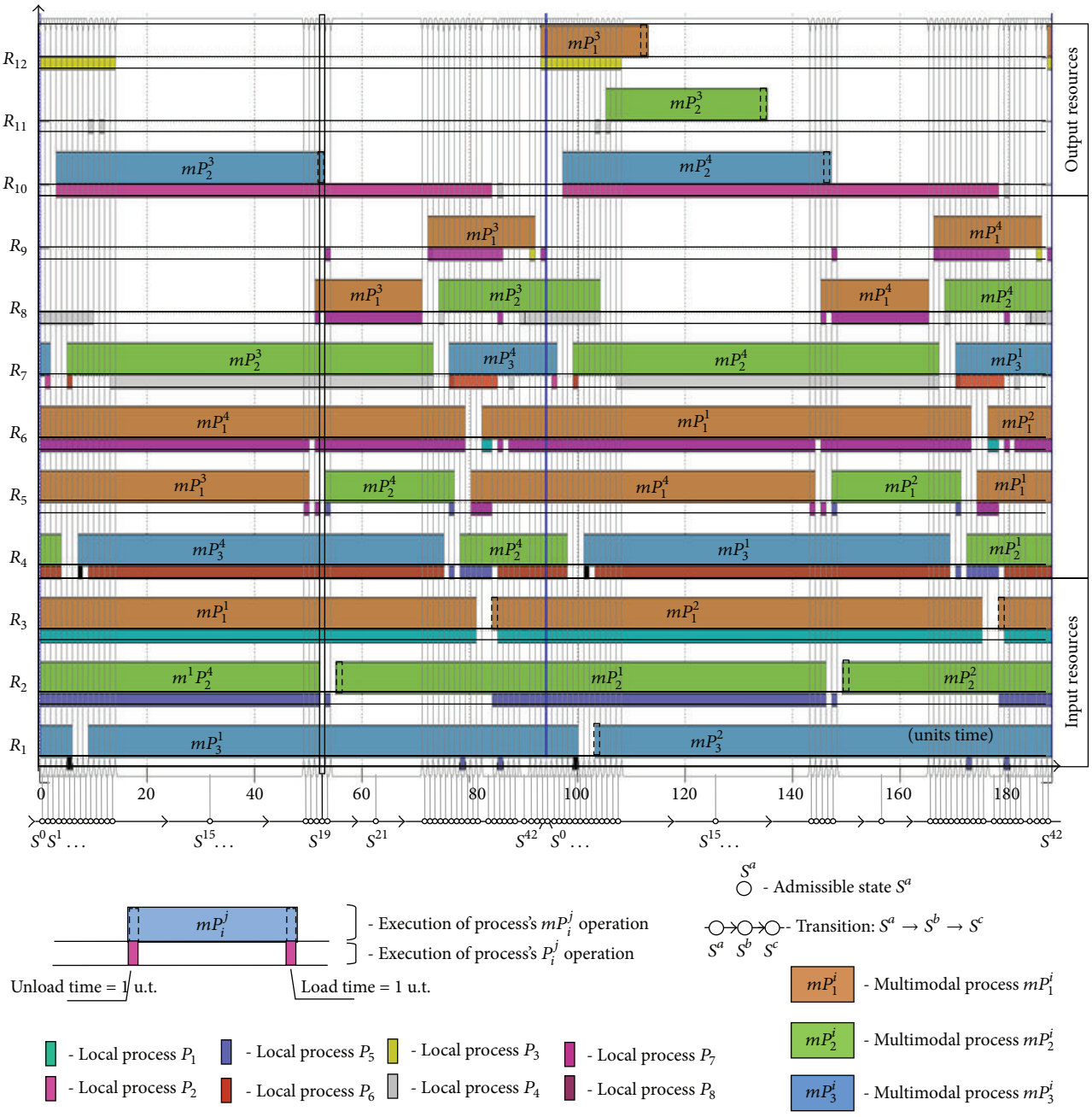


FIGURE 6: AGVs fleet cyclic schedule matching the multiproduct manufacturing for system with changed multimodal routes.

one descendant [16] (a descendant is the consecutive state after S^x). The presence of state S^x included simultaneously into two digraphs $(G_{W,1}, G_{W,2})$ would require the existence of at least two descendants of the state S^x.

Although there are no mutual states at the level of the space of states $\mathscr{P}$, the different projections of states upon lower levels of behavior as well as their components (space of allocation, semaphores, etc.) may be elements belonging to numerous digraphs at the same time.

An example of such a situation is shown in Figure 9, where the idea of a multilevel model of the cyclic steady states space $\mathscr{P}$ has been applied. The figure shows the projection of the space $\mathscr{P}$ upon the level of local processes behavior $\mathscr{P}_{SL}$ (the second level in Figure 9) and upon the space of allocation $\mathscr{A}$ (the lowest level in Figure 9). The elements of the space $\mathscr{A}$ are the allocations A^r of the local states $Sl^r = (A^r, Z^r, Q^r)$ occurring in the space $\mathscr{P}_{SL} = (V_{SL}, E_{SL})$ (where $Sl^r \in V_{SL}$), which at the same time is the projection of the space of states $\mathscr{P}$.

It is worth mentioning that some allocations of the space $\mathscr{A}$ are simultaneously mutual for several states. For example, the allocation $A^2 \in \mathscr{A}$ is at the same time an element of the state $Sl^3 = (A^2, Z^3, Q^3)$ and the state $Sl^4 = (A^2, Z^4, Q^4)$. In practice, it means that in both states the local processes

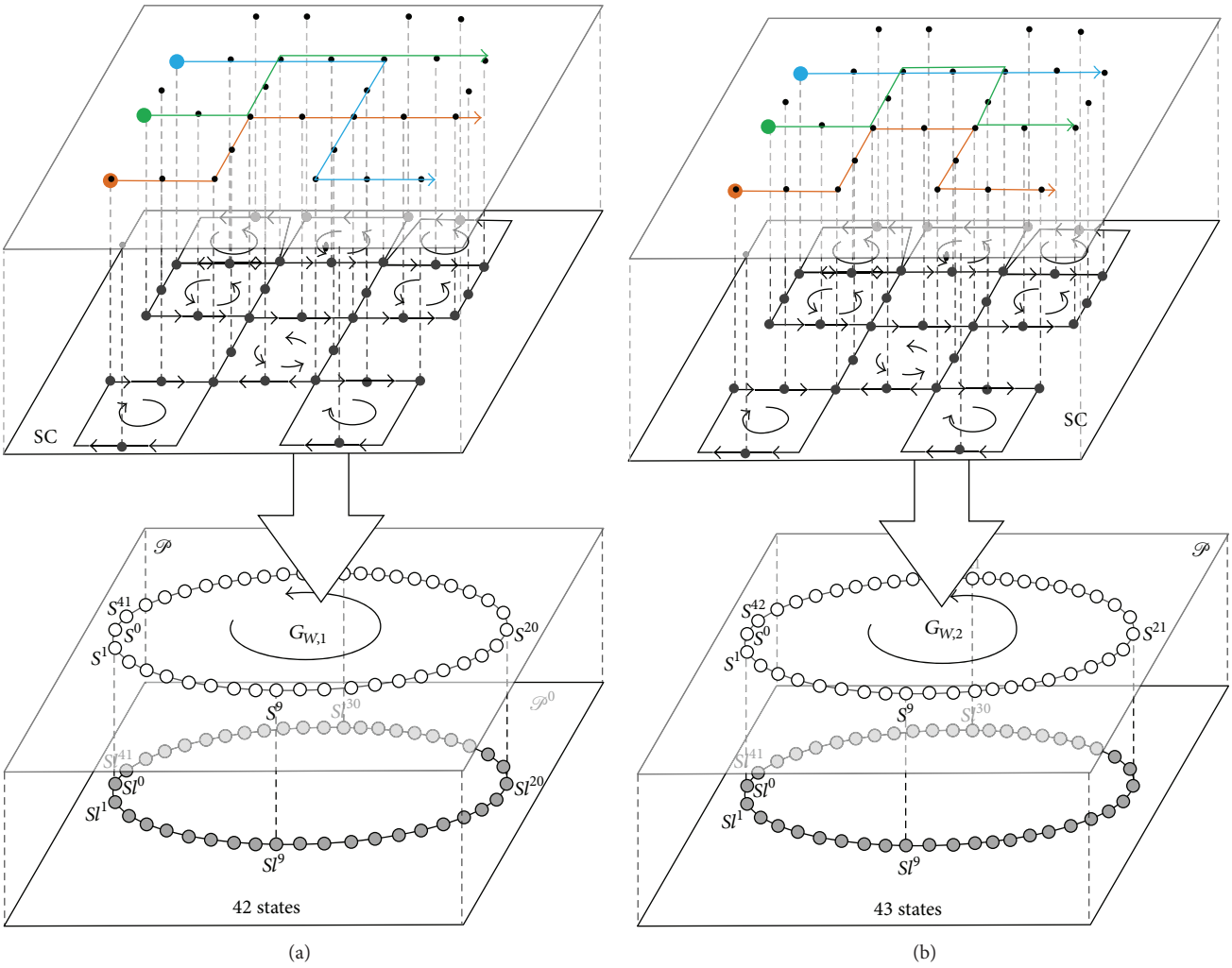


FIGURE 7: The state space for structures with different multimodal processes routes.

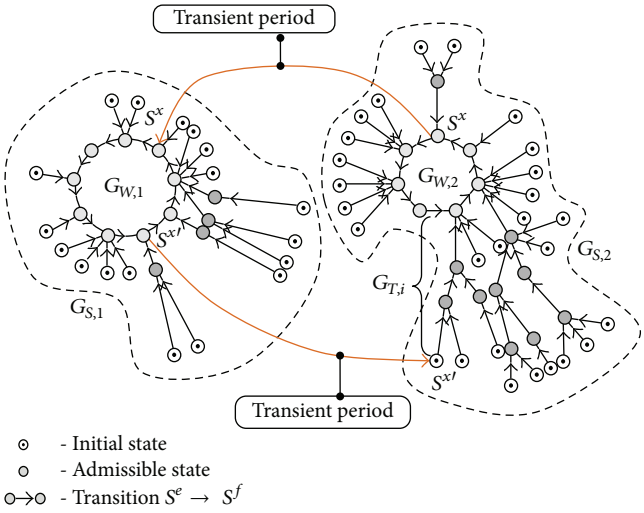


FIGURE 8: Illustration of the cyclic steady state space composed of two vortexes.

(representing, e.g., means of transport such as AGVs) are allocated on the same resources. However, since there are various access rights (determined by Z^3, Q^3 and Z^4, Q^4) their further implantation will be different.

The existence of states of common allocation can be used for the transitions between various behavior digraphs. The states Sl^3 and Sl^4 differ only with the sequence of semaphores and indexes. Thus, if in the state Sl^3 with allocation A^2, the form of Z^3, Q^3 changes into the form of Z^4, Q^4, we will attain the state Sl^4 leading to another cycle.

The modification of the state Sl^3 into Sl^4 by changing the form of semaphores and indexes allows for an indirect transition between cyclic processes of the space $\mathscr{P}^0$. The direct transition is possible as a result of modifying the semaphores and indexes of the states $Sl^1 = (A^1, Z^1, Q^1)$ and $Sl^2 = (A^1, Z^2, Q^2)$ mutually sharing the allocation A^1. Therefore, the presence of states of mutual allocation implies the possibility of changing the set of cyclic steady states.

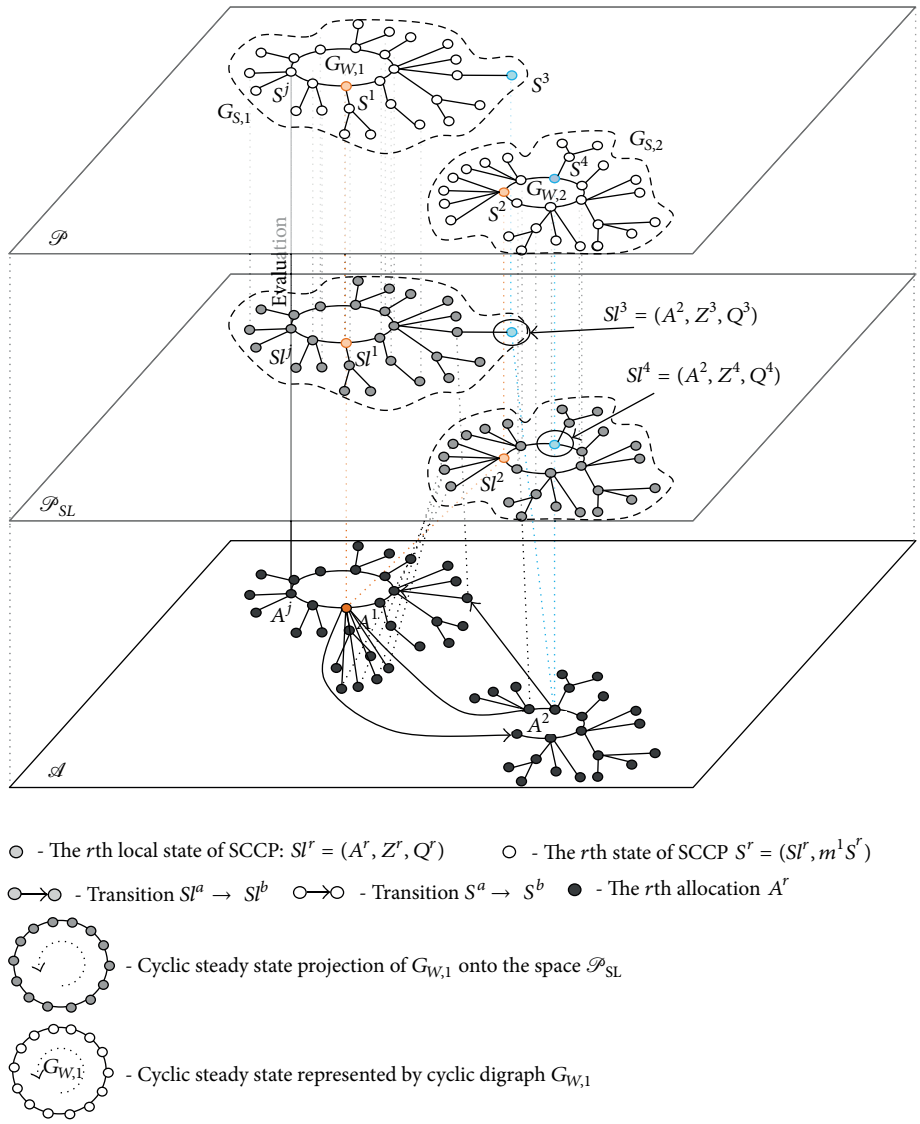


Figure 9: Illustration of cyclic steady state space projections, $\mathscr{P}_{SL}$ and $\mathscr{A}$, for behavior digraphs from Figure 8.

The considered transitions between Sl^3 and Sl^4 as well as Sl^1 and Sl^2 refer mainly to digraphs occurring only at the local level (i.e., space $\mathscr{P}_{SL}$). In case of digraphs $G_{S,1}$, $G_{S,2}$ of the space $\mathscr{P}$ a similar procedure should be followed. These states are the projections of the corresponding states from the space $\mathscr{P}$. The state Sl^1 is the projection of the state $S^1 = (Sl^1, m^1S^1)$, which is an element of the digraph $G_{W,1}$, and the state $S^2 = (Sl^2, m^1S^2)$ is the projection of the digraph $G_{W,2}$.

If, among states projecting upon Sl^1 and Sl^2, there are states (e.g., states Sl^1 and Sl^2) with mutual allocation m^1A^x of the multimodal processes, the transition between $G_{W,1}$ and $G_{W,2}$ (and therefore also between $G_{S,1}$ and $G_{S,2}$) is possible as a result of modifications of corresponding semaphores and indexes (from the form of m^1Z^1, m^1Q^1 into the form of m^1Z^2, m^1Q^2).

The states S^1 and S^2 are characterized by mutual allocation of both local and multimodal processes. By means of modifying semaphores and indexes in the state S^1, a direct transition from digraph $G_{W,1}$ to digraph $G_{W,2}$ is possible.

In practice, the change of semaphores and indexes related to them means simply the change of the control rules (change of signaling) of the system, the moment the processes are allocated properly (compatible with S^1). The solution of this kind neither stops the implementation of the process (work of AGVs, technological routes) nor creates the need for changing their allocation (shift). In case of the modification of the state S^6 the transition is immediate and noninvasive; it only requires the change of control.

To sum up the above considerations, we can say that the transition between the two digraphs $G_{W,1}$ and $G_{W,2}$ is possible if they include states characterized by mutual allocation at

every behavior level. This observation leads to the following two properties.

Property 1. Digraph $G_{W,2}$ is reachable from the digraph $G_{W,1}$ (which is denoted by $G_{W,1} \rightarrow G_{W,2}$) if there are states $S^a \in V_{W,1}$ and $S^b \in V_{S,2}$ (where $V_{W,1}$ is the set of states of the digraph $G_{W,1}$ and $V_{S,2}$ is the set of states of the digraph $G_{S,2}$) sharing mutual allocation of local and multimodal processes: $A^a = A^b; m^l A^a = m^l A^b$, for $l = 1 \cdots lp$.

Property 2. Two digraphs $G_{W,1}$ and $G_{W,2}$ are mutually reachable (which is denoted by $G_{W,1} \leftrightarrow G_{W,2}$) if $G_{W,1} \rightarrow G_{W,2}$ and $G_{W,2} \rightarrow G_{W,1}$.

In this approach, the problem of digraphs reachability calls for an answer to the following question: Are there two states $S^a \in V_{W,1}$ and $S^b \in V_{S,2}$, sharing the same allocations $A^a = A^b; m^l A^a = m^l A^b$, for $l = 1 \cdots lp$, among states included into $G_{S,1}$ and $G_{S,2}$? If such states exist, the transition between the states is possible as a result of modifications of semaphores and indexes. In this context the direct transition is defined as

$$\begin{aligned} &\cdots \longrightarrow S^{a_1} \longrightarrow S^{a_2} \longrightarrow \cdots \longrightarrow S^{a_{(x-1)}} \\ &\quad \longrightarrow \left(S^{a_x} \rightsquigarrow S^{b_x}\right) \longrightarrow S^{b_{(x+1)}} \longrightarrow \cdots \end{aligned} \tag{16}$$

and the indirect transition as

$$\begin{aligned} &\cdots \longrightarrow S^{a_1} \longrightarrow \cdots \longrightarrow \left(S^{a_x} \rightsquigarrow S^{d_x}\right) \longrightarrow \cdots \\ &\quad \longrightarrow S^{d_{ld}} \longrightarrow S^{b_j} \longrightarrow S^{b_{(j+1)}} \cdots, \end{aligned} \tag{17}$$

where one has the following: $S^{a_1}, \ldots, S^{a_{(x-1)}}, S^{a_x}, \ldots, S^{a_{la}}$—states of digraph $G_{W,1}$, $S^{b_1}, \ldots, S^{b_{(x-1)}}, S^{b_x}, \ldots, S^{b_{lb}}$—states of digraph $G_{W,2}$, $S^{d_1}, \ldots, S^{d_x}, S^{b_{(x+1)}} \ldots, S^{d_{lb}}$—states included into the transitory digraph leading G_T to $G_{W,2}$, $S^{a_1} \rightarrow S^{a_2}$—transition between states with SCCP assumption and transitory function δ [16], S^{a_x}, S^{b_x}—states with mutual allocation, and $S^{a_x} \rightsquigarrow S^{b_x}$—transition from the state S^{a_x} to S^{b_x} as a result of states modification, consisting of a sequence change of semaphores and indexes from Z^{a_x}, Q^{a_x} to Z^{b_x}, Q^{a_x}.

3.2.2. Searching Algorithm. In the situation when digraphs $G_{W,1}$ and $G_{W,2}$ and transitory processes leading to them are known (in other words, vortexes $G_{S,1}$ and $G_{S,2}$ (15) are known), determining the states with mutual allocation is not a complex task. Due to a small number of states (in practice no more than a few hundred states per vortex), all that has to be done is to successively compare all potential variants. An algorithm corresponding to such an approach looks as Algorithm 1.

In Algorithm 1, one has the following: $G_{S,1} = (V_{S,1}, E_{S,1})$, $G_{S,2} = (V_{S,2}, E_{S,2})$—input data, vortexes (15), S^a, S^b—states (15) included in the digraph $G_{W,1}, G_{S,2}$, and AC—set of pairs (S^a, S^b) of states with mutual allocation.

The result of Algorithm 1 is the set AC including pairs of states (S^a, S^b) with mutual allocations. The states can be used to determine the direct and indirect transitions between digraphs $G_{W,1}$ and $G_{W,2}$. Therefore, the existence of a nonempty set AC means a positive answer to the posed question.

To make things simple, an assumption can be made that we take into consideration that direct transitions and the digraphs have the same number of states (denoted by ld); thus computational complexity of Algorithm 1 is expressed by the function $f(ld) = ld^2$.

Algorithm 1 makes it possible to evaluate the mutual reachability of only two digraphs $G_{W,1}$ and $G_{W,2} \in$ DC (set of all cyclic digraphs existing in the space $\mathscr{P}$). In case of evaluating the mutual reachability of all digraphs from the set DC, it is necessary to make a search of this kind for a pair of processes. The computational complexity in such case amounts to

$$f(ld, dc) = \frac{1}{2}\left(dc^2 - dc\right) \cdot ld^2, \quad \text{where } dc = |\mathrm{DC}|. \tag{18}$$

Owing to polynomial character of the function of computational complexity, the problem of evaluating the mutual reachability of behavior digraphs is not a difficult one. All the computational effort is focused on the stage of determining cyclic digraphs (set DC), that is, solving the problem (10). In order to do so, methods described in [11, 12, 14–16] can be applied.

Among the available methods, however, there is a deficit of those which can help determine transient states leading to cyclic digraphs (i.e., digraphs G_T). Their usability is therefore limited to evaluating direct transitions (where no information on transient states (processes) is required).

In order to determine the transient states, the method of vortex generation can be used. Vortexes determined with use of this method include cyclic processes as well as all transient processes leading to them. The example below illustrates the use of this method for evaluating mutual reachability of behavior digraphs.

3.3. Numerical Example. Consider two AGVs supporting multiproduct manufacturing flows and their SCCP models shown in Figures 2 and 7(b). Let us assume that the two series manufacturing of three different products follow the routes distinguished by different colored (green, orange, and blue) bold lines.

Each workstation can process only unique work-piece from a unique product kind at a given instance. Successive work-pieces following the product route of a given product kind while taking into account local material handling routes are treated as subsequent streams of manufacturing processes (in a given kind of product manufacturing flow) and are modeled as streams of multimodal processes. For instance, see streams of mP_1, mP_2, and mP_3 from Figure 2 as follows:

$$\begin{aligned} mp_1^k &= \left(\left(R_3, R_{13}, R_6\right), \left(R_{17}, R_5\right), \left(R_{21}, R_8\right), \left(R_{22}, R_{11}\right)\right), \\ &\qquad k = 1, \ldots, 4, \\ mp_2^k &= \left(\left(R_2, R_{20}, R_5\right), \left(R_{24}, R_4\right), \left(R_{28}, R_7\right), \left(R_{29}, R_{10}\right)\right), \\ &\qquad k = 1, \ldots, 4, \end{aligned}$$

```
function StatesCoAll (G_{S,1} = (V_{S,1}, E_{S,1}), G_{S,2} = (V_{S,2}, E_{S,2}))
    AC = ∅
    for all S^a ∈ V_{W,1}
        for all S^b ∈ V_{S,2}
            if (A^a = A^b) & (m^1 A^a = m^1 A^b) &...& (m^{lp} A^a = m^{lp} A^b) then
                AC = AC ∪ (S^a, S^b)
            end
        end
    end
    return AC
end
```

Algorithm 1

$$mp_3^k = ((R_1, R_{27}, R_4), (R_{28}, R_7), (R_{25}, R_8), (R_{18}, R_9), (R_{15}, R_{12})), \quad k = 1, \ldots, 4, \tag{19}$$

and Figure 7(b) as follows:

$$\begin{aligned} mp_1^k &= ((R_3, R_{13}, R_6), (R_{17}, R_5), (R_{21}, R_8), (R_{18}, R_9), (R_{15}, R_{12})), \quad k = 1, \ldots, 4, \\ mp_2^k &= ((R_2, R_{20}, R_5), (R_{24}, R_4), (R_{28}, R_7), (R_{25}, R_8, R_{22}, R_{11})), \quad k = 1, \ldots, 3, \\ mp_3^k &= ((R_1, R_{27}, R_4), (R_{28}, R_7), (R_{29}, R_{10})), \quad k = 1, \ldots, 3. \end{aligned} \tag{20}$$

Both systems use the same fleet of 13 AGVs. A fleet supports work-pieces movements between workstations along the given manufacturing routes. The AGVs flows are modeled by streams P_1^1, P_2^1, P_2^2, P_2^3, P_3^1, P_4^1, P_4^2, P_5^1, P_5^2, P_6^1, P_6^2, P_7^1, and P_8^1 of local processes P_1–P_8. Each workstation (in general system resource) can be serviced only by one AGV at the moment. That means that both kinds of flow, that is, products and AGVs flows, are synchronized by a mutual exclusion protocol.

Given are the two cyclic schedules of multiproduct manufacturing supported by AVGs (see Figures 4 and 6) and corresponding to them two cyclic digraphs (Figures 7(a) and 7(b)). Due to this schedules in first system four work-pieces mP_i^1, mP_i^2, mP_i^3, mP_i^4 of each product ($i = 1, 2, 3$) are simultaneously processed in one cycle and one work-piece of each product is outputted at each 80 u.t. In second system four work-pieces of mP_1^k and three work-pieces of mP_2^k, mP_3^k product are simultaneously processed in one cycle and one work-piece of each product is outputted at each 96 u.t.

As mentioned before, operation times from the schedules considered take into account periods required for work-piece processing as well as material handling operations, that is, load/unload periods (1 u.t. for the work-piece load and 1 u.t. for its unload; see Figure 4). Moreover, the assumed time lags Δtm (see (9) and constraints from Table 2) enable to take into account the work-pieces transportation between workstations. In the case considered all time lags are the same and equal to $\Delta tm = 1$. Moreover, it is assumed that the input to R_1, R_2, and R_3 and the output of R_{10}, R_{11}, and R_{12} workstations are loaded and unloaded by dedicated conveyors.

The operation times mT and T of considered systems are defined in Tables 3 and 4. The timing $x_{i,j}^k$, $mx_{i,j}^k$ of commencement of operations following the constraints (8), (9) are presented in Tables 5 and 6.

The values presented in Tables 3–6 are the results of solving the problem (10), respectively, for the system from Figure 2 and Figure 7(b). In order to achieve this objective, OzMozart constraint programming environment was applied. The problem solving time in both cases did not exceed 3 seconds (Intel Core 2 Duo 3 GHz, RAM 4 GB).

During the analysis of digraphs of the attained schedules, it turned out that it is possible to smoothly change the behavior (with use of the state S^{18}) from the one presented in Figure 4 into the behavior from Figure 7. The transition was illustrated in Figures 11 and 12. In practice, a transition of this kind means the possibility of changing the current production order into a new one with no need to stop the system.

Therefore, a question may be posed whether an inverse situation is possible, when from the behavior represented by digraph $G_{W,2}$, behavior $G_{W,1}$ is attained (return to $G_{W,1}$). In other words, the question is the following: Are the considered behavior digraphs mutually reachable, $G_{W,1} \leftrightarrow G_{W,2}$?

According to Property 2 it is possible if $G_{W,1} \rightarrow G_{W,2}$ and $G_{W,2} \rightarrow G_{W,1}$. The schedule shown in Figure 10 illustrates the reachability $G_{W,1} \rightarrow G_{W,2}$. In order to determine whether $G_{W,2} \rightarrow G_{W,1}$ is possible, it must be stated (Property 1) if there are states of mutual allocations in digraphs $G_{W,2}$ and $G_{S,1}$ ($G_{S,1}$—digraph including $G_{W,1}$ and transitory path of states leading to it).

In order to determine these states, Algorithm 1 was applied. The use of Algorithm 1 depended upon the knowledge of digraph $G_{S,1}$. It should be emphasized that the solution of the problem (10) makes it possible to determine the cyclic schedule and digraph $G_{W,1}$ corresponding to it. It does not allow, however, determining states belonging to transitory processes. In the discussed case digraph $G_{S,1}$ is

Table 3: Times (units) of operations executions of local and multimodal processes from Figure 2.

$k = 1, 2, 3, 4$	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}	$R_{13} \div R_{33}$
mt_1^k	—	—	77	—	64	77	—	20	—	—	20	—	1
mt_2^k	—	77	—	20	10	—	30	—	—	30	—	—	1
mt_3^k	77	—	—	49	—	—	20	50	20	—	—	10	1
t_1^1	—	—	76	—	—	2	—	—	—	—	—	—	1
t_2^1	—	—	—	—	4	18	—	1	1	—	—	—	1
t_2^2	—	—	—	—	1	1	—	1	1	—	—	—	1
t_2^3	—	—	—	—	3	1	—	1	1	—	—	—	1
t_3^1	—	—	—	—	—	—	—	—	1	—	—	1	1
t_4^1	—	—	—	—	—	—	4	1	—	1	1	—	1
t_4^2	—	—	—	—	—	—	1	1	—	1	1	—	1
t_5^1	1	61	—	1	1	—	—	—	—	—	—	—	1
t_5^2	1	1	—	1	1	—	—	—	—	—	—	—	1
t_6^1	—	—	—	1	—	—	1	—	—	—	—	—	1
t_6^2	—	—	—	1	—	—	1	—	—	—	—	—	1
t_7^1	—	—	—	—	—	—	1	—	—	34	—	—	1
t_8^1	1	—	—	1	—	—	—	—	—	—	—	—	1

Table 4: Times (units) of operations executions of local and multimodal processes from Figure 7(b).

$k = 1, 2, 3, 4$	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}	$R_{13} \div R_{33}$
mt_1^k	—	—	91	—	64	91	—	20	20	—	—	20	1
mt_2^k	—	91	—	20	24	—	68	30	—	—	30	—	1
mt_3^k	91	—	—	68	—	—	20	50	—	—	—	—	1
t_1^1	—	—	90	—	—	2	—	—	—	—	—	—	1
t_2^1	—	—	—	—	4	28	—	1	1	—	—	—	1
t_2^2	—	—	—	—	1	1	—	1	1	—	—	—	1
t_2^3	—	—	—	—	1	1	—	18	1	—	—	—	1
t_3^1	—	—	—	—	—	—	—	—	1	—	—	15	1
t_4^1	—	—	—	—	—	—	1	1	—	1	1	—	1
t_4^2	—	—	—	—	—	—	60	1	—	1	1	—	1
t_5^1	1	62	—	1	1	—	—	—	—	—	—	—	1
t_5^2	1	1	—	6	1	—	—	—	—	—	—	—	1
t_6^1	—	—	—	13	—	—	1	—	—	—	—	—	1
t_6^2	—	—	—	66	—	—	1	—	—	—	—	—	1
t_7^1	—	—	—	—	—	—	1	—	—	81	—	—	1
t_8^1	1	—	—	1	—	—	—	—	—	—	—	—	1

unknown. Yet, the state of cyclic processes can be determined from digraph $G_{W,1}$ by means of searching for parents of the states $V_{W,1}$ not belonging to this set. Therefore, next state function δ defined in [16] was applied.

Based on the determined states, a search for mutual allocation states was carried out. An example of such states are the states S^{19}, S^{40}. A change of cyclic steady states based on these states (consisting of changing priority rules, semaphores and indexes) was illustrated in Figures 11 and 12.

As shown in Figure 12 the considered cyclic steady states are mutually reachable ($G_{W,1} \rightarrow G_{W,2}$ and $G_{W,2} \rightarrow G_{W,1}$). In general case, however, mutual reachability is not always guaranteed. For instance, consider SCCP composed of four cyclic processes shown in Figure 13(a). The cyclic steady state space $\mathscr{P}$ consists of two vortexes $G_{S,1}$ and $G_{S,2}$; see Figure 13(b). There are five different transient periods linking $G_{W,1}$ and $G_{W,2}$; see orange arcs. There is not, however, any transient period linking $G_{S,2}$ and $G_{S,1}$; that is, there are not any states shared by $G_{W,2}$ and $G_{S,1}$ while following Property 1. Moreover, both direct mutual reachability and indirect mutual reachability are not guaranteed in general case of a cyclic steady state space.

4. Related Work

Many algorithms for the scheduling and routing of AGVs have been proposed. However, most of the existing results are applicable to systems with a small number of AGVs,

Table 5: The timing $x_{i,j}^k$, $mx_{i,j}^k$ of commencement of operations of local and multimodal processes from Figure 2.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}
mx_1^1	—	—	−10	—	146	68	—	211	—	—	232	—
mx_2^1	—	55	—	144	133	—	165	—	—	190	—	—
mx_3^1	9	—	—	87	—	—	137	158	209	—	—	230
x_1^1	—	—	−9	—	—	68	—	—	—	—	—	—
x_2^1	—	—	—	—	66	47	—	−8	−6	—	—	—
x_2^2	—	—	—	—	45	71	—	48	49	—	—	—
x_2^3	—	—	—	—	47	−7	—	51	70	—	—	—
x_3^1	—	—	—	—	—	—	—	—	68	—	—	−10
x_4^1	—	—	—	—	—	—	−7	−2	—	71	0	—
x_4^2	—	—	—	—	—	—	−2	70	—	−6	72	—
x_5^1	69	−9	—	57	53	—	—	—	—	—	—	—
x_5^2	−9	53	—	64	62	—	—	—	—	—	—	—
x_6^1	—	—	—	3	—	—	5	—	—	—	—	—
x_6^2	—	—	—	55	—	—	57	—	—	—	—	—
x_7^1	—	—	—	—	—	—	34	—	—	36	—	—
x_8^1	5	—	—	7	—	—	—	—	—	—	—	—

Table 6: The timing $x_{i,j}^k$, $mx_{i,j}^k$ of commencement of operations of local and multimodal processes from Figure 7(b).

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}
mx_1^1	—	—	−10	—	174	82	—	239	260	—	—	281
mx_2^1	—	55	—	172	147	—	193	262	—	—	293	—
mx_3^1	9	—	—	101	—	—	170	—	—	191	—	—
x_1^1	—	—	−9	—	—	82	—	—	—	—	—	—
x_2^1	—	—	—	—	80	51	—	−9	−1	—	—	—
x_2^2	—	—	—	—	49	85	—	51	53	—	—	—
x_2^3	—	—	—	—	51	−7	—	53	72	—	—	—
x_3^1	—	—	—	—	—	—	—	—	91	—	—	−1
x_4^1	—	—	—	—	—	—	−7	−5	—	85	11	—
x_4^2	—	—	—	—	—	—	13	74	—	11	103	—
x_5^1	78	−10	—	76	53	—	—	—	—	—	—	—
x_5^2	−9	53	—	78	76	—	—	—	—	—	—	—
x_6^1	—	—	—	−9	—	—	5	—	—	—	—	—
x_6^2	—	—	—	9	—	—	76	—	—	—	—	—
x_7^1	—	—	—	—	—	—	1	—	—	3	—	—
x_8^1	5	—	—	7	—	—	—	—	—	—	—	—

offering a low degree of concurrency. With a drastically increased number of AGVs in recent applications (e.g., in the order of a hundred in a container handling system), the combinatorial nature of the fleet scheduling and assignment problems rapidly increases the complexity. For instance, it appears clearly through constraints which relate to assigning a fleet to a route. Suppose that there are n fleet and m routes, the number of binary decisions variables is $m \times n$ and theoretically, there are $2^{n\times m}$ solutions to be considered. In fact, according to computational complexity, this problem is NP-hard [2]. A number of papers are concerned with the fleet assignment problem [1–3, 8] and maintenance planning [6, 7, 21]. However, few papers have considered the combination of fleet assignment and maintenance planning [2, 22]. Most publications on fleet assignment have been focused on the assignment of fleets to minimize the fleet operation cost [1, 3, 8]. Some papers on maintenance planning have been focused on planning maintenance tasks to minimize the maintenance cost [6, 7, 21]. Few papers have considered the integration of fleet assignment and maintenance planning [2, 22]. The inventory policy with preventive maintenance consideration is mostly studied for production system [6, 23, 24]. Because of unpredicted situations, rescheduling is required to deal with online decision making. Finally, scheduling and rescheduling fleet assignment and maintenance planning and inventory policy are implemented as a decision support system [22].

Because most of real-life available manufacturing processes manifest their cyclic steady state the alternative

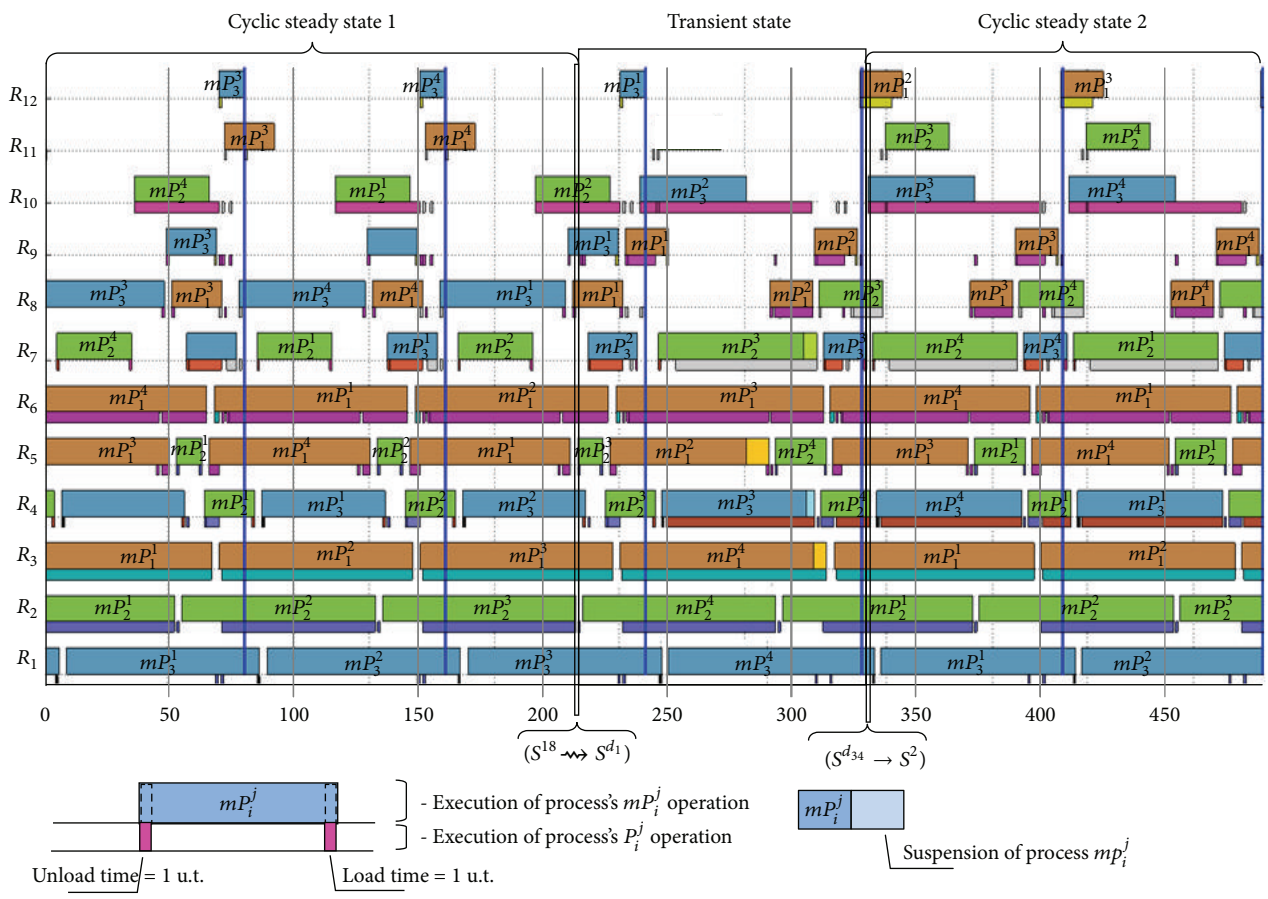


Figure 10: The indirect transition between schedules from Figures 4 and 6.

approach to AGVs fleet dispatching and scheduling, for example, following from the SCCP concept [1, 9, 25] can be considered as well. Many models and methods have been proposed to solve the cyclic scheduling problem so far. Most of these are oriented at finding a minimal cycle or maximal throughput while assuming deadlock-free processes flow. The approaches trying to estimate the cycle time from cyclic processes structure and the synchronization mechanism employed (i.e., mutual exclusion instances) are quite unique.

In that context our main contribution is to propose a new modeling framework enabling to state and answer the following questions: Does there exist a control procedure (e.g., a set of dispatching rules and an initial state) guaranteeing an assumed steady cyclic state subject to SCCP's structure constraints? Does an SCCP's structure exist such that a steady cyclic state can be achieved?

Is the cyclic steady sate space empty? In the case the space is not empty the next question is do the cyclic steady states are mutually reachable?

The last question refers to the problem of multimodal processes rescheduling. Most of so far proposed approaches to the rescheduling of processes executed within multimodal transportation networks environment [17, 26–28] are only limited to the deadlock-free cases. It should be noted that, however, the evolutionary algorithms driven approaches frequently used [26–28] are not effective in cases allowing deadlocks occurrence. The approach proposed allows finding deadlock-free transitions between the reachable cyclic steady states while avoiding the deadlocks in the course of the transient state generation.

5. Conclusions

The complicated problem which integrates AGVs fleet assignment, routing, and scheduling, generating a suboptimal cyclic schedule for multiproduct manufacturing, is novelty of this research. Such integrated AGVs dispatching problems can be seen as a special case of the cyclic blocking flowshop one, where the jobs might block either the machine or the AGV at the processing time. Therefore the main class of problems of AGVs fleet match-up scheduling with reference schedule of assumed production flow belong to the NP-hard ones.

Besides the above mentioned AGVs dispatching/planning issues the research objective regards a quite large class of digital and/or logistics networks that share common properties even though they have huge intrinsic differences. In other words, besides AGVs that can be treated as a kind of internal transport, other emerging trends concerning the logistics (e.g., supply chains management) and city traffic

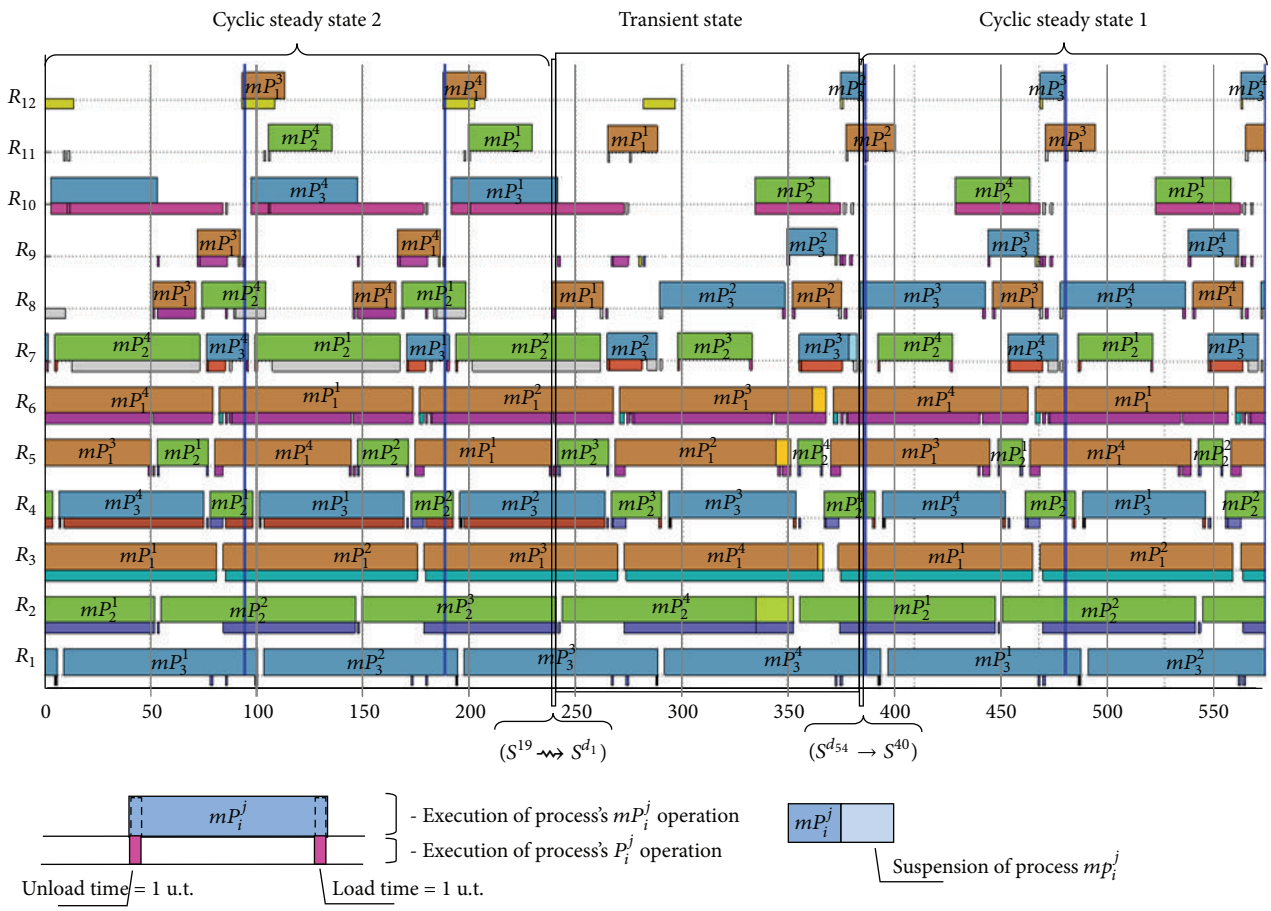


Figure 11: The indirect transition between schedules from Figures 6 and 4.

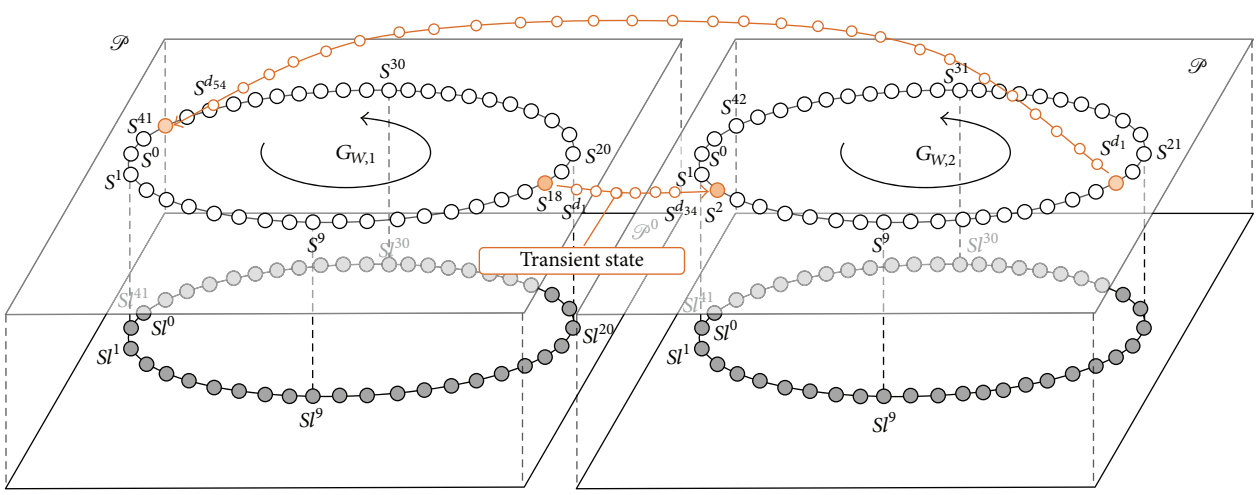
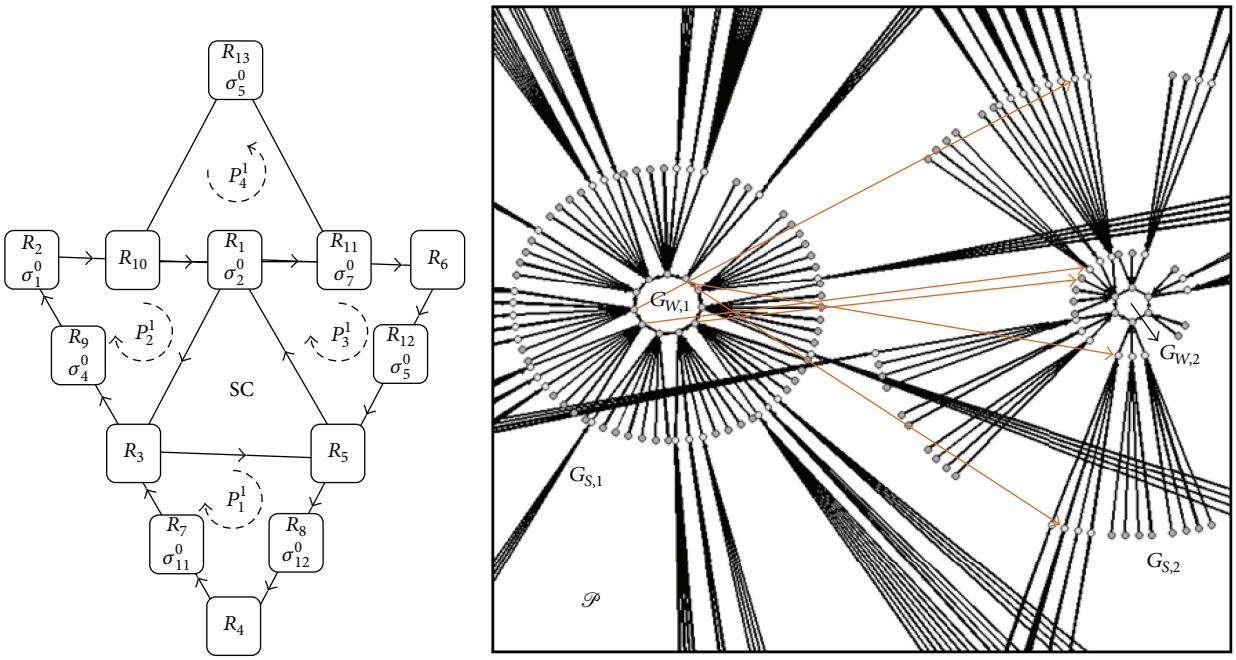


Figure 12: The indirect transition between cyclic steady states: $G_{W,1} \rightarrow G_{W,2}$ and $G_{W,2} \rightarrow G_{W,1}$.

(e.g., infrastructure of the public transport) issues can be modeled. For instance, the relevant multimodal processes of passengers traveling between destination points in an environment of local processes encompassing the subway lines network might be considered as well. In other words, the passenger's itinerary including different metro lines can be considered as a plan of a multimodal process routing within a metro network.

The proposed declarative approach aimed at AGVs fleet scheduling stated in terms of constraint satisfaction problem



(a) (b)

Figure 13: Illustration of SCCP composed of four local cyclic processes (a) and cyclic steady state space $\mathscr{P}$ including transient states depicting mutual reachability $G_{W,1} \rightarrow G_{W,2}$ (b).

representation provides a unified method for performance evaluation of local as well as supported by them multimodal processes.

In general case, however, there is not any guarantee that the considered SCCP model of AGVS has a nonempty cyclic steady states space as well as that in case this space is not empty, any direct or indirect mutual reachability can be observed. That means that the decidability of the problem of the space $\mathscr{P}$ reachability and mutual reachability among its cyclic steady states plays a pivotal role in multimodal processes rescheduling.

References

[1] J. Abara, "Applying integer linear programming to the fleet assignment problem," *Interfaces*, vol. 19, pp. 4–20, 1989.

[2] L. W. Clarke, C. A. Hane, E. L. Johnson, and G. L. Nemhauser, "Maintenance and crew considerations in fleet assignment," *Transportation Science*, vol. 30, no. 3, pp. 249–260, 1996.

[3] M. D. D. Clarke, "Irregular airline operations: a review of the state-of-the-practice in airline operations control centers," *Journal of Air Transport Management*, vol. 4, no. 2, pp. 67–76, 1998.

[4] N. G. Hall, C. Sriskandarajah, and T. Ganesharajah, "Operational decisions in AGV-served flowshop loops: fleet sizing and decomposition," *Annals of Operations Research*, vol. 107, no. 1–4, pp. 189–209, 2001.

[5] S. Hoshino and H. Seki, "Multi-robot coordination for jams in congested systems," *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 808–820, 2013.

[6] A. Chelbi and D. Ait-Kadi, "Analysis of a production/inventory system with randomly failing production unit submitted to regular preventive maintenance," *European Journal of Operational Research*, vol. 156, no. 3, pp. 712–718, 2004.

[7] S. Deris, S. Omatu, H. Ohta, L. C. Shaharudin Kutar, and P. Abd Samat, "Ship maintenance scheduling by genetic algorithm and constraint-based reasoning," *European Journal of Operational Research*, vol. 112, no. 3, pp. 489–502, 1999.

[8] C. A. Hane, C. Barnhart, E. L. Johnson, R. E. Marsten, G. L. Nemhauser, and G. Sigismondi, "The fleet assignment problem: solving a large-scale integer program," *Mathematical Programming*, vol. 70, no. 2, pp. 211–232, 1995.

[9] L. Qiu, W.-J. Hsu, S.-Y. Huang, and H. Wang, "Scheduling and routing algorithms for AGVs: a survey," *International Journal of Production Research*, vol. 40, no. 3, pp. 745–760, 2002.

[10] B. Trouillet, O. Korbaa, and J.-C. Gentina, "Formal approach of FMS cyclic scheduling," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 37, no. 1, pp. 126–137, 2007.

[11] E. Levner, V. Kats, D. Alcaide López De Pablo, and T. C. E. Cheng, "Complexity of cyclic scheduling problems: a state-of-the-art survey," *Computers and Industrial Engineering*, vol. 59, no. 2, pp. 352–361, 2010.

[12] T. Von Kampmeyer, *Cyclic scheduling problems [Ph.D. dissertation]*, Fachbereich Mathematik/Informatik, Universität Osnabrück, 2006.

[13] M. Polak, P. B. Z. Majdzik, and R. Wójcik, "The performance evaluation tool for automated prototyping of concurrent cyclic processes," *Fundamenta Informaticae*, vol. 60, no. 1-4, pp. 269–289, 2004.

[14] G. Bocewicz and Z. Banaszak, "Declarative approach to cyclic scheduling of multimodal processes," in *EcoProduction and*

Logistics, P. Golińska, Ed., pp. 203–238, Springer, Heidelberg, Germany, 2013.

[15] G. Bocewicz and Z. Banaszak, "Declarative approach to cyclic steady states space refinement: periodic processes scheduling," *International Journal of Advanced Manufacturing Technology*, vol. 67, no. 1–4, pp. 137–155, 2013.

[16] G. Bocewicz, R. Wójcik, and Z. Banaszak, "Cyclic Steady State Refinement," in *Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence*, A. Abraham, J. M. Corchado, S. Rodríguez González, and J. F. de Paz Santana, Eds., vol. 91 of *Advances in Intelligent and Soft Computing*, pp. 191–198, Springer, Berlin, Germany, 2011.

[17] M. Dridi, K. Mesghouni, and P. Borne, "Traffic control in transportation systems," *Journal of Manufacturing Technology Management*, vol. 16, no. 1, pp. 53–74, 2005.

[18] J.-S. Song and T.-E. Lee, "Petri net modeling and scheduling for cyclic job shops with blocking," *Computers and Industrial Engineering*, vol. 34, no. 2-4, pp. 281–295, 1998.

[19] A. Che and C. Chu, "Optimal scheduling of material handling devices in a PCB production line: problem formulation and a polynomial algorithm," *Mathematical Problems in Engineering*, vol. 2008, Article ID 364279, 21 pages, 2008.

[20] G. Bocewicz, I. Nielsen, and Z. Banaszak, "AGVsfleet match-up scheduling with production flow considerations," *Engineering Applications of Artificial Intelligence*. In press.

[21] N. Papakostas, P. Papachatzakis, V. Xanthakis, D. Mourtzis, and G. Chryssolouris, "An approach to operational aircraft maintenance planning," *Decision Support Systems*, vol. 48, no. 4, pp. 604–612, 2010.

[22] W. El Moudani and F. Mora-Camino, "A dynamic approach for aircraft assignment and maintenance scheduling by airlines," *Journal of Air Transport Management*, vol. 6, no. 4, pp. 233–237, 2000.

[23] N. Rezg, S. Dellagi, and A. Chelbi, "Joint optimal inventory control and preventive maintenance policy," *International Journal of Production Research*, vol. 46, no. 19, pp. 5349–5365, 2008.

[24] T. S. Vaughan, "Failure replacement and preventive maintenance spare parts ordering policy," *European Journal of Operational Research*, vol. 161, no. 1, pp. 183–190, 2005.

[25] M. Sharma, "Control classification of automated guided vehicle systems," *International Journal of Engineering and Advanced Technology*, vol. 2, no. 1, pp. 191–196, 2012.

[26] B. F. Chaar and S. Hammadi, "Evolutionary approach to the regulation of the traffic of a system of multimodal transport," *Journal Europeen des Systemes Automatises*, vol. 38, no. 7-8, pp. 901–931, 2004.

[27] A. K.-S. Wong, *Optimization of container process at multimodal container terminals [Ph.D. thesis]*, Queensland University of Technology, Brisbane, Australia, 2008.

[28] S. Zidi and S. Maouche, "Ant colony optimization for the rescheduling of multimodal transport networks," in *Proceedings of the IMACS Multiconference on Computational Engineering in Systems Applications*, vol. 1, pp. 965–9971, 2006.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 260351, 6 pages
http://dx.doi.org/10.1155/2013/260351

Research Article

Photovoltaic Power Prediction Based on Scene Simulation Knowledge Mining and Adaptive Neural Network

Dongxiao Niu, Yanan Wei, and Yanchao Chen

School of Economics and Management, North China Electric Power University, Beijing 102206, China

Correspondence should be addressed to Yanan Wei; weiyanan010@163.com

Received 20 March 2013; Revised 3 September 2013; Accepted 3 September 2013

Academic Editor: Chuandong Li

Copyright © 2013 Dongxiao Niu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Influenced by light, temperature, atmospheric pressure, and some other random factors, photovoltaic power has characteristics of volatility and intermittent. Accurately forecasting photovoltaic power can effectively improve security and stability of power grid system. The paper comprehensively analyzes influence of light intensity, day type, temperature, and season on photovoltaic power. According to the proposed scene simulation knowledge mining (SSKM) technique, the influencing factors are clustered and fused into prediction model. Combining adaptive algorithm with neural network, adaptive neural network prediction model is established. Actual numerical example verifies the effectiveness and applicability of the proposed photovoltaic power prediction model based on scene simulation knowledge mining and adaptive neural network.

1. Introduction

Facing the increasingly severe problem of tradition energy consumption and environment pollution, solar power is attracting more and more attention. With the development of grid-connected photovoltaic power (PV) generation, it has been regarded as a kind of effective way to make full use of solar energy, which is economic and environmental [1]. Recently, the grid-connected photovoltaic power system has been widely used around the world. However, due to the variations of light intensity and temperature, solar power owns the characters of volatility and intermittece, which are not conducive to stable operation of power grid. Therefore, it is important to predict the output power with less forecast error, to reduce the influence of PV power on grid, and to improve security and stability of power system [2–4].

In the early period, trend extrapolation, regression analysis, time series [5], and so on are the main methods to power prediction. But these methods have been difficult to meet the requirements of modern industry. Recently, artificial intelligence method becomes hot spot. Many new techniques and methods such as fuzzy method, expert system, artificial neural network (ANN), wavelet analysis, and support vector machine (SVM), and so forth, have been developed quickly [6]. Among them, it is recognized that ANN is a more effective way for short-term load forecasting [7] and has made many successful applications [8]. Based on data of light intensity and temperatures, some researches [9] use ANN to forecast PV power. However, there is little research on the overall consideration about various factors. Besides light intensity and temperature, there are numerous factors to be considered, including angles of incidence, conversion efficiency, installation angles of PV array, atmospheric pressure, and some other random factors. This paper comprehensively considers various factors to forecast photovoltaic power. For amount factors, the above parameters would require lots of computational resources, and ANN might suffer from multiple local minima easily. What is more results lacking clarity need to be explained. Thus, this paper introduces a new concept: scene simulation knowledge mining (SSKM) aimed to analyze the influencing factors and simulate the output power of PV.

This paper is organized as follows. In Section 2, the influencing factors of PV and the relationship between factors and output power are analyzed. The SSKM is used to cluster all elements in Section 3, and an adaptive algorithm is put forward to establish an improved adaptive ANN model. To prove the accuracy and practicability of the method, some calculating examples are given in Section 4.

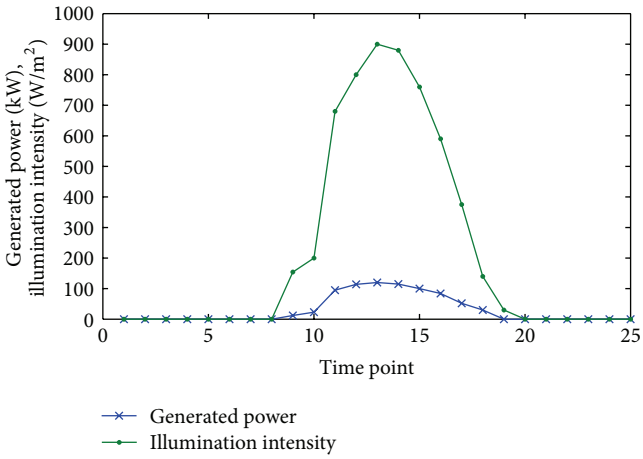


Figure 1: Photovoltaic generation power and light intensity.

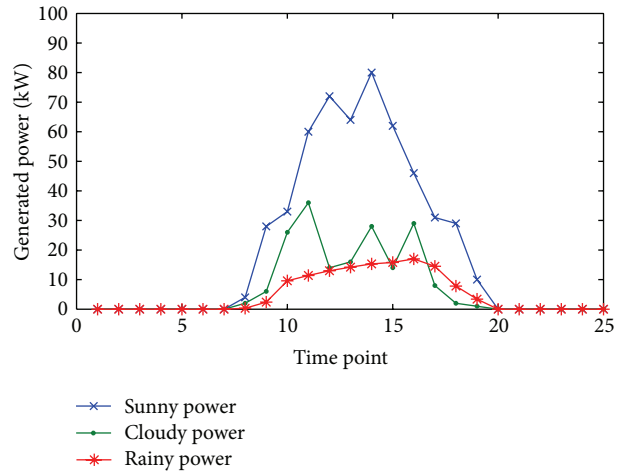


Figure 2: Photovoltaic power on different day types.

2. Factors Analysis

Many factors influence the generating capacity of PV grid system. Actually, because of various conditions, it could not be determined in advance one by one and is not necessary to be distinguished meticulously. Some influences can be combined into several correction groups, added with certain safety coefficient.

Influenced by many factors, change of PV generation capacity is a nonstationary random process with obvious cyclicity. An obvious feature for PV system is that the output time series is highly autocorrelated. Almost all PV grid inverters run with a relatively stable power conversion efficiency at maximum power point tracking (MPPT) model. Its output power is highly correlated. Although the efficiency of power conversion and photoelectric conversion changes over time, during system life cycle, the variation is relatively small, so much so that in short-term prediction can be considered as constant.

Therefore, the power conversion efficiency of inverter, photoelectric conversion efficiency of PV array, and area of the PV array can be ignored, since they are implicitly included in electricity data. Based on the above analysis, light intensity, day type, temperature, season, and the output data are the major factors taken into consideration in this paper.

2.1. Light Intensity. A method of calculating output power per unit area can be found in [10]. Consider

$$P_s = \eta SI\left(1 - 0.005\left(t_0 + 25\right)\right), \tag{1}$$

where η donates photovoltaic power conversion efficiency; the area of photovoltaic power supply is S; light intensity is I; and the environment temperature is t_0. Figure 1 is the photovoltaic generation power and light intensity curve. We can see from Figure 1 that the variation regularity of power curve and light intensity curve is similar. The change trend of photovoltaic generation power can be mapped to the change of light intensity. Therefore, this paper takes light intensity as an input value of the forecasting model.

2.2. Day Type. Generally, PV system mainly outputs power from 7:00 to 18:00. Figure 2 shows the power curves at a PV system in sunny, cloudy, and rainy days. It is easy to recognize that there are large differences among different day types.

For sunny day type, changes of electricity curves can broadly reflect the intensity of sun irradiation in a day. When the weather becomes rainy, solar irradiation intensity decreases obviously. If the input parameters ignore these changes of radiation intensity, the forecast will be inaccurate. So, some appropriate variables to reflect weather change correctly as well as corresponding variation of PV are needed.

With the development of weather forecast, the prediction model ought to take daily weather forecast information into consideration to deal with different day types. But, weather parameters are generally given by vague description, such as sunny, cloudy, rain, moderate rain, or heavy rain. So, plenty of statistical analyses on historical electricity have to be done to decide how to change vague and uncertain description of day types to accurate information, which is accepted by prediction model. Thus, In order to improve the prediction accuracy, PV data needs to be classified into three groups: sunny, cloudy, and rainy.

2.3. Temperature. The changes of atmospheric temperature can influence photovoltaic power system to a certain degree. Though historic data of photovoltaic reflects similarity between power curve and day types, the changes of temperature can reflect tiny change of curve height in the same day. So, we should take atmospheric temperature as an input variable.

Figure 3 shows the relationship between average day PV power and temperature of a photovoltaic system, in which the day type is sunshine. From Figure 3, we can know that in the same day type, the higher the average temperature is, the larger the day output power is.

2.4. Season. Season also plays an important role in photovoltaic output. The output of photovoltaic modules is changing with solar radiation intensity, and solar radiation intensity is different in diverse seasons.

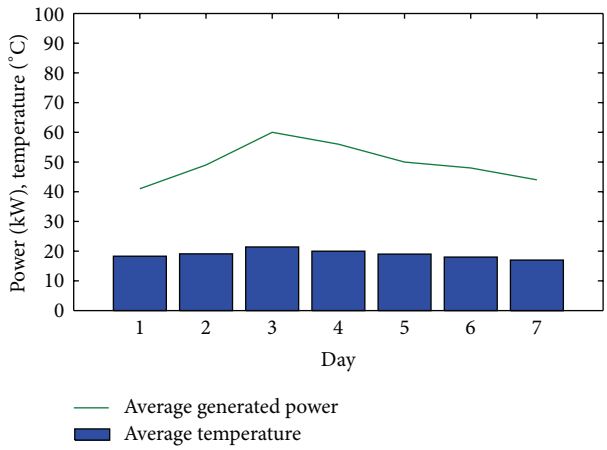


Figure 3: Influence of temperature on photovoltaic power.

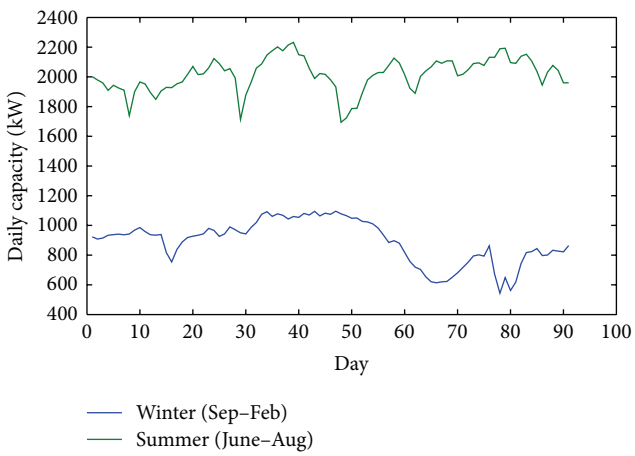


Figure 4: Photovoltaic generation power in different season.

Besides, the photovoltaic system is installed in different locations where the climatic condition is variable, which make the effect degree of season on photovoltaic output the different. According to NASA, we can get atmosphere data of diverse districts. The data reflects that temperature and solar radiation intensity vary a lot in different months. Especially in summer, the numerical values of temperature and solar radiation are much higher than that in winter. So, we can infer that the output of photovoltaic system varies in variable seasons.

Based on the above analysis, we make statistics choosing electric production data from June to August of 2010 and from December 2010 to February 2011. We aggregate data of two to three months and make comparison in accordance with day arrays. In Figure 4, we can find a law that the electric production in summer is higher than that in winter. On the one hand, the solar radiation intensity is much stronger in summer; on the other hand, the solar radiation time is also much longer than that in winter, which makes the photovoltaic modules work longer and more effective. At the same time, we can see the electric production in summer fluctuates a lot more than in winter, which means that the output of photovoltaic system is largely influenced by rainy days.

This section mainly analyzes several influential factors aimed at the uncertainty of photovoltaic output, including day type, season, temperature, and solar radiation. Through the comparison and analysis of historic data, we can approximately know their effect on the photovoltaic output. In the following section, we will build an accurate and reasonable forecast model based on the results above.

3. Prediction Model Based on Scene Simulation Knowledge Mining and Adaptive Neural Network

For photovoltaic power, we mainly consider the influence of light intensity, season, day type, and temperature. According to scene simulation knowledge mining, select high similarity historical data and meteorological environment as learning sample to complete scene simulation knowledge mining as the first step. Neural network prediction model can simulate arbitrary complex nonlinear mapping with the advantage of being intelligent. Here adaptive neural network is chosen to forecast photovoltaic power.

3.1. Scene Simulation Knowledge Mining Technique. By the above analysis, we know that the influence degree of meteorological factors on photovoltaic power is different. Light intensity, day type, temperature, and season can make different degrees of influence on photovoltaic power, and the power is often a result of combined action of various meteorological factors [11–13]. How to effectively deal with various meteorological factors is important to improve the prediction accuracy.

According to scene simulation knowledge mining technique, take the day characteristic similarity and the former trend similarity as index to choose similar scene as input value of forecast model.

Definition 1. Characteristic similarity for two days i and j is defined as follows:

$$O_{ij} = \frac{\sum_{k=1}^{m} x_{ik} y_{jk}}{\sqrt{\sum_{k=1}^{m} x_{ik}^2 \sum_{k=1}^{m} y_{ik}^2}}, \tag{2}$$

where O_{ij} is characteristic similarity. It is angle cosine between two eigenvectors in m-dimensional space which reflects the distance between two-day characters in m-dimensional space.

Definition 2. Former trend similarity of i, j is defined as follows:

$$F_{ij} = \frac{2\left[E\left(\overline{X_i}\,\overline{X_j}\right) - E\left(\overline{X_i}\right)E\left(\overline{X_j}\right)\right]}{D\left(\overline{X_i}\right) + D\left(\overline{X_j}\right)}. \tag{3}$$

Former trend similarity refers to average k day power curve similarity before the two days. Suppose k average power sequence before i, j is $(\overline{x_{ik}}, \overline{x_{i(k-1)}}, \ldots, \overline{x_{i1}})$, $(\overline{x_{jk}}, \overline{x_{j(k-1)}}, \ldots, \overline{x_{j1}})$. These values can be recognized as sampling points of, respectively, probability distribution random vector.

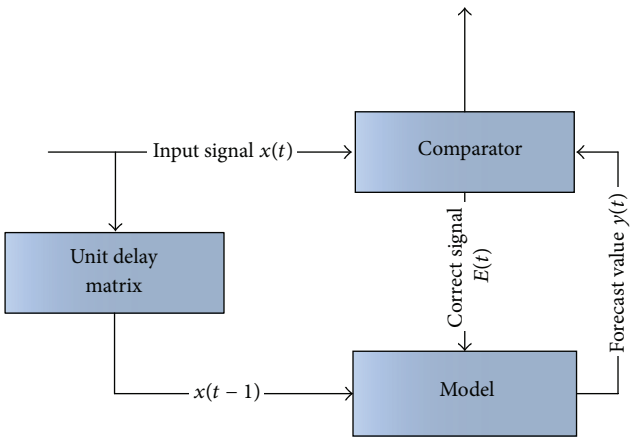


Figure 5: Block diagram of an adaptive system.

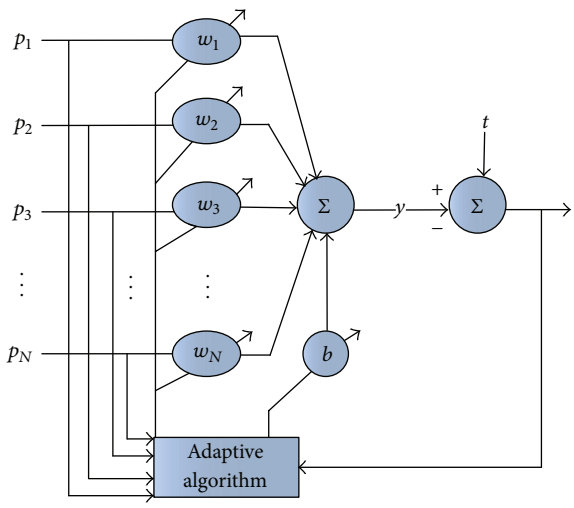


Figure 6: Adaptive artificial neural network.

3.2. Adaptive Neural Network. According to the latest situation to automatically adjust model so as to achieve satisfied prediction effect which is very important, adaptive prediction method continuously automatically adjusts model structure and parameters based on prediction deviation which actually forms a closed loop feedback [14]. A typical adaptive system block diagram is shown in Figure 5.

According to model and actual value of $t-1$ time and parameters of t time, we can predict output value $y(t)$ of t time. $E(t)$ is D-value of $y(t)$ and $x(t)$. If $E(t) = 0$, model parameter needs no correction. Otherwise, we amend the model parameters in order to track environment change.

Works of the literature [15] introduce the principle and application of adaptive neural network algorithm. Adaptive neural network can choose new paradigm for neural network training. Neuron structure of adaptive neural network is shown in Figure 6.

The output of Figure 6 is

$$y = \sum_{j=1}^{R} p_i w_{i,j} + b_i. \tag{4}$$

Adaptive neural network adopts the minimum mean square error (MSE) learning rule, namely, Windrow Hoff (WH) algorithm, to adjust weights and thresholds. For a given N set of training samples; $\{p_1, t_1\}$, $\{p_1, t_1\}, \ldots, \{p_N, t_N\}$, the basic implementation of WH learning rule is to find the best w and b, to minimize mean square error of each output neuron. The mean square error of neuron is

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2, \tag{5}$$

where t is the target value and y is the predicted value. Strives for partial derivatives of mean square error, we can get the optimal solution of w and b, respectively. Consider

$$\begin{aligned} \frac{\partial \mathrm{MSE}}{\partial w} &= \frac{\partial\left((1/N)\sum_{i=1}^{N}(t_i - y_i)^2\right)}{\partial w}, \\ \frac{\partial \mathrm{MSE}}{\partial b} &= \frac{\partial\left((1/N)\sum_{i=1}^{N}(t_i - y_i)^2\right)}{\partial b}. \end{aligned} \tag{6}$$

Setting type (6) equal to 0, extreme value point of mean square error can be calculated. Because of the continuous correction of model parameter, adaptive neural network is much faster than feed-forward network.

Adaptive neural network is able to obtain a set of continuous data, make accurate predictions, automatically discard old data, and study new data in the process of learning [16].

The adaptive data training process is as follows [17, 18].

(1) Define the error function as

$$E = \frac{1}{N}\sum_{k1=1}^{N}(t_{k1} - y_{k1})^2, \tag{7}$$

where N is the size of learning array; t_{k1} is target value; y_{k1} is predicted value. After learning weight w, make a prediction.

(2) Set studied weight and threshold values as initial values of new training array's weight and threshold, the new training function is

$$E = \frac{1}{N}\sum_{k2=1}^{N+1}(t_{k2} - y_{k2})^2. \tag{8}$$

(3) Start new forecast based on the studied weight in step (2). Proceed the above process until learning all of the data and making accurate forecast. This method can effectively deal with nonstatic data. Learning data for neural network is historical load and meteorological data in this paper.

4. Numerical Example Analysis

This power output prediction model of PV power generation system uses Mat lab programming implementation and selects Guangdong photovoltaic power generation system as an example. The precision time is 1 h, the time from 7:00 to

Table 1: Prediction error for different models.

Time	Winter			Spring			Summer			Fall		
	BP	NP	MODEL	BP	NP	MODEL	BP	NP	MODEL	BP	NP	MODEL
7	—	—	—	120.5	85.4	45.3	50.3	47.1	15.4	142.3	46.2	18.1
8	36.2	40.5	20.4	51.5	34	10.3	46.1	40.5	5	60.4	40	9.4
9	36	24	12.3	20.3	25	12.5	30.8	21.5	4	30.4	32.6	14.5
10	16.4	13.4	10.8	42.6	16	12.4	22.7	20.5	2.4	35.4	23.4	14.2
11	41.8	20.4	16.7	34.8	26.4	10.8	16.8	20.4	3.5	28.6	25	10.4
12	69	26.4	25.7	60.7	21.4	5.6	20.7	10.6	1.6	52	20.8	8.6
13	157.8	34.6	17.3	20.4	13.8	20.4	18.6	11.5	1.2	18.2	19.3	15.4
14	27.5	30.7	15	38.6	11.7	26.7	17.3	9.7	1	35.4	15.7	21.6
15	98.4	50.4	24.6	79.1	25.9	22.4	12.4	8.6	1.3	60.7	22.3	20
16	39	20	10.4	25.7	20.8	17.3	19.4	10.5	2.4	20.7	15.8	11.8
17	32.4	30	8.4	20	15.3	10.8	20.6	15.4	4.1	13	14	14
18	35.4	20.9	6.7	31	10.8	5.4	21.7	9.4	5.4	27.6	12.6	12.8
Average	53.6	28.3	15.3	45.4	25.5	16.7	24.8	18.8	3.9	43.7	24.0	14.2

Table 2: Prediction error for different day styles.

Time	Winter			Spring			Summer			Fall		
	Rainy day	Cloudy day	Sunny day	Rainy day	Cloudy day	Sunny day	Rainy day	Cloudy day	Sunny day	Rainy day	Cloudy day	Sunny day
7	—	—	—	52.4	34	20.4	30.4	22	15.4	35.4	20	10.5
8	36.2	20.5	12.5	30.4	25.1	11	24.5	15.4	0.5	23.4	21.4	2.4
9	36	24	12.7	32.4	23	12.3	11.5	16.4	0.8	31.4	22	4.6
10	37.6	23.7	10.4	35	21	10.4	20.4	12	2.1	30	21.4	0.8
11	41.8	20.7	16.5	34.8	26.4	14	15.3	10.5	1.5	28.6	20.1	10.5
12	30.5	26.4	15.4	30.4	22.4	10.5	12.5	13	1.4	27.5	15.2	5.4
13	25.4	24.5	18.4	31.2	15.4	11.4	10.5	5.6	1.6	30.4	10.5	12
14	27.5	20.1	12	24.3	13.4	5.4	25.6	8.4	2.4	25	5.4	4.5
15	35.1	23.5	11	38.1	20.7	8.6	24.1	8.6	1.8	34	15.4	16.4
16	39	24	10.3	25.7	22	7.1	29.5	10.7	0.7	29	18.6	11.4
17	32.8	22	11.7	30.4	15.3	10.8	26.4	11	1.4	27	14.2	10.5
18	33.4	20.4	6.7	30	14.5	6.3	22.7	13	1.8	34.5	10.4	2.4
Average	34.1	22.7	12.5	32.9	21.1	10.7	21.1	12.2	2.6	29.7	16.2	7.6

18:00. The sun radiation from 7 p.m. to 6 a.m. is 0, so it is not included in the study [19, 20].

In order to validate the proposed prediction model, one day was randomly chosen from each season of the year 2010, that is, D1 = December 25 for winter, D2 = April 20 for spring, D3 = July 15 for summer, and D4 = October 23 for fall. Three models are chosen to be predicted, respectively. Model 1 is traditional BP neural network forecasting model without the proposed scene simulation knowledge mining. Model 2 is ordinary adaptive neural network forecasting model. Model 3 is the proposed adaptive neural network prediction model based on scene simulation knowledge mining. The following table is PV power prediction error from different prediction models. As can be seen from Table 1, for the same season, prediction error of traditional BP neural network is much bigger, the adaptive neural network is relatively small, and the prediction result of the proposed adaptive neural network prediction model based on scene simulation knowledge mining is the best. For the proposed prediction method, analysis results in different seasons, comprehensive prediction error in summer (3.9%) is the minimum, and error in winter (14.2%) is the maximum.

In order to further assess the forecasting capability of the proposed adaptive neural network prediction model based on scene simulation knowledge mining, simulations are carried out for three different days—sunny day (SD), cloudy day (CD), and rainy day (RD) from each season, and the results obtained from the proposed model are presented in Table 2 where we can observe the lower values of MAPE in SDs compared to CDs and RDs.

Through the effective analysis of influencing factors of PV power output, combined with the improved adaptive neural network prediction model and scene simulation knowledge mining technique, a photovoltaic power prediction model based on scene simulation knowledge mining and adaptive neural network is proposed. The benefit of the proposed

approach is that it does not require complex modeling and complicated calculation; forecast under different weather types can be carried out using only historical power data and weather data. The test results proved validity and accuracy of the proposed approach; the proposed approach can be used to forecast the power output of photovoltaic system precisely.

5. Conclusions

With the development of the grid-connected photovoltaic power (PV), it has been regarded as a kind of effective way to make full use of solar energy, which is economic and environmental. Influenced by light, temperature, atmospheric pressure, and some other random factors, photovoltaic power has characteristics of volatility and intermittece. Accurately forecasting photovoltaic power can effectively improve security and stability of power grid system. We comprehensively analyze the influence of light intensity, day type, temperature, and season on photovoltaic power in this paper. According to proposed scene simulation knowledge mining technique, similar meteorological factors and power are taken as input value in prediction model. Combining advanced adaptive neural network and scene simulation knowledge mining, photovoltaic power prediction model based on scene simulation knowledge mining and adaptive neural network is established. Guangdong photovoltaic power generation system is selected to verify the proposed model. The test results proved validity and accuracy of the proposed approach; the proposed approach can be used to forecast the power output of photovoltaic system precisely.

Acknowledgments

This work was partially supported by the Natural Science Foundation of China (71071052) and the Fundamental Research Funds for the Central Universities (12QX22).

References

[1] D. Niu, Y. Wei, Y. Shi, and H. R. Karimi, "A novel evaluation model for hybrid power system based on vague set and Dempster-Shafer evidence theory," *Mathematical Problems in Engineering*, Article ID 784389, 12 pages, 2012.

[2] D. Niu, Y. Wei, and L. Fan, "Social comprehensive benefit evaluation for hybrid power system including thermal power, wind power and photovoltaic power," *Energy Education Science and Technology B*, vol. 5, no. 2, pp. 1113–1120, 2013.

[3] R.-J. Wai and C.-Y. Lin, "Active low-frequency ripple control for clean-energy power-conditioning mechanism," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3780–3792, 2010.

[4] I.-S. Kim, M.-B. Kim, and M.-J. Youn, "New maximum power point tracker using sliding-mode observer for estimation of solar array current in the grid-connected photovoltaic system," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 4, pp. 1027–1035, 2006.

[5] A. Yona, T. Senjyu, T. Funabashi et al., "Application of neural network to one-day-ahead 24 hours generating power forecasting for photovoltaic system," *Intelligent Systems Applications to Power Systems*, vol. 1-2, pp. 423–428, 2007.

[6] H. Zhao, Z. Ren, and W. Huang, "Short-term load forecasting considering weekly period based on periodical auto regression," *Proceedings of the Chinese Society of Electrical Engineering*, vol. 17, no. 3, pp. 211–216, 1997.

[7] X. Li, M. Luo, and K. Feng, "Comparison of neural network methods for short-term load forecasting," *Relay*, vol. 35, no. 6, pp. 49–53, 2007.

[8] A. Prias, B. Buchenel, and Y. Jaccard, "Heterogeneous artificial neural network for short term electrical load forecasting," *IEEE Transactions on Power Systems*, vol. 11, no. 1, pp. 456–463, 1996.

[9] A. Yona, T. Senjyu, and T. Funabashi, "Application of recurrent neural network to short-term-ahead generating power forecasting for photovoltaic system," in *Proceedings of the IEEE Power Engineering Society General Meeting*, June 2007.

[10] W. Guo, J. Liu, Y. Ma et al., "Optimal algorithm of electric power system's short-term load forecasting based on radial function neural network," *Power System Protection and Control*, vol. 36, no. 23, pp. 45–48, 2008.

[11] C. Kang, A. Zhou, P. Wang et al., "Impact analysis of hourly factors in short-term load forecasting and its processing strategy," *Power System Technology*, vol. 30, no. 7, pp. 5–11, 2006.

[12] X. Liu, D. Luo, J. Yao et al., "Short-term load forecasting based on load decomposition and hourly weather factors," *Power System Technology*, vol. 33, no. 12, pp. 94–100, 2009.

[13] H. Qin, W. Wang, H. Zhou et al., "Short-term electric load forecast using human body amenity indicator," *Proceedings of the CSU-EPSA*, vol. 18, no. 2, pp. 63–66, 2006.

[14] D. Wu, "Discussion on various formulas for forecasting human comfort index," *Meteological Science and Technology*, vol. 31, no. 6, pp. 370–372, 2003.

[15] A. Sinha, "Short term load forecasting using artificial neural network," *IEEE Transactions on Power Systems*, pp. 548–553, 2000.

[16] C. Kang, Q. Xia, and B. Zhang, "Review of power system load forecasting and its development," *Automation of Electric Power Systems*, vol. 28, no. 17, pp. 1–11, 2004.

[17] H. Yoo and R. Pimmel, "Short term load forecasting using a self-supervised adaptive neural network," *IEEE Transactions on Power Systems*, vol. 14, no. 2, pp. 779–784, 1999.

[18] J. Wu and H. Yan, "Short-term load forecasting technique based on adaptive optimal fuzzy logic system," *Automation of Electric Power Systems*, vol. 23, no. 17, pp. 35–37, 1999.

[19] J. Xie and G. Tang, "Fuzzy inference forecasting of power load with self-learning function," *Journal of Southeast University*, vol. 28, no. 4, pp. 156–160, 1998.

[20] D. Singh and S. P. Singh, "Self organization and learning methods in short term electric load forecasting: a review," *Electric Power Components and Systems*, vol. 30, no. 10, pp. 1075–1089, 2002.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 869356, 12 pages
http://dx.doi.org/10.1155/2013/869356

Research Article

Using Software Dependency to Bug Prediction

Peng He,[1] Bing Li,[2] Yutao Ma,[2] and Lulu He[2]

[1] *State Key Lab of Software Engineering, School of Computer, Wuhan University, Wuhan 430072, China*
[2] *State Key Lab of Software Engineering, School of Computer, Research Center of Complex Network, Wuhan University, Wuhan 430072, China*

Correspondence should be addressed to Peng He; penghe@whu.edu.cn

Received 12 March 2013; Revised 16 August 2013; Accepted 16 August 2013

Academic Editor: Chuandong Li

Copyright © 2013 Peng He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Software maintenance, especially bug prediction, plays an important role in evaluating software quality and balancing development costs. This study attempts to use several quantitative network metrics to explore their relationships with bug prediction in terms of software dependency. Our work consists of four main steps. First, we constructed software dependency networks regarding five dependency scenes at the class-level granularity. Second, we used a set of nine representative and commonly used metrics—namely, centrality, degree, PageRank, and HITS, as well as modularity—to quantify the importance of each class. Third, we identified how these metrics were related to the proneness and severity of fixed bugs in Tomcat and Ant and determined the extent to which they were related. Finally, the significant metrics were considered as predictors for bug proneness and severity. The result suggests that there is a statistically significant relationship between class's importance and bug prediction. Furthermore, betweenness centrality and out-degree metric yield an impressive accuracy for bug prediction and test prioritization. The best accuracy of our prediction for bug proneness and bug severity is up to 54.7% and 66.7% (top 50, Tomcat) and 63.8% and 48.7% (top 100, Ant), respectively, within these two cases.

1. Introduction

During the software development and maintenance procedures, bug (defect) is one of the most important forces to drive the improvement of software quality. It is well-known that software engineering is a systematic and disciplined approach to developing software. In detail, it applies computer science, engineering principles and practices to the generation, operation, and maintenance of software systems. There are many key processes in software engineering. In particular, Software maintenance and upgrade play a vital role in software engineering. It has many advantages, such as improving the efficiency of programming, reducing the cost of maintenance, and promoting the development of software systems. In fact, most development efforts and expenditures are allocated to this stage. We know that the majority of software projects today are becoming increasingly large and complex. While maintenance is considered as an ongoing process throughout the software life cycle, according to maintenance activities account for over 70% of the total life cycle costs of a software product. Large development project incurs a sizable number of bug reports every day, more and more efforts should be involved to resolve these burdened problems. Thus the effective and efficient solutions to bug prediction and test prioritization are urgent for both open source software and proprietary software.

Due to notable discoveries in the fast evolving field of complex networks and the dramatic increase of scale and complexity of real-world software systems, more and more researches in software engineering have also focused on representing the topology of software systems with network theory. Software represents one of the most diverse and sophisticated human made systems; however, little is known about the actual structure and quantitative properties of (large) software systems. In the context of combining complex networks theories with software engineering practices, the research for bug prediction has already made several discoveries over the past years. In this paper we propose to use some quantitative metrics from network sciences to actualize bug prediction of complex systems and further enhance software engineering practices.

Given the advantages of open source, (i.e., the openness of source codes and the available data repositories), the following work is conducted on open-source software. Open-source software usually maintains a bug repository (e.g., Bugzilla (http://www.bugzilla.org/), GNANTS (http://www.gnu.org/software/gnats/), and JIRA (http://www.atlassian.com/software/jira/)). With the increasing scale of software project, new features are added and more bugs are led into the system. According to the statistics [1], Eclipse on average 37 bugs was submitted, and more than 3 person-hours are spent on handling it per day; Mozilla discovers more than 300 bugs. However, potential bugs are often far more than that have been submitted in a system, and more serious. How to quickly retrieve much more bugs or serious bugs is the overarching goal of this work, specifically which metrics can be adapted to resolve this problem.

The rest of this paper is organized as follows. Section 2 is a review of related work. In Section 3, the preliminary theories of software networks and network metrics are hold and the research questions are presented, and Section 4 focuses on the research approach and the construction of software network. Section 5 shows the whole progress of our experiment. After that, we discuss the empirical results and several limitations. In Section 7, a conclusion for practice and research is made.

2. Related Work

Defect prediction models are used to support software inspection and to improve software quality by locating possible defects. For software bug prediction, many software metrics have been proposed. The most commonly-used ones are CK Object-oriented metrics, also traditional (e.g., LOC), and process (e.g., code churn) metrics are often used in some literature [2]. With the increasing maturity of complex network theory, network metrics are of more concerns, such as centrality and are used to defect prediction [3–5].

2.1. Software Network. Many real-world software systems are extremely complex and regarded as complex systems, in which software entities (e.g., objects, classes, and packages) are abstracted as nodes and the dependencies between two nodes as links or edges. Dependencies essentially represent the information flow within a software system and exert at least some influences on the overall success and quality of the product [6].

Characterizing large or ultralarge scale (ULS) software system as a complex networks is rational, and a great quantity of literature [7–12] has already demonstrated some software networks characteristics, like small-world phenomenon and scale-free distribution of degree. Likewise, with regard to software being built up out of many interacting units and subsystems (software entities) at multiple granularity, different levels of granularity software networks (component, feature, class, method etc.) have been constructed to study by researchers [13].

2.2. Network Metrics. Many metrics have been defined in both complex networks and social network analysis. Centrality [14, 15] was used to assess the relative importance of nodes in a given network. The simplest one is *degree centrality*, known as the number of connections a node has to other nodes. In a social context, degree centrality can be explained in terms of the potential impact of a node on other nodes. However, degree centrality does not capture the global position of a node in the network in terms of how important a node is to all other nodes, so further measures are proposed such as *closeness centrality*, *betweenness centrality*, and *eigenvector centrality*.

Modularity [15, 16] is a desirable characteristic for software systems. The modularity of software architecture contributes to the sustainability of large scale software projects by fostering the decoupling and cohesiveness of software development. Particularly, as the software evolves over time, modularity might even facilitate its maintainability and expendability. In some literature, besides, in-degree and out-degree [8] presenting the complexity and reuse were used, respectively. PageRank and HITS [17] sorting algorithm were used to handle the directed network.

Additionally, Zimmermann and Nagappan [3] on Windows Server 2003 where the authors leveraged dependency relationships between software entities captured using social network metrics to predict whether they are likely to have defects. They found that network metrics perform significantly better than source code metrics at predicting defects. Premraj and Herzig. [18] presented a replication study based on Z&N work, and found that network metrics undoubtedly perform much better than code metrics in the stratified random sampling method.

2.3. Network Metrics for Application. Meneely et al. [4] examined collaboration structure with the developer network derived from code churn information to predict failures at the file level and developed failure prediction models using test and postrelease failure data. The result indicated that a significant correlation exists between file-based developer network metrics and failures. Crowston et al. [5] identified the core-periphery structure and applied these measures to the interactions around bug fixing for 116 SourceForge projects. Pinzger et al. [19] investigated the relationship between the fragmentation of developer contributions, measured by network centrality metrics and the number of postrelease failure with a developer-module network. Authors claimed that central modules were more likely to be failure prone than those located in surrounding areas of the network. Shin et al. [20] used network analysis to discriminate and predict vulnerable code locations. They evaluated that network metrics can discriminate between vulnerable and neutral files and predict vulnerabilities. Sureka et al. [21] derived a collaboration network from a defect tracking system and applied network analysis to investigate the derived network for the purpose of risk and vulnerability analysis. They demonstrated that important information of risk and vulnerability can be uncovered using network analysis techniques.

Dependencies exist between various pieces of component. Modification to components with little regard to dependencies may have an adverse impact on the quality of the latter. Zimmermann and Nagappan [3] proposed to use

network analysis on software dependency network, helping identify central program units that are more likely to conceal defects. Perin et al. [22] used PageRank for ranking classes of Pharo Smalltalk system based on a dependency graph representing class inheritance and reference. Bhattacharya et al. [23] constructed software networks at source code and module level to capture its evolution and estimate bug severity, prioritize refactoring efforts, and predict defect-prone release. Steidl et al. [24] used different network analysis metrics on dependency network to retrieve central classes and demonstrated that the results can compete with the suggestions of experienced developers. Zanetti et al. [15] studied bug reporter centrality to validate that the centrality of bug reporters is greatly helpful for bug triaging procedures.

To the best of our knowledge, bug prediction is a problem that is still an open question; hence we expect that our study can provide a supplement to the existing solutions. Although the prior researches on using network metrics to predict failure proneness or bug severity, our study leverages more relationships between classes in analyzing the implications of dependencies. Additionally, prior researches did not consider both as is done in this paper. We constructed software network at class-level granularity and then used network metrics to analyze bug proneness and severity. Differing from the existing work, we make some contributions as follows.

(1) Five dependency scenes are considered in our class-level software network: inheritance, field, method, return, and parameter dependency, yet part of these relationships were taken into account in most of existing researches.

(2) A comparison between different metrics is made firstly to explore the extent to which they reflect bug quantity and severity, after that to model the predictors with significant metrics to improve software development. Some researchers directly assemble various network metrics without filtration; thus it is inevitable to bring biases so that reducing the accuracy. Our work just bridges this gap by a comparison in the first place.

(3) We select betweenness centrality and out-degree metrics for bug proneness and severity prediction. Just as 10% efforts are paid, an ideal F-value acquired about bug proneness and severity is up to 54.7% and 66.7% and 63.8% and 48.7%, respectively, which is comparable with the result in [4, 25].

3. Preliminary Study

3.1. Software Networks. There are all kinds of networks around us, such as social networks (e.g., friendship networks and scientific collaboration networks), technological networks (e.g., the Internet and WWW), and biological networks (e.g., neural networks and interaction of proteins in yeast). Surprisingly, the underlying structures of these networks have completely different statistical features than those of regular and random networks. They all belong to complex networks. These discoveries have served to draw together many disparate domains into an emerging science of network science.

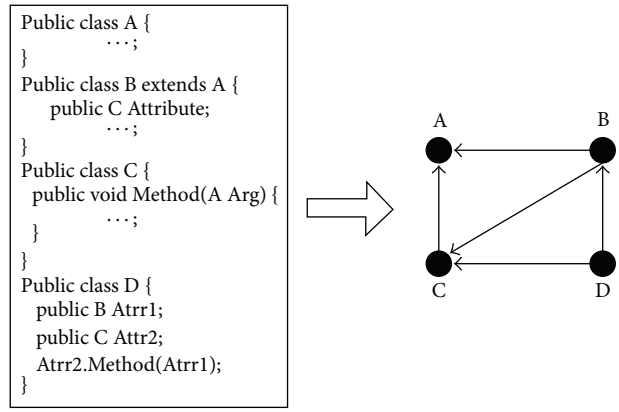


Figure 1: A simple example of software network.

With the rapid development of software technology and the pervasive of Internet, software scale and complexity so sharply increase that developers could not control the system in the global perspective; then software plan and quality cannot be guaranteed. In order to overcome the challenges, some pioneers of complex systems try to introduce graph theory that nodes and edges are used to simplify the structure. To our surprise, software network also shows the basic characteristics of complex networks. These exciting discoveries are paid attentions by researchers from software engineering. Through the interdisciplinary between complex network and software engineering, an approach has been emerging by abstracting a software system as a network, that is, software network.

Software network is an interdisciplinary outcome based on the combination of network science theory and software engineering practice. When constructing a software network, the source code should be handled in reverse engineering methods, (i.e., compiled the codes into a xml file, then derived the topology structure from the xml file). Figure 1 is an example of the fragments of software networks.

3.2. Network Metrics. Network metrics treat software entities as nodes in a graph and characterize them on the basis of their dependencies with other entities. As opposed to code metrics, network metrics take into account the interactions between entities, thus modelling the flow of software information. In this paper, node centrality, PageRank and HITS, in-/out-and total degrees, and modularity ratio metrics are involved, then those significant metrics were used to predict bug proneness and severity.

3.2.1. Eigenvector Centrality (EC). It is a measure of the influence of a node in a network. It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of current node than equal connections to low-scoring nodes and are computed by the adjacency matrix. For a given network (graph) $G = (V, E)$ with $|V|$ number of nodes, let $A = (a_{ij})$

be the adjacency matrix; if vertex i is linked to j, so $a_{ij} = 1$, and $a_{ij} = 0$ otherwise. The centrality score of vertex i can be defined as

$$x_i = \frac{1}{\lambda} \sum_{j \in M(i)} x_j = \frac{1}{\lambda} \sum_{j \in G} a_{ij} x_j, \quad (1)$$

where $M(i)$ is a set of the neighbors of i and λ is a constant. Only the greatest eigenvalue results in the desired centrality measure, the ith component of the eigenvector, then gives the centrality score of the vertex i in the network [26].

3.2.2. Betweenness Centrality (BC). It is based on the total number of shortest paths between all possible pairs of nodes that pass through a node [27]. It quantifies frequencies that a node acts as a bridge along the short path between two other nodes. If $g_i^{(st)}$ are all shortest path node s and t which pass through node i and $n^{(st)}$ is the shortest path between s and t, n is the total of nodes:

$$\mathrm{BC}_i = \frac{\sum_{s<t} g_i^{(st)} / n^{(st)}}{(n-1)(n-2)}. \quad (2)$$

3.2.3. Closeness Centrality (CC). It concerned the farness to all other nodes. Thus, it is defined as the inverse of the sum of all distances to all other nodes [28]. If $d(i, j)$ is the distance between node i and j, so $i's$ centrality is

$$\mathrm{CC}_i = \frac{1}{\sum_{j=1}^{V} d(i, j)}. \quad (3)$$

3.2.4. PageRank (PR). It is a variant of the Eigenvector centrality measure, which is a probability distribution used to represent the likelihood that while a person randomly clicking on links will arrive at a particular page. The rank value indicates the importance of a page, which is used here to denote the importance of a class:

$$\mathrm{PR}_i = \frac{1-q}{N} + q \sum_{j \in V(i)} \frac{\mathrm{PR}(j)}{\mathrm{out}D(j)}, \quad (4)$$

where q is a damping factor ($q = 0.85$), $V(i)$ is the set of nodes that link to i, $\mathrm{out}D(j)$ is the number of outgoing edges on node j, and N is the total number of nodes in a network.

3.2.5. HITS. It is also originally designed to rank web pages, which calculates two scores: authority and hub scores. The authority score is computed as the sum of the scaled hub scores that point to that page. The hub score is the sum of the scaled authority scores of the pages it points to. Here the authority score as a experimental metric is used.

3.2.6. Degree (D). In the case of directed network, it is meaningful to distinguish in-degree (the number of incoming links) from out-degree (the number of outgoing links). For software networks, in-degree is a representation of code reuse, and out-degree is design complexity. Nodes with large in-degree are reused and with large out-degree are more complex in some contexts. In our paper, we use $\mathrm{in}D(i)$, $\mathrm{out}D(i)$, and D representing in-degree, out-degree, and total degree of a node, respectively.

3.2.7. Modularity Ratio (MR). Modularity is a metric used to evaluate the quality of partitioning community in a network by Newman and Girvan [29]. A system consists of many packages, and each package has a large number of classes or subpackages. For a given definition of modules or clusters and their underlying network structure, its respective degree of modularity is defined by

$$\mathrm{Q} = \frac{\sum_i^n e_{ii} - \sum_i^n a_i b_i}{1 - \sum_i^n a_i b_i}, \quad (5)$$

where e_{ij} is the fraction of all edges that link nodes in module i to nodes in module j, $a_i = \sum_j^n e_{ij}$, $b_i = \sum_j^n e_{ji}$ (the sum of column and row resp.), while n is the total number of existing modules. According to this equation, we define a MR of modular i as follows:

$$\mathrm{MR}_i = e_{ii} - a_i b_i. \quad (6)$$

3.3. Network Metrics for Application. Before presenting the research questions and the details of our approach, we give two terminologies used henceforth in this paper, which indicate the practical application of our work.

(i) *Bug proneness*: Bug proneness is treated as a quality measure of software entries in this paper. An intuitive understanding is that it represents the probability that an entry (i.e., class) will become buggy in the process of debugging. The higher the bug proneness of a class is, the more likely it will be given priority to testing. For our purpose, we valuate it via the number of bugs and fixed classes.

(ii) *Bug severity*: This terminology describes the impact of a bug on the entire system. Bug tracking systems classify the severity into several levels, such as critical, major, minor, and trivial. However, severity and priority is not the same concept. In some cases, a bug of critical severity but the priority to resolve it is low as this will happen rarely in real scenario. Here we would like to investigate that the importance of a class in the software network is related to the severity of bug caused by it.

With these metrics and application scenes, the following four research questions are proposed to guide the design of the case study. Questions 1–3 investigate the properties of software networks, whereas question 4 predicts the bug proneness and severity with the significant metrics.

RQ1: Is the position of classes in the software network related to the bug proneness?

RQ2: Is the position of classes in the software network related to the bug severity?

RQ3: If so, which metrics are more significant?

RQ4: How well does the approach to predicting bug proneness/severity with significant metrics work?

4. Approach

In our approach, we mainly consider two Open Source projects—Tomcat and Ant—written in Java as our research subjects in the Apache Software Foundation (ASF) (http://ant.apache.org/) and (http://tomcat.apache.org/). The rationale is threefold. (1) They are well-known and stable projects; each has undergone a number of major release cycles and is still under active development. (2) The source code revision archives and SVN commit-log, dating back several years, and bug reports are available; it is a good chance for rewarding experience in Open Source projects. (3) The choice of Java programming language is limited by the tools developed to construct software networks, and we are interested in understanding open source software written in Java.

We observed the information provided by Bug tracking systems (Bugzilla and JIRA) and SVN commits; found that most bugs have given out the detailed classes and some have already posted the modified fragment of source codes. This information determines the feasibility of our approach. Besides, the dependencies between classes which are directed should not be ignored. Therefore, we decide to abstract the directed but unweighted software networks at the class level.

Let us assume that $G = (V, E)$ is a software network definition, $V = \{v_i\}$ is the set of all classes, and $E = \{e_{ij}\}$ is the set of all dependencies. We distinguish between different kinds of dependencies as follows.

(i) v_i implementing/extending the interface/class v_j is *inheritance dependency*.

(ii) v_i having a field of type v_j is *field dependency*.

(iii) v_i calling a method of v_j is *method dependency*.

(iv) A method of v_i returning an object of type v_j is *return dependency*.

(v) A method of v_i taking an object of v_j as a parameter is *parameter dependency*.

A bug report yet offers many fields to help developers understand this issue; one of which is severity assigned based on how severely a bug affects the program by administrators. Table 1 shows eight levels of bug severity and their ranks in the Bugzilla. An objective for software providers is not only to minimize the total number of bugs but to ensure that bugs' severity as low as possible [23]. The approach mainly consists of four phases: (1) compiling the source code files to extract the class-level directed software networks regarding the dependency requirements; (2) exporting SVN commits and integrating them with bug reports, to obtain our necessary bug-class association relationships; (3) calculating a set of network metrics in the established software networks, then acquire their relationships with the number of bugs and bug severity; (4) finally, applying the significant metrics to the analysis of bug prediction. The framework is shown in Figure 2.

Table 1: Levels of bug severity and their ranks in the Bugzilla bug tracking system.

Bug severity	Rank
Blocker	8
Critical	7
Regression	6
Major	5
Normal	4
Minor	3
Trivial	2
Enhancement	1

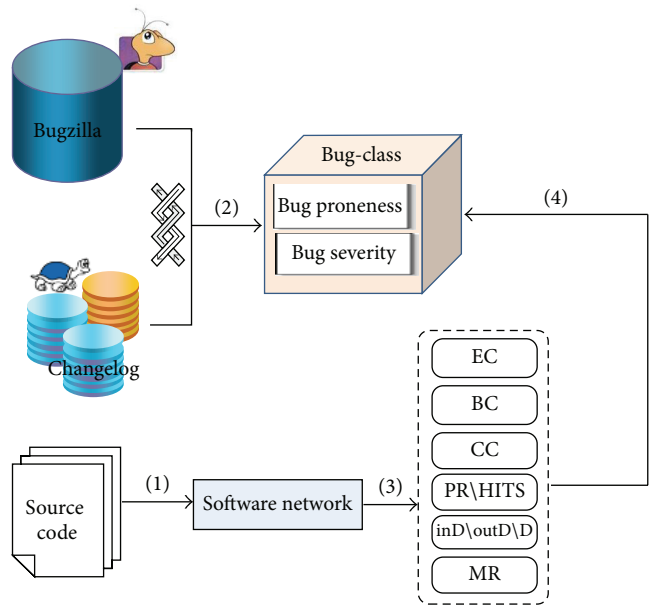


Figure 2: The framework of our approach.

5. Case Study

This section presents the subject projects used for our study and the relevant data we collected. With the experimental results, the proposed questions are answered in turn.

5.1. Data. For our analysis, we collect a number of types of data. We gather the open source software data from source code repositories and bug database and determine the rank of each bug. In this paper, we choose Tomcat 7 and Ant 1.8.4 as our experimental subjects. As a successful open source project, both Tomcat and Ant maintain are a publicly available source. To construct expected software network, source codes are compiled to xml file; then software network is established by parsing the xml file. In the subsequent part, we simplified Tomcat software network as TSN and Ant software network as ASN. There are 35 versions for Tomcat 7. The latest version is continuously renovating so far, thereby only the last 34 versions are adopted in our work. But few differences in the number of nodes/edges between each version. A likely explanation for this is that Tomcat is a

Table 2: The statistics: The number of bugs and fixed classes nodes/edges of software network.

	Tomcat	Ant
#bugs	475	480
#classes	422(21%)	389(29%)
#SN_nodes	2015	1345
#SN_edges	9573	3937
#community	19	20

relatively mature open source software project with decade development history. Of course, not all projects keep this phenomenon. As we known that some projects have a nearly super linear growth [13]. At last, stable Tomcat 7.0.29 version is used to analyze, consists of 2015 nodes and 9573 edges, 19 communities. 1345 nodes and 3937 edges, 20 communities for Ant.

The bug reports are comprised of two sets, one used for the experiment and the other for prediction. That is, the first thirty versions of Tomcat are utilized for progress, (2) and (3), yet the last four versions for prediction. Owing to Ant data refers to only one version; eighty percent of Ant data is used for experiment and the remaining twenty percent for prediction. Note that studies of software defect rely on links between bug databases and commit changes, and this information plays an important role in measuring quality and predicting defects. Some prior researches suggested that there are some missing links between bugs and bug-fix commit logs. They proposed automatic approaches to recover missing links [30, 31] and found that the missing links lead to biased defect information, and it affects prediction performance. However, some authors argued that a bug may not be a bug but a feature or an enhancement [32, 33]. This misclassification also introduces bias in bug prediction models. While considering the missing links, the false position of bug classification increases. Hence only bugs always reported as a bug in the bug tracker are taken into account. Once effective bugs are identified, one checks the change logs by searching for bug ID and calculating the similarity of summary text. The reason for using heuristics is that most of missing links are developed due to misclassification in our datasets. The statistics of the experimental data are shown in Table 2. There are some test classes that are not taken into account because they are duplicate. In bug 53062, for example, both class *org.apache.catalina.connector.TestResponse* and *org.apache.catalina.connector.Response* appear, but they represent the same object in our study. Note that less than thirty percent of the classes are involved; it is confirmed that most of the bugs occur in a small number of classes. So all we should do is to have a guiding detection in a system.

5.2. Results

RQ1: Is the Position of Classes in the Software Network Related to the Bug Proneness? To answer this research question, first of all, the nodes are divided into different groups by metric value in descending order, and each group keeps the same number of nodes. The reason for doing this is to facilitate the exploration for relations between the metrics and bugs. Owing to the last 15 nodes are never fixed in TSN; thus they are excluded to keep each group of 200 nodes. For ASN, in which nodes are divided into eleven groups evenly. The number of fixed classes and involved bugs in each group are recorded. With this information, the answer to $Q1$ is not hard to hold.

The number of fixed classes and involved bugs for different metrics in each group are illustrated in Figures 3 and 4. In Figure 3, EC, in$D$, D, and HITS have a significantly negative correlation (the bigger the average metric value of a class, the more bug-prone); BC and out$D$ show a skew distribution; see Figures 3(a), 3(b), 3(e), 3(f), 3(g), and 3(h). There are some differences in Figure 4; BC, out$D$, and D show a significant correlation but do not an obviously skew distribution; see Figures 4(b), 4(f), and 4(g). The results of other metrics are fluctuated, especially in groups six, eight, and nine.

MR shows the same relationship with fixed classes and bugs. A few differences from the above metrics are that the distribution of bugs is higher than classes' and more fluctuated. In subfigure (i), the x-axis is the rank of modular (community); the larger modularity ratio, the higher its rank is. An overall positive trend is that the larger metric value of a class is, the more it will be bug-prone, and the more it is likely to be modified is shown, especially, BC, out$D$, and D metrics.

RQ2: Is the Position of Classes in the Software Network Related to the Bug Severity? Understanding and characterizing the relationships between network metrics and bug severity in open-source software projects is also a very interesting problem. Although the value of metric is positively related to bug proneness, whether it is related to bug severity, the answer may be not, because there exist a lot of bugs in a class, but all of them are not serious. We conduct the next experiment to validate this research question with the severity rank given in Table 1. The results in Figure 5 show that the relationship deviates what was expected. CC keeps stable in both projects; the overall distribution trend is increased first and then decreased. The top row depicts the severity of Tomcat, and the below row does Ant.

Rank 5 and 7 are two prominent breakpoints marked by the dotted line in each subfigure; they refer to the major and critical bugs, respectively. Bugs with rank 8 have the large metric value in TSN, on the contrary, this kind of bugs have the relatively low value in ASN. Blocker bugs should be avoided as much as possible in software engineering practices. Once this kind of bug occurs and is found the origin is where, it may trigger that a lot of other parts are modified, or the problem is only a small mistake but affects an vital node.

RQ3: If So, Which Metrics Are More Significant? Different metrics measure the importance of a node from different aspects in a network. According to the results obtained from *RQ1*, *RQ2*, we know that not all the metrics have the same expected relationship; some metrics may be more suitable for bug proneness prediction and others for severity prediction. In this section, another work is conducted to analyze how

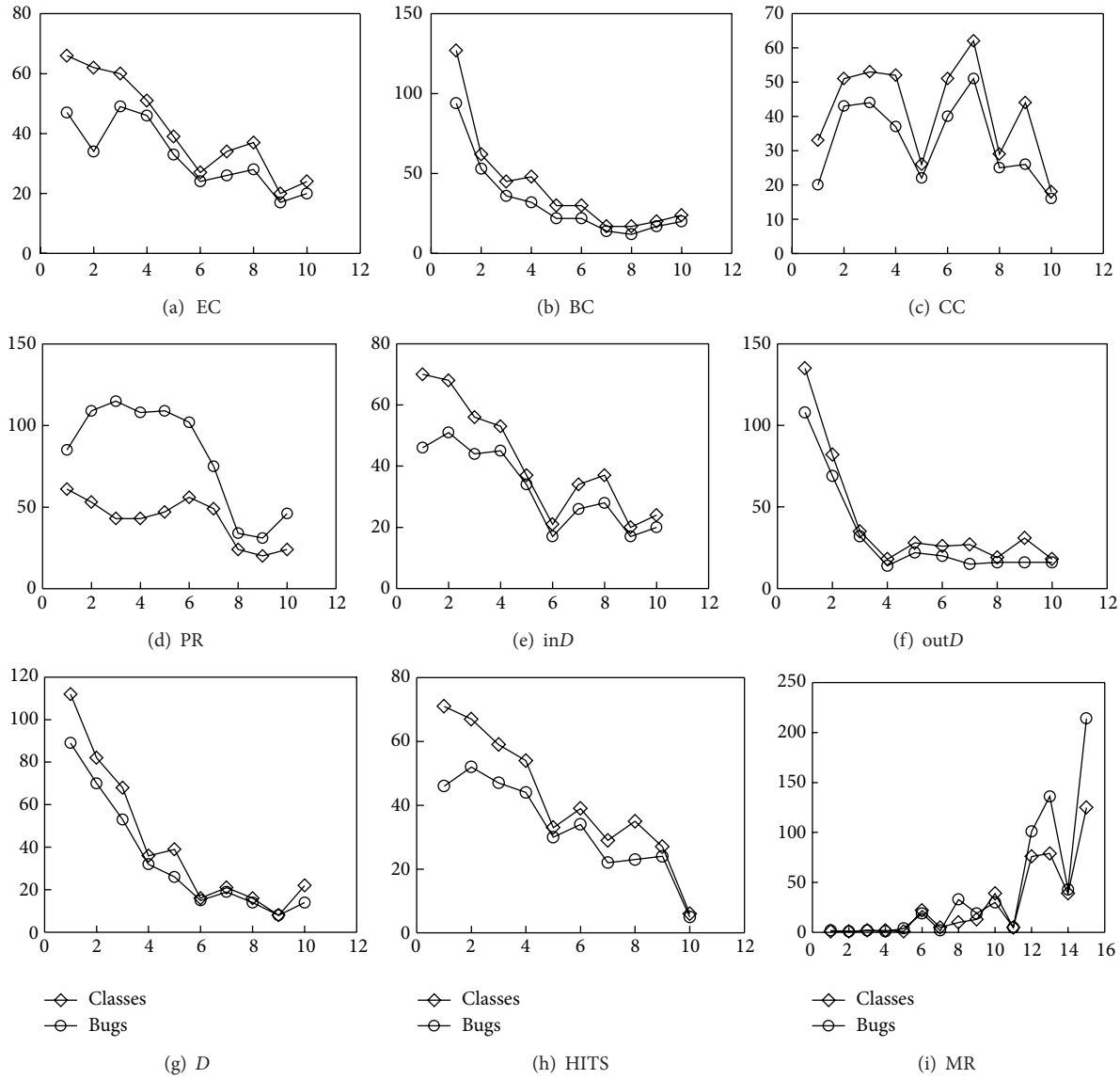


Figure 3: Tomcat (the x-axis is the group ID, and y-axis is the number of classes and bugs).

significant these metrics are and which metrics are better. Three typical correlation analysis methods are used to resolve this problem: Pearson, Kendall, and Spearman (Sig. $P < 0.01$). Pearson is widely used in statistics to measure the degree of the relationship between linear related variables, but both variables should be normally distributed before using it. Kendall and Spearman are nonparametric tests that does not assume any assumptions related to certain distribution. But the former basically gives the impression of being a permutation test of sorts; the latter is essentially based on the principle of least squares.

Table 3 gives the correlation coefficient between metric value and bug proneness on eight metrics. For Tomcat, the coefficient of out$D$ is minimum with 0.552, and D is maximum with 0.967, yet CC and PR metrics are not correlated. Although the correlations with out$D$ in three methods are minimum, the skew distribution curve in Figure 3(f) represented that most of the bugs and classes are fixed in the first two groups. On the other hand, as a result of no skew phenomenon for Ant, out$D$ is more significant than other metrics in three methods. However, CC, EC, in$D$, PR, and HITS are all not significant. The maximum of out$D$ is up to 0.952. The differences mentioned above suggest that treating these metrics should be unequally and with caution. Straightforward to apply all metrics will likely mislead interpretations. Consequently, BC and out$D$ are the suitable metrics to represent the bug proneness and severity. Both of them will be used for the next prediction.

RQ4: How Well Does the Approach to Predicting Bug Proneness/Severity with Significant Metrics Work? Lastly, once we have understood and characterized the relationships between network metrics and bug proneness or severity in two open source projects, what are the effects, if any, on software

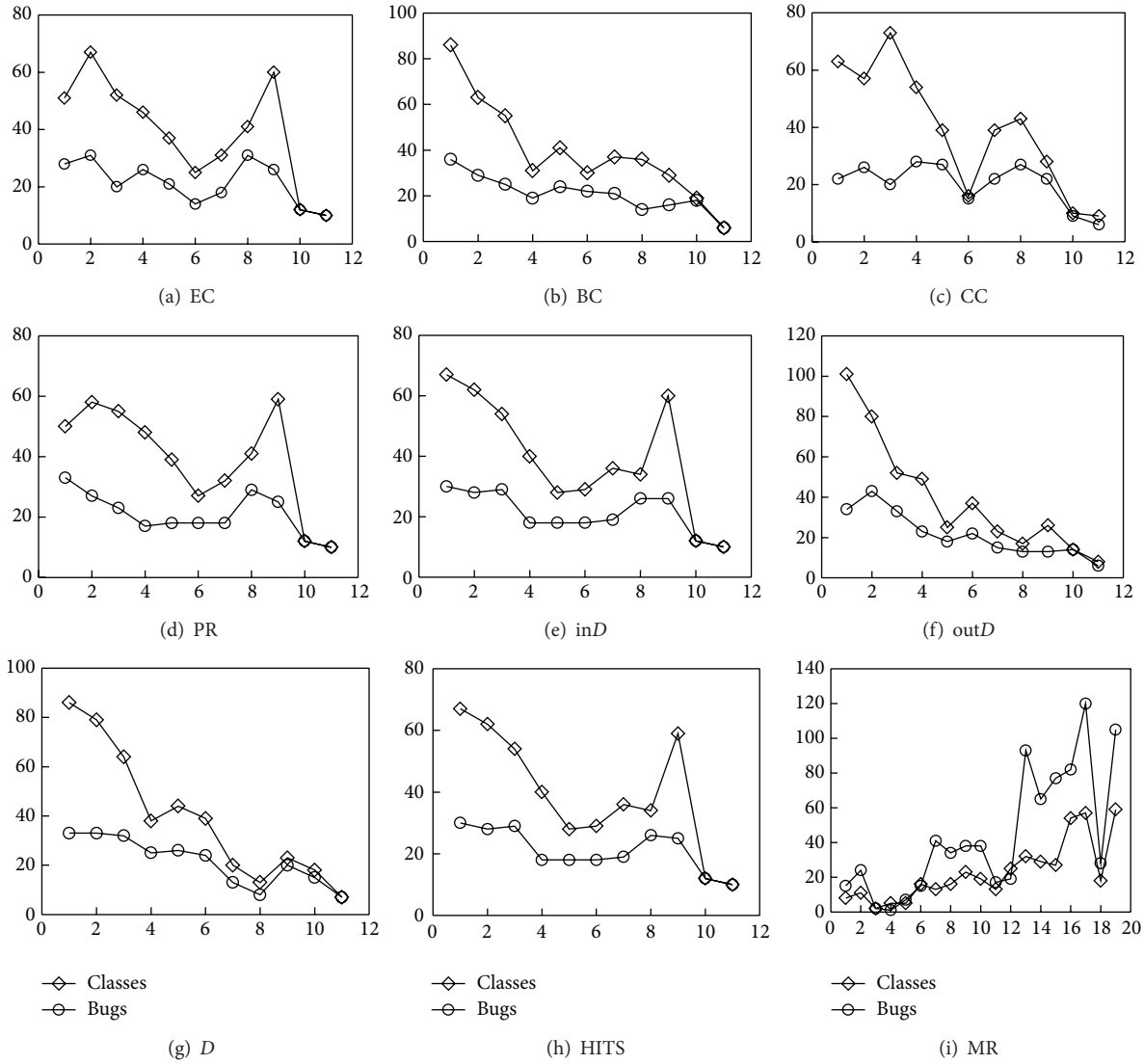


Figure 4: Ant (the x-axis is the group ID, and y-axis is the number of classes and bugs).

Table 3: Bug proneness: the correlation coefficient with eight network metrics.

	Bug proneness (*Sig.* $P < 0.01$)							
	BC	CC	D	EC	in$D$	out$D$	PR	HITS
Pearson-TSN	−0.804	—	−0.903	−0.852	−0.852	−0.752	−0.728	**−0.933**
Kendall-TSN	−0.764	—	**−0.899**	−0.689	−0.689	−0.552	—	−0.733
Spearman-TSN	−0.888	—	**−0.967**	−0.879	−0.879	−0.656	—	−0.903
Pearson-ASN	−0.891	—	−0.895	—	—	**−0.915**	—	—
Kendall-ASN	−0.782	—	−0.807	—	—	**−0.844**	—	—
Spearman-ASN	−0.909	—	−0.916	—	—	**−0.952**	—	—

The bold value refers to the maximum correlation coefficient of different analysis methods per project by comparing the eight metrics.

quality? Or what are the benefits to software engineering practices?

We have learnt that BC and out$D$ metrics are remarkable to represent bug proneness and severity with the above experiments. Whether this conclusion is workable or not, we should validate it through bug proneness and severity prediction with the test data. There are 67 effective bugs in test data, and 87 fixed classes related to these bugs for Tomcat and 135 bugs and 149 fixed classes for Ant. Given the experimental requirements, bug severity is divided into two categories, one is slight category in which bugs' rank is less than 4, but rather the other is severe category. Table 4 shows the resulting

predictive F-value, from Top 50 to 200 for Tomcat and from Top 50 to 125 for Ant, using selected metrics on software networks. The reason for top k within a range of 200 and 125 for these two cases is to ensure the selected nodes from the first group. F-value is calculated by integrating precise and recall as

$$F = \frac{2 * \text{precise} * \text{recall}}{\text{precise} + \text{recall}}. \quad (7)$$

The BCD metric is a combination of BC and out$D$, used to compare with each of them and follows the rule of BCD = BC + out$D$. For Tomcat, BCD metric performs better on fixed classes prediction than the others, and the maximum is 0.3316 when returning the top 100 nodes. While all the nodes in group ID = 1 are selected, three metrics obtain the same accuracy 0.2857. When it comes to the number of bugs, the maximum is 0.547 when using out$D$ metric, and just top 50 classes are needed to check. In this column, out$D$ and BCD metrics are better than BC metric in all cases. The last column is the results of bug severity. The finding shows out$D$ performs far better than the other metrics, and the F-value is nearly more than twice. The best case is 0.6667 at top of 50 nodes.

For Ant, BCD metric performs better than the other two on fixed classes and bugs when returning top 50 classes. Except this case, out$D$ has the highest accuracy at top 100 and 125 and obtains the best result when returning top 100 classes; the maximum of classes, bugs, and severity are 0.3534, 0.6383, and 0.4865, respectively. Throughout the prediction, out$D$ is the most suitable metric for predicting the bug proneness and bug severity in a directed class-level software network.

Finally, few concentrations are devoted to analyze the average human consumptions and needed comments when developer wants to overcome one kind of bug. In Figure 6 one observes that more serious bugs need more people to participate, and these people do more discussions before these bugs are identified as fixed, which is consistent with the common experience except rank = 1. A reasonable explanation for this exception is that the enhancement bug has the lowest difficulty that many developers are able to find the problem and offer their own suggestions or revisions.

6. Discussion

RQ1: From the perspective of node importance, the more important a node is, the greater its probability of being modified is, and more bugs would be involved during the software development process. The central classes (nodes) form the skeleton of the entire system, especially the nodes with large betweenness centrality (BC) and outgoing links (out$D$). The former category of nodes plays an important role in bridging the different modules to ensure the normal execution of the entire system. Such nodes are also named "broker" in network science. The latter nodes are more complex for they aggregate behaviors from many other nodes. They give some implications to software testing. Nodes with high BC mean that they are more important in many contexts and have significant external responsibility. Bugs occurred in these nodes will influence the reachability between modules. On the other hand, nodes with a high out$D$ mean that they aggregate many behaviors of other nodes and have a significant internal complexity. They are more likely to be fixed and contain bugs. In this part, PR and CC metrics are not significant. In summary, our results are consistent with the work in [34]; some metrics are related to bug proneness, and the others are not.

We conclude that the importance of different metrics of a node plays an inconsistent role in the analysis of bug proneness. It is clear that BC and out$D$ are suitable to test prioritization when the developers want to know which classes should be paid more attentions. In some special scenarios, for example, the node importance and out-degree can also be applied to bug location with regard to the relationship between them.

RQ2, RQ3: the hypothesis that a class would be apt to high bug proneness for great metric value, and the bugs would be more serious in this class, is not always true. Fortunately, Table 3 shows the average of BC, D, and out$D$ metrics has a significant correlation with software bug proneness. The most likely explanation for strong correlations among some metrics and bug proneness is that the classes were divided into different groups having the same number of classes in descending order during the experiment. This treatment can also represent that the average metric value of preceding group is larger than the succeeding one and is corresponding to the top-k recommendation in the subsequent section. The results show an overall trend between groups so that stronger than the ones reported generally.

In Figure 5, major, regression and critical bugs have a relatively high value in both projects. Moreover, this phenomenon represented by major bug is more obvious. Additionally, Bugs raised from the broker will hinder the transfer of information between modules and function scheduling of entire system. As long as the modules cannot work smoothly, the quality of system cannot be guaranteed. A class with high outgoing links has more potential bugs, but the severity of the bugs is higher. Such classes are usually the central component in a system or the central nodes in a network.

As the members of a project team, who are more concerned about whether they can find serious bugs sometimes. In other words, they hope to find more bugs quickly and efficiently, while more severe bugs in a limited period of time. Our results provide an appropriate method to alleviate this challenge by BC and out$D$ metrics.

RQ4: by the contrast among eight network metrics, BC and out$D$ are selected to predict bug proneness and severity. Overall, compared to other metrics, a significant advantage of out$D$ is shown in our results. It can be applied to identify which classes should be modified prior to others. The result also indicates that severe bugs usually have strong internal complexity.

Based on our work, we acquire some meaningful answers to the proposed four questions in Section 3. However, there are still some potential limitations and threats to the validity of our work.

(i) All datasets used in our experiments are collected from open source projects Apache Tomcat 7 and Ant. We know that there are many other available

Table 4: F-value: fixed classes, bugs, and bug severity.

	Classes			Bugs			Severity		
	BC	out$D$	BCD	BC	out$D$	BCD	BC	out$D$	BCD
Top 50	0.2043	**0.2920**	0.2628	0.3760	**0.5470**	0.4786	0.2917	**0.6667**	0.4583
Top 100	0.2567	0.3102	**0.3316**	0.4072	0.4431	**0.4551**	0.3288	**0.5069**	0.3836
Top 150	0.2616	**0.2954**	0.2954	0.3594	**0.3779**	0.3687	0.2857	**0.4184**	0.2959
Top 200	0.2857	**0.2857**	0.2857	0.3071	0.3071	**0.3521**	0.2439	**0.3740**	0.2846
Top 50	0.1307	0.2111	**0.2312**	0.1946	0.4221	**0.4322**	0.1481	**0.2613**	0.2613
Top 100	0.2651	**0.3534**	0.3213	0.3489	**0.6383**	0.5957	0.2270	**0.4865**	0.4649
Top 125	0.3139	**0.3358**	0.3285	0.4385	**0.5923**	0.5846	0.3143	**0.4381**	0.4286

The bold value refers to the best F-value, identified by comparing three metrics (BC, out$D$, and BCD) at the top K, of predicting fixed classes, bugs, and bug severity, respectively.

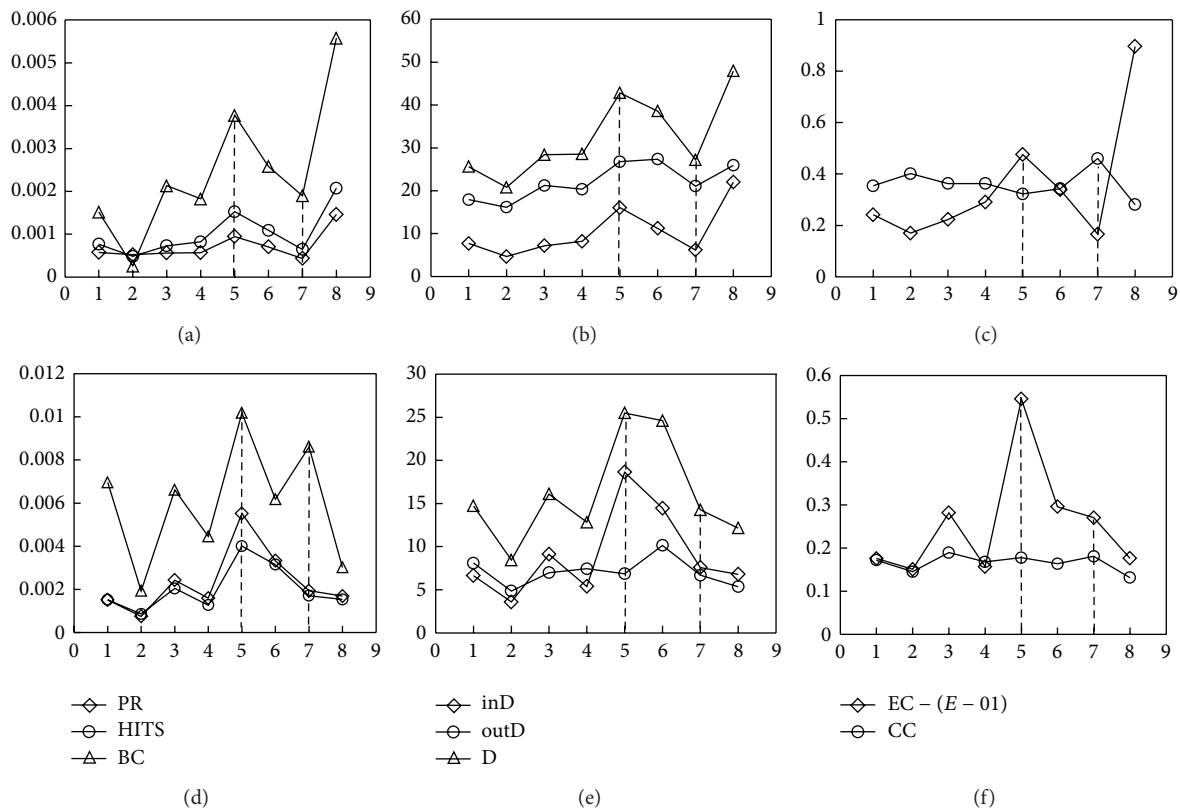


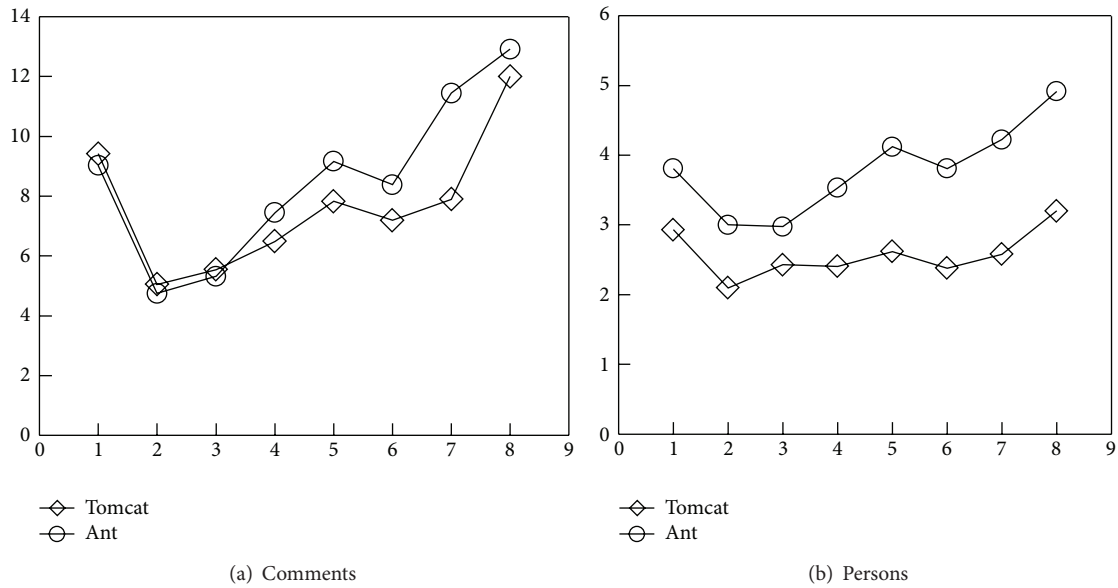
Figure 5: Bug severity (Tomcat-top/Ant-bottom, the x-axis is bug rank, and y-axis is the mean of metric).

software repositories that are helpful to bug analysis. We would like to improve our approach by combining more software resources, such as mailing-list, forum messages, in the future work, to enrich the findings.

(ii) A limitation of our work is that we choose projects written in Java, because of the tool used to construct software network can only deal with java source code. Whether our conclusion can explain projects written in other language or not, it still needs to evaluate.

(iii) Theoretically, software networks will evolve over time; therefore differences are inevitable between multiple versions. In this paper, we utilize a stable version of each project as the construction standard of software network, instead of considering the structural difference between them. Although such treatment is rough, it will not affect the final experimental results.

(iv) As mentioned at the beginning, software system can be characterized from multiple granularity, which is a multi-granularity software network. We investigate the bug proneness and severity through network metrics in class level. If the results will be more



(a) Comments (b) Persons

Figure 6: Human factor (the x-axis is bug rank).

delightful from the other granularity, it is an attractive theme.

7. Conclusion

We constructed a class-level software networks and introduced nine representative and commonly-used network metrics to bug prediction. An empirical study has been conducted on open-source project Apache Tomcat 7 from version 7.0.0 to 7.0.33 and Ant 1.8.4 from Feb 2002 to November 2012. Our analysis demonstrated that there is a statistically significant relation between class's importance and the probability that the class is to be modified and buggy. Also, class's importance is related to the severity of raised bugs. The result also showed that just about ten percent of efforts paid, and the accuracy of our prediction for bug proneness and bug severity can be up to 54.7% and 66.7%, 63.8% and 48.7%, respectively, when successively returning the top 50 and 100 classes within these two cases.

We expect that our findings are insightful and can be used to support the design and development of software, helping engineers in assessing the risk of adding or dropping a feature in the case of supporting existing dependencies between classes. We also believe that our approach can be leveraged in the analysis of bug prediction, test prioritization of others open source softwares. Finally, our findings also provide additional empirical evidence on the importance of dependencies between classes to researchers in the social network analysis domain.

The next work will mainly focus on two aspects. On the one hand, we will collect more open source projects (i.e., Eclipse, Mozilla, or projects deployed on SourceForge) to validate the generality of our approach. On the other hand, we will further take into account the human factors, by the state of software development, are a process of human participation. An exploration about what impact will be brought by sociotechnical congruence on bug prediction is urgent and meaningful.

Acknowledgment

This work is supported by the National Basic Research Program of China no. 2014CB340401, National Natural Science Foundation of China nos. 61273216, 61272111, 61202048, and 61202032, Wuhan Planning Project of Science and Technology no. 201210621214, the Open Foundation of Jiangsu Provincial Key Laboratory of Electronic Business no. JSEB2012-02, and Zhejiang Provincial Natural Science Foundation of China no. LQ12F02011.

References

[1] J. Anvik, "Automating bug report assignment," in *Proceedings of the 28th International Conference on Software Engineering 2006 (ICSE '06)*, pp. 937–940, May 2006.

[2] D. Radjenovica, M. Herickob, R. Torkarc, and A. Živkovičb, "Software fault prediction metrics: a systematic literature review," *Information and Software Technology*, vol. 55, pp. 1397–1418, 2013.

[3] T. Zimmermann and N. Nagappan, "Predicting defects using network analysis on dependency graphs," in *Proceedings of the 30th International Conference on Software Engineering 2008 (ICSE '08)*, pp. 531–540, May 2008.

[4] A. Meneely, L. Williams, W. Snipes, and J. Osborne, "Predicting failures with developer networks and social network analysis," in *Proceedings of the 16th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (SIGSOFT 2008/FSE '08)*, pp. 13–23, November 2008.

[5] K. Crowston, K. Wei, Q. Li, and J. Howison, "Core and periphery in Free/Libre and Open Source software team communications," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06)*, p. 118, January 2006.

[6] T. Zimmerman, N. Nagappan, K. Herzig, R. Premraj, and L. Williams, "An empirical study on the relation between dependency neighborhoods and failures," in *Proceedings of the 4th IEEE International Conference on Software Testing, Verification, and Validation (ICST '11)*, pp. 347–356, March 2011.

[7] S. Valverde, R. F. Cancho, and R. V. Solé, "Scale-free networks from optimal design," *Europhysics Letters*, vol. 60, no. 4, pp. 512–517, 2002.

[8] C. R. Myers, "Software systems as complex networks: structure, function, and evolvability of software collaboration graphs," *Physical Review E*, vol. 68, no. 4, Article ID 046116, 15 pages, 2003.

[9] G. Concas, M. Marchesi, S. Pinna, and N. Serra, "Power-laws in a large object-oriented software system," *IEEE Transactions on Software Engineering*, vol. 33, no. 10, pp. 687–708, 2007.

[10] P. Louridas, D. Spinellis, and V. Vlachos, "Power laws in software," *ACM Transactions on Software Engineering and Methodology*, vol. 18, no. 1, article 2, 2008.

[11] B. Li, Yutao Ma, and J. Liu, "Advances in the studies on complex networks of software systems," *Advances in Mechanics*, vol. 38, no. 6, pp. 805–814, 2008.

[12] B. Li, H. Wang, Z. Y. Li, K. Q. He, and D. H. Yu, "Software complexity metrics based on complex networks," *Acta Electronica Sinica*, vol. 34, pp. 2371–2375, 2006.

[13] W. Pan, B. Li, Y. Ma, and J. Liu, "Multi-granularity evolution analysis of software using complex network theory," *Journal of Systems Science and Complexity*, vol. 24, no. 6, pp. 1068–1082, 2011.

[14] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: generalizing degree and shortest paths," *Social Networks*, vol. 32, no. 3, pp. 245–251, 2010.

[15] M. S. Zanetti, I. Scholtes, C. J. Tessone, and F. Schweitzer, "Categorizing bugs with social networks: a case study on four open source software communities," in *Proceedings of the 35th International Conference on Software Engineering (ICSE '13)*, pp. 1032–1041, 2013.

[16] M. S. Zanetti and F. Schweitzer, "A network perspective on software modularity," in *Proceedings of the International Workshop on Complex Sciences in the Engineering of Computing Systems*, pp. 175–186, 2012.

[17] C. Ding, X. He, P. Husbands, H. Zha, and H. D. Simon, "PageRank, HITS and a unified framework for link analysis," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 353, 354, 2002.

[18] R. Premraj and K. Herzig, "Network versus code metrics to predict defects: a replication study," in *Proceedings of the 5th International Symposium on Empirical Software Engineering and Measurement (ESEM '11)*, pp. 215–224, September 2011.

[19] M. Pinzger, N. Nagappan, and B. Murphy, "Can developer-module networks predict failures?" in *Proceedings of the 16th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (SIGSOFT 2008/FSE '08)*, pp. 2–12, November 2008.

[20] Y. Shin, A. Meneely, L. Williams, and J. A. Osborne, "Evaluating complexity, code churn, and developer activity metrics as indicators of software vulnerabilities," *IEEE Transactions on Software Engineering*, vol. 37, no. 6, pp. 772–787, 2011.

[21] A. Sureka, A. Goyal, and A. Rastogi, "Using social network analysis for mining collaboration data in a defect tracking system for risk and vulnerability analysis," in *Proceedings of the 4th India Software Engineering Conference (ISEC '11)*, pp. 195–204, February 2011.

[22] F. Perin, L. Renggli, and J. Ressia, "Ranking software artifacts," in *Proceedings of the 4th Workshop on FAMIX and Moose in Reengineering (ICSM '10)*, pp. 1–4, 2010.

[23] P. Bhattacharya, M. Iliofotou, I. Neamtiu, and M. Faloutsos, "Graphbased analysis and prediction for software evolution," in *Proceedings of the International Conference on Software Engineering (ICSE '12)*, pp. 419–429, 2012.

[24] D. Steidl, B. Hummel, and E. Juergens, "Using network analysis for recommendation of central software classes," in *Proceedings of the 19th Working Conference on Reverse Engineering*, pp. 93–102, 2012.

[25] T. Wolf, A. Schroter, D. Damian, and T. Nguyen, "Predicting build failures using social network analysis," in *Proceedings of the International Conference on Software Engineering (ICSE '09)*, pp. 1–11, May 2009.

[26] M. E. J. Newman, *Networks*, Oxford University Press, Oxford, UK, 2010.

[27] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, 1994.

[28] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.

[29] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, Article ID 026113, 15 pages, 2004.

[30] A. Bachmann, C. Bird, F. Rahman, P. Devanbu, and A. Bernstein, "The missing links: bugs and bug-fix commits," in *Proceedings of the 18th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE '10)*, pp. 97–106, November 2010.

[31] R. Wu, H. Zhang, S. Kim, and S. C. Cheung, "ReLink: recovering links between bugs and changes," in *Proceedings of the 19th ACM SIGSOFT Symposium on Foundations of Software Engineering (SIGSOFT/FSE '11)*, pp. 15–25, September 2011.

[32] G. Antoniol, K. Ayari, M. Di Penta, F. Khomh, and Y. G. Guéhéneuc, "Is it a bug or an enhancement? A text-based approach to classify change requests," in *Proceedings of the Conference of the Center for Advanced Studies (CASCON '08)*, ACM, October 2008.

[33] K. Herzig, S. Just, and A. Zeller, "It's not a bug, it's a feature: how misclassification impacts bug prediction," in *Proceedings of the 35 International Conference on Software Engineering*, pp. 392–401, 2013.

[34] R. Tonelli, G. Concas, M. Marchesi, and A. Murgia, "An analysis of SNA metrics on the Java Qualitas Corpus," in *Proceedings of the 4th India Software Engineering Conference (ISEC '11)*, pp. 205–213, February 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 264504, 8 pages
http://dx.doi.org/10.1155/2013/264504

Research Article

ACO-Based Routing Algorithm for Cognitive Radio Networks

Peng Huang,[1,2] Feng Lin,[3] and Jiliu Zhou[3]

[1] *College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China*
[2] *School of Automotive and Electronic Engineering, Xichang College, Xichang 615000, China*
[3] *School of Computer Science, Sichuan University, Chengdu 610065, China*

Correspondence should be addressed to Feng Lin; lf1975@gmail.com

Received 11 March 2013; Accepted 9 June 2013

Academic Editor: Tingwen Huang

Copyright © 2013 Peng Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cognitive Radio Networks (CRNs) are an outstanding solution to improve efficiency of spectrum usage. Secondary users in cognitive networks may select from a set of available channels to use provided that the occupancy does not affect the prioritized licensed users. However, CRNs produce unique routing challenges due to the high fluctuation in the available spectrum as well as diverse quality-of-service (QoS) requirements. In CRNs, distributed multihop architecture and time varying spectrum availability are some of the key factors in design of routing algorithms. In this paper, we develop an ant-colony-optimization- (ACO-) based on-demand cognitive routing algorithm (ACO-OCR), jointly consider path and spectrum scheduling, and take advantage of the availability of multiple channels, to improve the delivery latency and packet loss rate. Then, an analytical framework based on M/G/1 queuing theory is introduced to illustrate the relay node queuing model. The performances of ACO-OCR have been evaluated by means of numerical simulations, and the experimental results confirm its effectiveness. Simulation results show that ACO-OCR outperforms other routing approaches in end-to-end path latency and package loss rate.

1. Introduction

According to a report by FCC in 2002, average utilization of many spectrum bands allocated through static assignment polices varies between 15% and 85% [1]. In order to improve the utilization efficiency of existing radio spectrum, so-called Cognitive Radio (CR) has emerged as a technique which allows unlicensed secondary users (SUs) to access the licensed spectrum when no licensed primary users (PUs) appear on the frequency band. Primary users use traditional wireless communication system with static spectrum allocation while secondary users use CRs to complete communication through spectrum opportunities without interfering with Pus activities. With CRs, the efficiency in spectrum utilization will be improved significantly [2].

In realizing Cognitive Radio Networks (CRNs), both occupied spectrum and participating nodes of an abstract wireless link are opportunistically determined by their instantaneous availabilities. To fully explore the potentials of CRNs, it is important to study routing in dynamic spectrum access system while considering the unique properties of cognitive environment. Routing in cognitive radio networks shows characteristics different from traditional network and there are important differences in research methods. Due to characteristics of dynamic spectrum access, CR nodes channel changes with time and space. The quick changing channels and PUs interferences will lead to route instability. Therefore, most existing routing algorithms could not work well in CRNs. In this paper, we focus on the local spectrum knowledge scenario of multihop CRNs. We propose a framework of spectrum aware on-demand routing based on ant colony optimization, which could find the globe optimal path with minimum end-to-end delay and max delivery rate. The main contributions of this paper are listed as follows.

(1) A distributed, on-demand, optimal dynamic multihop routing algorithm for CRNs, named ACO-OCR, is proposed by introducing improved ACO in routing determination.

(2) A preemptive resume priority (PRP) M/G/1 queuing network model to characterize the spectrum usage behaviors between primary and secondary users in CR networks is proposed.

(3) A series of experimentations are conducted to validate the performance of the proposed algorithm. The results have demonstrated that ACO-OCRs have better performance in end-to-end path latency and average package loss rate when compared with JSORP and CAODV.

The rest of the paper is organized as follows. Section 2 describes the related works of distributed routing algorithms of CRNs, and Section 3 presents the design of ACO-OCR. In Section 4 we give the node queuing analysis. In Section 5, we present out simulation results, while Section 6 concludes the paper.

2. Related Works

One of the main differences between ad hoc networks and traditional CRNs is frequency distribution, which is static in traditional ad hoc networks but dynamic in CRNs. The frequency distribution in CRNs varies with the PUs working condition. Routing algorithms for CRNs should utilize the flexibility of CRs and deal with the critical challenges that do not exist in the traditional ad hoc networks. Generally, there are two main kinds of routing scenario in CRNs: the CR nodes with full spectrum knowledge and with local spectrum knowledge. In the former case, all the CR nodes have a spectrum occupancy map of the CRNs, or there is a central control entity, which could indicate over time and space the channel availabilities [3]. On the other hand, CR nodes with local knowledge are that nodes locally construct spectrum availability information.

Under local spectrum knowledge scenario, Cheng et al. [4, 5] propose an approach based on local spectrum knowledge and integrating consideration of the switching delay and back off delay along the path. With full consideration of all possible delays during a multihop transmission through Cognitive Radio Network, the authors develop metrics and mechanism of spectrum assignment. Cheng's approach has a good performance and many algorithms compare with it. Yang et al. [6] analyze and model per node delay and the path delay in multihop Cognitive Radio Network. Then, they propose a framework of local coordination based routing and spectrum assignment, which consists of one protocol for routing path and one scheme for neighborhood region. In brief, the proposals [4–6] are "greedy-like" approaches and have the potential of ending up in a bad locally optimal solution. The Spectrum Aware Mesh Routing (SAMER) proposal [7] opportunistically routes traffic across paths with higher spectrum availability and quality via a new routing metric. It balances between long-term route stability and short-term opportunistic performance. However, this algorithm assumes that nodes are able to communicate with each other with several possible disjoint channels at the same time. In addition, this routing algorithm works based on link state routing, which requires time to converge and create a network topology map. Thus, this approach may not be suitable to CRNs. Shiang and van der Schaar [8] propose a distributed resource-management algorithm that allows network nodes to exchange information and that explicitly considers the delays and cost of exchanging the network information over multihop cognitive radio networks. In this algorithm, spectrum sensing plays an essential role in all SUs. If the detection is highly unreliable, the collisions between the SUs and PUs may happen more frequently. As a result, the overall spectral efficiency cannot be improved. Han et al. [9] present provably good distributed algorithms for simultaneous channel allocation of individual links and packet-scheduling, in Software-Defined Radio (SDR) wireless networks. Unfortunately, this algorithm can only achieve a fraction of the maximum achievable throughput in the worst case.

Cacciapuoti et al. [10] propose CAODV which avoids regions of PU activity during both route formation and packet discovery without requiring any dedicated control channel. However, CAODV broadcast route requests messages to all available channels which increase the protocol overhead and route discovery time. In [11], by introducing preservation regions around primary receivers, a modified multihop routing protocol is proposed for the cognitive users. Under this protocol, there are possibly some SUs that can never be served. Thus, to design more robust and scalable routing algorithm for CRN and take into account the efficiency of optimization, we propose a more dynamic and more practical ACO-based routing algorithm for CRNs.

3. Routing Algorithm (ACO-OCR)

Some assumptions of ACO-OCR are as follows. First, a common channel [12] which enables CR nodes to transmission control packets is used. Meanwhile, CR nodes can adjust the working frequency band to transmit data. Second, each node should maintain lists of locally available channels that are not occupied by primary users. Thus, each node can individually detect its spectrum opportunity (SOP) [13], which is a set of frequency bands currently unoccupied and available. Third, in our mechanism, there are two kinds of ants: forward ants and backward ants. The forward ant executes path search function and the backward ant establishes the pheromone table.

The route establishment is listed as follows.

(1) When the source node has data to transmit, route discovery process starts. The source node sends forward ant which uses ACO mechanism to choose the next node through common control channel. Because CR nodes have multichannel, traditional ant optimization formula should be adapted. When forward ants arrived in node i, the probabilities of selection of next node and channel are calculated in formula (1) or (2) as follows:

$$s = \begin{cases} \arg \max\limits_{j \in \text{allowed}_k} \left\{ \left[\tau_{ijl}^{\alpha}(t) \cdot \eta_{ijl}^{\beta}(t) \right] \right\}, & \text{if } q \le q_0, \\ S, & \text{otherwise}, \end{cases} \tag{1}$$

$$S = \begin{cases} \dfrac{\tau_{ijl}^{\alpha}(t) \cdot \eta_{ijl}^{\beta}(t)}{\sum_{s \in \text{allowed}_k} \tau_{is}^{\alpha}(t) \cdot \eta_{is}^{\beta}(t)}, & j \in \text{allowed}_k, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where q is a random number generated by forward ant obeying uniform distribution in $[0, 1]$. q_0 is a predefined threshold parameter ($0 \le q_0 \le 1$). If $q \le q_0$, the ants use prior knowledge to choose next node and channel. $\tau_{ijl}(t)$ is pheromone concentration parameter of node i to node j through channel l. $\eta_{ijl}(t)$ is the "heuristic visibility" and is defined as follows:

$$\eta_{ijl}(t) = \frac{K_1}{t_{si}} + \frac{K_2}{t_{qj}}, \tag{3}$$

where t_{si} represents the channel switch delay of node i. t_{qj} is queuing delay of node j. K_1, K_2 are the weight coefficients of parameters t_{si} and t_{qj}, respectively, and $K_1 + K_2 = 1$.

(2) When forward ants arrived at destination node, the backward ants are generated. The backward ants return the source node and update the local pheromone concentration. The local pheromone concentration is updated using

$$\tau_{ijl}(t + \Delta t) = (1 - \rho) \cdot \tau_{ijl}(t) + \rho \cdot \Delta\tau_{ijl}(t, t + \Delta t), \tag{4}$$

where ρ is coefficient of pheromone evaporation and $0 < \rho < 1$.

(3) Once all the backward ants back to the source node, a simple local search scheme is applied. The scheme has potential to further reduce the path delay and accelerate convergence of ACO algorithm. Assume the path which a forward ant has established as a vertex set $v = \{v_1, v_2, \ldots, v_i\}$. The procedure of the local search for ant k's path, we call the first path, in the ACO-OCR is described as follows.

Step 1. Choose another ant j's path, we call the second path, which has less path latency.

Step 2. For each path vertex of the first path, try to find same vertex in path vertex of the second path.

Step 3. If two paths have same vertex, two paths are broken into two segments: the beginning part which contains the same vertex and the remaining part of the path. Then, the first path is obtained by combining the beginning part of the first path with the ending part of the second selected tour. At the same time, the second new path can be obtained by inversing the previous operation. All the same vertex should be tested.

Step 4. Compute the path latency, if the latency is shorter or the path has fewer vertex. The new path is accepted.

(4) The global update rule is implemented to the global optimized path. The global update of pheromone concentration is implemented using

$$\tau(r, s) = (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta\tau(r, s),$$
$$\Delta\tau(r, s) = \begin{cases} (L_{\text{gb}})^{-1}, & \text{if } (r, s) \in \text{global optimized path}, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where α presents the evaporation coefficient of pheromone and $0 < \alpha < 1$. L_{gb} is the global optimized path up to now.

(5) If the route is selected, another kind of global update is made; the pheromone concentration in node is cleared.

4. Performance Analysis

In this section, we present the queuing analytical mode of a cognitive relay node. We assume that connections have same priority and follow the first-come-first-served (FCFS) scheduling principle. Each of the nodes has an infinite buffer for storing fixed-length packets.

Assume that there are parallel channels (indexed from 0 to n) available for transmissions by primary and secondary users and that primary users can access these channels at any time according to a certain protocol and a secondary sense the spectrum and tries to access available frequency bands. For ease of analysis, let $\{N(t), t \ge 0\}$ denote the neighbor node forward data rate process, which follows Poisson distribution with parameter λ_f. If there is no route to destination, the data flow packets will be dropped; otherwise the packets will be sent to next node (Figure 1).

We assume that the duration of one data flow t_{dd} follows negative exponential distribution with parameter λ_{dd} and mean μ_{dd}^{-1}, where $\lambda_{dd} = \mu_{dd}^{-1}$, and assume that the time for searching forward t_{cs} follows negative exponential distribution with parameter λ_{cs} and mean μ_{cs}^{-1}, where $\lambda_{cs} = \mu_{cs}^{-1}$. Let $1/\mu_{dd} = E[T_{dd}]$ and $1/\mu_{cs} = E[T_{cs}]$. Y denotes the time of a dataflow being looking for forwarding opportunity. Obviously, $Y = t_{dd} + t_{cs}$. The distribution, distribution density function and expectation of Y are shown in (6), (7), and (8), respectively:

$$F(y) = 1 - \left(\frac{\lambda_{dd}}{\lambda_{dd} - \lambda_{cs}} \cdot e^{-y\lambda_{cs}} - \frac{\lambda_{cs}}{\lambda_{dd} - \lambda_{cs}} \cdot e^{-y\lambda_{dd}} \right), \tag{6}$$

$$f(y) = \lambda_{dd}\lambda_{cs} \left(\frac{e^{-y\lambda_{dd}}}{\lambda_{cs} - \lambda_{dd}} + \frac{e^{-y\lambda_{cs}}}{\lambda_{dd} - \lambda_{cs}} \right), \tag{7}$$

$$E(y) = \frac{\lambda_{dd}}{\lambda_{cs}(\lambda_{cs} - \lambda_{dd})} + \frac{\lambda_{cs}}{\lambda_{dd}(\lambda_{cs} - \lambda_{dd})}, \tag{8}$$

$$E(y^2) = \frac{2\lambda_{cs}}{\lambda_{dd}(\lambda_{cs} - \lambda_{dd})} + \frac{2\lambda_{dd}}{\lambda_{cs}(\lambda_{dd} - \lambda_{cs})}. \tag{9}$$

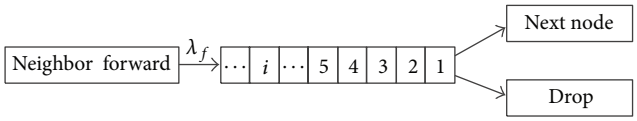


Figure 1: Node buffer queue.

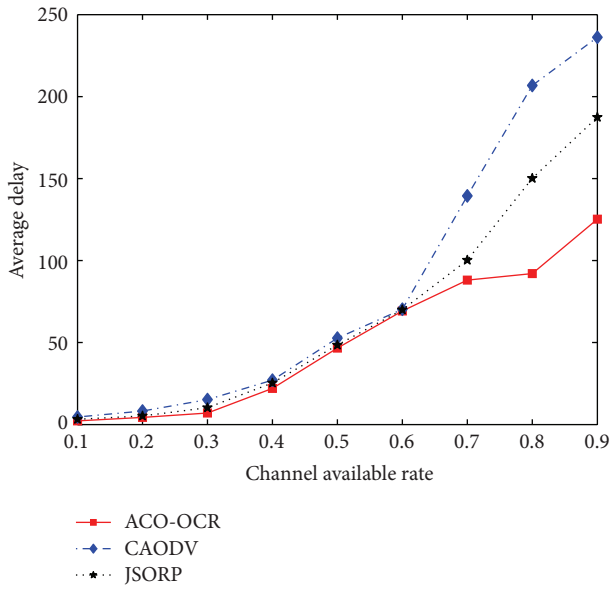


Figure 2: Average delay when channel number = 2.

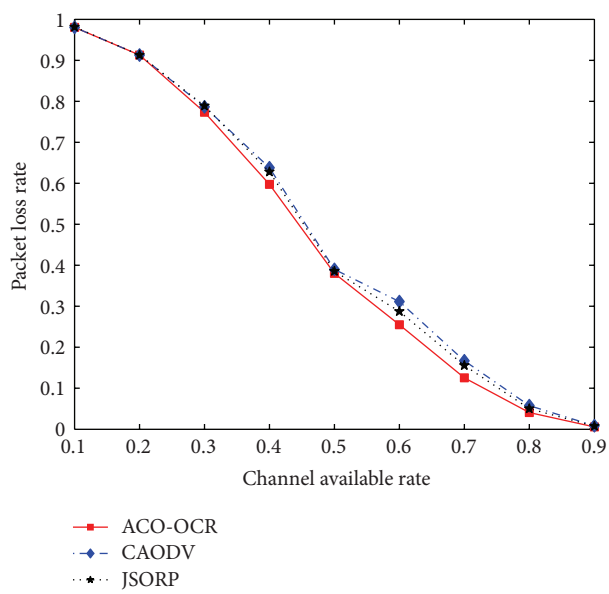


Figure 3: Average packet loss rate when channel number = 2.

So, the model becomes an M/G/1 queuing network model. Let $T'_n\,(n \geq 1)$ denote ending of the nth data flow service. Define $Q_n = Q_{T'_n\,(n\geq 1)}$ as the dataflow remaining in the system after the nth dataflow leaves the node and A_n as the

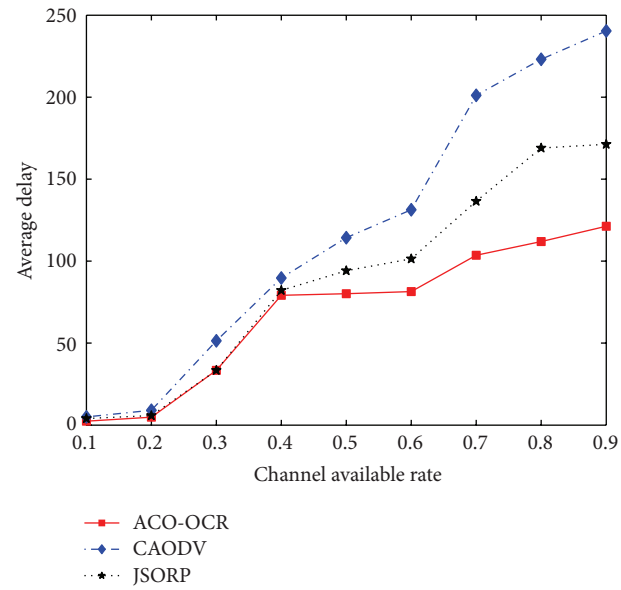


Figure 4: Average delay when channel number = 3.

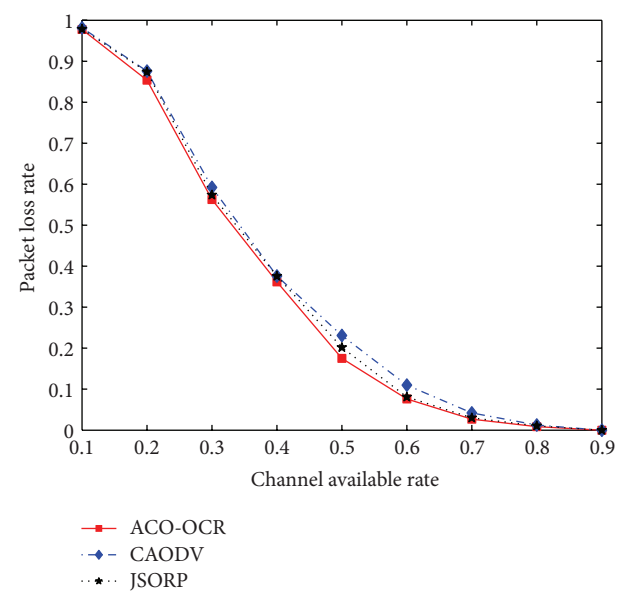


Figure 5: Average packet loss rate when channel number = 3.

arrived dataflow number in nth dataflow forward time; for $n > 0$, we have

$$Q_{n+1} = \begin{cases} Q_n - \varepsilon\left(Q_n\right) + A_{n+1} & \left(Q_n \geq 1\right) \\ A_{n+1} & \left(Q_n = 0\right). \end{cases} \tag{10}$$

Let L denote the expectation of packet number, $L = E(Q_n)$. From (9), when $Q_n \geq 1$, we can get

$$E\left(Q_{n+1}\right) = E\left(Q_n\right) - E\left[\varepsilon\left(Q_n\right)\right] + E\left(A_{n+1}\right). \tag{11}$$

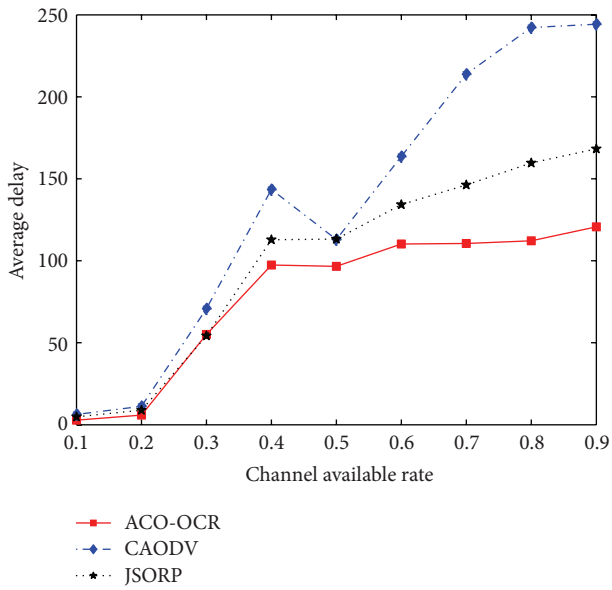


Figure 6: Average delay when channel number = 4.

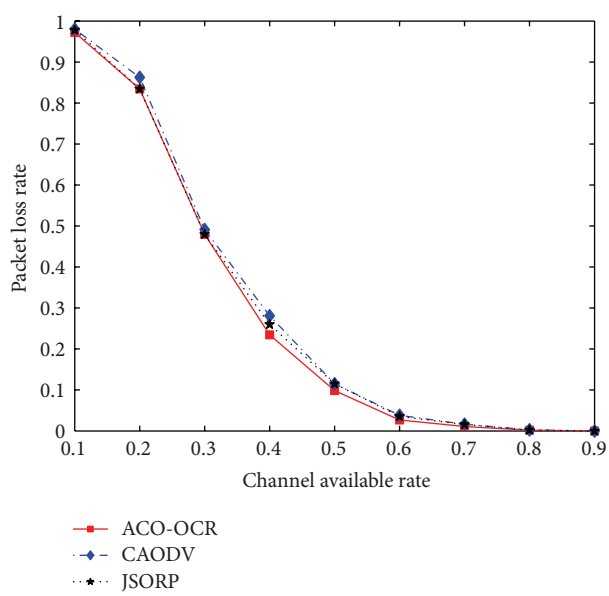


Figure 7: Average packet loss rate when channel number = 4.

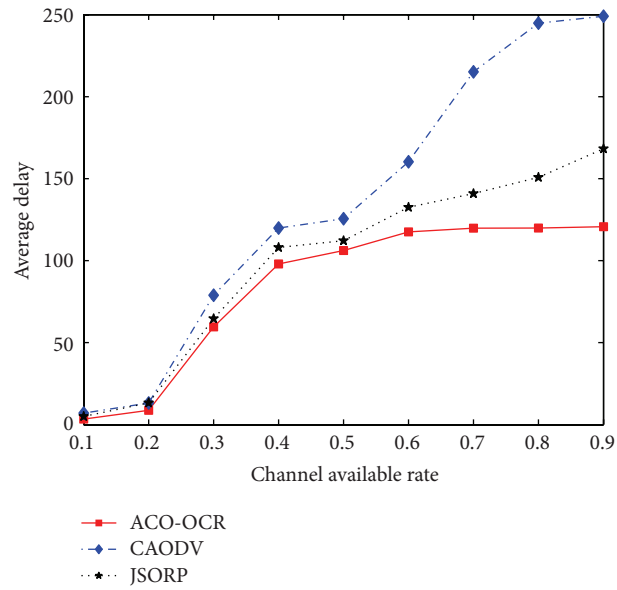


Figure 8: Average delay when channel number = 5.

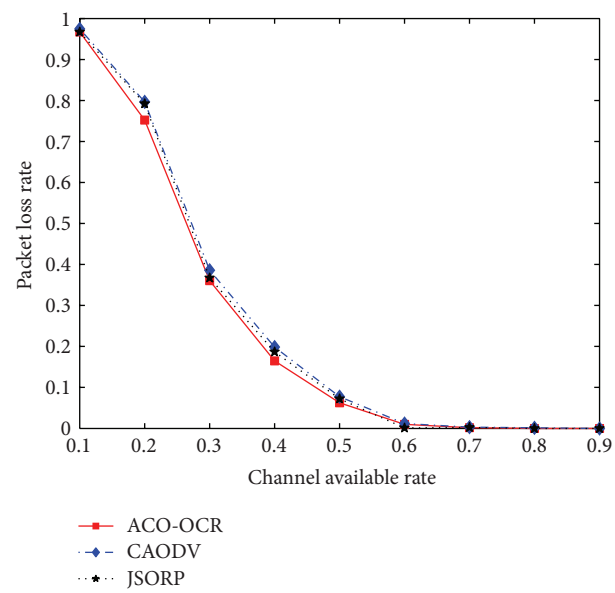


Figure 9: Average packet loss rate when channel number = 5.

Let $E(Q) = \lim_{n\to\infty} E(Q_n)$, and then $\lim_{n\to\infty} E(Q_{n+1}) = \lim_{n\to\infty} E(Q_n)$, and we have

$$E(A_{n+1}) = \int_0^\infty E[N(t)]\,dF(t) = \int_0^\infty \lambda_f t\,dF(t) = \rho. \tag{12}$$

In (11), let $n \to \infty$, and we can get

$$1 - \lim_{n\to\infty} P\{Q_n = 0\} = \rho. \tag{13}$$

Taking mathematical expectation on both sides after square of (13), we can establish

$$\begin{aligned} E(Q^2) = {} & E(Q^2) + E[\varepsilon(Q)] + E(A^2) - 2E(Q) \\ & - 2E(A)E[\varepsilon(Q)] + 2E(Q)E(A). \end{aligned} \tag{14}$$

From (14), we can get

$$E(Q) = \frac{E\left[\left(A^2\right) - 2\rho^2 + \rho\right]}{2 - 2\rho}, \tag{15}$$

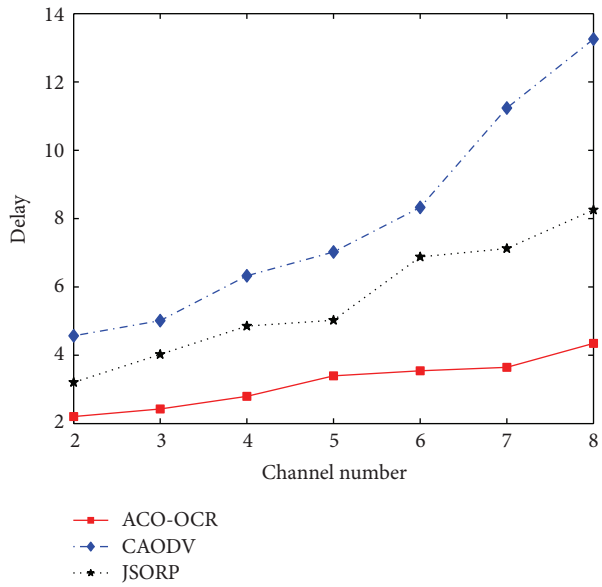


Figure 10: Average delay when channel available rate = 0.1.

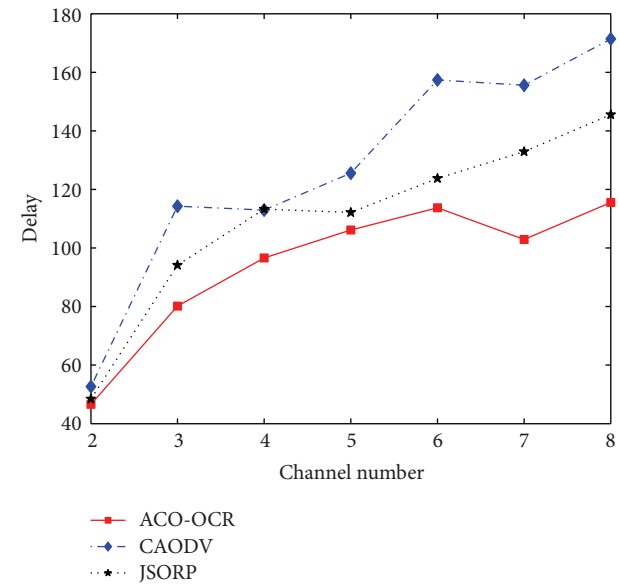


Figure 12: Average delay when channel available rate = 0.5.

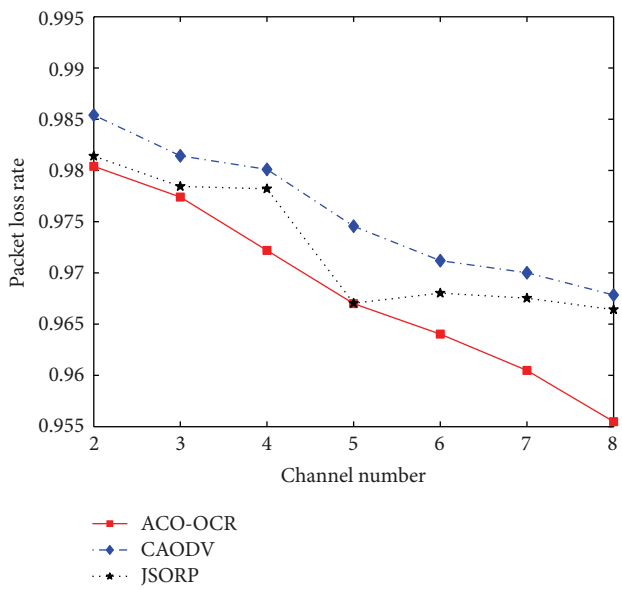


Figure 11: Average packet loss rate when channel available rate = 0.1.

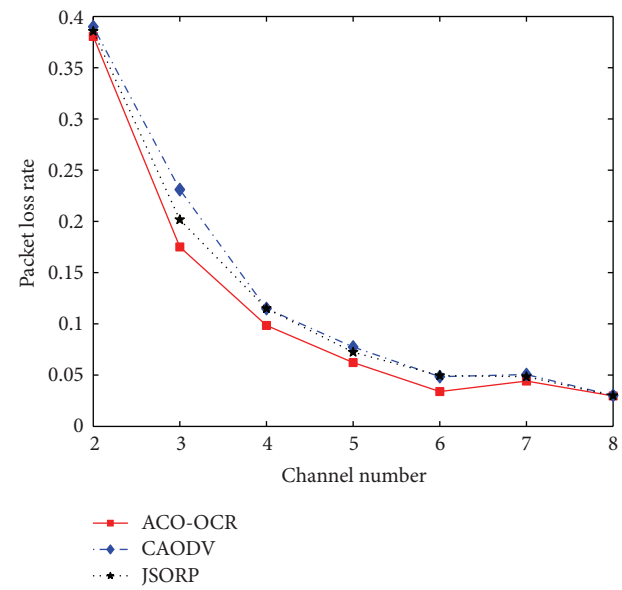


Figure 13: Average packet loss rate when channel available rate = 0.5.

where

$$E\left[\left(A^2\right)\right] = E\left[N^2(F)\right] = \int_0^{\infty} E\left[N^2(F) \mid F = y\right] dF(y) = \lambda^2 E\left(y^2\right) + \lambda E(y). \tag{16}$$

Furthermore, taking (16) to (15), we can get the expectation of average packet number $E(Q)$

$$E(Q) = \rho + \frac{\lambda_f{}^2 E\left[y^2\right]}{2(1-\rho)}. \tag{17}$$

According to Little formula, average lingering time W is

$$W = \frac{L}{\lambda_f} = \frac{1}{E(y)} + \frac{\lambda_f E\left[y^2\right]}{2(1-\rho)}. \tag{18}$$

Similarly, we can get average queue size L_q and average waiting time W_q as follows:

$$L_q = L - \rho = \frac{\lambda_f{}^2 E\left[y^2\right]}{2(1-\rho)}, \tag{19}$$

$$W_q = \frac{L_q}{\lambda_f} = \frac{\lambda_f E\left[y^2\right]}{2(1-\rho)}. \tag{20}$$

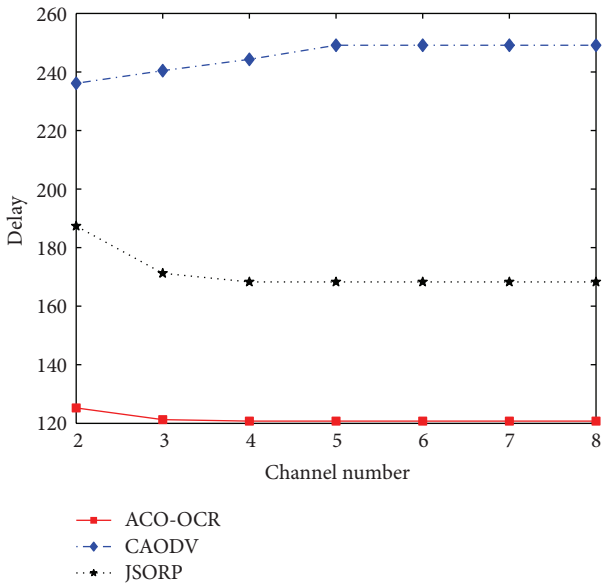


Figure 14: Average delay when channel available rate = 0.9.

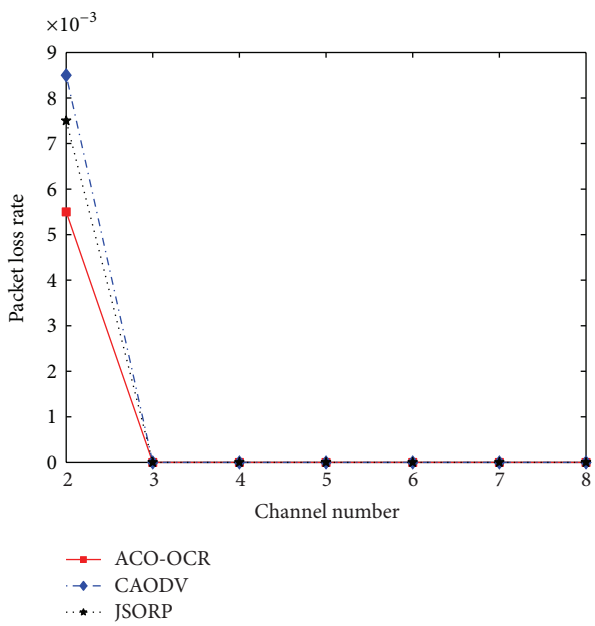


Figure 15: Average packet loss rate when channel available rate = 0.9.

5. Simulations and Results

In this section, we evaluate the effectiveness of ACO-OCR to improve the end-to-end delay and packet loss rate. We build our simulation model with C++ language. In 200×200 square, we randomly deploy 30 CR nodes which have communication radius of 30. The dataflow consists of several packets and each packet has 1024 bytes. The expectation of numbers of packets in dataflow is 100.

We change the numbers of frequency band and channel avail rate and then conduct the simulation. The numbers of each CR node of frequency bands varies from 2 to 8. The available rate of each band varies from 0.1 to 0.9. Then, we compared our algorithm with CAODV [10] and JSORP [4]. From Figures 2, 3, 4, 5, 6, 7, 8, and 9, we can see that with increase of channel available rate, the delay becomes longer and the average packet loss rate becomes lower. When the channel available rate is low, the performances of the three algorithms are almost the same. The reason is that there is not much chance to send the packet. When the channel available rate is bigger than 0.6, the ACO-OCR has better performance than CAODV and JSORP. That is because when the channel available rate is larger, the nodes have more opportunities to send data, so the packet loss rate is decreased and with more packet to send, the delay is increased. In Figures 10, 11, 12, and 13, we set packet loss rate to 0.1 and increase the channel number from 2 to 8; the same result is derived. In Figures 14 and 15, when the channel available rate is high, the average delay and packet loss rate have slight changes. That is because under the high channel available rate, the algorithm already finds the best route, and the increasing of channel number could have little effect on routing. Due to consideration of queuing delay of node buffer, the performances of JSORP and ACO-OCR are better than CAODV. Because routing metrics of ACO-OCR employ weighted approach, when the path to the destination is long, it is more likely to choose a better relay node. Thus, the global delay is lower.

6. Conclusion

Routing is one of the important problems in cognitive radio. In this paper, we propose an ACO-based on-demand routing algorithm to tackle this problem. With the framework of ACO, we implement several operations so as to make ACO capable of routing and channel assignment in multihop cognitive radio networks. Our approach jointly considers the path and channel decisions and thus the end-to-end path latency is minimized. Then, we derive the queuing delay and average queuing length of a cognitive relay node. We validate the effectiveness of the algorithm by thorough simulation and find that our approach provides better performance than JSOR and CAODV in both spectrum distribution varying and channel availability varying environment. This suggests that ACO may be more suitable to solve similar problems than other computing techniques. In the future, we will develop tailor-made operators for ACO to get even better performance and also compare our algorithm with a broader class of algorithms.

Acknowledgments

This work is supported by National Natural Science Foundation of China (no. 61272448), Doctoral Fund of Ministry of Education of China (no. 20110181130007), Science and Technology Supporting Plan of Sichuan (nos. 2011RZ0004 and 2012GZ0005), and Xichang College Natural Science Foundation XA1201.

References

[1] FCC, "Spectrum policy task force report," ET Docket No. 02-155, 2002.

[2] J. Mitola, "Cognitive radio architecture evolution," *Proceedings of the IEEE*, vol. 97, no. 4, pp. 626–641, 2009.

[3] FCC, FCC 08-260, Unlicensed Operation in the TV Broadcast Bands, 2008.

[4] G. Cheng, W. Liu, Y. Li, and W. Cheng, "Spectrum aware on-demand routing in cognitive radio networks," in *Proceedings of the 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '07)*, pp. 571–574, Dublin, UK, April 2007.

[5] G. Cheng, W. Liu, Y. Li, and W. Cheng, "Joint on-demand routing and spectrum assignment in cognitive radio networks," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 6499–6503, Glasgow, UK, June 2007.

[6] Z. Yang, G. Cheng, W. Liu, W. Yuan, and W. Cheng, "Local coordination based routing and spectrum assignment in multi-hop cognitive radio networks," *Mobile Networks and Applications*, vol. 13, no. 1-2, pp. 67–81, 2008.

[7] I. Pefkianakis, S. H. Y. Wong, and S. Lu, "SAMER: spectrum aware mesh routing in cognitive radio networks," in *Proceedings of the 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '08)*, pp. 1–5, Chicago, Ill, USA, October 2008.

[8] H. P. Shiang and M. van der Schaar, "Distributed resource management in multihop cognitive radio networks for delay-sensitive transmission," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 941–953, 2009.

[9] B. Han, V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan, "Distributed strategies for channel allocation and scheduling in software-defined radio networks," in *Proceedings of the 28th Conference on Computer Communications (IEEE INFOCOM '09)*, pp. 1521–1529, Rio de Janeiro, Brazil, April 2009.

[10] A. S. Cacciapuoti, C. Calcagno, M. Caleffi, and L. Paura, "CAODV: routing in mobile ad-hoc cognitive radio networks," in *Proceedings of the IFIP Wireless Days (WD '10)*, pp. 1–5, Venice, Italy, October 2010.

[11] S. W. Jeon, N. Devroye, M. Vu, S. Chung, and V. Tarokh, "Cognitive networks achieve throughput scaling of a homogeneous network," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5103–5115, 2011.

[12] S. Krishnamurthy, M. Thoppian, S. Venkatesan, and R. Prakash, "Control channel based MAC-layer configuration, routing and situation awareness for cognitive radio networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM '05)*, vol. 1, pp. 455–460, Atlantic City, NJ, USA, October 2005.

[13] C. Xin, B. Xie, and C. C. Shen, "A novel layered graph model for topology formation and routing in dynamic spectrum access networks," in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 308–317, Baltimore, Md, USA, November 2005.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 842614, 5 pages
http://dx.doi.org/10.1155/2013/842614

Research Article

Modeling Computer Virus and Its Dynamics

Mei Peng,[1] Xing He,[2] Junjian Huang,[3] and Tao Dong[4]

[1] *College of Mathematical and Computer Science, Yangtze Normal University, Chongqing 400084, China*
[2] *College of Computer Science, Chongqing University, Chongqing 400030, China*
[3] *School of Computer Science, Chongqing University of Education, Chongqing 400067, China*
[4] *College of Software and Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*

Correspondence should be addressed to Mei Peng; pmgs@qq.com

Received 27 March 2013; Accepted 9 June 2013

Academic Editor: Tingwen Huang

Copyright © 2013 Mei Peng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Based on that the computer will be infected by infected computer and exposed computer, and some of the computers which are in suscepitible status and exposed status can get immunity by antivirus ability, a novel coumputer virus model is established. The dynamic behaviors of this model are investigated. First, the basic reproduction number R_0, which is a threshold of the computer virus spreading in internet, is determined. Second, this model has a virus-free equilibrium P^0, which means that the infected part of the computer disappears, and the virus dies out, and P^0 is a globally asymptotically stable equilibrium if $R_0 < 1$. Third, if $R_0 > 1$ then this model has only one viral equilibrium P^*, which means that the computer persists at a constant endemic level, and P^* is also globally asymptotically stable. Finally, some numerical examples are given to demonstrate the analytical results.

1. Introduction

Computer virus is a malicious mobile code which including virus, Trojan horses, worm, and logic bomb. It is a program that can copy itself and attack other computers. And they are residing by erasing data, damaging files, or modifying the normal operation. Due to the high similarity between computer virus and biological virus [1], various computer virus propagation models are proposed [2–4]. This dynamical modeling of the spread process of computer virus is an effective approach to the understanding of the behavior of computer viruses because on this basis, some effective measures can be posed to prevent infection.

The computer virus has a latent period, during which individuals are exposed to a computer virus but are not yet infectious. An infected computer which is in latency, called exposed computer, will not infect other computers immediately; however, it still can be infected. Based on these characteristics, delay is used in some models of computer virus to describe that although the exposed computer does not infect other computers, it still has infectivity [5, 6]. Yang et al. [7, 8] proposed an SLB and SLBS models; in these models, the authors considered that the computer virus has latency, and the computer also has infectivity in the period of latency. However, they do not show the length of latency and take into account the impact of artificial immunization ways such as installing antivirus software. And the newly entered in the internet from the susceptible status to exposed status, the contact rate is the same as that of susceptible status entering into infected status. In this paper, a novel model of computer virus, known as SEIR model, is put forward to describe the susceptible computer which can be infected by the other infected or exposed computer and come into the exposed status. In the SEIR model, based on artificial immunity, we consider the bilinear incidence rate for the latent and infection status. Assume that the computers which newly entered the internet are susceptible, the computers correspond with exposed computers, and their adequate contact rate is denoted by β_1, and computers also correspond with infected computers, and their adequate contact rate is denoted by β_2. So, the fraction of the computer which newly entered the internet will enter the class R by anti-virus software; the fraction of computers contact with exposed and infected computer will stay latent before becoming infectious and enter the class E. It is shown that the dynamic behavior of the proposed model is determined by a threshold R_0, and this

model has a virus-free equilibrium P^0, and P^0 is a globally asymptotically stable equilibrium if $R_0 < 1$; if $R_0 > 1$ this model has only one viral equilibrium P^*, and it is globally asymptotically stable.

This paper is organized as follows. Section 2 formulates a novel computer virus mode. Section 3 proves the global stability of the virus-free equilibrium. Section 4 discusses the stability of the viral equilibrium. In Section 5, numerical simulations are given to present the effectiveness of the theoretic results. Finally, Section 6 summarizes this work.

2. Model Formulation

At any time, a computer is classified as internal and external depending on weather it is connected to internet or not. At that time, all of the internet computers are further categorized into four classes: (1) susceptible computers, that is, uninfected computers and new computers which connected to network; (2) exposed computers, that is, infected but not yet broken-out; (3) infectious computers; (4) recovered computers, that is, virus-free computer having immunity. Let $S(t)$, $E(t)$, $I(t)$, $R(t)$ denote their corresponding numbers at time t, without ambiguity; $S(t)$, $E(t)$, $I(t)$, $R(t)$ will be abbreviated as S, E, I, R, respectively. The model is formulated as the following system of differential equations:

$$\begin{aligned} S' &= (1-p)N - \beta_1 SI - \beta_2 SE - pS - \mu S, \\ E' &= \beta_1 SI + \beta_2 SE - kE - \alpha E - \mu E, \\ I' &= \alpha E - rI - \mu I, \\ R' &= pS + kE + rI, \end{aligned} \tag{1}$$

$$N(t) = S(t) + E(t) + I(t) + R(t). \tag{2}$$

We may see that the first three equations in (1) are independent of the fourth equation, and therefore, the fourth equation can be omitted without loss of generality. Hence, system (1) can be rewritten as

$$\begin{aligned} S' &= A - \beta_1 SI - \beta_2 SE - aS, \\ E' &= \beta_1 SI + \beta_2 SE - bE, \\ I' &= \alpha E - cI. \end{aligned} \tag{3}$$

Therefore,

$$\begin{aligned} a &= p + \mu, \qquad b = k + \alpha + \mu, \\ c &= r + \mu, \qquad (1-p)N = A, \end{aligned} \tag{4}$$

where N denotes the rate at which external computers are connected to the network; p denotes the recovery rate of susceptible computer due to the anti-virus ability of network; k denotes the recovery rate of exposed computer due to the anti-virus ability of network; β_1 denotes the rate at which, when having a connection to one infected computer, one susceptible computer can become exposed but has not broken-out; β_2 denotes the rate of which, when having connection to one exposed computer, one susceptible computer can become exposed; α denotes the rate of the exposed computers cannot be cured by anti-virus software and broken-out; r denotes the recovery rate of infected computers that are cured; μ denotes the rate at which one computer is removed from the network. All the parameters are nonnegative.

Moreover, all feasible solutions of the system (3) are bounded and enter the region D, where

$$D = \left\{(S, E, I) \in R_+^3 \mid S \geq 0,\ E \geq 0,\ I \geq 0,\ S + E + I \leq \frac{A}{a}\right\}. \tag{5}$$

Referring to [9], we define the basic reproduction number of the infection as

$$R_0 = \frac{A(\beta_1 \alpha + \beta_2 c)}{abc}. \tag{6}$$

For system (3), there always exists the virus-free equilibrium which is $P^0(A/a, 0, 0)$; if $R_0 > 1$, then there also exists a viral equilibrium $P^*(S^*, E^*, I^*)$.

Therefore,

$$\begin{aligned} S^* &= \frac{A}{aR_0}, \\ E^* &= \frac{A(R_0 - 1)}{bR_0}, \\ I^* &= \frac{A\alpha(R_0 - 1)}{bcR_0}. \end{aligned} \tag{7}$$

3. The Virus-Free Equilibrium and Its Stability

Theorem 1. *P^0 is locally asymptotically stable if $R_0 < 1$. Whereas P^0 is unstable if $R_0 > 1$.*

Proof. The characteristic equation of (3) at P^0 is given by

$$\det\begin{pmatrix} \lambda + a & -\beta_2 S & -\beta_1 S \\ 0 & \lambda - (\beta_2 S - b) & \beta_1 S \\ 0 & \alpha & \lambda + c \end{pmatrix} = 0, \tag{8}$$

which equals to

$$(\lambda + a)\left[\lambda^2 - \left(\beta_2 S^0 - b - c\right)\lambda - bc(R_0 - 1)\right] = 0. \tag{9}$$

Then, (9) has negative real part characteristic roots:

$$\begin{aligned} \lambda_1 &= -a, \\ \lambda_{2,3} &= \frac{(\beta_2 S - b - c) \pm \sqrt{(\beta_2 S - b - c)^2 + 4abc(R_0 - 1)}}{2}, \end{aligned} \tag{10}$$

where

$$\beta_2 S - b - c < 0. \tag{11}$$

When $R_0 < 1$, there are no positive real roots of (9) and thus P^0 is a local asymptotically stable equilibrium. While $R_0 > 1$, (9) has positive real roots, which means P^0 is unstable.

The proof is completed. □

Theorem 2. *P^0 is globally asymptotically stable with respect to D if $R_0 < 1$.*

Proof. Let $L = ((\beta_1 c + \beta_2 \alpha)/bc)E + \beta_2 I/c$.

Obviously

$$L > 0, \tag{12}$$

thus

$$\begin{aligned} L' &= \frac{(\beta_1 c + \beta_2 \alpha)}{bc} E' + \frac{\beta_2}{c} I' \\ &= \frac{(\beta_1 c + \beta_2 \alpha)}{bc} (\beta_1 SI + \beta_2 SE - bE) + \frac{\beta_2}{c} (\alpha E - cI) \\ &= \frac{(\beta_1 c + \beta_2 \alpha)}{bc} (\beta_1 I + \beta_2 E) S - \frac{(\beta_1 c + \beta_2 \alpha)}{bc} bE \\ &\quad + \frac{\beta_2 \alpha E}{c} - \beta_2 I \\ &= \frac{(\beta_1 c + \beta_2 \alpha)}{bc} (\beta_1 I + \beta_2 E) S - \beta_1 E - \beta_2 I \\ &= (\beta_1 E + \beta_2 I) \left[\frac{(\beta_1 c + \beta_2 \alpha)}{bc} S - 1 \right] \\ &= (\beta_1 E + \beta_2 I) \left[\frac{1}{bc} \cdot \frac{A}{a} \cdot \frac{R_0 abc}{A} - 1 \right] \\ &= (\beta_1 E + \beta_2 I)(R_0 - 1) < 0. \end{aligned} \tag{13}$$

The proof is completed. □

4. The Viral Equilibrium and Its Stability

Theorem 3. *P^* is locally asymptotically stable if $R_0 > 1$.*

Proof. The Jacobin matrix of system (3) about P^* is given by

$$J^* = \begin{pmatrix} -aR_0 & -\beta_2 S^* & -\beta_1 S^* \\ a(R_0 - 1) & \beta_2 S^* - b & \beta_1 S^* \\ 0 & \alpha & -c \end{pmatrix}, \tag{14}$$

which equals to

$$f(\lambda) = a_0 \lambda^3 + a_1 \lambda^2 + a_2 \lambda + a_3 = 0, \tag{15}$$

where

$$\begin{aligned} a_0 &= 1, \\ a_1 &= aR_0 - (\beta_2 S^* - b - c) > 0, \\ a_2 &= abR_0 + acR_0 - a\beta_2 S^* \\ &> abR_0 + acR_0 - a(b + c) \\ &= a(b + c)(R_0 - 1) > 0, \end{aligned} \tag{16}$$

where

$$\begin{aligned} \beta_2 S^* &< b + c, \\ a_3 &= abc(R_0 - 1) > 0. \end{aligned} \tag{17}$$

Thus,

$$\begin{aligned} \Delta_1 &= a_1 > 0, \\ \Delta_2 &= \begin{vmatrix} a_1 & 1 \\ a_3 & a_2 \end{vmatrix} = a_1 a_2 - a_3 > 0, \\ \Delta_3 &= \begin{vmatrix} a_1 & 1 & 0 \\ a_3 & a_2 & a_1 \\ 0 & 0 & a_3 \end{vmatrix} = a_3 (a_1 a_2 - a_3) > 0. \end{aligned} \tag{18}$$

According to the Hurwitz criterion, all roots of (15) have negative real pats. Thus, the claimed result follows. The proof is completed. □

The following result can be proved in the same way (see [9]).

Theorem 4. *P^* is uniquely globally asymptotically stable if $R_0 > 1$.*

Proof. The Jacobin matrix of system (3) about P^* is given by

$$J^* = \begin{pmatrix} -aR_0 & -\beta_2 S^* & -\beta_1 S^* \\ a(R_0 - 1) & \beta_2 S^* - b & \beta_1 S^* \\ 0 & \alpha & -c \end{pmatrix}. \tag{19}$$

The second compound matrix $J^{[2]}$ of the Jacobin matrix can be calculated as follows (see [10, 11]):

$$J^{[2]} = \begin{pmatrix} -aR_0 + \beta_2 S^* - b & \beta_1 S^* & \beta_1 S^* \\ \alpha & -aR_0 - c & -\beta_2 S^* \\ 0 & a(R_0 - 1) & \beta_2 S - b - c \end{pmatrix}. \tag{20}$$

Set P as the following diagonal matrix:

$$P(x) = \left(1, \frac{E}{I}, \frac{E}{I}\right). \tag{21}$$

Denote that

$$P_f P^{-1} = \operatorname{diag}\left(0, \frac{E'}{E} - \frac{I'}{I}, \frac{E'}{E} - \frac{I'}{I}\right). \tag{22}$$

Therefore, the matrix $B = P_f P^{-1} + PJ^{[2]}P^{-1}$ can be written in the following block form:

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \tag{23}$$

with

$$B_{11} = -aR_0 + \beta_2 S^* - b,$$
$$B_{12} = \frac{I}{E}\beta_1 S^* (1,1),$$
$$B_{21} = \frac{E}{I}(\alpha, 0)^T,$$
$$B_{22} = \begin{pmatrix} \frac{E'}{E} - \frac{I'}{I} - (aR_0 + c) & -\beta_2 S^* \\ a(R_0 - 1) & \frac{E'}{E} - \frac{I'}{I} + (\beta_2 S * -b - c) \end{pmatrix}, \tag{24}$$

thus

$$\mu_1(B_{11}) = \beta_2 S^* - aR_0 - b,$$
$$\begin{aligned} \mu_1(B_{22}) = \max\Big\{ & \frac{E'}{E} - \frac{I'}{I} - aR_0 - c + aR_0 - a, \\ & \frac{E'}{E} - \frac{I'}{I} + \beta_2 S * -b - c - \beta_2 S^* \Big\} \\ = {} & \frac{E'}{E} - \frac{I'}{I} - c - a. \end{aligned} \tag{25}$$

The vector norm $\|\cdot\|$ in $R^3 \cong R^{\binom{3}{2}}$ is choosen as

$$\|(u, v, w)\| = \max\{|u|, |v + w|\}. \tag{26}$$

The Lozinskii measure $\mu(B)$ with respect to $\|\cdot\|$ is as follows (see [12]):

$$\mu(B) \le \sup\{g_1, g_2\}, \tag{27}$$

where

$$\begin{aligned} g_1 &= \mu_1(B_{11}) + |B_{12}| = \beta_2 S^* - aR_0 - b + \frac{I}{E}\beta_1 S^*, \\ g_2 &= \mu_1(B_{22}) + |B_{21}| = \frac{E'}{E} - \frac{I'}{I} - a - c + \frac{E}{I}\alpha. \end{aligned} \tag{28}$$

From (3), we find that

$$\begin{aligned} \frac{I}{E}\beta_1 S^* &= \frac{E'}{E} - \beta_2 S^* + b, \\ \frac{I'}{I} &= \frac{E}{I}\alpha - c, \end{aligned} \tag{29}$$

thus

$$\begin{aligned} g_1 &= \frac{E'}{E} - aR_0, \\ g_2 &= \frac{E'}{E} - a. \end{aligned} \tag{30}$$

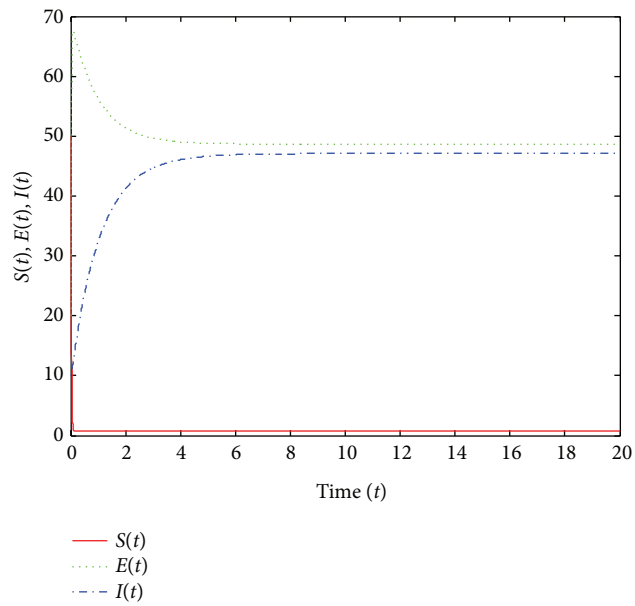


Figure 1: Dynamical behavior of system (3). Time series of susceptible, exposed, and infectious computers $S(t)$, $E(t)$, $I(t)$ with $R_0 > 1$.

Relations (28)–(30) imply that

$$\mu(B) \le \frac{E'}{E} - a. \tag{31}$$

Thus,

$$\frac{1}{t}\int_0^t \mu(B)\,d\tau \le \frac{1}{t}\int_0^t \left(\frac{E'}{E} - a\right) d\tau = \frac{1}{t}\ln\frac{E(t)}{E(0)} - a. \tag{32}$$

If $R_0 > 1$, then the virus-free equilibrium is unstable by Theorem 1. Moreover, the behavior of the local dynamic near D_0 as described in Theorem 1 implies that the system (3) is uniformly persistent in D; that is, there exists a constant $c_1 > 0$ and $T > 0$, such that $t > T$ implies that

$$\begin{aligned} &\lim_{t\to\infty} \inf S(t) > c_1, \\ &\lim_{t\to\infty} \inf E(t) > c_1, \\ &\lim_{t\to\infty} \inf I(t) > c_1, \\ &\lim_{t\to\infty} \inf [1 - S(t) - E(t) - I(t)] > c_1. \end{aligned} \tag{33}$$

For all $(S(0), E(0), I(0) \in D)$ (see [13, 14]),

$$\overline{q} = \lim_{t\to\infty} \sup \sup_{x\in K} \frac{1}{t}\int_0^t \mu(B)\,d\tau \le -\frac{a}{2} < 0. \tag{34}$$

The proof is complete. □

5. Numerical Examples

For the system (3), Theorem 2 implies that the virus dies out if $R_0 < 1$, and Theorem 4 implies that the virus persists if $R_0 > 1$. Now, we present two numerical examples.

Let $p = 0.5$, $\mu = 0.02$, $k = 0.4$, $\alpha = 0.6$, $r = 0.6$, $N = 100$, $\beta_1 = 0.7$, $\beta_2 = 0.8$, then $R_0 = 13.8 > 1$ and $\beta S^* < b + c$; Figure 1 shows the solution of system (3) when $R_0 > 1$. We

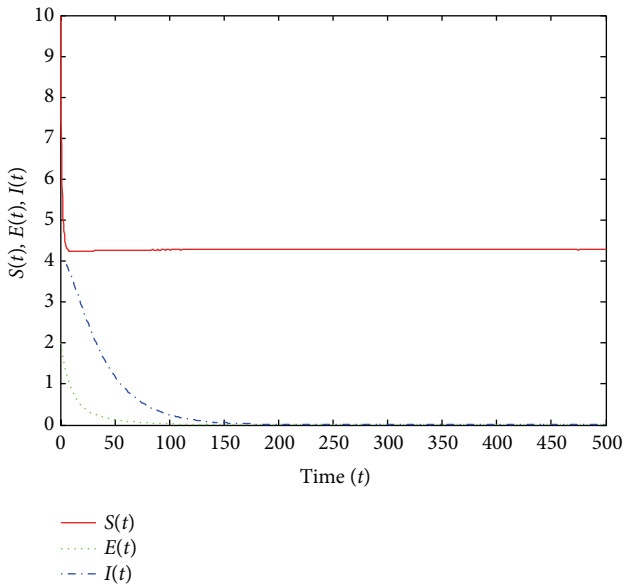


Figure 2: Dynamical behavior of system (3). Time series of susceptible, exposed, and infectious computers $S(t)$, $E(t)$, $I(t)$ with $R_0 < 1$.

can see that the viral equilibrium P^* of system (3) is globally asymptotically stable.

Let $p = 0.7$, $\mu = 0.001$, $k = 0.02$, $\alpha = 0.09$, $r = 0.04$, $N = 10$, $\beta_1 = 0.002$, $\beta_2 = 0.003$, then $R_0 = 0.1808 < 1$ and $\beta_2 S^0 < b + c$; Figure 2 shows the solution of system (3) when $R_0 < 1$. We can see that the virus-free equilibrium P^0 of the system (3) is globally asymptotically stable.

6. Conclusion

We assume that the virus process has a latent period and in these times the infected computers have infectivity also. A compartmental SEIR model for transmission of virus in computer network is formulated. In this paper, the dynamics of this model have been fully studied.

The results show that we should try our best to make R_0 less than 1. The most effective way is to increase the parameters p, k, r and decrease β_1, β_2, α and so on. Maybe in such way, the computer virus can be well predicted and thus controlled.

Acknowledgment

The work described in this paper was supported by the Science and Technology Project of Chongqing Education Committee under Grant KJ130519.

References

[1] C. Sun and Y.-H. Hsieh, "Global analysis of an SEIR model with varying population size and vaccination," *Applied Mathematical Modelling*, vol. 34, no. 10, pp. 2685–2697, 2010.

[2] L.-P. Song, Z. Jin, and G.-Q. Sun, "Modeling and analyzing of botnet interactions," *Physica A*, vol. 390, no. 2, pp. 347–358, 2011.

[3] J. Ren, X. Yang, L.-X. Yang, Y. Xu, and F. Yang, "A delayed computer virus propagation model and its dynamics," *Chaos, Solitons & Fractals*, vol. 45, no. 1, pp. 74–79, 2012.

[4] B. K. Mishra and S. K. Pandey, "Dynamic model of worms with vertical transmission in computer network," *Applied Mathematics and Computation*, vol. 217, no. 21, pp. 8438–8446, 2011.

[5] X. Han and Q. Tan, "Dynamical behavior of computer virus on Internet," *Applied Mathematics and Computation*, vol. 217, no. 6, pp. 2520–2526, 2010.

[6] Q. Zhu, X. Yang, and J. Ren, "Modeling and analysis of the spread of computer virus," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 5117–5124, 2012.

[7] L. X. Yang, X. Yang, Q. Zhu, and L. Wen, "A computer virus model with graded cure rates," *Nonlinear Analysis: Real World Applications*, vol. 14, no. 1, pp. 414–422, 2013.

[8] L. X. Yang, X. Yang, L. Wen, and J. Liu, "A novel computer virus propagation model and its dynamics," *International Journal of Computer Mathematics*, vol. 89, no. 17, pp. 2307–2314, 2012.

[9] P. van den Driessche and J. Watmough, "Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission," *Mathematical Biosciences*, vol. 180, pp. 29–48, 2002.

[10] M. Fiedler, "Additive compound matrices and an inequality for eigenvalues of symmetric stochastic matrices," *Czechoslovak Mathematical Journal*, vol. 24(99), pp. 392–402, 1974.

[11] J. S. Muldowney, "Compound matrices and ordinary differential equations," *The Rocky Mountain Journal of Mathematics*, vol. 20, no. 4, pp. 857–872, 1990.

[12] G. Butler, H. I. Freedman, and P. Waltman, "Uniformly persistent systems," *Proceedings of the American Mathematical Society*, vol. 96, no. 3, pp. 425–430, 1986.

[13] H. I. Freedman, S. G. Ruan, and M. X. Tang, "Uniform persistence and flows near a closed positively invariant set," *Journal of Dynamics and Differential Equations*, vol. 6, no. 4, pp. 583–600, 1994.

[14] P. Waltman, "A brief survey of persistence in dynamical systems," in *Delay Differential Equations and Dynamical Systems (Claremont, CA, 1990)*, S. Busenberg and M. Martelli, Eds., vol. 1475, pp. 31–40, Springer, Berlin, Germany, 1991.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 430945, 9 pages
http://dx.doi.org/10.1155/2013/430945

Research Article

An Evaluative Study on the Choice of Regional Strategic Emerging Industry Based on DEMATEL: Heilongjiang Province as an Example

Xiuying Guo,[1,2] Xiaofeng Hui,[1] and Junxiang Wang[2]

[1] *School of Management, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China*
[2] *College of Mathematics, Heilongjiang Institute of Technology, Harbin, Heilongjiang 150050, China*

Correspondence should be addressed to Xiuying Guo; guoxiuying6@163.com

Received 29 March 2013; Accepted 22 May 2013

Academic Editor: Tingwen Huang

Copyright © 2013 Xiuying Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Currently, the convergence of choosing regional strategic emerging industry is very serious in our country. And affected by its developmental stage, at present, the research of the methodology for choosing and evaluating regional strategic emerging industry is not very mature and lacks systematic and dynamic study. By expanding the DEMATEL method to the field of the complex weighted network, comparing the regional industry to a dynamic system and establishing the DEMATEL improved model based on industry input-output table, it is possible to realize the systematicness and dynamic nature of choosing regional strategy emerging industry.

1. Introduction

Throughout the history, the economic crisis will promote the development of emerging industries every time, and in various countries around the world, the emerging industries are of strategic importance to deal with the financial crisis, construct the new economical growth point, and preempt the development opportunities of the future. In September, 2009, Premier Jiabao Wen presided over three forums about strategic emerging industry development and "the concept of strategic emerging industry" was proposed for the first time. In October, 2010, the State Council determined seven industries as strategic emerging industries in China: energy conservation and environment protection, new generation of information technology, biotechnology, high-end equipment manufacturing, new energy, new materials, and alternative fuel vehicles. In March, 2012, Premier Jiabao Wen emphasized in the "government work report," to further promote the healthy development of strategic emerging industries, promoting the optimization and upgrading of industrial structure [1].

At present, strategic emerging industry has entered a stage of deepening development, which means, as the core of all levels of local government, establishing regional strategic emerging industries and overall promoting the upgrading of regional industrial structure and then promoting the sustainable development of social economy. Due to the economic development level of various regions in China and resource endowments have differences, regional industrial structure has its own characteristics, and its industrial gradual progress has obvious regional characteristics [2]. Therefore, regional practice must first be selected scientifically before regional strategic emerging industry will be developed. At present, led by the planning of country strategic emerging industry, proceeding from the separate condition, advantage, and characteristic, the provinces (cities, regions) of our country propose the strategic emerging industry planning during the period of "12th Five-Year Plan." However, after sorting out the strategic emerging industry planning in thirty provinces (cities, regions) of our country, the author found that during the period of "12th Five-Year Plan," the convergence in the choice of the regional strategic emerging industries is very serious, as shown in Figure 1.

The number of region selecting new material is twenty-nine, the region selecting new energy is twenty-six, the number of selecting biological is twenty-five; these three ones are ranked as the top three. The convergence in the

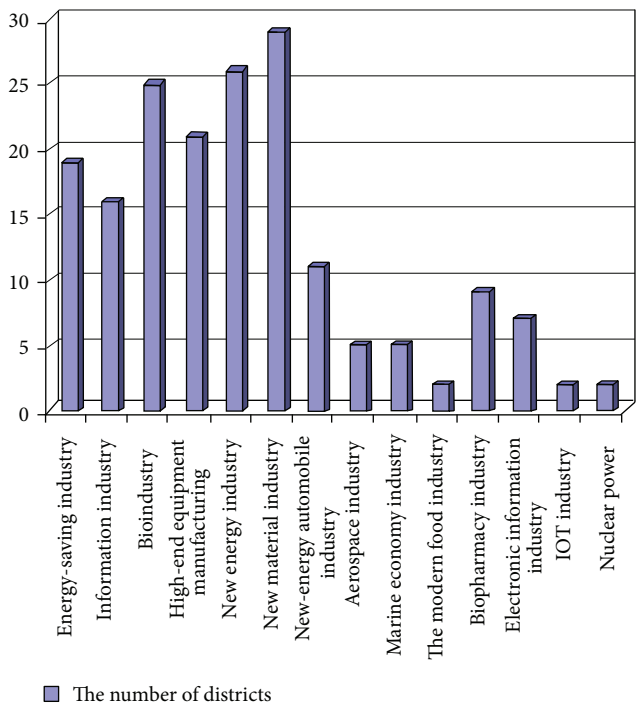


Figure 1: Selected case charts of the strategic emerging industry in thirty provinces (cities) of our country.

choice of the regional strategic emerging industries can easily result in vicious competition, redundant construction, waste of resources, and other issues of excess production capacity. Regional strategic emerging industry is never a simple regional of the country strategic emerging industries, the difference between the two is not only in amount, but also in quality. Therefore, how to accurately and scientifically select and evaluate the regional strategic emerging industry will have important theoretical and practical meanings.

At present, many scholars are carrying on the theoretical and empirical research of the strategic emerging industries: first, Qiao and Yang [3] and Xiong and Zeng [4] established an evaluation system from the perspective of coupling the emerging industries and traditional industries. Second, He [5], Li [6], and Luo [7] and Hu et al. [8] constructed the choice model based on the Analytic Hierarchy Process (AHP), fuzzy comprehensive and the combination of "AHP-IE—PCA" empowerment evaluation method. Third, Liu and Cheng [9] and Liu [10] used comparative analysis to propose the suggestion of choosing mechanism. Forth, Ao and Liu [11] come up with the choosing principle, basic principle and several kinds of typical selective methods of the strategic emerging industry from the theoretical level.

It is not difficult to see that scholars for regional strategic emerging industry choice mainly focused on the importance of strategic emerging industry, connotation and characteristic, mechanism of formation and development, selection principle, and cultivation measures discussion of theory; empirical research is mainly to have a set of strategic emerging industries evaluation of each area; method is selection the method of simple transplantation for leading industry, lacking regional strategic emerging industry choice and the choice of system science method and evaluation system. As regional industrial system is a complicated dynamic system, and strategic emerging industry has strong correlation effect and characteristics of sustainable development, this paper will be based on the complicated network and input-output theory, with the aid of DEMATEL improved model as an example to Heilongjiang province regional strategic emerging industry selection research, and will do dynamic gathered qualitative research for selection result.

2. The Choice of Regional Strategic Emerging Industry Based on the Improved DEMATEL Model

2.1. Industrial Complex Network and the Introduction of the Theory of Input and Output. Regional industrial system, being the most complicated subsystem of national economic system construction, must composite complex system theory and macroeconomic quantity analysis to make overall planning [12]. Industrial system is mainly related to the input and output of the interindustry; so with the help of the input and output theory, we need the theory of input and output to study some characteristics of the regional industrial system. Input-output theory is founded by the Harvard University professor Wassily Leontief in 1936, and it mainly studies the quantitative analysis of the input-output interdependence relationship of each part in the economic system. This interindustry perplexing relationship of supply and demand generally is succinctly explained by the input-output table.

2.2. Based on DEMATEL Improved Model of the Input-Output Table and Empirical Research. DEMATEL is an analysis method, using graph theory and matrix tools for system analysis, proposed by the American scholar Gabus and Fontela. By constructing the direct affect matrix between system factors, apply the matrix to calculate the center degree and reason degree of each factor, and then determine the importance of various factors in the system. This method is more effective for dealing with complicated social economic problems, especially for the problem of unclear relationships of system factor [13]. In recent years, many scholars apply DEMATEL to study complex system. Tzeng et al. [14] use DEMATEL, structure analysis method (AHP), and fuzzy integral to construct a selection evaluation model of online learning project. Liang and Ma [15] apply DEMATEL method to analyze the foundation engineering construction risk. Zhu and Yang [16] bring DEMATEL and entropy according to the autocorrelation of customers' demand and the influence of market competition evaluation to correct the basic important factor of the customers' needs and integrate the processing results of the two sides, then we can get more objective final important factor of the customers' needs.

On the basis of using DEMATEL method for reference, this paper extends the DEMATEL method to the weighted field of complex network, confirms network supporting matrix using the improved DEMATEL method, studies the

relations of every industrial department based on input-output matrix, then determines the preselected strategic emerging industry whose centrality and reason degree are high. The DEMATEL improved model based on complex network is described as follows.

Step 1. Generated standardized effect matrix D.

We regard every industrial department of the national economy as one point and regard input-output relationship of each department as one border. Take, for example, input-output table of 42 industry departments of Heilongjiang province in 2007, we regard each industry department as an influencing factor and regard the input and output values of every industrial departments as weighted interrelation between every factor, a_{ij}, and get direct affect matrix $A = (a_{ij}), (i, j = 1, 2, \ldots, 42)$. By setting up the table of 42 industrial departments corresponding to sequence number, as shown in Table 1, we represent the industry name with sequence number in the latter research.

According to the direct affect matrix A, with the help of

$$D = sA \quad (s > 0), \tag{1}$$

construct standardized affect matrix D, where $d_{ij} = sa_{ij}$, $(i, j = 1, 2, \ldots, N)$ and "S" is called the scale factor, $0 < S <$ Sup. We usually make

$$\text{Sup} = \max\left\{\frac{1}{\max_{1\le i\le N}\sum_{j=1}^{N}\left|a_{ij}\right|}, \frac{1}{\max_{1\le j\le N}\sum_{i=1}^{N}\left|a_{ij}\right|}\right\}. \tag{2}$$

According to (1) and (2) calculation methods, we use MATLAB 7.1 to calculate the ratio factor of Heilongjiang province standardization impact matrix D $s = 7.65036819 + E - 9$ and get the standardized affect matrix D.

Step 2. Construct total impact matrix T.

The sum of rows of matrix D is called the outgoing degree of i industry, which expresses the direct impact total intensity of the industry i, as $d_i^{\text{out}} = \sum_{j=1}^{N} d_{ij}$. The sum of columns of matrix D column is called incoming degree of industry j, which expresses the direct impact total intensity of the industry j, as $d_j^{\text{in}} = \sum_{j=1}^{N} d_{ij}$. However, for the actual issue, we not only consider the direct effect between the industries, but we also need to consider the indirect effects between the industries, and the chain effects brought by an industry change. The total effect matrix T expresses the comprehensive summation of the direct and indirect effects between factors, in order to determine the final effect of each factor relative to the highest level factor in the system. That is,

$$T = C + C^2 + \cdots + C^n = C\frac{I - C^{n-1}}{I - C}. \tag{3}$$

Since $0 \le C_{ij} \le 1$, so, when $n \to \infty$, $C^{n-1} \to 0$,

$$T = C\frac{I}{I - C} = C(I - C)^{-1}. \tag{4}$$

Table 1: The table of 42 industries of national economy in Heilongjiang province corresponding to sequence number.

Industrial sequence number	Industry name
1	Agriculture forestry, animal husbandry and fishery
2	Coal mining and coal washing industry
3	Oil and natural gas industry
4	Metal mines
5	Nonselected industry
6	Food manufacturing and tobacco processing
7	Textile crafts and others
8	Textile, leather, and products industry
9	Wood processing and furniture manufacturing industry
10	Paper printing and stationery and sports goods manufacturing industry
11	Petroleum processing, coking, and nuclear fuel processing
12	The chemical industry
13	Nonmetallic mineral products industry
14	Metal smelting and rolling processing industry
15	Metal products industry
16	General, special equipment manufacturing industry
17	Transportation equipment manufacturing
18	Electrical machinery
19	Equipment manufacturing industry in communications equipment, computers, and other electronic equipment metal
20	mine and other mining manufacturing instrumentations and cultures
21	Office machinery manufacturing industry
22	Manufacturing and others
23	Waste
24	Electricity, heat production and supply industry
25	Gas production and supply industry
26	Water production, and supply industry
27	Construction
28	Transportation and warehousing industry
29	Postal industry
30	Information transmission, computer service, and software industry
31	Wholesale and retail trade industry
32	Accommodation and catering industry
33	Finance and insurance industry
34	Real estate
35	Leasing and business services
36	Tourism
37	Research and experimental development industry
38	Integrated technology services
39	Other social services
40	Education
41	Health, social security, and social welfare
42	Culture, sports, and entertainment industry
	Public management and social organization

Table 2: The table of 42 industries impacted degree in Heilongjiang province in 2007.

Industrial serial number	Impact degree
1	0.95071
2	0.30522
3	0.9977
4	0.095245
5	0.04165
6	0.42161
7	0.037194
8	0.035844
9	0.087075
10	0.11412
11	0.71392
12	0.75186
13	0.13191
14	0.67531
15	0.13036
16	0.58628
17	0.15669
18	0.23295
19	0.079387
20	0.095469
21	0.017872
22	0.025863
23	0.66337
24	0.014136
25	0.046386
26	0.059348
27	0.62216
28	0.049898
29	0.11265
30	0.52904
31	0.13655
32	0.18218
33	0.096514
34	0.15404
35	0.008932
36	0.031676
37	0.014577
38	0.12878
39	0.021051
40	0.029709
41	0.027365
42	0.001848

According to the formula in (4), after calculation, we can get the total effect matrix $T$of 42 industries of Heilongjiang province in 2007.

Step 3. Calculate the impacted degree matrix T_r and the affected degree matrix T_c.

The row sum and column sum of sum-impact matrix T are known as the impact matrix T_r and the affected matrix T_c, thereinto,

$$T_r(i) = \sum_{j=1}^{n} t_{ij},$$
$$T_c(j) = \sum_{i=1}^{n} t_{ij}. \tag{5}$$

By the formula in (5), we calculated the impact degree of 42 industries of Heilongjiang province in Table 2, affected degree of 42 industries of Heilongjiang province in Table 3.

Step 4. Calculate centrality degree matrix M and reason degree matrix R.

The impact matrix T_r plus the affected matrix T_c is the centrality degree matrix M, namely,

$$M = T_r + T_c. \tag{6}$$

M_i is the centrality degree of ith industry, representing the importance of ith industry in regional industrial system, namely,

$$M_i = T_r(i) + T_c(i). \tag{7}$$

The impacted matrix T_r minus the affected matrix T_c is the reason degree matrix R, namely,

$$R = (T_r - T_c). \tag{8}$$

R_i is the reason degree of ith industry, which represents causes or results of ith industry in input-output process of the industrial system factors, namely,

$$R_i = T_r(i) - T_c(i). \tag{9}$$

If the reason degree $R_i > 0$, then it shows that the ith industry is more influential than other industries, and the industry i is reason industry; if the reason degree $R_i < 0$, then it shows that the ith industry is more affected than other industries, and the industry i is called the outcome industry. By the formula in (6) and (8), we calculate the top 10 industrial centrality degrees in Table 4 and the reason degree table of the corresponding industry in Table 5.

Step 5. Draw industry cause-result diagram.

First, establish Descartes coordinate system by the centrality degree to be the abscissa and the reason degree to be the ordinate. Second, mark the position of the industry in the coordinate system in Figure 2. Third, choose the strategic emerging industry according to the centrality degree and reason degree to analyze the various roles of the industries in the regional industry system, and give some suggestions to the actual industrial system.

Table 3: The table of 42 industries affected degree in Heilongjiang province in 2007.

Industrial serial number	Affected degree
1	1.0523
2	0.1989
3	0.2731
4	0.014714
5	0.021051
6	1.1157
7	0.048682
8	0.069762
9	0.11041
10	0.08791
11	0.81487
12	0.58536
13	0.11039
14	0.19893
15	0.076224
16	0.50038
17	0.24168
18	0.12866
19	0.018562
20	0.025923
21	0.010428
22	0.004718
23	0.59301
24	0.016373
25	0.037949
26	1.2815
27	0.40812
28	0.056323
29	0.076689
30	0.2504
31	0.2205
32	0.076427
33	0.10506
34	0.10007
35	0.020194
36	0.025845
37	0.033515
38	0.15067
39	0.12273
40	0.17842
41	0.025338
42	0.12669

2.3. The Analysis of DEMATEL Results and the Suggestion of Strategic Emerging Industry Selection. As Table 2 shows, $0.001848 \leq T_r(i) \leq 0.9977$, the expected value is 0.228915 and the standard deviation is 0.281140. Among these data,

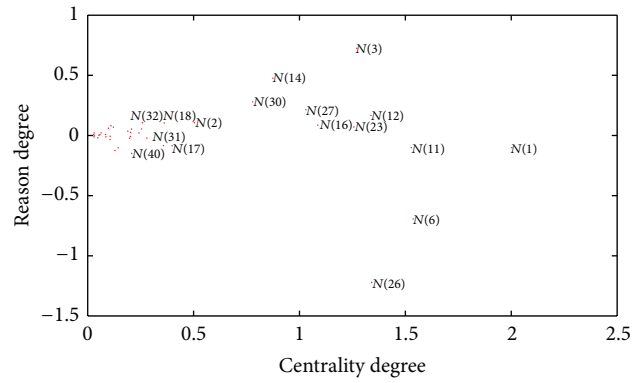


Figure 2: The cause-result diagram of 42 industries in Heilongjiang province.

the largest impacted degree is oil and natural gas exploitation industry, followed by agriculture, hunting, and chemical industry. We can see that the industry influenced degree of Heilongjiang province is less than 1, but the standard deviation is relatively large, which means Heilongjiang Province industrial agglomeration level is low and the distribution of industries is scattered. Agriculture and energy industries are the two major industries of Heilongjiang province.

As Table 3 shows, $0.004718 \leq T_c(j) \leq 1.2815$, the expected value is 0.228916 and the standard deviation is 0.315526. We can see that the standard deviation of industrial impacted degree is larger than the industrial affected degree, that is, to say, the affected degree varies greatly. Food manufacturing and tobacco processing industry, animal husbandry, and fishery are rounding out the top three industries impacted degree, and they are mainly concentrated in primary industry and secondary industry.

As Table 4 shows, $0.0283 \leq M_i \leq 2.00301$, Heilongjiang province's industry centrality degree is low, lacking the core industry in region system. The centrality degree industries such as agriculture, hunting, petroleum processing, coking and nuclear fuel processing, transportation, and warehousing industries are more larger and can be taken as the development base of strategic emerging industry for their comprehensive influence in Heilongjiang province.

As Table 5 shows, the reason degrees of 24 industries are positive values, $0.002027 \leq R_i \leq 0.7246$. They become the reason factors to influence the development of other industries. The top 2 of 24 industries impacted degree including oil and natural gas industry and metal smelting and rolling processing industry which have more influence than other industries are the cause of industry. And the reason degrees of 24 industries are negative values, $-0.124842d \leq R_i \leq -0.002237$. The last 2 of 18 industries impacted degree including construction industry and food manufacturing and tobacco processing industry which are affected more largely by other industries are the result industry. This kind of industry departments should put more attention to the development and changes of their related industries and adjust their development strategies timely.

In view of the above analysis, the higher degree of influence in Heilongjiang Province mainly comprises agriculture,

Table 4: The table of top 10 industries centrality degree in Heilongjiang province in 2007.

Industry number	Industry name	Centrality
1	Agriculture forestry animal husbandry, and fishery	2.003
6	Food manufacturing and tobacco processing	1.5373
11	Petroleum processing, coking, and nuclear fuel processing	1.5288
26	Construction	1.3408
12	The chemical industry	1.3372
3	Oil and natural gas industry	1.2708
23	Electricity, heat production, and supply industry	1.2564
16	General special equipment manufacturing industry	1.0867
27	Transportation and warehousing industry	1.0303
14	Metal smelting and rolling processing industry	0.87424

Table 5: The table of reason degree in Heilongjiang province in 2007 (the top 10 industries centrality degree).

Industry number	Industry name	Reason degree
1	Agriculture forestry, animal husbandry, and fishery	−0.10156
6	Food manufacturing and tobacco processing	−0.69412
11	Petroleum processing, coking, and nuclear fuel processing	−0.10095
26	Construction	−1.2221
12	The chemical industry	0.1665
3	Oil and natural gas industry	0.72461
23	Electricity, heat production, and supply industry	0.07036
16	General special equipment manufacturing industry	0.085897
27	Transportation and warehousing industry	0.21404
14	Metal smelting and rolling processing industry	0.47637

forestry, animal husbandry and fishery, oil and equipment manufacturing. Therefore, for Heilongjiang province, on the basis of international and domestic scientific and technological progress and industrial development objective trends to be taken full advantage of, the primary industry and the advantages of the secondary industry should be given full play, from the specific situation in Heilongjiang province, combined with the major needs of the current economic and social development; the strategic emerging industry selection should be determined. Specific recommendations are as follows.

(1) Relying on the centrality, the agriculture, forestry, animal husbandry, and fisheries would be ranked first, in Heilongjiang province; bioindustry and farm machinery and equipment manufacturing industry can be an important part of the strategic emerging industries. There are 200 million mu of arable land in Heilongjiang province and is the country's total arable land ninth, and it is one of the most important grain production bases in China. However, in Heilongjiang Province, there is still raw grain economy and the agricultural lack of scientific and technological content and industrial innovation. Therefore, the resource advantages of agriculture should be given full play, in order to accelerate the upgrading of industrial structure of agriculture, to strengthen the deepening agricultural production processing and development of upstream and downstream biological industry chain, and vigorously to develop green products, the formation of the ecological cycle of bioclusters. At the same time, in Heilongjiang province, the agricultural investment in the equipment manufacturing industry should be further strengthened, to make their own province of farm machinery and equipment manufacturing industry and to gradually realize the industrialization of agriculture, ultimately the formation of the organic combination of TIAC.

(2) Relying on the degrees of the reasons, mining industry on oil and natural gas, as well as electric power, the heat production and the supply industry ranked in first number, in Heilongjiang province, the led "oil economy" should be changed gradually; the formation of a new pattern of wind energy, hydropower, biomass, and other new types of clean energy-based diversified energy development should be formed. Carry out the riding implementation strategies, and develop with the large-scale backbone enterprises. Based on the straight play of the leading role of large enterprise, the upgrading of the energy industry should be promoted accelerately. Actively participate in the restructuring and industrial upgrading of the central enterprises, do a good job to build together and to form a complete set, explore together with the central enterprises to form the diversification of the new energy industry base and industrial base so as to realize the transformation of economy of the resource-based cities, especially the resource-exhausted cities.

(3) Rely on the common use in the higher central degrees and causes of them, and also depend on the special equipment

manufacturing, actively seize the chances of the policy in the aircraft, rail transportation and automobile industry. In Heilongjiang Province we should focus on the manufacturing industry of the construction of the transportation equipment. There is a good base on transportation equipment manufacture in Heilongjiang province; Dongan group has become a major research and production base for aviation engine and helicopter transmission system; there are, 4G1 and 4G9, two series engine manufacturing technology and products in the automobile engine company of Dong' an; there are four production bases in North car group with Qi, Ha, Mu, and Da, and there is the e-design manufacturing and export base of railway freight and railway cran of China, and the domestic market share is more than 20%.

(4) In the mining province, there are varieties of mineral resources abundant in Heilongjiang province. Now, 134 kinds of minerals of various kinds have been discovered, accounting for 57.2% of the 234 kinds of various minerals that have been found in China. There are 87 species that have been identified reserves of mineral, and it has been identified that there are 87 kinds of mineral reserves, accounting for 37.7% of the number of the resources of the mineral reserves. Therefore, the new material industry may be taken as strategic emerging industry in Heilongjiang province, which has important strategic significance for promoting the upgrading of industrial structure and transformation of economic development.

In January 2011, "Heilongjiang Province National Economic and Social Development 12th Five-Year Plan" is proposed by the Heilongjiang Provincial Party Committee; the synopsis to develop new materials industry, biological industry, new energy equipment manufacture, new farm machinery and equipment manufacturing, transportation equipment manufacturing industry, and five strategic emerging industries is pointed out clearly, and it is consistent with the strategic emerging industry in this paper.

3. The Cluster Analysis of Regional Emerging Industries

The cultivation of strategic emerging industry cluster is an important way develop regional economic transformation and industrial upgrading, so, we should appraise the standard of strategic emerging industry accumulation dynamically after we have chosen and determined the strategic emerging industries.

It is important to give the major construction to the strategic emerging industries which have good aggregation and form regional leading and characteristic industry by excavating their upstream and downstream industries. This action will drive the regional economic development. In addition, it is equally important to analyze the reasons of the strategic emerging industries which have poor aggregation and realize the reasonable distribution.

3.1. Analysis of the Industry Concentration Degree Index. In its narrow sense, the degree of industrial concentration is used to express the largest proportion index of the largest entrepreneur among the whole national economy or all the economic activities of enterprises. It is a concept which is used to describe the characteristics of the market structure and the market power of large enterprises. Usually, it is meant by the proportion of the total indexes of several enterprises arranged in the front of one industry among the whole industry. The bigger ratio the is, the higher the industry concentration is. But in its broad sense, the industry concentration can be referred to some industry, some certain owned enterprises, and the proportion of some enterprises in the whole economy. The calculation formula is

$$CR_n = \frac{\sum (X_i)_n}{\sum (X_i)_N}, \quad N > n, \tag{10}$$

CR_n is the industry concentration of the previous biggest enterprises; X_i is the ith enterprise output, production, sales, sales volume, employees number, the total assets, and so on; n is the number of the previous biggest enterprises in the industry; N is the total number of enterprises in the industry.

In this paper, we choose $n = 4$ and get the data of industrial concentration in accordance with the data of the "statistical yearbook of Heilongjiang province" in Table 6.

3.2. Analysis of the Industrial Concentration Index. The concentration index of industry (EG index) is to solve the distortion of spatial Gini index, combined with Geffen Dahl (H), proposed by Elilsion and Glaeser [17], a new industrial concentration index which measures the geographical space and reflects the difference of regional economic development. The assumption is as follows: an economy (country or region) in a certain industry with N, and M a geographical distribution in the economy area, the calculation formula is

$$r_i = \frac{G_i - \left(1 - \sum_{j=1}^{M} x_j^2\right) H_i}{\left(1 - \sum_{j=1}^{M} x_j^2\right)\left(1 - H_i\right)}, \tag{11}$$

where $H_i = \sum_{j=1}^{N} z_j^2$, $G_i = \sum_{j=1}^{M} (s_{ij} - x_j)^2$.

$$r_i = \frac{\sum_{j=1}^{M} \left(s_{ij} - x_j\right)^2 - \left(1 - \sum_{j=1}^{M} x_j^2\right) \sum_{j=1}^{N} z_j^2}{\left(1 - \sum_{j=1}^{M} x_j^2\right)\left(1 - \sum_{j=1}^{N} z_j^2\right)}, \tag{12}$$

S_{ij} is the proportion of which the industry output of industry i in reginal j accounts for national output, x_j is the proportion of which all industries output in reginal j accounts for total national output of all industries. r_i is the industrial concentration degree of industry i, the greater the value of r_i is (maximus is 1), the more concentrated the industry i is. G_i represents a spatial Gini index of industry i, H_i is on behalf of the Herfindahl index of industry i, z_j is enterprise j in the industrial share of total employment. By the formula in (12), the Heilongjiang province strategic emerging industry industrial agglomeration index is shown in Table 7.

3.3. Analysis of Strategic Emerging Industry Agglomeration in Heilongjiang Province. (1) The degree of industrial concentration of Heilongjiang province in five strategic emerging

Table 6: Strategic emerging industry of Heilongjiang industrial concentration.

Year	Industry				
	New material	Biology	New energy equipment manufacturing industry	New farm machinery and equipment manufacturing industry	Transportation equipment manufacturing industry
2008	0.325716	0.314621	0.700157	0.311452	0.715621
2009	0.412326	0.378612	0.709826	0.368610	0.778952
2010	0.510165	0.465365	0.714054	0.390874	0.841048

Table 7: Strategic emerging industry of Heilongjiang's industrial agglomeration index.

Year	Industry				
	New material	Biology	New energy equipment manufacturing industry	New farm machinery and equipment manufacturing industry	Transportation equipment manufacturing industry
2008	0.210071	0.063821	0.09072	0.031925	0.227509
2009	0.241346	0.069672	0.141869	0.059078	0.266527
2010	0.289908	0.086536	0.189489	0.120672	0.293065

industries and its industry aggregation index are showing a rising trend, but the agglomeration level is not high; the scope of CR_n is 0.314621–0.841048, and the highest amplification of industrial concentration is the advanced material, annual amplification mean reaches, which is closely related with the lower economic development in Heilongjiang province which is located in the frontier of the motherland, the overall level of economic development is not high. As for the advanced material, one kind of emerging industry has the same starting point in different provinces. Therefore, Heilongjiang province developed this area faster, but the industrial concentration is not very high, reached 0.510165 in 2010, which required us to invest more in a high level and increase its concentration ratio. Among them, the industry concentration of transportation equipment manufacturing industry concentration degree index was the highest, reached 0.841048 in 2010, which fitted the rapid development of transportation equipment manufacturing industry trend in recent years. Therefore, our province should concentrate more on strength, seize the opportunity, depend on the relatively strong foundation of equipment manufacturing industry, and focus on supporting the potential of the transportation equipment manufacturing industry, which results in the rapid development.

(2) The degree of industrial concentration of Heilongjiang province is normally low, only reached 0.031925–0.293065 and all less than 0.3, which declares that the Heilongjiang province, which has advantage in agriculture, states in a weaker area comparing to other provinces. And its agriculture machinery equipment manufacture's concentration index is only 0.120672, which declares that its productivity of agriculture machinery equipment manufacture is very poor, which cannot fit the demand of modern agriculture productivity. However, the greatest increasing trend is gratifying, which separately reached 45.9% and 104.3% of the two years. It also means that the Heilongjiang province has realized the importance and feasibility to develop the new kind of agriculture machinery equipment manufacture. The concentration index of biological industry is low, because the biological industry belongs to the resource intensive industries the phenomenon of all over the country repeat construction is pretty serious, which is more difficult to form in a particular province of industrial agglomeration. New material industry and new energy equipment manufacturing's industry concentrations have increased steadily, thanks to the advantage of abundant resources in Heilongjiang province.

4. Conclusion

Though the policy support is essential to strategic emerging industry, to achieve the objective of adaptation to local conditions, upgrading of industrial structure, and rapid and healthy economic development, it is very important to choose the suitable industry according to the characteristics, level, and realistic situation. In this paper, improved DEMATEL model based on input-output and complex network theory is adopted, which can play a comprehensive evaluation effect, by fully relying on the advantages of traditional industries and fully considering the coupling relationship between them. Taking Heilongjiang province as an example to choose 2007 input-output analysis of regional strategic emerging industry, the selection results are basically the same with the "12th Five-Year Plan" in 2011. The results can show that the the selection and model analysis are in conformity with the development reality and reflect its science, objectivity, and reasonableness. In addition, a dynamic follow-up index test on industry concentration and aggregation is established with the industrial development data from 2008 to 2010. Considering the sustainable development of strategic emerging industry, the ecoefficiency and dynamic industrial evaluation will be continued in the future.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (Grant nos. 71173060, 71102131), the Social Sciences Foundation of Heilongjiang Province (project no. 10E078), the National Natural Science Foundation of Heilongjiang Province (Grant nos. G201004), and the Social Science research Foundation of Heilongjiang Province Education Department (project no. 12512278).

References

[1] Premier Jiabao Wen, "Government Work Report," China Network, 2012, http://www.china.com.cn/.

[2] L. Q. Guo, "International Comparison and Enlightenment in the adjustment of the declining industry," *Economic Review*, vol. 4, pp. 14–16, 2004.

[3] F. L. Qiao and J. Yang, "Research on selection and evaluation of strategic emerging industry in Liaoning Province," *Journal of Shenyang University of Technology*, vol. 4, pp. 268–273, 2010.

[4] Y. Q. Xiong and D. Zeng, "The cultivation and development of strategic emerging industries: based on the perspective of the traditional industries," *Emerging Industry Research*, vol. 4, pp. 49–54, 2011.

[5] Z. C. He, "Strategic emerging industry choice and evaluation and empirical analysis," *Public Understand Science*, vol. 12, pp. 62–67, 2010.

[6] H. M. Li, "Regional strategic emerging industry selection evaluation," *Journal of North China Water Conservancy and Hydropower College (Social Sciences Edition)*, vol. 4, pp. 91–93, 2011.

[7] D. S. Luo, "The establishment of evaluation index system of industrial characteristics and thinking based," *Journal of City Economy*, vol. 1, pp. 7–9, 2012.

[8] Z. H. Hu, C. Q. Li, and X. Y. Qing, "Based on the AHP-IE-PCA 'combination weighting method of strategic emerging industry selection model'," *Public Understand Science*, vol. 7, pp. 104–110, 2011.

[9] Z. Y. Liu and H. S. Cheng, "Strategic emerging industry cluster development and network characteristics," *Reform*, vol. 5, pp. 36–42, 2010.

[10] Y. Liu, "Experience and Enlightenment of Japan in the development of strategic emerging industries," *City Development Strategy*, vol. 6, pp. 2–25, 2010.

[11] Y. C. Ao and J. X. Liu, "Theoretical thoughts of the selection of regional strategic new industries," *The Commercial Era*, vol. 21, pp. 116–117, 2012.

[12] Z. L. Xing, "Study on regional industrial structure network model based on complex network theory," *Industrial Technology and Economy*, vol. 2, pp. 19–30, 2012.

[13] R. Liu and Z. F. Zhou, "Study on the evaluation index system of credit risks of enterprise based on DEMATEL," *Management Experts Academic Edition*, vol. 1, pp. 22–30, 2012.

[14] G. Tzeng, C. Chiang, and C. Li, "Evaluating intertwined effects in e-learning programs: a novel hybrid MCDM model based on factor analysis and DEMATEL," *Expert Systems with Applications*, vol. 32, no. 4, pp. 1028–1044, 2007.

[15] Y. K. Liang and Z. D. Ma, "Construction risk analysis of foundation engineering based on DEMATEL method," *Journal of Engineering Management*, vol. 2, pp. 164–167, 2010.

[16] C. Y. Zhu and M. S. Yang, "Modifying the importance ratings of customers' requirements correction based on DEMATEL and Entropy," *Industrial Engineering and Management*, vol. 2, pp. 97–101, 2012.

[17] G. Ellison and E. L. Glaeser, "Geographic concentration in U.S. manufacturing industries:a dartboard approach," *Journal of Political Economy*, vol. 105, pp. 889–927, 1997.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 762861, 10 pages
http://dx.doi.org/10.1155/2013/762861

Research Article

Impulsive Consensus for Leader-Following Multiagent Systems with Fixed and Switching Topology

Zhi-Wei Liu,[1,2] Zhi-Hong Guan,[2] and Hong Zhou[1]

[1] *Department of Automation, Wuhan University, Wuhan 430072, China*
[2] *College of Automation, Huazhong University of Science and Technology, Wuhan 430074, China*

Correspondence should be addressed to Zhi-Hong Guan; zhguan@mail.hust.edu.cn

Received 29 March 2013; Accepted 2 June 2013

Academic Editor: Chuandong Li

Copyright © 2013 Zhi-Wei Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper studied the consensus problem of the leader-following multiagent system. It is assumed that the state information of the leader is only available to a subset of followers, while the communication among agents occurs at sampling instant. To achieve leader-following consensus, a class of distributed impulsive control based on sampling information is proposed. By using the stability theory of impulsive systems, algebraic graph theory, and stochastic matrices theory, a necessary and sufficient condition for fixed topology and sufficient condition for switching topology are obtained to guarantee the leader-following consensus of the multiagent system. It is found that leader-following consensus is critically dependent on the sampling period, control gains, and interaction graph. Finally, two numerical examples are given to illustrate the effectiveness of the proposed approach and the correctness of theoretical analysis.

1. Introduction

During the past several decades, the consensus problem of the multiagent system has drawn a great deal of attentions because of its broad applications in many domains, including distributed coordination [1], synchronization of dynamical networks [2], distributed filtering [3], and load balancing [4]. The basic idea of consensus is to design a distributed control such that the team of agents can achieve a state agreement only by locally available information without central control stations. Consensus problem has been addressed in various situations, such as time delay [5], switching topology [6], asynchronous algorithms [4, 7], nonlinear algorithms [8, 9], quantized data [4, 10], noisy communication channel [11], and second-order model [12, 13].

Inspired by some biological systems and engineering applications, the leader-following consensus problem has received a lot of interest. The leader is a special agent whose motion is independent of all other agents and thus is followed by all other agents. It has been widely used in many applications [14, 15]. For the first-order multiagent systems, Jadbabaie et al. [16] considered a leader-following consensus problem and discussed the convergence properties of the leader-follower systems. Cao and Ren [17] studied a leader-following consensus problem with reduced interaction for both first- and second-order multiagent systems. Su et al. [18] studied a flocking algorithm with a virtual leader. Zhu and Cheng [19] considered leader-following consensus of second-order agents with multiple time-varying delays. Meng et al. [20] studied the leaderless and leader-following consensus algorithms with communication and input delays under a directed network by the Lyapunov theorems and the Nyquist stability criterion.

In recent years, owing to the development of digital sensors and the constraints of transmission bandwidth of networks, many control systems can be modeled by continuous-time systems together with discrete sampling. Therefore, it is significant to design the distributed control for continuous-time multiagent systems based on sampled information. There are a few reports [20–25] dealing with this problem, where the control inputs regulate the velocity of each agent continuously over the sampling period.

On the other hand, impulsive dynamical systems exhibit continuous evolutions typically described by ordinary differential equations and instantaneous state jumps or impulses.

It is also well known that the impulsive control is more efficient than one of continuous control in many situations. The examples include ecosystems management [26], orbital transfer of satellite [27], and optimal control of economic systems [28]. The main idea of impulsive control is to instantaneously change the state of a system when some conditions are satisfied. During the last few decades, it has been widely applied into the synchronization problems of complex dynamical networks [29–31], which can be regarded as first-order multiagent systems with nonlinear dynamics. In many real-world system, agents are governed by both position and velocity dynamics. The impulsive control for second-order multiagent system was studied in [32, 33], where both velocity and position are instantaneously changed by impulsive control, but position cannot change quickly in many situation. Therefore, it is more reasonable to only regulate the velocity of each agent to reach consensus [34, 35]. In [34], we designed impulsive velocity-control for multiagent systems with fixed topology to achieve consensus. In [35], an impulsive control was proposed in which the current position data of its neighbours and the past position data of its own state were utilised to regulate the velocity of agents.

This paper aims to investigate the consensus problem of leader-following multiagent systems by using impulsive control which only regulates the velocity of agents. Our main contributions are summarised as follows. First, a necessary and sufficient condition under fixed topology is derived, and it is found that the leader-following consensus in multiagent systems with sampling information can be reached if and only if the sampled period is bounded by critical values which depend on control gains and the interaction graph. Second, a sufficient condition under switching topology is obtained, and it is shown that the impulsive interval is restricted by an upper bound which depends on control gains, the diagonal element of the Laplacian matrix, and the connections between agents and leader. The two key difference between this paper and our earlier work [34] are that the leader-following case is taken into account and that this paper considers multiagent systems under switching topology.

The remainingpart of the paper is organized as follows. In Section 2, some necessary mathematical preliminaries are given. Main results of this paper, that is, the convergence of the distributed impulsive control under fixed and switching topology, are presented in Sections 3 and 4. In Section 5, some illustrative numerical examples are given. Concluding remarks are finally stated in Section 6.

2. Problem Formulation

Let $\mathbb{R}$ and $\mathbb{C}$ denote the set of real numbers and complex numbers, respectively. For $A = (a_{ij})_{m\times m} \in \mathbb{R}^{m\times m}$, $\lambda_1(A)$, $\lambda_2(A), \ldots, \lambda_m(A)$ are the eigenvalues of A, $\rho(A)$ represent the spectral radius of A. The identity matrix of order n is denoted as I_n (or simply I if no confusion arises). For $\gamma \in \mathbb{C}$, $\mathrm{Re}(\gamma)$ and $\mathrm{Im}(\gamma)$ are the real and imaginary part of γ, respectively. $\mathbf{1}_n = (1, 1, \ldots, 1)^T$ is the column vector. $\mathbf{0}_{n\times m}$ denotes the $n\times m$ matrix with all elements equal to zero.

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ be a directed graph (digraph) with the set of nodes $\mathcal{V} = \{1, 2, \ldots N\}$, the set of edges $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$, and the weighted adjacency matrix $\mathcal{A} = (a_{ij})_{N\times N}$. In the digraph $\mathcal{G}$, node i represents the agent i, and an edge in $\mathcal{G}$ is denoted by an ordered pair $\{j, i\}$. $\{j, i\} \in \mathcal{E}$ if and only if the agent i can directly receive information from the jth agent. In this case, the jth agent is the neighbor of the ith agent. The set of neighbors of the ith agent is denoted by $\mathcal{N}_i = \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$. All elements of adjacency matrix are nonnegative. For $i, j \in \mathcal{V}$, $j \in \mathcal{N}_i \Leftrightarrow a_{ij} > 0$, and assume that $a_{ii} = 0$, $i \in \mathcal{V}$. A directed path in a digraph $\mathcal{G}$ is an ordered sequence $v_1, v_2, \ldots, v_k$ of agents such that any ordered pair of vertices appearing consecutively in the sequence is an edge of the digraph, that is, $(v_i, v_{i+1}) \in \mathcal{E}$, for any $i = 1, 2, \ldots, k-1$. A directed tree is a digraph, where there exists an agent, called the root, such that any other agent of the digraph can be reached by one and only one path starting at the root. $\mathcal{T}_{\mathcal{G}} = \{\mathcal{V}_{\mathcal{T}}, \mathcal{E}_{\mathcal{T}}\}$ is a directed spanning tree of $\mathcal{G}$, if $\mathcal{T}_{\mathcal{G}}$ is a directed tree and $\mathcal{V}_{\mathcal{T}} = \mathcal{V}$. The Laplacian matrix $L(\mathcal{G}) = (l_{ij})_{N\times N}$ of $\mathcal{G}$ is defined as

$$l_{ij} = \begin{cases} -a_{ij}, & i \neq j, \\ \displaystyle\sum_{k=1, k\neq i}^{N} a_{ik}, & i = j. \end{cases} \tag{1}$$

Given a matrix $\mathbf{P} = (p_{ij}) \in \mathbb{R}^{N\times N}$, the digraph (without self-link) of $\mathbf{P}$ denotes by $\mathcal{G}(\mathbf{P})$, which is the digraph with node set $\mathcal{V} = \{1, 2, \ldots, N\}$ such that there is an edge in $\mathcal{G}(\mathbf{P})$ from j to i if and only if $p_{ij} \neq 0$. The matrix A is nonnegative, that is, $A \geq 0$, if all element of A is non-negative. The matrix $A, B \in \mathbb{R}^{N\times N}$, $A \geq B$ denote $A - B \geq 0$. The nonnegative matrix A is row stochastic if all of its row sum are equal to 1. The row stochastic matrix $A \in \mathbb{R}^{N\times N}$ is called indecomposable and aperiodic (SIA) if $\lim_{k\to\infty} A^k = \mathbf{1}_N y^T$, where y is some $N \times 1$ column vector.

Consider that a multiagent system consists of N identical agents indexed by $1, 2, \ldots, N$, which is described by

$$\dot{x}_i(t) = v_i(t), \qquad \dot{v}_i(t) = u_i(t), \tag{2}$$

where $i = 1, 2, \ldots, N$, $x_i(t) \in \mathbb{R}$, $v_i(t) \in \mathbb{R}$ are the position and velocity states of the agent i, respectively. $u_i(t) \in \mathbb{R}^n$ is a control input for $i = 1, 2, \ldots, N$. The static leader for the system (2) is a static agent represented by $x_0(t) = x_0$, where $x_0 \in \mathbb{R}$. The edges between the agents and the leader is unidirectional; namely, there are only partial agents that can obtain information from the leader. It is also assumed that each agent can only obtain information from other agents or the leader at sampling times.

This paper focuses on the problem of designing $u_i(t)$, $i = 1, 2, \ldots, N$ based on sampling information to make all N agents converge to a static leader.

Definition 1. The leader-following consensus of the multiagent system (2) with static leader is said to be achieved if

$$\lim_{t\to+\infty} x_i(t) = x_0, \quad \lim_{t\to+\infty} v_i(t) = 0, \quad i \in \mathcal{V}, \tag{3}$$

for any initial state.

3. Leader-Following Consensus under Fixed Topology

In this section, the leader-following consensus problem under fixed topology is considered. The interaction between agents in this part is described by a fixed digraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$, and the connections between agents and leader are described by $b_i \in \mathbb{R}$, $b_i > 0$ if and only if the agent i can obtain information from the leader, otherwise, $b_i = 0$.

In order to achieve the leader-following consensus of the multiagent system (2) with sampled information under fixed topology, the impulsive control for the agent i, is designed as

$$u_i(t) = -\sum_{k=1}^{+\infty} p_1 \left[\left(\sum_{j\in N_i} l_{ij}\left(x_j(t) - x_i(t)\right) + b_i\left(x_i(t) - x_0\right) \right) + p_2 v_i(t) \right] \delta(t - t_k), \tag{4}$$

where $i \in \mathcal{V}$, the sampling time sequence $\{t_k\}_{k=1}^{\infty}$ satisfies $t_{k+1} - t_k = h$ ($h \in \mathbb{R}$ is sampled period) and $\lim_{k\to\infty} t_k = \infty$, p_1, $p_2 > 0$ are the control gain to be determined, and $\delta(\cdot)$ is the Dirac impulsive function.

Equivalently, the multiagent system (2) with impulsive controller (4) can be rewritten as follows:

$$\begin{gathered} \dot{x}_i(t) = v_i(t), \quad \dot{v}_i(t) = 0, \quad t \in (t_k, t_{k+1}], \\ \Delta v_i(t_k) = -p_1\left(\sum_{j\in\mathcal{V}} l_{ij} x_j(t_k) + b_i\left(x_i(t_k) - x_0\right)\right) - p_2 v_i(t_k), \end{gathered} \tag{5}$$

where $\Delta v_i(t_k) = v_i(t_k^+) - v_i(t_k)$, $v_i(t_k^+) = \lim_{t\to t_k^+} v_i(t_k)$. For simplicity, it is assume that $v_i(t)$ is left continuous at t_k.

Remark 2. From (5), the control input of each agent only uses the information from its neighbors at sampling instants and are only applied at sampling instants. This is quite different from the previously mentioned works, where the control inputs are applied continuously. The velocity of the agent is instantaneously changed at sampling times. This is feasible when the operating time of the impulsive controller is much smaller than the sampled period.

Lemma 3. *The multiagent system* (2) *with impulsive control* (4) *achieves leader-following consensus asymptotically if and only if* $\rho(P) < 1$, *where*

$$\begin{gathered} P = \begin{pmatrix} I_N & hI_N \\ -p_1(L+B) & (1-p_2)I_N - p_1 h(L+B) \end{pmatrix}, \\ B = \begin{pmatrix} b_1 & & & \\ & b_2 & & \\ & & \ddots & \\ & & & b_N \end{pmatrix}. \end{gathered} \tag{6}$$

Proof. Let $\hat{x}_i(t) = x_i(t) - x_0(t)$, for $i \in \mathcal{V}$ and note that $\sum_{j\in\mathcal{V}} l_{ij} x_0 = 0$, system (5) can be rewritten as follows:

$$\begin{gathered} \dot{\hat{x}}_i(t) = v_i(t), \quad \dot{v}_i(t) = 0, \quad t \in (t_k, t_{k+1}], \\ \Delta v_i(t_k) = -p_1\left(\sum_{j\in\mathcal{V}} l_{ij}\hat{x}_j(t_k) + b_i\hat{x}_i(t_k)\right) - p_2 v_i. \end{gathered} \tag{7}$$

From (7), one has

$$\begin{gathered} \hat{x}_i(t_{k+1}) = \hat{x}_i(t_k) + h v_i(t_k^+), \\ v_i(t_{k+1}) = v_i(t_k^+), \\ v_i(t_{k+1}^+) = (1-p_2) v_i(t_{k+1}) \\ - p_1\left(\sum_{j\in\mathcal{V}} l_{ij}\hat{x}_j(t_{k+1}) + b_i\hat{x}_i(t_{k+1})\right). \end{gathered} \tag{8}$$

From (8), one has

$$\begin{aligned} v_i(t_{k+1}^+) &= (1-p_2) v_i(t_k^+) \\ &\quad - p_1 h \sum_{j\in\mathcal{V}} l_{ij} v_j(t_k^+) - p_1 b_i h v_i(t_k^+) \\ &\quad - p_1\left(\sum_{j\in\mathcal{V}} l_{ij}\left(\hat{x}_j(t_k)\right) + b_i\hat{x}_i(t_k)\right). \end{aligned} \tag{9}$$

Then, the evolution of $\hat{x}_i(t_k)$, $v_i(t_k)$ under impulsive control (4) can be described as follows:

$$\begin{aligned} \hat{x}_i(t_{k+1}) &= \hat{x}_i(t_k) + h v_i(t_k^+), \\ v_i(t_{k+1}^+) &= (1-p_2) v_i(t_k^+) \\ &\quad - p_1 h \sum_{j\in\mathcal{V}} l_{ij} v_j(t_k^+) - p_1 b_i h v_i(t_k^+) \\ &\quad - p_1\left(\sum_{j\in\mathcal{V}} l_{ij}\left(\hat{x}_j(t_k)\right) + b_i\hat{x}_i(t_k)\right). \end{aligned} \tag{10}$$

Let $\hat{x}(k) = (\hat{x}_1(t_k), \ldots, \hat{x}_N(t_k))^T$ and $v(k) = (v_1(t_k^+), \ldots, v_N(t_k^+))^T$. Then, the multiagent system (2) achieves leader-following consensus, if and only if $\lim_{k\to+\infty}\hat{x}(k) = 0$, $\lim_{k\to+\infty} v(k) = 0$.

Equivalently, (10) can be rewritten as follows:

$$\begin{pmatrix} \hat{x}(k+1) \\ v(k+1) \end{pmatrix} = P \times \begin{pmatrix} \hat{x}(k) \\ v(k) \end{pmatrix}. \tag{11}$$

Therefore, it is easy to obtain the result by the stability theory of discrete-time systems. □

The following lemmas and definition are needed for the subsequent development.

Lemma 4 (bilinear transformation theorem [36]). *Polynomial $R(z)$ (of degree d) is Schur stable if and only if $Q(z)$ is Hurwitz stable, where*

$$R(z) = (z-1)^d Q\left(\frac{z+1}{z-1}\right). \tag{12}$$

For complex polynomial $R(z)$, let

$$R(i\omega) = m(\omega) + in(\omega), \tag{13}$$

where $m(\omega)$, $n(\omega) \in \mathbb{R}$ and i is the imaginary unit.

Lemma 5 (see [37, 38]). *The complex polynomial $R(z) = z^2 + az + b$, where $\alpha \in \mathbb{C}$ and $\beta \in \mathbb{C}$, is Hurwitz stable if and only if* $\mathrm{Re}(a) > 0$ *and* $\mathrm{Re}(a)\,\mathrm{Im}(a)\,\mathrm{Im}(b) + \mathrm{Re}^2(a)\mathrm{Re}^2(b) - \mathrm{Im}^2(b) > 0$.

Next theorem will show what kind of interaction topology can reach leader-following consensus and how to determine the control gains p_1, p_2 and sampling period h.

Theorem 6. *The multiagent system* (2) *with impulsive control* (4) *under fixed topology achieves the leader-following consensus asymptotically if and only if*

$$h < \min_{1<i<N} \frac{2p_2^2(2-p_2)\,\mathrm{Re}(\lambda_i)}{\mathrm{Im}^2(\lambda_i)\,p_1(p_2-2)^2 + \mathrm{Re}^2(\lambda_i)\,p_1 p_2^2}, \tag{14}$$

where λ_i, $i = 1, 2, \ldots, N$ are the eigenvalues of $L + B$.

Proof. Let the γ be an eigenvalue of matrix P. Then,

$$\begin{aligned}
&\det(\gamma I_{2N} - P) \\
&\quad = \det\begin{pmatrix} (\gamma-1)I_N & -hI_N \\ p_1(L+B) & \gamma I_N - (1-p_2)I_N + p_1 h(L+B) \end{pmatrix} \\
&\quad = \prod_{i=1}^{N}\left(\gamma^2 + \gamma(p_2 - 2 + p_1 h\lambda_i) + 1 - p_2\right).
\end{aligned} \tag{15}$$

Let

$$Q_i(\gamma) = \gamma^2 + \gamma(p_2 - 2 + p_1 h\lambda_i) + 1 - p_2, \quad i \in \mathscr{V}. \tag{16}$$

Then, we only need to prove that polynomials $Q_i(\gamma)$ for $i \in \mathscr{V}$ are Schur stable.

Let

$$\begin{aligned}
R_i(\sigma) &= (\sigma-1)^2 Q_i\left(\frac{\sigma+1}{\sigma-1}\right) \\
&= p_1 h\lambda_i\sigma^2 + 2p_2\sigma + 4 - 2p_2 - p_1 h\lambda_i.
\end{aligned} \tag{17}$$

Let

$$R'(\sigma) = \sigma^2 + \frac{2p_2}{p_1 h}\lambda_i'\sigma + \left(\frac{4}{p_1 h} - \frac{2p_2}{p_1 h}\right)\lambda_i' - 1, \tag{18}$$

where $\lambda_i' = 1/\lambda_i'$. Then, according to Lemma 4, polynomials $R_i'(\sigma)$, for $i = 1, 2, 3, \ldots, N$, are Hurwitz stable if and only if polynomials $Q_i(\gamma)$ for $i = 1, 2, 3, \ldots, N$ are Schur stable.

It can be proved by Lemma 5 that $R_i(\sigma)$ is Hurwitz stable if and only if (14) is satisfied. Therefore, $\rho(P) < 1$ if and only if (14) is satisfied. The proof is thus completed. □

Remark 7. It can be observed from the inequality (14) that the real and imaginary part of the eigenvalues of $L + B$, the sampling period h, and two control gains p_1 and p_2 play important roles in achieving consensus. $p_2 < 2$ and $\mathrm{Re}(\lambda_i(L+B)) > 0$, for $i \in \mathscr{V}$, are necessary conditions for leader-following consensus. It is easy to see that the critical value of h increases as p_1 decreases.

Remark 8. Let $\widetilde{\mathscr{G}} = \{\widetilde{\mathscr{V}}, \widetilde{\mathscr{E}}, \widetilde{\mathscr{A}}\}$ with $\widetilde{\mathscr{V}} = \{0, 1, 2, \ldots, N\}$, and the Laplace matrix is

$$\widetilde{L} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ -b_1 & l_{11}+b_1 & l_{12} & \cdots & l_{1N} \\ -b_2 & l_{21} & l_{22}+b_2 & \cdots & l_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -b_N & l_{N1} & l_{N2} & \cdots & l_{NN}+b_N \end{pmatrix}. \tag{19}$$

Note that

$$E^{-1}\widetilde{L}E = \begin{pmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & L+B \end{pmatrix}, \tag{20}$$

where

$$E = \begin{pmatrix} 1 & \mathbf{0}_N^T \\ \mathbf{1}_N & I_N \end{pmatrix} \tag{21}$$

is an invertible matrix. $\mathrm{Re}(\lambda_i) > 0$, for $i \in \mathscr{V}$ imply that $\widetilde{L}$ has a simple eigenvalue 0, and all the other eigenvalues have positive real parts. This implies that the graph $\widetilde{\mathscr{G}}$ contains a spanning tree. The root of the spanning tree is the leader.

Remark 9. How to choose a suitable control gain p_1 and p_2 when the sampling period h is given. According to Theorem 6, $p_2 < 2$ is a necessary condition for consensus. Therefore, one can choose p_2 from $(0, 2]$, and then compute

$$\Theta = \min_{i\in\mathscr{V}} \frac{2p_2^2(2-p_2)\,\mathrm{Re}(\lambda_i)}{(\mathrm{Im}(\lambda_i)(p_2-2))^2 h + hp_2^2\mathrm{Re}^2(\lambda_i)}. \tag{22}$$

Then, one can choose p_1 from $(0, \Theta)$.

4. Leader-Following Consensus under Switching Topology

In this section, the leader-following consensus under switching topology is considered. The interaction between agents at sampling time t_k is described by time-varying digraph $\mathscr{G}(t) = \{\mathscr{V}, \mathscr{E}(t), \mathscr{A}(t)\}$, where $\mathscr{A}(t) = (a_{ij}(t))_{N\times N}$ and the connections between agents and leader at time t are described by $b_i(t)$, $b_i(t) > 0$ if and only if the agent i can obtain information from the leader at time t; otherwise, $b_i(t) = 0$.

In order to achieve leader-following consensus under switching topology, the impulsive control input is designed as

$$\begin{aligned}
u_i(t) = -\sum_{k=1}^{+\infty} p_1 \Bigg[\Bigg(&\sum_{j\in N_i} l_{ij}(t)\left(x_j(t) - x_i(t)\right) \\
&+ b_i(t)\left(x_i(t) - x_0\right)\Bigg) + p_2 v_i \Bigg] \delta(t - t_k),
\end{aligned} \tag{23}$$

where $i = 1, 2, \ldots, N$. Let $\widetilde{\mathcal{G}}(t) = \{\widetilde{\mathcal{V}}, \widetilde{\mathcal{E}}(t), \widetilde{\mathcal{A}}(t)\}$ with $\widetilde{\mathcal{V}}(t) = \{0, 1, 2, \ldots, N\}$ and

$$\widetilde{\mathcal{A}}(t) = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ b_1(t) & 0 & a_{12}(t) & \cdots & a_{1N}(t) \\ b_2(t) & a_{21}(t) & 0 & \cdots & a_{2N}(t) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_N(t) & a_{N1}(t) & a_{N2}(t) & \cdots & 0 \end{pmatrix}. \quad (24)$$

Let $\widetilde{L}(t) = (\widetilde{l}_{ij})_{(N+1)\times(N+1)}$ denotes the Laplace matrix of $\widetilde{\mathcal{G}}$. Equivalently, the multiagent system (2) with the impulsive controller (23) can be rewritten as follows:

$$\begin{aligned} &\dot{x}_i(t) = v_i(t), \quad \dot{v}_i(t) = 0, \quad t \in (t_k, t_{k+1}], \\ &\Delta v_i(t_k) = -p_1 \sum_{j\in\mathcal{V}} \widetilde{l}_{ij}(t) x_j(t_k) - p_2 v_i(t_k), \end{aligned} \quad (25)$$

where $i = 0, 1, \ldots, N$.

Remark 10. Note that the communication among agents only occurs at sampling times. This implies that interation graph does not contain any edges $\mathcal{G}(t) = 0$ where $t \neq t_k$.

Similar to the discussion in Section 4, one has

$$\begin{aligned} x_i(t_{k+1}) &= x_i(t_k) + h v_i(t_k^+), \\ v_i(t_{k+1}^+) &= (1 - p_2) v_i(t_k^+) - p_1 \sum_{j=0}^{N} \widetilde{l}_{ij}(t_k) x_j(t_k) \\ &\quad - p_1 h \sum_{j=0}^{N} \widetilde{l}_{ij}(t_k) v_j(t_k^+). \end{aligned} \quad (26)$$

Let $\widetilde{x}_i(k) = x_i(t_k)$, $\widetilde{v}_i(k) = x_i(t_k) + \alpha v_i(t_k^+)$, where $\alpha = 2h/p_2$.

It is easy to know that the network (2) achieves leader-following consensus, if $\widetilde{x}_i(k) \to \beta$ and $\widetilde{v}_i(k) \to \beta$, for some $\beta \in \mathbb{R}$, $i \in \widetilde{\mathcal{V}}$.

From (26), one has

$$\begin{aligned} \widetilde{x}_i(k+1) &= \left(1 - \frac{p_2}{2}\right) \widetilde{x}_i(k) + \frac{p_2}{2} \widetilde{v}_i(k), \\ \widetilde{v}_i(k+1) &= \frac{p_2}{2} \widetilde{x}_i(k) + \left(1 - \frac{p_2}{2}\right) \widetilde{v}_i(k) \\ &\quad - \left(\frac{2}{p_2} - 1\right) p_1 h \sum_{j=0}^{N} l_{ij}\left(\widetilde{x}_j(t_k)\right) \\ &\quad - p_1 h \sum_{j=0}^{N} l_{ij} \widetilde{v}_j(k). \end{aligned} \quad (27)$$

Let $\widetilde{x}(k) = (\widetilde{x}_0^T(k), \widetilde{x}_1^T(k), \widetilde{x}_2^T(k), \ldots, \widetilde{x}_N^T(k))^T$ and $\widetilde{v}(k) = (\widetilde{v}_0^T(k), \widetilde{v}_1^T(k), \widetilde{v}_2^T(k), \ldots, \widetilde{v}_N^T(k))^T$; then,

$$\begin{pmatrix} \widetilde{x}(k+1) \\ \widetilde{v}(k+1) \end{pmatrix} = \mathbf{P}(k) \times \begin{pmatrix} \widetilde{x}(k) \\ \widetilde{v}(k) \end{pmatrix}, \quad (28)$$

where

$$\mathbf{P}(k) = \begin{pmatrix} \left(1 - \frac{p_2}{2}\right) I & \frac{p_2}{2} I \\ \frac{p_2}{2} I - \left(\frac{2}{p_2} - 1\right) p_1 h \widetilde{L}(k) & \left(1 - \frac{p_2}{2}\right) I - p_1 h \widetilde{L}(k) \end{pmatrix}. \quad (29)$$

Before moving on, the following lemmas are needed.

Lemma 11 (see [16]). *Let $m \geq 2$ be a positive integer, and let $P_1, P_2, \ldots, P_m$ be non-negative $N \times N$ matrices with positive diagonal entries; then, $P_1 P_2 \cdots P_m \geq \varepsilon(P_1 + P_2 + \cdots + P_m)$, where $\varepsilon > 0$ can be specified from matrices P_i, $i = 1, 2, \ldots, m$.*

Lemma 12 (see [39]). *Let $P_1, P_2, \ldots, P_k \in \mathbb{R}^{N\times N}$ be a finite set of SIA matrices with the property that for each sequence $P_{i_1}, P_{i_2}, \ldots, P_{i_j}$ of positive length, the matrix product $P_{i_1} P_{i_2} \cdots P_{i_j}$ is SIA. Then, for each infinite sequence $P_{i_1}, P_{i_2}, \ldots, P_{i_j}$, there exists a column vector y such that*

$$\lim_{j\to\infty} P_{i_1} \times P_{i_2} \times \cdots P_{i_j} = \mathbf{1}_N y^T. \quad (30)$$

Lemma 13 (see [40]). *Suppose that $\mathbf{P} \in \mathbb{R}^{N\times N}$ is a row stochastic matrix with positive diagonal elements. If the digraph $\mathcal{G}(\mathbf{P})$ has a directed spanning tree, then $\mathbf{P}$ is SIA.*

Lemma 14. *Let*

$$\mathbf{P} = \begin{pmatrix} (1-\alpha) I & \alpha I \\ \beta I - \mu_1 L & (1-\beta) I - \mu_2 L \end{pmatrix} \quad (31)$$

be non-negative matrix, where $\alpha, \mu_1, \mu_2 > 0$. If L is a Laplace matrix of a digraph $\mathcal{G}$, which has a directed spanning tree, then $\mathbf{P}$ is a row stochastic matrix and the digraph of $\mathbf{P}$ contains a directed spanning tree.

Proof. It is easy to check the non-negative matrix $\mathbf{P}\mathbf{1}_{2N} = \mathbf{1}_{2N}$. Then, $\mathbf{P}$ is a row stochastic matrix. Let $\mathcal{G}(\mathbf{P})$ denote the digraph of $\mathbf{P}$. Then, the Laplace matrix of $\mathcal{G}(\mathbf{P})$ is

$$L(\mathbf{P}) = \begin{pmatrix} \alpha I & -\alpha I \\ -\beta I + \mu_1 L & \beta I + \mu_2 L \end{pmatrix}. \quad (32)$$

Let γ_i, $i = 1, 2, \ldots, N$ denote the eigenvalues of L.

Let λ be an eigenvalue of matrix $\mathbf{P}$; then, one has

$$\begin{aligned} &\det \begin{pmatrix} \lambda I - \alpha I & \alpha I \\ \beta I - \mu_1 L & \lambda I - \beta I - \mu_2 L \end{pmatrix} \\ &\quad = \prod_{i=1}^{N} \left((\lambda - \alpha)(\lambda - \beta - \mu_2 \gamma_i) - \alpha(\beta - \mu_1 \gamma_i)\right). \end{aligned} \quad (33)$$

Let $Q(\lambda) = (\lambda - \alpha)(\lambda - \beta - \mu_2\gamma_i) - \alpha(\beta - \mu_1\gamma_i)$. Then,

$$Q(0) = \alpha\mu_1\gamma_i + \alpha\mu_2\gamma_i. \quad (34)$$

Therefore, from (34), $\lambda = 0$ only if $\gamma_i = 0$.

When $\gamma_i = 0$,

$$Q(\lambda) = \lambda^2 - \lambda(\alpha + \beta). \tag{35}$$

Thus, when $\gamma_i(L) = 0$, the solutions of $Q(\lambda) = 0$ are $\lambda = 0$ and $\lambda = \alpha + \beta$. On the other hand, if $\mathcal{G}$ contains a spanning tree, L only has one simple eigenvalue equal to zero. Therefore, $L(\mathbf{P})$ only has one simple eigenvalue equal to zero, which implies that the digraph of $\mathbf{P}$ has a spanning tree. The proof is completed. □

Theorem 15. *If there exists a positive integer l, the union of $\widetilde{\mathcal{G}}(t_k)$ across $k \in [k_0, k_0 + l]$ contains a directed spanning tree, for any non-negative integer k_0, and*

$$\begin{aligned} h &< \frac{{p_2}^2}{2p_1(2-p_2)\Delta}, \quad p_2 \le 1, \\ h &< \frac{2-p_2}{2p_1\Delta}, \quad 1 < p_2 < 2, \end{aligned} \tag{36}$$

where $\Delta = \max_{i\in\mathcal{V}, k\in\mathbb{N}}\{l_{ii}(k) + b_i(k)\}$; then, the multiagent system (2) *achieves the leader-following consensus.*

Proof. Let

$$\mathbf{P}(k) = \begin{pmatrix} P_{11}(k) & P_{12}(k) \\ P_{21}(k) & P_{22}(k) \end{pmatrix}, \tag{37}$$

where $\mathbf{P}(k)$ is defined in (29). From (1), one has $l_{ij} < 0$, for $i \neq j$. Then, the following statements are satisfied:

(i) $P_{11}(k)$ is nonnegative if and only if $p_2 < 2$;

(ii) $P_{12}(k)$ is nonnegative if and only if $p_2 > 0$;

(iii) $P_{21}(k)$ is nonnegative if and only if

$$\frac{p_2}{2} - \left(\frac{2}{p_2} - 1\right) p_1 h (l_{ii}(k) + b_i(k)) > 0, \quad \text{for } i \in \mathcal{V}; \tag{38}$$

(iv) $P_{22}(k)$ is nonnegative if and only if

$$\left(1 - \frac{p_2}{2}\right) - p_1 h (l_{ii}(k) + b_i(k)) > 0, \quad \text{for } i \in \mathcal{V}. \tag{39}$$

If $0 < p_2 < 2$, then $2/p_2 - 1 > 0$ and $1 - p_2/2 > 0$. Note that $l_{ij} + b_i(k) \ge 0$ and $p_1 > 0$. Then, the following four statements are satisfied when $p_2 < 2$.

(i) If $l_{ij} + b_i(k) = 0$, then we have

$$\begin{aligned} &\frac{p_2}{2} - \left(\frac{2}{p_2} - 1\right) p_1 h (l_{ii}(k) + b_i(k)) > 0, \\ &\left(1 - \frac{p_2}{2}\right) - p_1 h (l_{ii}(k) + b_i(k)) > 0, \end{aligned} \tag{40}$$

(ii) If $l_{ij} + b_i(k) > 0$, and

$$h < \frac{p_2/2}{(2/p_2 - 1)\, p_1 (l_{ii}(k) + b_i(k))}, \tag{41}$$

then we have

$$\frac{p_2}{2} - \left(\frac{2}{p_2} - 1\right) p_1 h (l_{ii}(k) + b_i(k)) > 0, \tag{42}$$

(iii) If $l_{ij} + b_i(k) > 0$, and

$$h < \frac{1 - p_2/2}{p_1 (l_{ii}(k) + b_i(k))}, \tag{43}$$

then we have

$$\left(1 - \frac{p_2}{2}\right) - p_1 h (l_{ii}(k) + b_i(k)) > 0. \tag{44}$$

According to the previous discussion, (38) is satisfied, if $p_2 < 2$, and

$$h < \frac{p_2/2}{(2/p_2 - 1)\, p_1 \Delta}. \tag{45}$$

Equation (39) is satisfied, if $p_2 < 2$, and

$$h < \frac{1 - p_2/2}{p_1 \Delta}. \tag{46}$$

Note that

$$\frac{p_2/2}{(2/p_2 - 1)\, p_1 \Delta} - \frac{1 - p_2/2}{p_1 \Delta} = \frac{2 - 2/p_2}{(2/p_2 - 1)\, p_1 \Delta}. \tag{47}$$

Then, (38) and (39) are satisfied if (36) holds. This implies that $\mathbf{P}_{11}(k)$, $\mathbf{P}_{12}(k)$, $\mathbf{P}_{21}(k)$, and $\mathbf{P}_{22}(k)$ are nonnegative. Then, $\mathbf{P}(k)$ is also nonnegative. Note that

$$\mathbf{P}(k)\, \mathbf{1}_{2N+2} = \begin{pmatrix} \mathbf{1}_{N+1} \\ \mathbf{1}_{N+1} - \dfrac{2}{p_2} p_1 h \widetilde{L}(k)\, \mathbf{1}_{N+1} \end{pmatrix}, \tag{48}$$

and $\widetilde{L}(k)\mathbf{1}_{N+1} = \mathbf{0}$. Then, $\mathbf{P}(k)\mathbf{1}_{2N+2} = \mathbf{1}_{2N+2}$, $\mathbf{P}(k)$ is a row stochastic matrix. Then, $\sum_{k=k_0}^{k_0+l} \mathbf{P}(k)$ and $\prod_{k=k_0}^{k_0+l} \mathbf{P}(k)$ are also a row stochastic matrix.

Note that

$$\begin{aligned} &\sum_{k=k_0}^{k_0+l} \mathbf{P}(k) \\ &\quad = (l+1) \\ &\quad \times \begin{pmatrix} \left(1 - \dfrac{p_2}{2}\right) I & \dfrac{p_2}{2} I \\ \dfrac{p_2}{2} I - \left(\dfrac{2}{p_2} - 1\right) p_1 h \mathbf{S}_{k_0}^l & \left(1 - \dfrac{p_2}{2}\right) I - p_1 h \mathbf{S}_{k_0}^l \end{pmatrix}, \end{aligned} \tag{49}$$

where $\mathbf{S}_{k_0}^l = (1/(l+1)) \sum_{k=k_0}^{k_0+l} \widetilde{L}(k)$.

The union of $\mathcal{G}(t_k)$ across $k \in [k_0, k_0 + l]$, for any nonnegative integer k_0 contains a directed spanning tree. This implies that the digraph with the Laplace matrix $\mathbf{S}_{k_0}^l$ also contains a directed spanning tree.

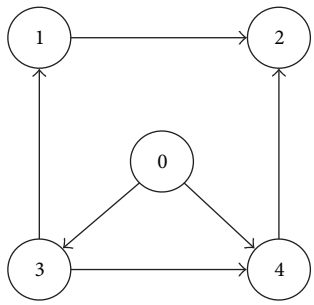


FIGURE 1: Fixed topology.

By Lemma 14, from (49), the digraph of $\sum_{k=k_0}^{k_0+l} \mathbf{P}(k)$ contains a spanning tree.

According to Lemma 11, one has

$$\prod_{i=k_0}^{k_0+l} \mathbf{P}(i) \geq \varepsilon \sum_{i=k_0}^{k_0+l} \mathbf{P}(i), \tag{50}$$

for some ε. This implies that the digraph of $\prod_{k=k_0}^{k_0+l} \mathbf{P}(k)$ also contains a spanning tree. It follows from Lemma 13 that $\prod_{k=k_0}^{k_0+l} \mathbf{P}(k)$ is SIA. By Lemma 12,

$$\lim_{k\to\infty} \mathbf{P}(k)\,\mathbf{P}(k-1)\cdots\mathbf{P}(0)\,x(0) = \mathbf{1}_{2(N+1)} y^T x(0). \tag{51}$$

The proof is thus completed. □

Remark 16. In this remark, we also show how to choose a suitable control gain p_1 and p_2 when the sampling period h is given. According to Theorem 15, $p_2 < 2$ is also required. Similar to Remark 9, one can choose p_2 from $(0, 2]$, and then compute

$$\Theta = \begin{cases} \dfrac{p_2^2}{2h(2-p_2)\Delta}, & p_2 \leq 1 \\ \dfrac{2-p_2}{2h\Delta}, & 1 < p_2 < 2. \end{cases} \tag{52}$$

Then, one can choose p_1 from $(0, \Theta)$.

5. Illustrative Examples

In this section, two illustrative numerical examples will be given to demonstrate the correctness of theoretical analysis.

5.1. Fix Topology. The communication topology is described as in Figure 1. The Laplacian matrix L and matrix B are given as follows:

$$L = \begin{pmatrix} 1 & 0 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}, \qquad B = \operatorname{diag}(0, 0, 1, 1). \tag{53}$$

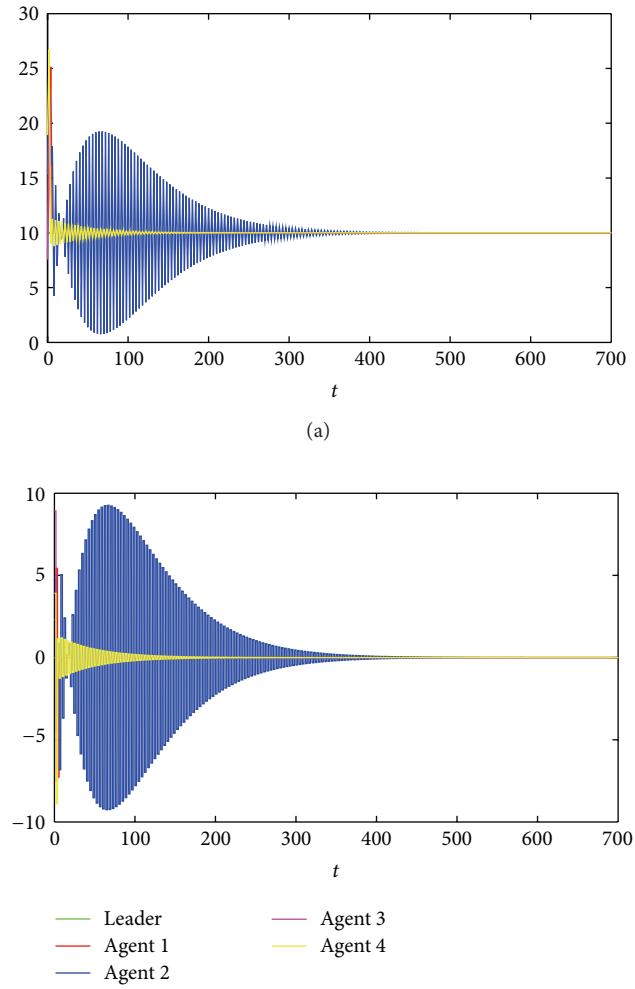


FIGURE 2: Trajectory of the multiagent system (2) under fixed topology, when $p_1 = 0.49$. Evolution of (a) x_i, and (b) v_i.

The eigenvalues of $L + B$ are $\lambda_1(L + B) = \lambda_2(L + B) = 1$, $\lambda_3(L + B) = \lambda_4(L + B) = 2$. Let $p_2 = 1$, $h = 2$; according to Theorem 6, the network can achieve leader-following consensus, if and only if

$$p_1 < \min_{i\in\mathcal{V}} \frac{2p_2^2(2-p_2)\operatorname{Re}(\lambda_i)}{(\operatorname{Im}(\lambda_i)(p_2-2))^2 h + h p_2^2 \operatorname{Re}^2(\lambda_i)} = 0.5. \tag{54}$$

Figure 2 shows that the leader-following consensus can be achieved when $p_1 = 0.49$. But it cannot be achieved when $p_1 = 0.51$ (as shown in Figure 3).

5.2. Switching Topology. In this subsection, the network topology switches from a set $\{\widetilde{\mathcal{G}}_1, \widetilde{\mathcal{G}}_2, \widetilde{\mathcal{G}}_3, \widetilde{\mathcal{G}}_4\}$ as shown in

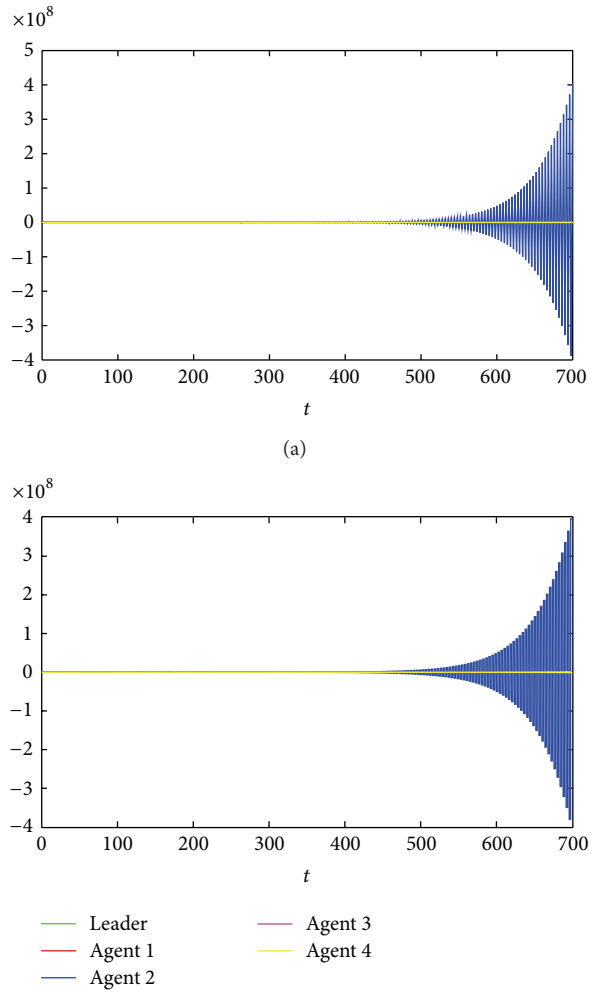


FIGURE 3: Trajectory of the multiagent system (2) under fixed topology, when $p_1 = 0.51$. Evolution of (a) x_i, and (b) v_i.

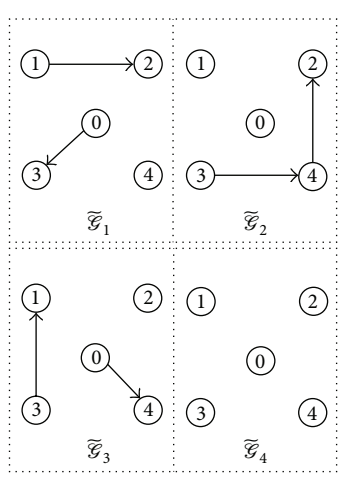


FIGURE 4: Switching topology: $\widetilde{\mathcal{G}}_i$, $i = 1, 2, 3, 4$.

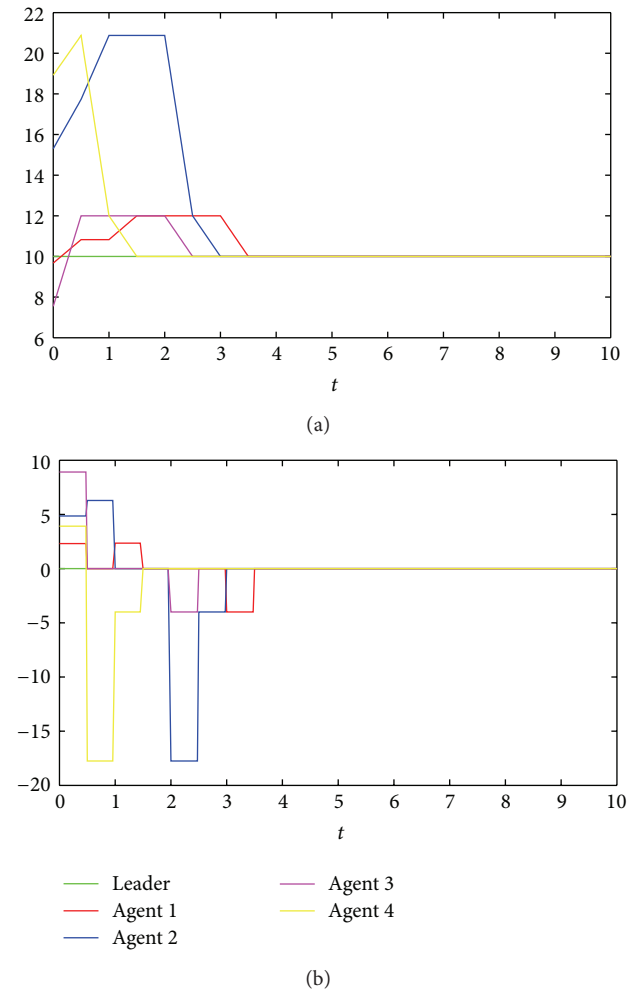


FIGURE 5: Trajectory of the multiagent system (2) under switched topology, when $p_1 = 0.95$. Evolution of (a) x_i, and (b) v_i.

Figure 4. The corresponding Laplacian matrices of $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_3$, $\mathcal{G}_4$ and matrices B_1, B_2, B_3, B_4 are

$$L_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad L_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix},$$
$$L_3 = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad L_4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{55}$$

$B_1 = \operatorname{diag}(0\ 0\ 1\ 0)$, $B_2 = \operatorname{diag}(0\ 0\ 0\ 0)$, $B_3 = \operatorname{diag}(0\ 0\ 0\ 1)$, and $B_4 = \operatorname{diag}(0\ 0\ 0\ 0)$. Assume that $\mathcal{G}(t_0) = \mathcal{G}_1$, $\mathcal{G}(t_1) = \mathcal{G}_2$, $\mathcal{G}(t_2) = \mathcal{G}_3$, $\mathcal{G}(t_3) = \mathcal{G}_4$, $\mathcal{G}(t_4) = \mathcal{G}_1, \ldots$ and $B(t_0) = B_1$, $B(t_1) = B_2$, $B(t_2) = B_3$, $B(t_3) = B_4$, $B(t_4) = B_1, \ldots$. Note that the union graph of $\widetilde{\mathcal{G}}_1$, $\widetilde{\mathcal{G}}_2$, $\widetilde{\mathcal{G}}_3$, and $\widetilde{\mathcal{G}}_4$, and $\max_{i\in\mathcal{N}, k\in\mathbb{N}}(l_{ii}(k) + b_i(k)) = 1$.

Let $p_2 = 1$, $h = 0.5$, according to Theorem 15, if

$$p_1 < \frac{{p_2}^2}{2h(2-p_2)\max_{i\in\mathcal{N},k\in\mathbb{N}}\{l_{ii}(k)+b_i(k)\}} = 1. \quad (56)$$

Figure 5 shows that the leader-following consensus can be achieved when $p_1 = 0.95$.

6. Conclusions

In this paper, the leader-following consensus problem of the multiagent system is considered. The impulsive control, which only needs sampled information and regulates the velocity of each agent at sampling times, is proposed for the leader-following consensus. Several new criteria are established for the leader-following consensus of the system under both fixed and switching topology. Illustrated examples have been given to show the effectiveness of the proposed impulsive control.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 61073026, 61170031, 61272069, and 61073025.

References

[1] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, Princeton Series in Applied Mathematics, Princeton University Press, Princeton, NJ, USA, 2009.

[2] Z.-H. Guan, Z.-W. Liu, G. Feng, and Y.-W. Wang, "Synchronization of complex dynamical networks with time-varying delays via impulsive distributed control," *IEEE Transactions on Circuits and Systems. I*, vol. 57, no. 8, pp. 2182–2195, 2010.

[3] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622–633, 2008.

[4] A. Kashyap, T. Başar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.

[5] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[6] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[7] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, 2008.

[8] Z. Lin, B. Francis, and M. Maggiore, "State agreement for continuous-time coupled nonlinear systems," *SIAM Journal on Control and Optimization*, vol. 46, no. 1, pp. 288–307, 2007.

[9] Q. Hui and W. M. Haddad, "Distributed nonlinear control algorithms for network consensus," *Automatica*, vol. 44, no. 9, pp. 2375–2381, 2008.

[10] Z.-W. Liu, Z.-H. Guan, T. Li, X.-H. Zhang, and J.-W. Xiao, "Quantized consensus of multi-agent systems via broadcast gossip algorithms," *Asian Journal of Control*, vol. 14, no. 6, pp. 1634–1642, 2012.

[11] T. Li and J.-F. Zhang, "Consensus conditions of multi-agent systems with time-varying topologies and stochastic communication noises," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2043–2057, 2010.

[12] P. Lin and Y. Jia, "Consensus of second-order discrete-time multi-agent systems with nonuniform time-delays and dynamically changing topologies," *Automatica*, vol. 45, no. 9, pp. 2154–2158, 2009.

[13] W. Yu, G. Chen, and M. Cao, "Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems," *Automatica*, vol. 46, no. 6, pp. 1089–1095, 2010.

[14] Y. Hong, J. Hu, and L. Gao, "Tracking control for multi-agent consensus with an active leader and variable topology," *Automatica*, vol. 42, no. 7, pp. 1177–1182, 2006.

[15] D. V. Dimarogonas, P. Tsiotras, and K. J. Kyriakopoulos, "Leader-follower cooperative attitude control of multiple rigid bodies," in *Proceedings of the American Control Conference*, pp. 801–806, Seattle, Wash, USA, 2008.

[16] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[17] Y. Cao and W. Ren, "Distributed coordinated tracking with reduced interaction via a variable structure approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 33–48, 2012.

[18] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 293–307, 2009.

[19] W. Zhu and D. Cheng, "Leader-following consensus of second-order agents with multiple time-varying delays," *Automatica*, vol. 46, no. 12, pp. 1994–1999, 2010.

[20] Z. Meng, W. Ren, Y. Cao, and Z. You, "Leaderless and leader-following consensus with communication and input delays under a directed network topology," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 41, no. 1, pp. 75–88, 2011.

[21] T. Li and J. Zhang, "Sampled-data based average consensus with measurement noises: convergence analysis and uncertainty principle," *Science in China F*, vol. 52, no. 11, pp. 2089–2103, 2009.

[22] Y. Cao and W. Ren, "Sampled-data discrete-time coordination algorithms for double-integrator dynamics under dynamic directed interaction," *International Journal of Control*, vol. 83, no. 3, pp. 506–515, 2010.

[23] Y. Zhang and Y.-P. Tian, "Consensus of data-sampled multi-agent systems with random communication delay and packet loss," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 939–943, 2010.

[24] H. Liu, G. Xie, and L. Wang, "Necessary and sufficient conditions for solving consensus problems of double-integrator dynamics via sampled control," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 15, pp. 1706–1722, 2010.

[25] Y. Gao and L. Wang, "Consensus of multiple double-integrator agents with intermittent measurement," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 10, pp. 1140–1155, 2010.

[26] X. Liu and K. Rohlf, "Impulsive control of a Lotka-Volterra system," *IMA Journal of Mathematical Control and Information*, vol. 15, no. 3, pp. 269–284, 1998.

[27] J. E. Prussing and J.-H. Chiu, "Optimal multiple-impulse time-fixed rendezvous between circular orbits," *Journal of Guidance, Control, and Dynamics*, vol. 9, no. 1, pp. 17–22, 1986.

[28] J. F. Eastham and K. J. Hastings, "Optimal impulse control of portfolios," *Mathematics of Operations Research*, vol. 13, no. 4, pp. 588–605, 1988.

[29] G. Zhang, Z. Liu, and Z. Ma, "Synchronization of complex dynamical networks via impulsive control," *Chaos*, vol. 17, no. 4, Article ID 043126, 2007.

[30] B. Liu, X. Liu, G. Chen, and H. Wang, "Robust impulsive synchronization of uncertain dynamical networks," *IEEE Transactions on Circuits and Systems. I*, vol. 52, no. 7, pp. 1431–1441, 2005.

[31] J. Lu, D. W. C. Ho, and J. Cao, "A unified synchronization criterion for impulsive dynamical networks," *Automatica*, vol. 46, no. 7, pp. 1215–1221, 2010.

[32] Q. Zhang, S. Chen, and C. Yu, "Impulsive consensus problem of second-order multi-agent systems with switching topologies," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 1, pp. 9–16, 2012.

[33] Y. Qian, X. Wu, J. Lu, and J.-A. Lu, "Second-order consensus of multi-agent systems with nonlinear dynamics via impulsive control," *Neurocomputing*, 2013.

[34] Z.-H. Guan, Z.-W. Liu, G. Feng, and M. Jian, "Impulsive consensus algorithms for second-order multi-agent networks with sampled information," *Automatica*, vol. 48, no. 7, pp. 1397–1404, 2012.

[35] Z.-W. Liu, Z.-H. Guan, X. Shen, and G. Feng, "Consensus of multi-agent networks with aperiodic sampled communication via impulsive algorithms using position-only measurements," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2639–2643, 2012.

[36] K. Ogata, *Discrete-Time Control Systems*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2nd edition, 1995.

[37] P. Parks and V. Hahn, *Stability Theory*, Prentice Hall, Englewood Cliffs, NJ, USA, 1993.

[38] W. Yu, W. X. Zheng, G. Chen, W. Ren, and J. Cao, "Second-order consensus in multi-agent dynamical systems with sampled position data," *Automatica*, vol. 47, no. 7, pp. 1496–1503, 2011.

[39] J. Wolfowitz, "Products of indecomposable, aperiodic, stochastic matrices," *Proceedings of the American Mathematical Society*, vol. 14, pp. 733–737, 1963.

[40] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 439362, 8 pages
http://dx.doi.org/10.1155/2013/439362

Research Article

User Demand Aware Grid Scheduling Model with Hierarchical Load Balancing

P. Suresh[1] and P. Balasubramanie[2]

[1] *Department of Information Technology, Kongu Engineering College, Perundurai, Erode, Tamil Nadu 638052, India*
[2] *Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, Tamil Nadu 638052, India*

Correspondence should be addressed to P. Suresh; sureshme@gmail.com

Received 15 March 2013; Accepted 19 May 2013

Academic Editor: Tingwen Huang

Copyright © 2013 P. Suresh and P. Balasubramanie. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Grid computing is a collection of computational and data resources, providing the means to support both computational intensive applications and data intensive applications. In order to improve the overall performance and efficient utilization of the resources, an efficient load balanced scheduling algorithm has to be implemented. The scheduling approach also needs to consider user demand to improve user satisfaction. This paper proposes a dynamic hierarchical load balancing approach which considers load of each resource and performs load balancing. It minimizes the response time of the jobs and improves the utilization of the resources in grid environment. By considering the user demand of the jobs, the scheduling algorithm also improves the user satisfaction. The experimental results show the improvement of the proposed load balancing method.

1. Introduction

Grid computing is a new computing paradigm which provides massive heterogeneous resources for solving very large applications in science and engineering [1]. It mainly deals with large scale computing and scientific problems. Grid is divided into two categories such as computational grid and data grid. A computational grid is a collection of hardware and software resources that provide an inexpensive access to all high end capabilities [2]. It supports sharing and coordinated use of computational resources which are distributed geographically [3]. Data grid focuses on controlling and accessing of massive source of data storage.

Task scheduling is a challenging task in grid environment because it is needed to exploit the capabilities of the grid resources. An efficient scheduling algorithm has to allocate the resources from various administrative domains to large number of tasks [4]. The load of resource needs to be considered to increase the performance, resource utilization, and throughput. Task scheduling involves mapping of "n" tasks to "m" resources. It is a NP-complete problem. Scheduling is done using an application called scheduler. The scheduler software enables an enterprise to schedule and in some cases monitor computer "batch" tasks [5].

Scheduling algorithms are categorized as application centric and resource centric. Scheduling algorithms adopting resource centric objectives aim to improve the performance of the system such as resource utilization and throughput. Scheduling algorithms adopting application centric objectives aim to minimize the waiting time and makespan.

Generally, load balancing algorithms are divided into two categories such as static and dynamic. Static load balancing algorithms are used to allocate the resources to tasks based on the current load. Dynamic load balancing algorithms allocate or reallocate resources at runtime based on the current load of the resources [6]. Load balancing algorithms aim to maximize the number of jobs completion, resource utilization, and minimizing the makespan. This section describes many task scheduling algorithms which are used to allocate resources to jobs.

Opportunistic load balancing (OLB) algorithm allocated the task to the next machine which is expected available in arbitrary order. This algorithm did not consider the expected execution time on the machine [7]. Minimum execution time

(MET) algorithm scheduled tasks by considering expected execution time regardless of availability of machine. It causes poor makespan and severe load imbalance [8]. Minimum completion time (MCT) algorithm allocated job in random order to the resource which has minimum completion time. But it leads to load imbalance [9].

Min-min algorithm calculated expected completion time for all the jobs in all machines. Task with minimum expected completion time is allocated to a machine which has minimum completion time. This algorithm failed to consider idleness of the machine [9, 10].

Max-min algorithm is similar to min-min algorithm. It finds the job with maximum expected completion time from set of unmapped jobs and allocates it to machine which has minimum completion time for the job [11].

Application demand aware algorithm considered the application's requirement and allocated the jobs to the appropriate resources. But results show poor resource utilization [12].

Suffrage heuristic algorithm calculated the suffrage value by finding the difference between its best minimum completion time and its second best minimum completion time. The task with highest suffrage value is selected and allocated to the machine with the minimum completion time. The assigned task is deleted, and the completion times for all the remaining tasks are updated. This process is repeated until all tasks are assigned to machines [13].

Prioritized user demand aware algorithm allocated jobs to the machines by considering user requirements of jobs. It results in more user satisfaction and better makespan, but data movement is not considered [14].

UDDA algorithm considered both user requirements and data requirements of the job. It aims to reduce the overall execution time of the jobs in the machines [15].

Yagoubi and Slimani developed a layered algorithm in which a dynamic load balancing approach is implemented in grid computing. Load balancing is performed based on a tree model. It is supported for heterogeneity and scalability [16].

Ludwig and Maollem proposed two new distributed swarm intelligence inspired load balancing algorithms. One algorithm is based on ant colony optimization, and the other algorithm is based on particle swarm optimization [17].

Cao et al. designed a load balancing algorithm in combination with intelligent agents and multiagent approaches. This algorithm assumed two levels such as local grid and global grid. In local grid, each agent is responsible for scheduling and load balancing, and in global grid, agent acts as service provider [18].

Alharbi et al. proposed a simple load balancing algorithm in which load is equally distributed. This algorithm used expected completion time to schedule the jobs [19].

Even though there are many scheduling algorithms that focus on user deadline and load balancing, there is a scope for scheduling algorithms that focuses on both of these factors. The main contribution of this paper is that a new scheduling algorithm has been proposed which focuses on both user deadline consideration and load balancing. This algorithm is compared with the min-min algorithm which serves as a benchmark scheduling algorithm and user demand aware (UDDA) scheduling algorithm which is proposed in our previous work [15]. The UDDA algorithm considers only the user deadline and do not consider the load balancing factor. The proposed algorithm considers both user satisfaction and load balancing.

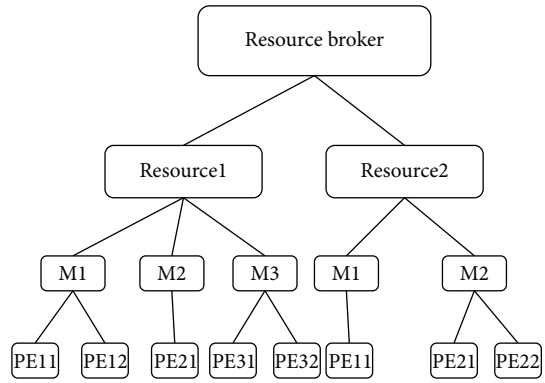


Figure 1: Proposed Architecture Model.

2. Materials and Methods

2.1. Problem Definition. Since grid environment consists of heterogeneous resources which are owned by multiple administrative domains, an efficient task scheduling is required for allocating tasks to machines efficiently. Since grid resources are dynamic, an effective load balancing algorithm is required to maintain load at resources. Most of the early developed scheduling algorithms do not consider the user deadline to complete the jobs. To improve the user satisfaction, the deadline of the jobs has to be considered.

The architecture followed in this work is given in Figure 1. Grid resource broker collects information about the resources such as capacity expressed in million instructions per second (MIPS), cost, and baud rate. Grid resource is an entity which is next to the grid resource broker in the hierarchy. It is responsible for maintaining load at the machines which are next to the resource in the hierarchy. Machines are responsible for scheduling and maintaining balanced load at the processing elements.

Users submit jobs in the form of gridlets to the machines. A gridlet is an entity which contains information such as length expressed in MI (million instructions), input file size (IF_i) expressed in MB, output file size (OF_i) expressed in MB, and the user who submits the job. This information is used to calculate expected execution time, data transfer time between the user and the resource which are considered to select the appropriate resources for the jobs.

The length of the gridlet is assumed to be calculated based on the number of instructions. The scheduling algorithm is applied to a batch of jobs submitted since it follows batch mode of scheduling, and the user deadline based selection policy is followed when jobs are waiting in queue at the scheduler.

2.2. Hierarchical Load Balancing Algorithm (HLBA). The proposed scheduling algorithm mainly aims for effective

```
Begin
    While there are jobs in unassigned list
    do
        Select job i which has minimum user deadline from unassigned list
        For all resources j in the resource list
        do
            EET(i, j) = Length_i / Capacity_j
            S_i = IF_i + OF_i
            DT(i, j) = S_i / B_j    where B_j is the baud rate of resource j expressed in Mbps
            RT(j) = Σ_{i=1}^{n} EET(i, j)    where n is number of jobs that are already in queue
            ECT(i, j) = RT(j) + EET(i, j) + DT(i, j)
        End for
    Calculate difference matrix such that
        DF(i, j) = D_i − ECT(i, j) for all i, j and if DF(i, j) < 0, then assign a maximum value to DF(i, j)
        where i represents the jobs and j is resource/machine/PE
    Classify the resources/machines/PEs using Algorithms 2, 3, 4
    Sort the list of resources/machines/PEs in ascending order based on DF(i, j)
    Assign the job i to the first resource/machine/PE j in under loaded list with load ≤ 1
    Update the load at each resource/machine/PE
    Remove the job i from unassigned list
    End While
End
```

Algorithm 1: Scheduling algorithm.

resource utilization and minimized makespan. So it considers both application and system aspects. It considers user deadline, expected completion time, data transfer time, and load of each resource. This algorithm calculates the load at the different levels. Since machine and PEs are arranged very close to the resource, data transfer time to machine/PE is negligible, the resources are geographically distributed, and data transfer time to the resource is considered.

Steps

(i) Users submit the jobs to the machines and add the jobs to unassigned list.

(ii) Assign the jobs among the processing elements (PEs) of the machine with satisfied user demand using Algorithm 1, and remove the job from unassigned list. If it is not possible, then forward the set of unassigned jobs to the resource.

(iii) Assign the jobs among the machines of the resource with satisfied user demand using Algorithm 1, and remove the job from unassigned list. If it is not possible, then forward the set of unassigned jobs to the resource broker.

(iv) Assign the jobs among the resources with satisfied user demand using Algorithm 1, and remove the job from unassigned list.

In this model, number of machines, number of tasks, and size of tasks are known prior. The following parameters are used in this algorithm.

$\mathrm{EET}(i, j)$: expected execution time of task i in resource j.

$\mathrm{ECT}(i, j)$: expected completion time of task i in resource/machine/PE_j.

$\mathrm{RT}(j)$: ready time of the resource/machine/PE_j.

D_i: user demand or deadline of the task i.

$\mathrm{DT}(i, j)$: data transfer time to resource j.

Length_i: length of the job i expressed in MI.

$\mathrm{Capacity}_j$: capacity of the resource/machine/PE_j.

AT_j: availability time of resource/machine/PE_j.

When the user submits jobs at the machines, along with the information such as length, deadline, input, and output file size, machine schedules jobs to their processing elements which are capable of completing the jobs within user deadline. If there is no suitable PE, then the jobs will be forwarded to resources. Then resources will schedule the jobs to machines. If there is no suitable PEs found at all machines under that resource, then resources will forward the jobs to the resource broker which will schedule the jobs to other resources.

Load Balancing. To ensure effective utilization of the resources/machines/PEs, load is calculated at different levels. Load of processing elements (PE) is calculated as follows:

$$\mathrm{PELoad}_j = \frac{\sum_{i=1}^{n} \mathrm{Length}_i}{\mathrm{MIPS}_j \times \mathrm{AT}_j}, \tag{1}$$

```
Begin
    For all resources
        if (RLoad < TRes)
            r.state = "underloaded"
            Add the resource to r.underloaded list
        else if (RLoad > TRes)
            r.state = "heavilyloaded"
            Add the resource to r.heavilyloaded list
        else
            r.state = "normallyloaded"
    End for
End
```

Algorithm 2: State allocation for resource.

where PELoad_j is load of the PE_j, Length_i is length of job i, n is jobs that are submitted to PE_j, and MIPS_j is capacity of PE_j.

Load of the machine is calculated as follows:

$$\mathrm{MLoad}_j = \frac{\sum_{i=1}^{n} \mathrm{Length}_i}{\mathrm{MCapacity}_j \times \mathrm{AT}_j}, \tag{2}$$

where

$$\mathrm{MCapacity}_j = \sum_{i=1}^{m} \mathrm{MIPS}_i, \tag{3}$$

where n is jobs that are submitted to $\mathrm{machine}_j$ and m is number of PEs under $\mathrm{machine}_j$.

Load of a resource is calculated as follows:

$$\mathrm{RLoad}_j = \frac{\sum_{i=1}^{n} \mathrm{Length}_i}{\mathrm{RCapacity}_j \times \mathrm{AT}_j}, \tag{4}$$

$$\mathrm{RCapacity}_j = \sum_{i=1}^{m} \mathrm{MCapacity}_i, \tag{5}$$

where n is jobs that are submitted to $\mathrm{resource}_j$ and m is number of machines under $\mathrm{resource}_j$.

Since grid is a collection of dynamic resources from different administrative domains, the resources/machines/PEs can be online and offline from the grid. So the status (either online or offline) of the resource/machine/PE must be updated periodically. To perform load balancing, the state of the resource/machine/PE must be checked when a job arrives or status of the resource/machine/PE is changed. The state is classified into three types such as heavily loaded, normally loaded, and under loaded. To assign the state of resource/machine/PE, the threshold value is calculated at different levels. Threshold value for resource is calculated as follows:

$$\mathrm{TRes} = \frac{\sum_{i=1}^{n} \mathrm{RLoad}_i}{n}, \tag{6}$$

where TRes is threshold value for resource and n is number of resources under resource broker. Threshold value for machine is calculated as follows:

$$\mathrm{TMac} = \frac{\sum_{j=1}^{m} \mathrm{MLoad}_j}{m}, \tag{7}$$

where TMac is threshold value for machine and m is total number of machines under a resource. Threshold value for PE is calculated as follows:

$$\mathrm{TPE} = \frac{\sum_{k=1}^{l} \mathrm{PELoad}_k}{l}, \tag{8}$$

where TPE is threshold value for PE and l is number of PEs under a machine.

Based on the threshold value of the load, the state is assigned for resource/machine/PE using the following criteria.

Algorithm 2 is used to categorize the resource based on their states. Algorithm 3 is used to categorize the machines based on their states. Algorithm 4 is used to categorize the processing elements based on their states. When a resource broker/resource/machine gets information about dynamicity (status) of the resource/machine/PE such as newly added or offline, then status is updated using Algorithm 5. Algorithm 6 is used to reschedule the jobs from heavily loaded resource/machine/PE to the resources/machines/PEs in under loaded list.

2.3. Simulation Setup. The proposed HLBA algorithm is simulated using gridsim5.0 toolkit. It is a more popular tool for modeling the resources and scheduling the jobs. It is also used to analyze algorithms on large scale distributed systems of heterogeneous resources. This tool has the facility for creating different classes of heterogeneous resources that can be aggregated using grid resource broker. This tool can be used for solving computational and data intensive applications.

The proposed HLBA algorithm is simulated for varying number of jobs from 100 to 600 in 16 resources. The characteristics of jobs and resources in grid hierarchy are given in Table 1.

```
Begin
    For all resources
        For all machines
            if (MLoad < TMac)
                m.state = "underloaded"
                Add the machine to m.underloaded list
            else if (MLoad > TMac)
                m.state = "heavilyloaded"
                Add the machine to m.heavily loaded list
            else
                m.state = "normallyloaded"
        End for
    End for
End
```

Algorithm 3: State allocation for machine.

```
Begin
    For all resources
        For all machines
            For all PEs
                if (PELoad < TPE)
                    pe.state = "underloaded"
                    Add it to underloaded list
                else if (PELoad > TPE)
                    pe.state = "heavilyloaded"
                    Add it heavilyloaded list.
                else
                    pe.state = "normallyloaded"
            End for
        End for
    End for
End
```

Algorithm 4: State allocation for PE.

Table 1: Characteristics of grid resources and jobs.

Number of resources	16
Number of machines	1–5
Number of PE's per machine	1-2
Number of jobs	100 to 600

The simulation is carried out in two ways. Firstly, the number of resources is kept as 16, the number of jobs is varied from 100 to 600, and the makespan, hit rate, and resource utilization are measured. Secondly, the number of jobs is considered as 512, and the number of resources is considered as 16 which is the benchmark for scheduling algorithms to evaluate their efficiency. The simulation results for these two ways are discussed in Results and Discussion section.

3. Results and Discussion

The performance of the proposed algorithm is evaluated using the parameters such as makespan, hit rate, miss rate, and resource utilization. Makespan is defined as maximum of completion time of the jobs at all resources:

$$\text{Makespan} = \max\left(\text{CT}_i\right), \tag{9}$$

where CT_i is defined as completion time of the last job at resource i.

The utilization of the resources is calculated using the following formulae:

$$\text{RU}_i = \text{load}_i \times 100. \tag{10}$$

Average resource utilization is as follows:

$$\text{ARU} = \frac{\sum_{i=1}^{n} \text{RU}_i}{n}, \tag{11}$$

where RU_i is utilization of the resource i, CT_i is completion time of the last job at resource i, ARU is average resource utilization expressed in percentage, and n is number of resources.

```
Begin
    if (Resource/Machine/PE is newly added)
        Calculate new threshold value at different levels
        Classify the resource/machine/PE based on Algorithms 2, 3, and 4
    Endif
    if (Resource/Machine/PE goes down)
        Retrieves the jobs which are allocated to the resource/Machine/PE and added to unassigned
        list
        Calculate new threshold value at different levels
        Classify the resource/machine/PE based on Algorithms 2, 3, and 4
    Endif
End
```

Algorithm 5: Status update algorithm.

```
Begin
    Select a job from the resource/machine/PE which goes down
    Find the resources/machines/PEs from under loaded list which satisfy user demand for the selected job
    Schedule the jobs among selected resources/machines/PEs using Algorithm 2.
End
```

Algorithm 6: Rescheduling algorithm.

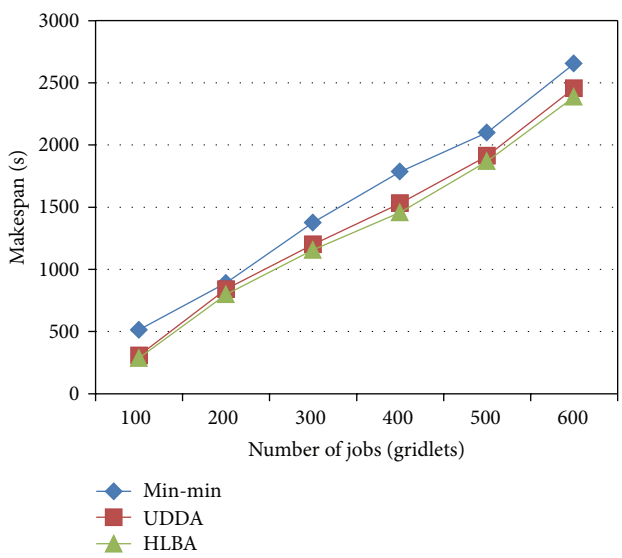


Figure 2: Performance analysis based on makespan (sec).

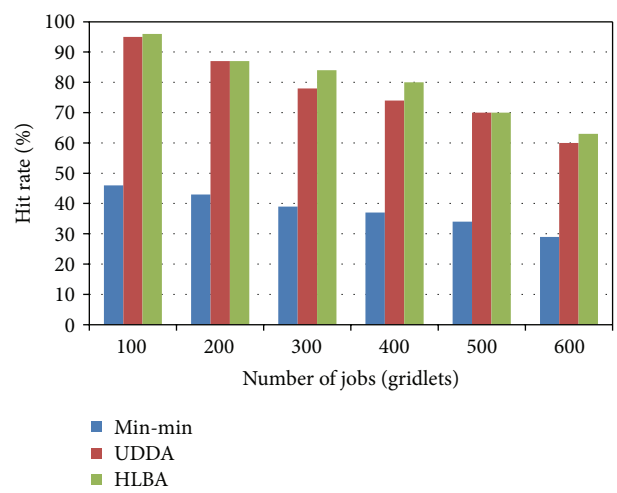


Figure 3: Performance analysis based on hit rate (%).

Hit rate is defined as number of jobs completed within user deadline:

$$\text{Hit rate} = \frac{J_{\text{comp}}}{J_{\text{sub}}} \times 100. \tag{12}$$

Miss rate is defined as number of jobs that could not be completed within user deadline:

$$\text{Miss rate} = \frac{J_{\text{fail}}}{J_{\text{sub}}} \times 100, \tag{13}$$

where J_{comp} is jobs that are completed within user deadline, J_{sub} is jobs that are submitted, and J_{fail} is jobs that could not be completed within user deadline.

The result of the proposed load balancing algorithm is analyzed and compared with min-min and UDDA algorithms. The factors such as resource/machine heterogeneity and job heterogeneity are considered. The improvement of the proposed algorithm is proved by comparing makespan, hit rate, miss rate, and resource utilization. The simulation results have been taken from various test cases. The proposed algorithm generates better results in most of the test cases. Figures 2–5 show the performance of proposed algorithm for varying number of jobs.

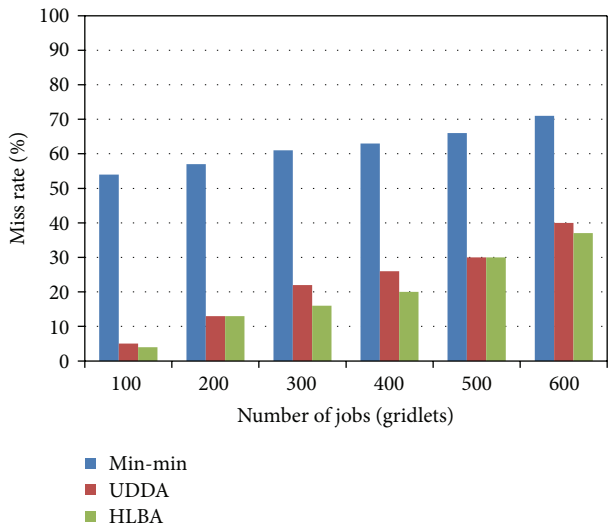


Figure 4: Performance analysis based on miss rate (%).

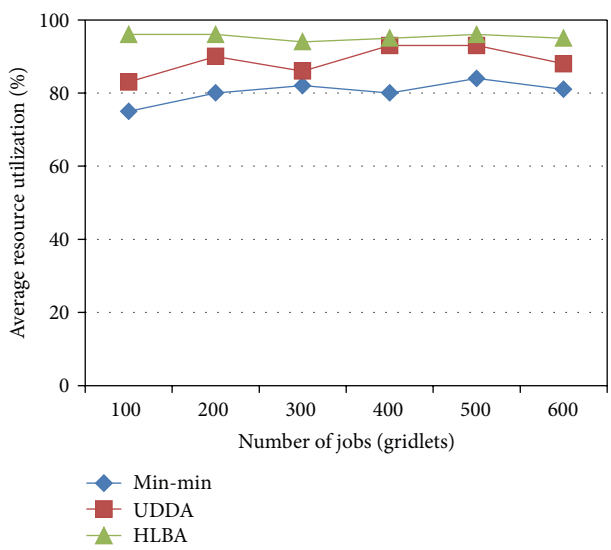


Figure 5: Performance analysis based on average resource utilization (%).

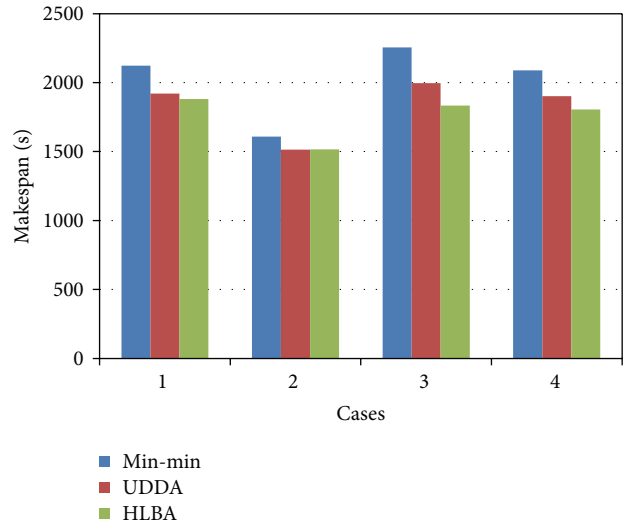


Figure 6: Performance analysis based on makespan (sec).

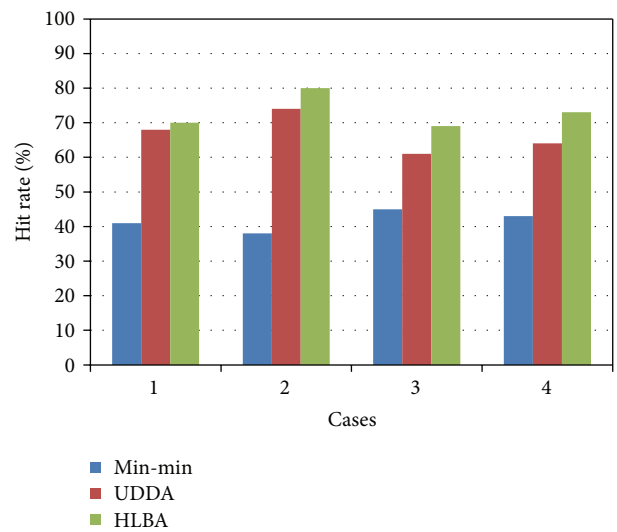


Figure 7: Performance analysis based on hit rate (%).

In Figure 2, the proposed hierarchical load balancing algorithm is compared with the min-min and UDDA algorithm for its efficiency of scheduling which can be measured by makespan expressed in seconds for varying number of jobs from 100 to 600. The performance of the proposed algorithm is better when compared to UDDA and min-min.

In Figure 3, the efficiency of the proposed HLBA algorithm based on hit rate is analyzed, and the analysis shows that the HLBA algorithm has highest hit rate than the other algorithms. Based on miss rate, the performance analysis is shown in Figure 4 which shows that the proposed HLBA has less miss rate than the other algorithms.

The performance analysis of the proposed HLBA and other algorithms min-min and UDDA based on the benchmark values for 512 jobs and 16 resources are elaborated in Figures 6–8. Figure 6 shows that the proposed HLBA has better makespan than min-min and UDDA.

Figures 7 and 8 show that the proposed HLBA has better hit rate and average resource utilization than min-min and UDDA.

4. Conclusion and Future Work

In this paper, hierarchical load balancing approach with user demand aware scheduling algorithm is proposed. This algorithm considers the dynamicity of the resources/machines/PEs, resource/machine heterogeneity, and task heterogeneity. Since this algorithm considers load of resource/machine/PE, the utilization of the resource/machine/PE is improved. By considering the user deadline of the job, the number of jobs completed within user deadline (hit rate) is increased, and user satisfaction is improved. When comparing with other

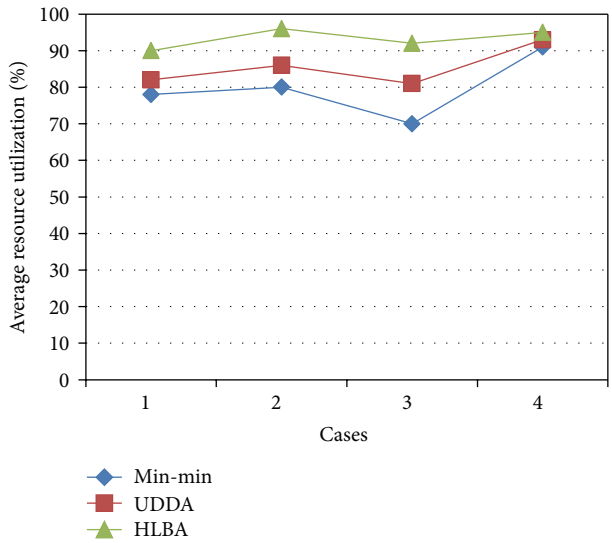


FIGURE 8: Performance analysis based on average resource utilization (%).

scheduling algorithms, the overall system performance is improved in the proposed scheduling algorithm. In future, some other user demands such as execution cost may be considered, and each machine can be thought as multiprogrammed system. In addition to that some more technologies can also be introduced such as GridFTP and fault tolerant techniques.

References

[1] Q. Zheng, C.-K. Tham, and B. Veeravalli, "Dynamic load balancing and pricing in grid computing with communication delay," *Journal of Grid Computing*, vol. 6, no. 3, pp. 239–253, 2008.

[2] I. Foster and C. Kesselman, *The Grid2: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, Boston, Mass, USA, 2nd edition, 2004.

[3] B. Yagoubi and M. Meddeber, "Distributed load balancing model for grid computing," *ARIMA Journal*, vol. 12, pp. 43–60, 2010.

[4] Y. Li, Y. Yang, M. Ma, and L. Zhou, "A hybrid load balancing strategy of sequential tasks for grid computing environments," *Future Generation Computer Systems*, vol. 25, no. 8, pp. 819–828, 2009.

[5] P. Keerthika and N. Kasthuri, "An efficient fault tolerant scheduling approach for computational grid," *American Journal of Applied Sciences*, vol. 9, no. 12, pp. 2046–2051, 2012.

[6] U. Karthick Kumar, "A dynamic load balancing algorithm in computational grid using fair scheduling," *International Journal of Computer Science Issues*, vol. 8, no. 5, article 1, 2011.

[7] T. D. Braun, H. J. Siegel, N. Beck et al., "Comparison study of static mapping heuristics for a class of meta-tasks on heterogeneous computing systems," in *Proceedings of the 8th Heterogeneous Computing Workshop (HCW '99)*, pp. 15–29, April 1999.

[8] H. Baghban and A. M. Rahmani, "A heuristic on job scheduling in grid computing environment," in *Proceedings of the 7th International Conference on Grid and Cooperative Computing (GCC '08)*, pp. 141–146, October 2008.

[9] Z. Qian and L. Zhen, "Design of grid resource management system based on divided min-min scheduling algorithm," in *Proceedings of the 1st International Workshop on Education Technology and Computer Science (ETCS '09)*, pp. 613–617, March 2009.

[10] X. He, X. Sun, and G. von Laszewski, "QoS guided Min-Min heuristic for grid task scheduling," *Journal of Computer Science and Technology*, vol. 18, no. 4, pp. 442–451, 2003.

[11] L. Wenzheng and Z. Wenyue, "An improved scheduling algorithm for grid tasks," in *Proceedings of the International Symposium on Intelligent Ubiquitous Computing and Education (IUCE '09)*, pp. 9–12, May 2009.

[12] J. Lin, B. Gong, H. Liu, C. Yang, and Y. Tian, "An application demand aware scheduling algorithm in heterogeneous environment," in *Proceedings of the 11th International Conference on Computer Supported Cooperative Work in Design (CSCWD '07)*, pp. 599–604, IEEE Xplore Press, April 2007.

[13] H. Izakian, A. Abraham, and V. Snášel, "Comparison of heuristics for scheduling independent tasks on heterogeneous distributed environments," in *Proceedings of the International Joint Conference on Computational Sciences and Optimization (CSO '09)*, pp. 8–12, April 2009.

[14] P. Suresh, P. Balasubramanie, and P. Keerthika, "Prioritized user demand approach for scheduling meta tasks on heterogeneous grid environment," *International Journal of Computer Applications*, vol. 23, article 1, 2011.

[15] P. Suresh and P. Balasubramanie, "User demand aware scheduling algorithm for data intensive tasks in grid environment," *European Journal of Scientific Research*, vol. 74, no. 4, pp. 609–616, 2012.

[16] B. Yagoubi and Y. Slimani, "Dynamic load balancing strategy for grid computing," in *Proceedings of the World Academy of Science, Engineering and Technology*, vol. 19, May 2006.

[17] S. A. Ludwig and A. Moallem, "Swarm Intelligence Approaches for Grid Load Balancing," *Journal of Grid Computing*, vol. 9, no. 3, pp. 279–301, 2011.

[18] J. Cao, D. P. Spooner, S. A. Jarvis, and G. R. Nudd, "Grid load balancing using intelligent agents," *Future Generation Computer Systems*, vol. 21, no. 1, pp. 135–149, 2005.

[19] F. Alharbi, "Simple scheduling algorithm with load balancing for grid computing," *Asian Transactions on Computers*, vol. 2, no. 2, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 372368, 11 pages
http://dx.doi.org/10.1155/2013/372368

Research Article

A Resilience Approach to Symbiosis Networks of Ecoindustrial Parks Based on Cascading Failure Model

Yu Zeng,[1,2] Renbin Xiao,[1] and Xiangmei Li[3]

[1] *Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*
[2] *School of Science, Hubei University of Technology, Wuhan 430068, China*
[3] *Wenhua College, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China*

Correspondence should be addressed to Renbin Xiao; rbxiao@ hust.edu.cn

Received 3 April 2013; Accepted 19 May 2013

Academic Editor: Tingwen Huang

Copyright © 2013 Yu Zeng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Evaluation and improvement of resilience in ecoindustrial parks have been the pressing issues to be addressed in the study of safety. In this paper, eco-industrial systems are extracted as symbiosis networks by using social network analysis first. We then construct a novel cascading failure model and propose an evaluation method of node importance according to the features of symbiosis networks of eco-industrial parks. Based on the cascading model, an effective new method, that is, the critical threshold, is put forward to quantitatively assess the resilience of symbiosis networks of eco-industrial parks. Some theoretical analysis is furthermore provided to the critical threshold. Finally, we take Jinjie eco-industrial system in Shanxi Province of China as a case to investigate its resilience. The key potential nodes are identified by using our model. We also find the respective relation among the resilience of symbiosis networks and the parameters in our cascading model. Theoretical analysis results and numerical simulations both show the optimal value of the tunable load parameter with which the strongest resilience level against cascading failures can be attained in symbiosis networks of eco-industrial parks.

1. Introduction

Complex systems are everywhere in the nature [1–5]. Their Security could have a big impact on the economic development and social stability, so it has attracted a great deal of attention [6–9]. Complex systems can be described as all kinds of networks. Complex networks have become an indispensable part of our life, such as social ecological networks, transportation networks, and power grid networks. Particularly, cascading failure phenomenon of complex networks has been the focus of many recent studies, for the reason that cascading failures are common phenomenon in real-life systems and can cause huge damages to our life and industry development [10–13]. Symbiosis networks of ecoindustrial parks (abbreviated as EIPs hereafter) are also typical networks including cascading phenomenon. In EIPs, the products produced by one enterprise could be the nutrients of other enterprises. Through this kind of cooperation, they form the industrial symbiosis networks, where there exist fairly close industrial symbiosis relations between or among products produced by different enterprises. In the symbiosis network of an EIP, when an upstream enterprise or link failure emerges, the insufficiency of supplying goods to the downstream enterprises will happen. Even though the failure emerges very locally in the ecoindustrial system, it will also quickly spread like a plague to large areas, causing serious damage to the whole system and even resulting in the global collapse. In this way, the entire symbiosis network can be largely affected due to cascading failures. Therefore, it is very necessary to study cascading failures in symbiosis networks of EIPs.

Recently, a plenty of EIPs are constructed in many places, such as Denmark, America, Japan, and Austria. In China, Guangdong Nanhai, Hunan Changsha, Xinjiang Shihezi, Shanghai Wujing, and others have also constructed some ecoindustrial parks. With the development of industrial ecology, the EIP is developing as a brand new research field [14–16]. The research on the complexity of ecoindustrial

systems has become popular and gradually made a breakthrough. Graph theory was adopted to introduce the flow of materials or energy in Choctaw ecoindustrial park first [17]. Furthermore, Sterr and Ott [18] introduced the flow path of materials or energy in the industrial region network of Rhine-Neckar and made clear the relationship between the enterprise inside the network and that outside the network. Graedel et al. [19] used statistical indexes of the food web from natural ecosystems for reference in his evaluations of EIPs. They assessed the business symbiosis in industrial systems through species richness and connectedness. Based on the work, the potential contribution of industrial symbiosis networks to environmental innovation was explored [16]. In order to measure the complexity characteristics of industrial symbiosis networks, network complexity of two cases, the Kalundborg system in Denmark and the Gongyi system in China were studied. The features of industrial symbiosis networks from the aspects of scale, aggregation, connectedness, complexity, and node influences were analyzed [20]. Different from these methods, a methodology for translating ecological quantitative analysis techniques to an industrial context was proposed. The methodology is demonstrated in the case of Burnside industrial park using the concepts of connectance and diversity. The demonstrated techniques can potentially aid in gaining an understanding of industrial symbiosis [21]. In addition, Posch [22] investigated whether industrial recycling networks or industrial symbiosis projects can be used as a starting point for broader intercompany cooperation for sustainable development.

Although much research on EIPs focuses on the exchange analysis of material or energy, little work has been published on the topology structure of symbiosis networks of EIPs. Especially to date, most researchers have investigated the cascading phenomenon in power grid networks and traffic networks [23, 24]. Little attention has been paid to the cascading phenomenon in symbiosis networks of EIPs uptil now. Facing the upsurge of ecoindustrial park construction, a set of scientific appraise index systems needs to be established. A challenging problem in industrial ecology is whether industrial symbiosis systems bear complexity features if the systems are described only from their mass or energy interlinkages. Moreover, various members play various roles in symbiosis networks of EIPs. If a critical component fail, the failure will spread very quickly, and it could trigger a more serious damage to symbiosis networks of EIPs due to cascading failures.

Taking the above existing problems into account, we use complex network theory to go into the interior of symbiosis networks and analyze the resilience against cascading failures in symbiosis networks of EIPs in this paper. We abstract EIPs as symbiosis networks and construct a novel cascading model for symbiosis network according to the EIP's characteristics. We use the ecological degree in EIPs for reference and raise a measure to quantitatively evaluate the power and status of nodes. Based on the cascading model, an effective new concept, that is, the critical threshold is put forward to quantitatively assess the resilience of symbiosis networks of EIPs. We take Jinjie ecoindustrial system in Shanxi Province of China as a case to investigate its resilience. Through our proposed model, some potential critical nodes which are sensitive to the ecological degree and the global connectivity of the symbiosis network but not so important intuitively are found. The respective relation among the resilience of symbiosis networks and the parameters is also found in our cascading model. Moreover, according to theoretical analysis results and numerical simulations, we obtain an interesting result; that is, when the value of the tunable load parameter is 1, the symbiosis network displays the strongest resilience level against cascading failures. These results in this paper may be very helpful for symbiosis networks of EIPs to enhance the resilience against cascading failures. Our work also may have practical implications for promoting the construction of EIPs.

The remainder of this paper is organised as follows. Symbiosis networks of ecoindustrial parks are established with social network analysis in Section 2. According to the construction method, an illustrative example is also given in this section. Section 3 proposes the cascading failure model and an evaluation method of node importance according to the features of ecoindustrial parks. Section 4 gives some theoretical analysis and numerical simulations to our model. Finally, Section 5 concludes with some discussions.

2. Symbiosis Network Construction of Ecoindustrial Parks

As one direct practical form of industrial ecology, ecoindustrial parks are becoming important research contents of industrial ecology. The ecoindustrial park (EIP) is defined as an industrial park in which enterprises cooperate with each other by using each other's by-products and wastes; through this kind of cooperation, they form a symbiosis network. Network analysis is a methodology to study objects as a part of a larger system, so it is a brand new feeling to study EIPs through this method. Here, the symbiosis network of an EIP is a network that consists of nodes (components) and the connections between them. In order to investigate the status of all components in EIPs and the resilience of symbiosis networks of EIPs, we will extract ecoindustrial parks as symbiosis networks with social network analysis. The detailed method of building symbiosis network models is as follows.

(1) Nodes and edges of symbiosis networks of EIPs: We confirmed the members of the parks first. The area products, by-products, wastes, and materials are the foundation nodes. Using the administrative boundary as the network boundary, we add some members outside the parks who exchange resources with members within the parks as the supplement nodes. Then we investigate the flow between members, such as exchanges of products, by-products, wastes, and materials. When there is a flow of matter or energy between any two members in the parks, we say that there is an edge between them.

(2) Adjacency matrices of symbiosis networks of EIPs: on the basis of confirmed nodes and edges among members, we establish a relational data collection. We use the directive dichotomous assessment system to

judge whether the relation exists or not and call the system the adjacency matrix. The adjacency matrix is a square matrix in which if there is one relation between longitudinal and transverse members, the crossover box should be filled with 1, that is, $a_{ij} = 1$, otherwise 0; that is, $a_{ij} = 0$. Obviously, if the matrix represents an undirected graph, then it is symmetric as $a_{ij} = a_{ji}$.

(3) Symbiosis network models of EIPs: when one node fails, it not only could lead to the insufficient supply of matter to the downstream enterprises but also decrease the matter outflow of the upstream enterprises at the inverse direction. Members in symbiosis networks cooperate with each other. So we regard symbiosis networks of EIPs as undirected networks. Furthermore, considering that the exchange of data of matter between nodes is difficult to obtain and the measurement standard could also not be unified, we therefore extract ecoindustrial parks as undirected and unweighted symbiosis networks which consist of nodes and edges.

Here, we take the Jinjie ecoindustrial park in Shanxi Province of China as a case to build a symbiosis network model. The EIP consists of four industrial systems: the industrial system of coal power, the industrial system of coal chemical, the industrial system of salt chemical, and the industrial system of glass building materials. This ecoindustrial system is designed to transform resources in situ, develop comprehensive utilization of materials, and produce minimal emissions. A lot of exchanges of matter or energy exist in Jinjie ecoindustrial system, and it is a relatively complex industrial symbiosis network. We choose products, by-products, wastes, and materials from a certain scale of enterprises whose annual output values are more than 2 million yuan as 75 nodes of the symbiosis network. These nodes come from four different industrial systems, respectively, including outsourcing coal, semicoke, coal tar, coal gangue, coal mine, methyl alcohol, solid sulfur, condensed water, salt, polyvinyl chloride, caustic soda, methane oxides, quartz sand, industrial silicon, glass, glass bricks, waste heat, waste water, waste residues, and waste gas. By collecting data and investigating the interrelation between members of Jinjie ecoindustrial park, we obtain the adjacency matrix. According to the adjacency matrix of the symbiosis network, we could obtain the symbiosis network model of Jinjie ecoindustrial park under UCINET 6.0, as is shown in Figure 1.

3. Model of Cascading Failure in Symbiosis Networks of Ecoindustrial Parks

In EIPs, the products produced by one enterprise could be the nutrients of other enterprises. Components in EIPs cooperate with each other. If an upstream enterprise or link failure emerges, it not only could lead to the insufficient supply of matter to the downstream enterprises but also decrease the matter outflow of the upstream enterprises at the inverse direction. The fairly close industrial symbiosis relations exist in symbiosis networks. In order to show the cascading phenomenon in symbiosis networks of ecoindustrial parks, we will focus on cascades triggered by the removal of a component. For the symbiosis network of an EIP, initially the network is in a stationary state in which the load of each node is smaller than its capacity. The production system with the EIP maintains the normal operationsm but the removal of a node will change the load balance and lead to the load redistribution over other nodes. Once the capacity of these nodes is insufficient to handle the extra load, this must be induce the node further breaking, triggering a cascade of overload failures and eventually a large drop in the performance of the network. Therefore, the removal of one node can cause serious consequences. In the following, we will construct a cascading failure model according to the characteristics of EIPs.

3.1. A New Cascading Failure Load Model. We define a network $G = (V, E)$ by a nonempty set of nodes $V = \{1, 2, \ldots, N\}$ and a nonempty set of edges $E = \{ij \mid i, j \in V\}$. In most of the previous cascade failure models, the load on a node (or an edge) was generally estimated by its degree or its betweenness [25, 26]. The degree method is inferior owing to its consideration of only a single node degree. Thus, it will lose much information in many actual applications. The betweenness method is not beneficial to its consideration of topological information for the whole network. In symbiosis networks of EIPs, considering that the load of a component is associated with its degree and the degrees of its neighbor components due to the close relation between supply and demand, we assume that the initial load L_i of node i being dependent on the degree of node i and the degrees of its neighbor nodes, the expression of which is defined as follows:

$$L_i = k_i^{\beta} + \left(\sum_{m \in H_i} k_m \right)^{\beta}, \tag{1}$$

where k_i and H_i are the degree of the node i and the set of its neighbor nodes, respectively, and β is the tunable parameter which controls the strength of the initial load of a node.

We know that each node of symbiosis networks has a capacity which is the largest load that the node can handle. In symbiosis networks of EIPs, the load capacity of each node is severely limited by cost. Therefore, we assume that the load capacity C_i of node i is proportional to its initial load; that is,

$$C_i = (1 + \alpha) L_i(0), \quad i = 1, 2, \ldots, N, \tag{2}$$

where $\alpha > 0$ is the tolerance parameter which characterizes the resistance to the attacks. For the symbiosis network of an EIP, the tolerance parameter α also reflects the construction cost of the EIP.

After node i is removed from the network, its load will be redistributed to its neighbor node j. We assume that

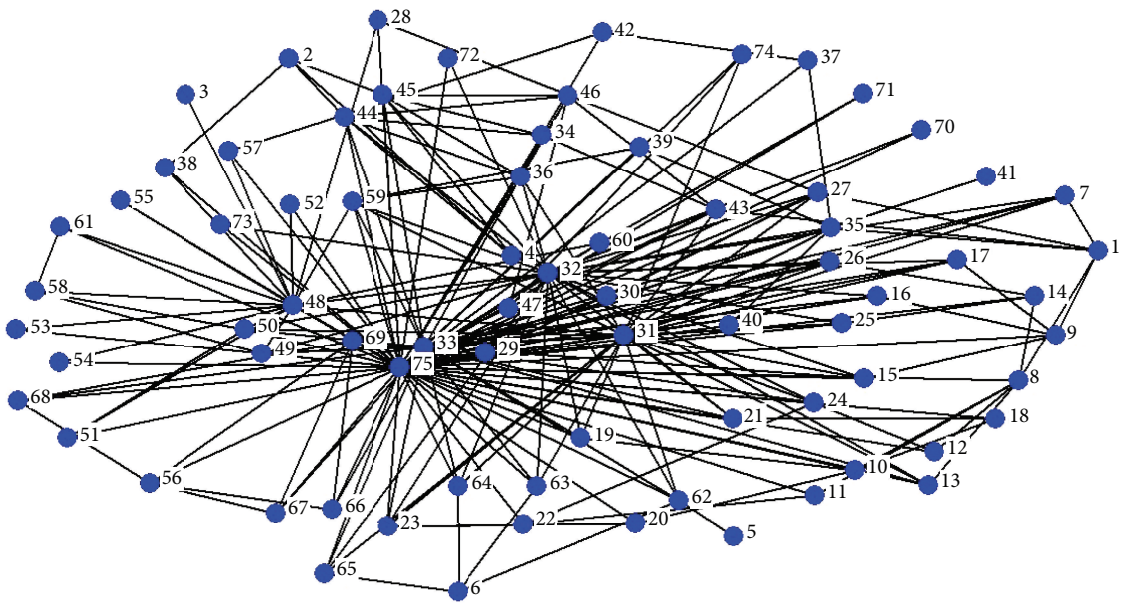


Figure 1: The symbiosis network representation of Jinjie ecoindustrial park.

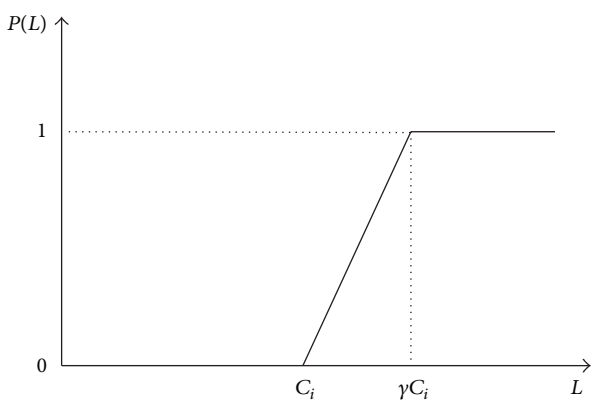


Figure 2: The removal probability $P(L)$ of node i, at which load L lasts for time T.

the additional load ΔL_{ij} received by the neighbor node j is proportional to its initial load; that is,

$$\Delta L_{ij} = \frac{L_j}{\sum_{f\in H_i} L_f} \times L_i \tag{3}$$

$$= \frac{k_j^{\beta} + \left(\sum_{f\in H_j} k_f\right)^{\beta}}{\sum_{f\in H_i}\left[k_f^{\beta} + \left(\sum_{l\in H_f} k_l\right)^{\beta}\right]} \times \left[k_i^{\beta} + \left(\sum_{f\in H_i} k_f\right)^{\beta}\right]. \tag{4}$$

The method of immediate removal of an instantaneously overloaded node is widely adopted in most of the previous cascade failure models [25, 26]. This method does not consider the overloaded node removal mechanism in real life. According to the actual situation of EIPs, a certain quantity of instantaneous node overload is permissible and the failure of a node is mainly caused by the accumulative effect of overload, especially for EIPs. In symbiosis networks of EIPs, the insufficient supply of goods induced by a small occasional incident usually has no distinct effect on the normal operation of the whole symbiosis networks. It is because that there exists a certain monitoring and control system in such networks. Once the node overload is detected, some effective measures are taken within the response time to decrease the capacity constraint violation and even can make the node load less than its capacity. In other words, the overloaded node is not necessarily removed. Therefore, we introduce a new method of the overloaded node removal to reflect ecoindustrial systems.

Suppose that $P(L)$ is the removal probability of a node, at which load L lasts for a period of time T, as shown in Figure 2. The probability density of $P(L)$ obeys uniform distribution of the interval $[0, T]$. The value of T has some relation to the response time of industrial symbiosis systems. The detailed expression of $P(L)$ is as follows:

$$P(L) = \begin{cases} 0, & L \le C_i, \\ \dfrac{L - C_i}{(\gamma - 1) C_i}, & C_i < L < \gamma C_i, \\ 1, & L \ge \gamma C_i, \end{cases} \tag{5}$$

where the parameter γ ($\gamma > 1$) may be different for various symbiosis networks and the value of γ determines to a certain extent the protection of those overload nodes. Obviously, the bigger the value of γ, the higher the cost of avoiding cascading failures. In (5), the three cases correspond to three states of nodes: normal state, overload state, and failure state.

In the simulations, the following iterative rule is adopted at each time t:

$$h_i(t) = \begin{cases} h_i(t-1) + \dfrac{P(L_i(t))}{T}, & L_i(t) > C_i, \\ 0, & L_i(t) \le C_i, \end{cases} \quad (6)$$

$$i = 1, 2, \ldots, N,$$

where $h_i(t)$ is the removal probability of node i at time t and $L_i(t)$ is the load of node i at time t.

We obtain $P(L_i(t))$ according to Figure 2 and (5) and derive the malfunction probability $h_i(t)$ from (6). According to the above process, the loads of all nodes are calculated at each time after an initial removal of a node induced by a malfunction. We then determine the nodes which will be removed from the network by comparing the malfunction probability and a random $\delta \in (0, 1)$. The cascading failure stops until the loads of all nodes are not larger than their corresponding capacity.

3.2. Node Importance Evaluation Based on Cascading Model. In the symbiosis network of an EIP, various nodes have different effects on the operation and management of ecoindustrial system, especially for the ecological degree which is employed to measure the relation of exchanging by-products and wastes. The failures of critical nodes are more likely to cause the collapse of the whole ecoindustrial system due to cascading failures. If we can provide priority protection for those critical nodes in advance, the resilience and security of symbiosis networks of EIPs can be improved. To better understand the cascading failure characteristics, in this section, we will discuss and propose a new evaluation method of node importance considering cascading failure for symbiosis networks of EIPs.

After the cascading process is induced by removing a node, the total number of residual nodes is called the network area. In order to find a suitable evaluation method of node importance, we define the following indicator by adopting the network area first:

$$R_i = 1 - \frac{N_i'}{N}, \quad (7)$$

where N is the number of all nodes before cascading failures and N_i' is the number of the nodes which can maintain the normal operations after cascading failures induced by removing node i in the whole symbiosis network. Obviously, the value of indicator R_i reflects the damage degree of node i to the global connectivity of symbiosis networks.

Then, the set V of all nodes in the symbiosis network is divided into two categories $V = V_1 \cup V_2$, where the set $V_2 = \{n_1, n_2, \ldots, n_{N_e}\}$ consists of waste nodes and by-product nodes which are all called ecological nodes, and the set $V_1 = \{m_1, m_2, \ldots, m_{N_p}\}$ consists of all other nodes which are called product nodes. The number of all nodes is $N = N_e + N_p$. We propose a parameter which is called the rate of ecological connectance R_e. It is defined as follows:

$$R_e = \frac{\sum_{a \in V_2} k_a}{N(N-1)/2}, \quad (8)$$

where k_a is the degree of node a and the summation is over the degrees of all ecological nodes. From (8), the higher the value of R_e is, the more the flow of the reciprocal utilization of by-products and wastes within EIP is.

We know that the development aim of EIPs is maximum utilization of matter and energy. To reflect the feature of symbiosis networks of EIPs which is different from other networks, now we define another indicator which can be used to characterize the node's impact on the ecological degree of symbiosis networks as follows:

$$I_i = 1 - \frac{R_e'}{R_e} = 1 - \frac{\sum_{b \in V_2 \cap V_i'} k_b}{\sum_{a \in V_2} k_a}, \quad (9)$$

where R_e and R_e' are the rates of ecological connectance before and after cascading failures in symbiosis networks of EIPs, respectively. V_i' is the set of residual nodes under the normal operations in the whole network after cascading breakdowns induced by removing node i. That is, the numerator of the fraction is the summation over the degree of all residual ecological nodes after cascading breakdowns caused by the removal of node i, and the denominator is the summation over the degree of all ecological nodes before cascading breakdowns. The indicator I_i characterizes the damage degree of node i to the rate of ecological connectance. The higher the value of indicator I_i, the greater the impact of node i on the ecological degree of EIPs.

In order to compare indicators R_i and I_i, we apply the two methods to the symbiosis network of Jinjie ecoindustrial park in Section 2 and analyze its cascading failure characteristics. In the symbiosis network of Jinjie ecoindustrial park, the set V_2 has 14 nodes, such as waste water, waste residue, and waste gas, and the set V_1 consists of other 61 nodes, such as outsourcing coal, salt, and quartz sand. In the simulations, we choose $\alpha = 0.2$, $\beta = 0.2$, and $\gamma = 0.3$. Every node in the network is removed one by one, and the corresponding results are calculated, for example, removing the node i and calculating R_i and I_i by (7) and (9) after the cascading process is over.

Figure 3 shows the values of two indicators for each node. We see that the change in I_i is more obvious than R_i. For example, the values of indicators R_6, R_7 are the same. It indicates that the removals of node 6 and node 7 lead to the same damage degree to the global connectivity of the symbiosis network. However, the difference in I_6 and I_7 illustrates that the two nodes' removal has a different impact on the ecological degree of the symbiosis network. Similarly, although the values of R_{74} and R_{75} are same, the removals of node 74 and node 75 cause different damage on ecological degree according to the values of I_{74} and I_{75}. The same cases also happen in other nodes. Furthermore, we also find that the removals of nodes 1, 2, 6, 7, 9–28, 33–39, 42–46, 48–51, 53, 56–59, 61–68, 74, and 75 do not cause cascading failures. It implies that their importance is relatively weak. Despite this, the removals of these nodes result in different impacts on the ecological degree according to indicator I_i. So we can declare that indicator I_i is more helpful in finding out the potential key nodes than indicator R_i. In other words, indicator I_i is a more suitable measure to evaluate the node importance and

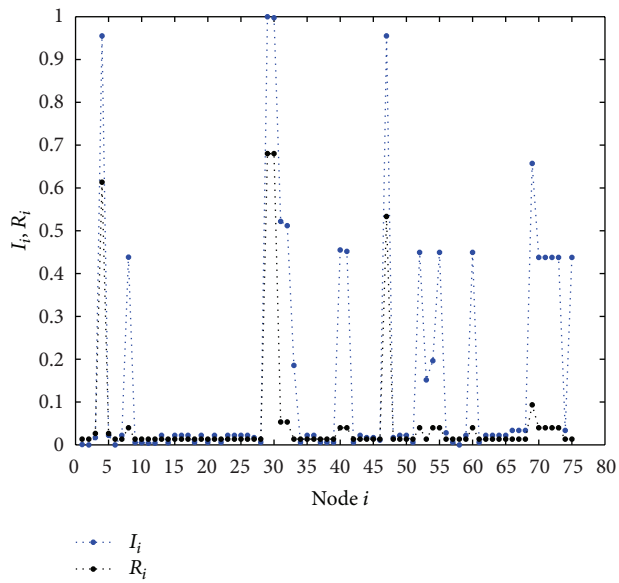


Figure 3: Comparison of two indicators for each node.

reflect the status and role of nodes for symbiosis networks of EIPs. It is because that this method can help us to find some potential critical nodes which are sensitive to the ecological degree of symbiosis networks and the efficiency but not so important intuitively.

With the above analysis, we choose indicator I_i as the measurement of node importance. All nodes are sorted according to the descending order of the values of indicator I_i. Table 1 represents the detailed cascading failure processes induced by the nodes that take the top 5 places in ranking. The result shows that node 29 is the most important node in the symbiosis network of Jinjie ecoindustrial park because the failure of node 29 (outsourcing coal) leads to the collapse of the whole symbiosis network. We also find that node 30 (waste water), node 4 (salt), node 47 (waste residue), node 69 (quartz sand), and node 31 (waste heat) rank 2nd, 3rd, 4th, and 5th respectively. The removals of these nodes result in great changes in the ecological degree. They also occupy a very important status. The simulation results are reasonably consistent with the actual situation. Therefore, in the operation and management of Jinjie ecoindustrial park, we should pay more attention to the enterprises in advance that product or deal with these critical nodes, such as waster water, salt, waste residue, waste heat, quartz sand, and especially outsourcing coal.

In addition, we also find that the highest degree or load node is not necessarily one of the most important nodes, and the low degree or load node may also occupy an important status. For instance, the load and the degree of node 30 are both low. Its removal leads to neighbor nodes 27, 75 malfunction first. This triggers a cascade of overload failures and eventually results in the whole network almost collapsing. As a result, there is a large drop in the ecological degree of the whole network after the cascading process is over. It is shown that node 30 is actually a potential key node. Therefore, the high or low ecological degree does not absolutely depend on the node degree or node load. Especially for symbiosis networks of EIPs, the exchange and recycling of by-products and wastes is the most essential and direct factor.

By applying our model to the symbiosis network of Jinjie ecoindustrial park, one can see that our new evaluation method of node importance characterizes the status of node well. This measure can help us to identify some potential critical nodes in symbiosis networks of EIPs. Through protecting these nodes in the operation and management in advance and making an effort to strengthen their construction, we can achieve the comprehensive utilization of matter and energy between components and even achieve the optimal resource allocation. The ecological construction will be promoted systematically with the step-by-step breakthrough method. The cost of pollution treatment not only will be decreased, but also the considerable economic and social benefits will be obtained. Thus, the resilience and security of symbiosis networks of EIPs are improved, especially at the same time the development of EIPs will be promoted.

4. Resilience Analysis for Symbiosis Networks of EIPs on Model

It is known that the safety of EIPs severely affects the economic development and the social stability. Therefore, it is very necessary to evaluate and improve resilience of symbiosis networks of EIPs. In this section, we will go into the interior of symbiosis networks and propose the concept of critical threshold to characterize the resilience of symbiosis networks against cascading failures by using our cascading model.

4.1. A Quantitative Measurement of Resilience for Symbiosis Networks of EIPs Based on Cascading Model. CF_i denotes the avalanche size, that is, the number of broken nodes induced by removing node i. Obviously, $0 \leq CF_i \leq N-1$. We remove every node in a network one by one and calculate the corresponding results, for example, removing the node i and calculating CF_i after the cascading process is over. Then, the normalized avalanche size is defined as follows:

$$CF^* = \frac{\sum_{i \in V} CF_i}{N(N-1)}, \tag{10}$$

where V represents the set of all nodes in the symbiosis network of an EIP.

From the proposed model, the load capacity C_i of the node i is proportional to its initial load and precisely $C_i = (1+\alpha)L_i(0)$, $i = 1, 2, \ldots, N$. Given a value of the tunable parameter β, when the value of the tolerance parameter α is sufficiently small, we can imagine that it is easy for the whole network to fully collapse in the case of an arbitrary node failure because the capacity of each node is limited. On the other hand, for sufficiently large α, since all nodes have the larger extra capacities to tolerate the load, no cascading failure occurs and the network maintains its normal and efficient functioning, but at the same time it will need greater network cost because the load capacity of each node is limited by the

Table 1: The cascading failure process and node weight ranking of the top 5 places.

Node number	Cascading failure process	R	I	Ranking	Weight
29	{27, 75} → {28, 30–32} → {35, 43} → {48–50, 59, 62–65} → {1, 5, 6, 33, 34, 36, 37, 39–41, 46, 51, 58} → {7–9, 42, 44, 45, 47, 61, 66–69} → {2, 4, 15–17, 56, 57, 74} → {38}	0.6800	1	1	0.0843
30	{27, 75} → {28, 29, 31, 32} → {35, 43, 48–50, 59, 62–65} → {1, 5, 6, 33, 34, 36, 37, 39, 40, 41, 46, 51, 58} → {7–9, 42, 44, 45, 47, 61, 66–69} → {2, 4, 15–17, 56, 57, 74} → {38}	0.6800	0.9976	2	0.0841
4	{2, 45} → {38, 44} → {32–34, 46, 48, 75} → {35, 36, 43, 47, 49, 50, 59, 62–65} → {1, 5, 6, 31, 37, 39, 40, 41, 51, 58, 74} → {7, 8, 9, 27, 42, 61} → {15–17, 28–30, 57}	0.6133	0.9551	3	0.0805
47	{46, 74} → {31, 75} → {36, 43} → {1, 32–34, 44, 45, 59} → {2, 4, 27, 35, 39, 48–50, 62–65} → {5, 6, 28–30, 37, 38, 40, 41, 51, 58} → {42, 61} → {57}	0.5333	0.9396	4	0.0786
69	{33, 56, 75} → {66, 67, 68} → {56}	0.1067	0.6573	5	0.0554

cost of the node. Thus, with the increase of the parameter α, there should be some crossover behavior of the system from large scale breakdown to no breakdown, going through small scale ones. Therefore, we use the crossover behavior to quantify the network resilience, that is, the critical threshold α_c, at which a phrase transition occurs from normal state to collapse. That is to say, when $\alpha > \alpha_c$, the ecoindustrial system maintains its normal and efficient functioning, while when $\alpha < \alpha_c$, CF^* suddenly increases from 0 and cascading failure emerges because the capacity of each node is limited, propagating the whole or part network to stop working. Therefore, α_c is the least value of protection strength to avoid cascading failure, which reflects the minimum anti-interference ability of the network. Apparently, the lower the value of α_c, the stronger the resilience of the network against cascading failure. The critical threshold α_c controls all the properties of the symbiosis network of an EIP and also quantifies the symbiosis network resilience by a transition from the normal state to collapse.

4.2. Theoretical Analysis. To explore the resilience against cascading failures for symbiosis networks of EIPs, some theoretical analysis is given below to discuss the critical threshold α_c. Taking two cases of $\gamma = 1$ and $\gamma > 1$ into account, we analyze the observed cascading phenomenon on symbiosis networks. The failure probability of an overload node is denoted by h. Based on our removal rule of an overload node, in order to avoid the occurrence of cascading failures for a neighbor node j of the removed node i, the following condition should be satisfied:

$$\begin{aligned} &L_j + \Delta L_{ij} < C_j, \quad \gamma = 1, \\ &\frac{L_j + \Delta L_{ij} - C_j}{\gamma C_j - C_j} < h, \quad \gamma > 1. \end{aligned} \tag{11}$$

According to (1) and (3) in our cascading model, the above condition (11) can be expressed as

$$\begin{aligned} &\left[k_j^{\beta} + \left(\sum_{m\in H_j} k_m\right)^{\beta}\right] + \left[k_i^{\beta} + \left(\sum_{n\in H_i} k_n\right)^{\beta}\right] \\ &\quad \times \frac{k_j^{\beta} + \left(\sum_{m\in H_j} k_m\right)^{\beta}}{\sum_{n\in H_i}\left[k_n^{\beta} + \left(\sum_{f\in H_n} k_f\right)^{\beta}\right]} \\ &\quad < (1+\alpha)\left[k_j^{\beta} + \left(\sum_{m\in H_j} k_m\right)^{\beta}\right], \quad \gamma = 1, \\ &\left(\left[k_j^{\beta} + \left(\sum_{m\in H_j} k_m\right)^{\beta}\right] + \left[k_i^{\beta} + \left(\sum_{n\in H_i} k_n\right)^{\beta}\right]\right. \\ &\quad \left.\times \frac{k_j^{\beta} + \left(\sum_{m\in H_j} k_m\right)^{\beta}}{\sum_{n\in H_i}\left[k_n^{\beta} + \left(\sum_{f\in H_n} k_f\right)^{\beta}\right]}\right) \\ &\quad \times \left((\gamma-1)(1+\alpha)\left[k_j^{\beta} + \left(\sum_{m\in H_j} k_m\right)^{\beta}\right]\right)^{-1} < h, \\ &\quad \gamma > 1. \end{aligned} \tag{12}$$

From (12), we get

$$\begin{aligned} &1 + \frac{\left[k_i^{\beta} + \left(\sum_{n\in H_i} k_n\right)^{\beta}\right]}{\sum_{n\in H_i}\left[k_n^{\beta} + \left(\sum_{f\in H_n} k_f\right)^{\beta}\right]} < 1+\alpha, \quad \gamma = 1, \\ &\frac{1 + \left[k_i^{\beta} + \left(\sum_{n\in H_i} k_n\right)^{\beta}\right] / \sum_{n\in H_i}\left[k_n^{\beta} + \left(\sum_{f\in H_n} k_f\right)^{\beta}\right]}{h\gamma - h + 1} < 1+\alpha, \\ &\quad \gamma > 1, \end{aligned} \tag{13}$$

$$\sum_{n\in H_i} k_n^{\beta} = \sum_{\tilde{k}=m}^{K} k_i P\left(\tilde{k} \mid k_i\right) \tilde{k}^{\beta}, \tag{14}$$

where m and K are the minimum degree and the maximum degree, respectively. $P(\tilde{k} \mid k_i)$ is the conditional probability that the node of degree k_i has a neighbor of degree $\tilde{k}$. It

is satisfied with the normalized condition and the balance condition which is as follows:

$$\sum_{\tilde{k}} P\left(\tilde{k} \mid k_i\right) = 1,$$
$$k_i P\left(\tilde{k} \mid k_i\right) P\left(k_i\right) = \tilde{k} P\left(k_i \mid \tilde{k}\right) P\left(\tilde{k}\right). \tag{15}$$

Based on the condition (15), equation (14) can be expressed as follows:

$$\sum_{n \in H_i} k_n{}^{\beta} = k_i \sum_{\tilde{k}=m}^{K} \frac{\tilde{k} P\left(\tilde{k}\right) \tilde{k}^{\beta}}{\langle k \rangle} = k_i \frac{\left\langle k^{\beta+1} \right\rangle}{\langle k \rangle}. \tag{16}$$

In the case of $\beta = 1$, we get

$$\sum_{n \in H_i} k_n = k_i \frac{\left\langle k^2 \right\rangle}{\langle k \rangle}. \tag{17}$$

Hence, based on (16) and (17), the condition (13) can be rewritten as follows:

$$1 + \frac{k_i{}^{\beta-1} \langle k \rangle}{k^{\beta+1}} < 1 + \alpha, \quad \gamma = 1,$$
$$\frac{1 + k_i{}^{\beta-1} \langle k \rangle / k^{\beta+1}}{h\gamma - h + 1} < 1 + \alpha, \quad \gamma > 1. \tag{18}$$

By the inequality (18), the critical threshold α_c can be gotten by considering the ranges of $\beta < 1$, $\beta = 1$, and $\beta > 1$, respectively, as follows:

when $\gamma = 1$, we have

$$\alpha_c = \begin{cases} \dfrac{m^{\beta-1} \langle k \rangle}{\left\langle k^{\beta+1} \right\rangle}, & \beta < 1, \\ \dfrac{\langle k \rangle}{\left\langle k^2 \right\rangle}, & \beta = 1, \\ \dfrac{K^{\beta-1} \langle k \rangle}{\left\langle k^{\beta+1} \right\rangle}, & \beta > 1, \end{cases} \tag{19}$$

and when $\gamma > 1$, we have

$$\alpha_c = \begin{cases} \dfrac{1 + m^{\beta-1} \langle k \rangle / \left\langle k^{\beta+1} \right\rangle}{h\gamma - h + 1} - 1, & \beta < 1, \\ \dfrac{1 + \langle k \rangle / \left\langle k^2 \right\rangle}{h\gamma - h + 1} - 1, & \beta = 1, \\ \dfrac{1 + K^{\beta-1} \langle k \rangle / \left\langle k^{\beta+1} \right\rangle}{h\gamma - h + 1} - 1, & \beta > 1. \end{cases} \tag{20}$$

For (18), we consider in the case of $\beta < 1$ that

$$\frac{m^{\beta-1} \langle k \rangle}{\left\langle k^{\beta+1} \right\rangle} = \frac{\langle k \rangle}{(1/N) \sum_{i=1}^{N} k_i^2 (k_i/m)^{\beta-1}} > \frac{\langle k \rangle}{(1/N) \sum_{i=1}^{N} k_i^2} = \frac{\langle k \rangle}{\left\langle k^2 \right\rangle}. \tag{21}$$

From the inequality (21), we find that $\alpha_c(\beta < 1) > \alpha_c(\beta = 1)$ when $\gamma = 1$ and $\gamma > 1$. Similarly, $\alpha_c(\beta > 1) > \alpha_c(\beta = 1)$ when $\gamma = 1$ and $\gamma > 1$. Compared with other values of the parameter β, the lowest value of α_c at $\beta = 1$ indicates that symbiosis systems reach the strongest resilience level against cascading failures. We know that the tunable parameter β controls the strength of the initial load of a node. Therefore, the strongest resilience at $\beta = 1$ implies an optimal initial load distribution rule, which may be useful for the EIP to avoid cascading failure triggered by production problem.

Moreover, according to (20), we find that α_c has a negative relationship with the value γ when the value β is given. That is to say, a bigger value γ leads to a stronger resilience against cascading failures. The value of γ determines to a certain extent the protection of those overload nodes. The bigger the value of the parameter γ, the higher the cost of avoiding cascading failures. This result will be very helpful for the choice on an optimal value of γ.

4.3. Numerical Simulation. The network structure plays an important role in the resilience of networks. For the symbiosis network of an EIP, if the tolerance parameter is able to be effectively selected, not only the construction cost of the ecoindustrial park will be saved, but also the overall resilience of the symbiosis network will be enhanced. Thus, many existing problems in the current planning and construction for many ecoindustrial parks will be solved. Therefore, we furthermore explore the relation between the critical threshold α_c and the other parameters in our cascading model. Figures 4–6 report the dynamical results of cascading failures in the symbiosis network of Jinjie ecoindustrial park with different values of the parameters where cascading failures are triggered by removing the nodes one by one in the set V.

We focus on the effect of the tolerance parameter α for the normalized avalanche size CF^* for different values of γ first. Figure 4 illustrates CF^* after cascading failures of all removed nodes, as a function of the tolerance parameter α under six different values of γ from Figures 4(a), 4(b), 4(c), 4(d), 4(e), and 4(f). From Figure 4, one can see that the value of CF^* increases rapidly as the decrease of the tolerance parameter from 1 to 0, which indicates that the cascading failures of symbiosis networks triggered by removing nodes can lead to serious results.

Figure 5 represents the relation between the critical threshold α_c and the tunable parameter β for different values of γ. One can see that α_c attains the minimal value when $\beta = 1$ on all the curves for different γ in Figure 5. From Figures 4-5, we also can see an interesting phenomenon in our cascading model. The symbiosis network displays its strongest resilience against cascading failures decided by the critical threshold α_c at $\beta = 1$ under six different values of γ. Therefore, the theoretical analysis result is further verified by the simulation results. These numerical simulation results agree well with the finding of theoretical analysis result. In addition, we see that the parameter γ plays an important role in the symbiosis network resilience against cascading failures in Figures 4-5.

The effect of the parameter γ on the critical threshold α_c is further reported in Figure 6. It is easy to find that the obtained α_c originating from the same value β has a negative

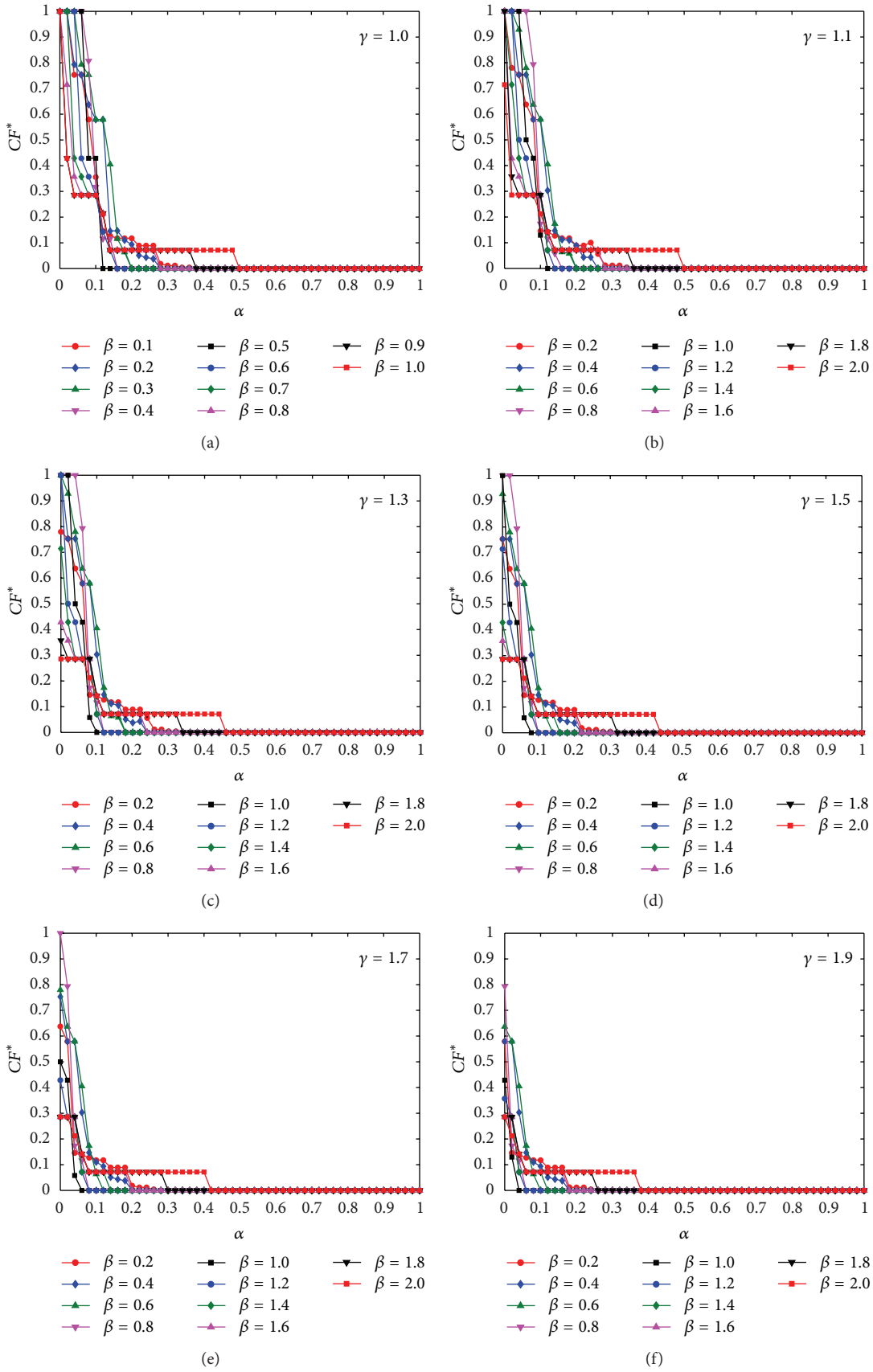


FIGURE 4: CF^* as a function of the tolerance parameter α for different values of β. (a), (b), (c), (d), (e) and (f) show six results when $\gamma = 1.0$, 1.1, 1.3, 1.5, 1.7, and 1.9, respectively.

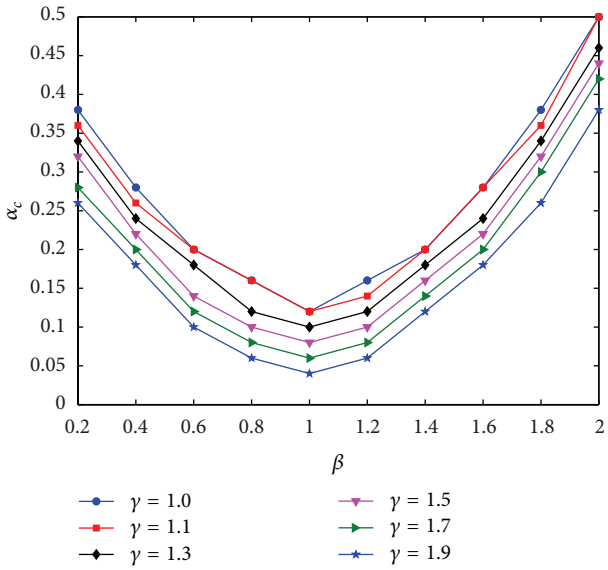


Figure 5: Relation between the critical threshold α_c and the tunable parameter β for different values of γ.

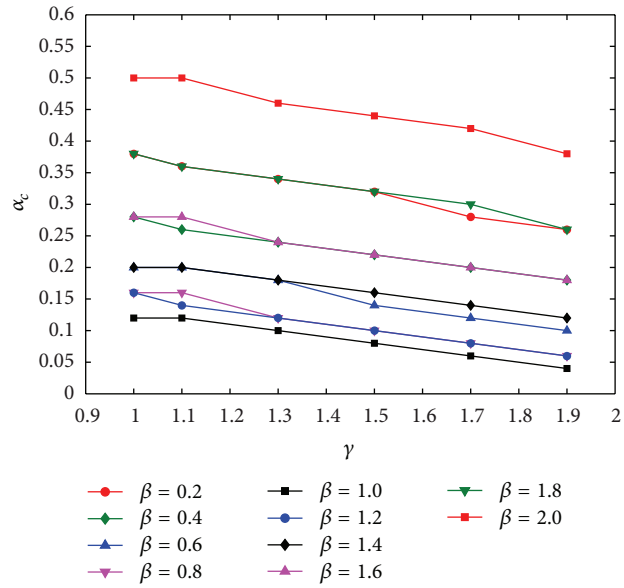


Figure 6: Relation between the critical threshold α_c and the parameter γ.

relationship with the parameter γ. From Figure 6, we can see that the curves of simulation results are in good agreement with the above theoretical analysis results and also just consistent with the actual situation. This relationship can thus supply information for monitoring and control to avoid cascading failures in the symbiosis network of Jinjie ecoindustrial park. Additionally, as is shown in Figure 6, all values on the curve $\beta = 1$ are smaller than the ones on other curves in the case of the same γ, which also verify the result that the symbiosis network reach their strongest resilience level at $\beta = 1$. Therefore, our model is reasonable and useful to improve the resilience of symbiosis networks of EIPs in the control and defense of the cascading failures.

5. Conclusions

In this paper, we construct a cascading model with tunable parameters for symbiosis networks of EIPs and explore cascading failure phenomenon in EIPs. Considering that industrial developments are usually planned to increase the recycling of wastes and by-products, we introduce the rate of ecological connectance of EIPs and propose a new evaluation method of node importance considering cascading failure according to the features of symbiosis networks of EIPs. This evaluation method can help us to find some potential critical nodes which are sensitive to the ecological degree and the functioning efficiency of networks but not so important intuitively. Furthermore, based on the proposed cascading model, we propose the critical threshold to quantify and further investigate the resilience of symbiosis networks of ecoindustrial parks. To illustrate the effectiveness and feasibility of the proposed model, we extract Jinjie ecoindustrial system in Shanxi Province of China as an undirected and unweighted symbiosis network. The cascading reaction behaviors on the symbiosis network are investigated under the cascading model.

Our results show those critical nodes in the symbiosis network of Jinjie EIP. In addition, some interesting and instructive results are obtained in our proposed cascading model. The theoretical analysis results and numerical simulations both show that the symbiosis network displays the strongest resilience level against cascading failure at $\beta = 1$ under the proposed cascading model. This result implies an optimal initial load distribution rule, which may be useful for the EIP to avoid cascading failure triggered by production problem. It is also seen that the critical threshold α_c has a negative relationship with the value γ. We know that the value of the tolerance parameter reflects the construction cost of an EIP. If the tolerance parameter is able to be effectively selected in construction and management of EIPs, we not only can save the construction cost of EIPs but also enhance the resilience for symbiosis networks of EIPs. It is very helpful for the construction of symbiosis networks of EIPs to select the optimal value of the parameters to attain the optimal resilience. With these results, many existing production problems in the planning and construction will be solved in EIPs.

Our work may have practical implications for protecting the key nodes effectively and avoiding cascading-failure-induced disasters in symbiosis networks. Moreover, the proposed cascading model has great generality for characterizing cascading-failure-induced disasters in the real world and provides a valuable method to many real-life networks. It is expected that the presented model may help us better understand the cascading phenomena in nature and enhance the resilience of the networks.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (no. 60974076).

References

[1] H. Li and R. B. Xiao, "A multi-agent virtual enterprise model and its simulation with Swarm," *International Journal of Production Research*, vol. 44, no. 9, pp. 1719–1737, 2006.

[2] S. M. Wagner and N. Neshat, "A comparison of supply chain vulnerability indices for different categories of firms," *International Journal of Production Research*, vol. 50, no. 11, pp. 2877–2891, 2012.

[3] S. P. Wen, Z. G. Zeng, and T. W. Huang, "H_∞ Filtering for neutral systems with mixed delays and multiplicative noises," *IEEE Transactions on Circuits and Systems Part II*, vol. 59, no. 11, pp. 820–824, 2012.

[4] S. P. Wen, Z. G. Zeng, and T. W. Huang, "Robust probabilistic sampling H_∞ output tracking control for a class of nonlinear networked systems with multiplicative noises," *Journal of the Franklin Institute*, vol. 350, pp. 1093–1111, 2013.

[5] S. P. Wen, Z. G. Zeng, and T. W. Huang, "Observer-based H_∞ control of discrete time-delay systems with random communication packet losses and multiplicative noises," *Applied Mathematics and Computation*, vol. 219, no. 12, pp. 6484–6493, 2013.

[6] R. Cohen, K. Erez, D. Ben-Avraham, and S. Havlin, "Resilience of the Internet to random breakdowns," *Physical Review Letters*, vol. 85, no. 21, pp. 4626–4628, 2000.

[7] R. Cohen, K. Erez, D. Ben-Avraham, and S. Havlin, "Breakdown of the internet under intentional attack," *Physical Review Letters*, vol. 86, no. 16, pp. 3682–3685, 2001.

[8] A. X. C. N. Valente, A. Sarkar, and H. A. Stone, "Two-peak and three-peak optimal complex networks," *Physical Review Letters*, vol. 92, no. 11, 4 pages, 2004.

[9] K. Burnard and R. Bhamra, "Organisational resilience: development of a conceptual framework for organisational responses," *International Journal of Production Research*, vol. 49, no. 18, pp. 5581–5599, 2011.

[10] K.-I. Goh, B. Kahng, and D. Kim, "Fluctuation-driven dynamics of the Internet topology," *Physical Review Letters*, vol. 88, no. 10, pp. 1087011–1087014, 2002.

[11] R. Albert, I. Albert, and G. L. Nakarado, "Structural vulnerability of the North American power grid," *Physical Review E*, vol. 69, no. 2, Article ID 025103, 4 pages, 2004.

[12] H. Pearson, "Protecting critical assets: the interdiction median problem with fortification," *Physical Review E*, vol. 37, no. 19, pp. 352–367, 2004.

[13] D.-O. Leonardo, J. I. Craig, B. J. Goodno, and A. Bostrom, "Interdependent response of networked systems," *Journal of Infrastructure Systems*, vol. 13, no. 3, pp. 185–194, 2007.

[14] E. A. Lowe and L. K. Evans, "Industrial ecology and industrial ecosystems," *Journal of Cleaner Production*, vol. 3, no. 1-2, pp. 47–53, 1995.

[15] J. Korhonen, "Industrial ecology in the strategic sustainable development model: strategic applications of industrial ecology," *Journal of Cleaner Production*, vol. 12, no. 8–10, pp. 809–823, 2004.

[16] M. Mirata and T. Emtairah, "Industrial symbiosis networks and the contribution to environmental innovation: the case of the Landskrona industrial symbiosis programme," *Journal of Cleaner Production*, vol. 13, no. 10-11, pp. 993–1002, 2005.

[17] A. J. P. Carr, "Choctaw Eco-Industrial Park: an ecological approach to industrial land-use planning and design," *Landscape and Urban Planning*, vol. 42, no. 2–4, pp. 239–257, 1998.

[18] T. Sterr and T. Ott, "The industrial region as a promising unit for eco-industrial development—reflections, practical experience and establishment of innovative instruments to support industrial ecology," *Journal of Cleaner Production*, vol. 12, no. 8–10, pp. 947–965, 2004.

[19] T. Graedel, B. R. Allenby, and H. Shi, *Industrial Ecology*, Tsinghua University Press, Beijing, China, 2004, (in Chinese).

[20] Y. M. Song and L. Shi, "Complexity measurements for Kalundborg and Gongyi industrial symbiosis networks," *Journal of Tsinghua University*, vol. 48, no. 9, pp. 61–64, 2008 (Chinese).

[21] R. A. Wright, R. P. Côté, J. Duffy, and J. Brazner, "Diversity and connectance in an industrial context: the case of burnside industrial park," *Journal of Industrial Ecology*, vol. 13, no. 4, pp. 551–564, 2009.

[22] A. Posch, "Industrial recycling networks as starting points for broader sustainability-oriented cooperation?" *Journal of Industrial Ecology*, vol. 14, no. 2, pp. 242–257, 2010.

[23] I. Dobson, B. A. Carreras, V. E. Lynch, and D. E. Newman, "Complex systems analysis of series of blackouts: cascading failure, critical points, and self-organization," *Chaos*, vol. 17, no. 2, Article ID 026103, 13 pages, 2007.

[24] J. F. Zheng, Z. Y. Gao, and X. M. Zhao, "Modeling cascading failures in congested complex networks," *Physica A*, vol. 385, no. 2, pp. 700–706, 2007.

[25] A. E. Motter and Y. Lai, "Cascade-based attacks on complex networks," *Physical Review E*, vol. 66, no. 6, Article ID 065102, 4 pages, 2002.

[26] D.-O. Leonardo and S. M. Vemuru, "Cascading failures in complex infrastructure systems," *Structural Safety*, vol. 31, no. 2, pp. 157–167, 2009.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 815035, 7 pages
http://dx.doi.org/10.1155/2013/815035

Research Article

Persistent Homology of Collaboration Networks

C. J. Carstens and K. J. Horadam

School of Mathematical and Geospatial Sciences, RMIT University, Melbourne, VIC 3001, Australia

Correspondence should be addressed to C. J. Carstens; corriejacobien.carstens@rmit.edu.au

Received 29 March 2013; Accepted 19 May 2013

Academic Editor: Tingwen Huang

Copyright © 2013 C. J. Carstens and K. J. Horadam. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Over the past few decades, network science has introduced several statistical measures to determine the topological structure of large networks. Initially, the focus was on binary networks, where edges are either present or not. Thus, many of the earlier measures can only be applied to binary networks and not to weighted networks. More recently, it has been shown that weighted networks have a rich structure, and several generalized measures have been introduced. We use persistent homology, a recent technique from computational topology, to analyse four weighted collaboration networks. We include the first and second Betti numbers for the first time for this type of analysis. We show that persistent homology corresponds to tangible features of the networks. Furthermore, we use it to distinguish the collaboration networks from similar random networks.

1. Introduction

Networks are a useful abstraction for many real-world systems. Some examples are the Internet, communication networks, biological networks, and social networks. Many of these networks are intrinsically weighted [1]. For instance, not all connections in social networks are equal; some people are close friends or family, whereas others are merely acquaintances. By modeling social networks as weighted networks we can analyze the richer structure of links with different strengths [1–3].

There are several approaches to analysing weighted networks. One is to suitably generalize measures for binary networks [2]. Another approach is finding the optimal threshold network for the property of interest. A threshold network is the binary network obtained by setting a threshold weight w^* and keeping only connections with weight higher than w^*. In [4], a range of different thresholds is scanned to find the optimal threshold for detecting a global community structure in a network. The authors regard changing the threshold as changing the resolution at which a network's structure is inspected.

In this paper, we take a different approach. Instead of finding the optimal threshold weight, which is often only optimal for a specific property, we study all different levels of resolution at once. To do so we use persistent homology, a recent technique from computational topology. The framework of persistent homology records structural properties and their changes for a whole range of thresholds. There are only a few other papers that use persistent homology to analyse networks [5–9] that we are aware of. Both [5, 6] use a different filtration for persistent homology from that of Lee et al.; in particular, they do not use persistent homology to analyse weighted networks. The filtration that we use is the same as in the work by Lee et al., where it was used to compare normal and abnormal brain networks [7–9]. In their work only the zeroth Betti numbers were used.

Here, we include the second and first Betti numbers for the first time. This leads to richer network measures. We show that the first Betti numbers correspond to tangible features of the network and use this richer form of persistent homology to distinguish structured networks from random networks.

2. Persistent Homology of Weighted Networks

In this section we introduce concepts from computational topology in the setting of networks. For a more elaborate introduction to persistent homology we refer to [10, 11].

2.1. Persistent Homology. Persistent homology computes the topological features of a filtration of a space. A filtration of a space can be thought of as the evolution of a space or a growing sequence of spaces. More formally a filtration of a space X is a nested sequence of subspaces beginning with the empty set and ending with X:

$$\emptyset = X_0 \subseteq X_1 \subseteq \cdots X_n = X. \tag{1}$$

See Figure 1(a) for an illustration of a filtration, where X is a triangle. Persistent homology computes the classical homology groups of spaces in such a filtration. In this paper we always use homology with $\mathbb{Z}_2$ coefficients. We write $H_i(X)$ for the ith homology group of X. (We are being sloppy with our notation here for increased readability but should in fact write $H_i(X;\mathbb{Z}_2)$.) The homology groups with coefficients in $\mathbb{Z}_2$ will always be of the form $H_i(X) \simeq \mathbb{Z}_2^{\beta_i}$ where β_i is the ith Betti number of X. We are mainly interested in computing these Betti numbers.

Using the inclusion maps $X_j \rightarrow X_{j+1}$ we can identify copies of $\mathbb{Z}_2$ in the homology groups $H_i(X_j)$ and $H_i(X_{j+1})$ of a filtration. This way we can record when a new copy is born, an existing copy persists or dies. The births and deaths correspond to changes in the topology of the filtration. These changes can be depicted as a barcode [11, 12], where the intervals $[b_k, d_k]$ correspond to filtration values of the birth and death of an element in the ith homology group. The longer a topological feature is present in the filtration, the longer we say it persists; see Figure 1(c).

Here, we will restrict our attention to the zero-, one- and two-dimensional homology of spaces. This will reduce our computations significantly, since we do not need to include parts of our space that are higher dimensional than two-dimensional. We will make this statement more precise in the following section.

It is well known that β_0 equals the number of connected components of a space [13]. β_1 and β_2 roughly count the number of loops and voids in a space. We will restrict our results to these dimensions, but there are Betti numbers for all positive $n \in \mathbb{N}$, corresponding to higher dimensional holes in a space. However, for finite spaces most of these will be zero since homology groups are zero in dimensions larger than the dimension of the space itself.

2.2. Weighted Network. A weighted graph is a graph $G = (V, E)$ together with a weight function $w : E \rightarrow \mathbb{R}$. As mentioned in the introduction, a weighted graph can be converted to an unweighted graph by keeping only the edges stronger than a certain threshold weight w^*. For a weighted graph G we will denote this threshold subgraph by $G(w^*)$. In every threshold subgraph all of the vertices of G are present.

Note that for two different thresholds $w_i^* > w_j^*$, we obtain an inclusion $G(w_i^*) \subseteq G(w_j^*)$. All edges that are present in $G(w_i^*)$ have weight larger than w_i^*, so larger than w_j^*, and thus they are included in $G(w_j^*)$. For a sequence of weights $w_0 > w_1 > \cdots > w_k$ we obtain a series of graphs and inclusions as follows:

$$\emptyset \subseteq G(w_0) \subseteq G(w_1) \subseteq \cdots \subseteq G(w_k) \subseteq G. \tag{2}$$

Such a sequence of graph inclusions is called a graph filtration.

Since a graph can be equipped with a topology to turn it into a a one-dimensional space, we can directly apply persistent homology to a graph filtration. We will then obtain nontrivial Betti numbers in dimensions zero and one only.

We can encode more of the topological information of the graph into a higher dimensional space, a simplicial complex. There are many different ways to construct a filtration of simplicial complexes from a graph filtration. A common choice is the clique complex since it reduces computational efforts [11, 14]. (The clique complex is also known as the Vietoris Rips complex and the flag complex. We use the term clique complex as it has more meaning in terms of social networks.)

We obtain the clique complex of a graph by "filling in" all cliques, that is, all complete subgraphs. A 3-clique will turn into a filled triangle and a 4-clique into a solid tetrahedron and similarly for higher dimensional cliques. A nice property of the clique complex is that cliques correspond to highly connected groups of nodes that may represent communities [4]. When computing the first Betti numbers of such a clique complex we count the number of loops in the complex. In the original graph a triangle is a loop and increases the first Betti number by one. In the clique complex all triangles are filled, and the loop is no longer there; see Figure 1(a). This means that all loops that we detect in the clique complex have four or more vertices. The simplest possible loop is formed by four vertices connected as a square with no diagonal connections.

A vertex is also known as a 0-simplex, an edge as a 1-simplex, a triangle as a 2-simplex, and a tetrahedron as a 3-simplex. A face of a simplex σ is a subsimplex of σ. For instance, a triangle has six faces, the three edges and three points in its boundary. A simplicial complex is a set of simplices such that any face of a simplex is also in the simplicial complex and such that the intersection of any two simplices is a face of both.

Let K be the clique complex of a graph G. The 0-skeleton of K is the simplicial complex consisting of just the vertices of G. The 1-skeleton of K is the set of all vertices and edges of G, that is, the graph itself. The 2-skeleton is the set of all vertices, edges, and triangles. In general the i-skeleton of a simplicial complex K is the subcomplex consisting of all j-simplices with $j \leq i$. We denote the i-skeleton by $K^{(i)}$. Notice that for G' a subgraph of G, the i-skeleton of the clique complex $K'^{(i)}$ is a subcomplex of $K^{(i)}$. This means we obtain a filtration of $K^{(i)}$ from a graph filtration of G. And in particular since all our clique complexes are finite dimensional, we obtain a filtration of K.

From the definition of homology we know that the ith homology groups of a simplicial complex and thus the ith Betti numbers are completely determined by the $(i+1)$-skeleton. In particular this means that to compute the zero-dimensional persistent homology of the clique complex of a graph, we only need the original graph filtration. This is not surprising; the graph contains all connectivity information. Filling in triangles cannot change the number of connected components. Moreover, making use of this fact we can reduce

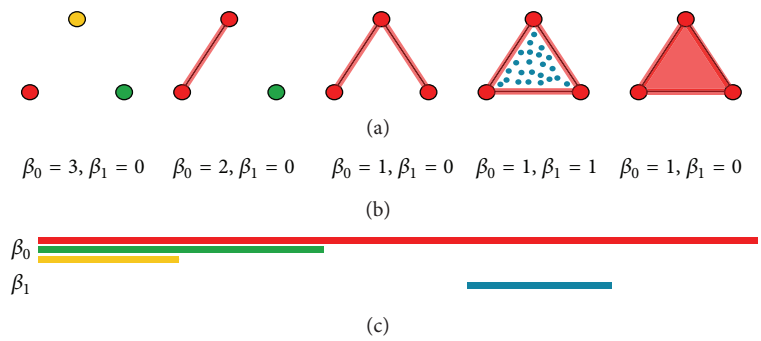


Figure 1: A filtration of a triangle (a). We start with three connected components. The yellow and the green components die in step two and three, but the red component persists the whole filtration. In the fourth step a loop is born, which dies in the final step of the filtration. The zeroth Betti number equals the number of connected components. The first Betti number equals the number of loops (b). We use a barcode to visualise the birth and death of the Betti numbers (c).

the computational times for computing Betti numbers in dimensions one and two, by only constructing the clique complex up to the 3-skeleton.

3. Collaboration Networks

We have applied persistent homology to four collaboration networks of scientists [15, 16]. These networks were obtained from http://www-personal.umich.edu/~mejn/netdata/. The four networks were constructed using four collections of papers. The vertices in the network correspond to the authors of the papers. There is a connection between two scientists if they are coauthors on at least one paper. These connections are assigned weights by taking into account how often scientist collaborate and how closely they collaborate. A paper contributes a weight to the connections between all of its authors, however the more authors a paper has the smaller the contributed weight. To be precise, a paper that has n authors contributes a weight of $1/(n-1)$. Strong connections correspond to people that collaborate often and in small groups.

Through this construction we obtain a network that has a very different weight distribution from a more traditional social network as described by Granovetter [3]. In the latter, one finds communities of strongly connected individuals and weak ties functioning as local bridges between communities.

Instead, in these collaboration networks, weak ties are necessarily part of communities. And in fact, the weaker the tie, the larger the community that it is part of. For example, let two scientists be connected by a weak tie with weight 0.125. This implies that they have coauthored a paper with at least seven other authors (they could also both have appeared on, e.g., two papers with 15 authors). Let us for simplicity assume this is the case. This paper with nine authors corresponds to a 9-clique in our network. All edges in this clique have weight larger or equal to 0.125. If we inspect edges with lower weight than 0.125 we find even more coauthors and larger cliques.

3.1. Collaboration Network of Network Scientists. We will use the network scientists data to explore β_0 and β_1 in detail. We have restricted our persistence computation to the clique complex of the largest connected component of this collaboration network. This component consists of 379 vertices and 914 edges. The weights in this component range from 0.125 to 4.75.

We will first discuss the zeroth Betti numbers of the clique complex filtration. As discussed in the previous section, we may restrict to the 1-skeleton of the complex for this computation, that is, the graph itself. We start our filtration with $w^* = 5$; all vertices of the graph are present but none of the edges are since none of the edges have weight larger than or equal to 5. We immediately find that $\beta_0 = 379$ since there are 379 connected components, all the individual vertices.

As we lower w^* in our filtration, more and more edges are added to the graph, and β_0 will decrease as the graph becomes more connected. Finally $G(0)$ is connected, so we will end with $\beta_0 = 1$. In Figure 3(a) we can see how β_0 responds to lowering the threshold weight w^*. We have also plotted the number of edges in the graph and noticed that the large decreases in number of connected components correspond to values of w^* where many edges are added.

The network is not connected while $w^* > 0.143$. Only after adding the 47 edges with weight equal to 0.143, the network becomes connected; see Table 1. Before adding these edges there are ten components, eight of these consisting of single nodes. We find that these nodes are all part of two 8-cliques; see Figure 2. These authors are only very loosely connected to the rest of the network. We expect that the largest connected component grows rapidly in the filtration and that further lowering the threshold corresponds to adding nodes that are in the periphery of the network. This requires further investigation.

We were curious to see if the zeroth Betti numbers could distinguish this collaboration network from random Erdös-Rényi graphs with the same number of nodes and edges and with the same weights assigned to the edges. We generated 1000 random graphs and used the Bottleneck distance [7, 17] between barcodes to compare the networks. We found that for the random graphs the average pairwise distance was 0.157 (s.d. 0.019) whereas the average distance from the collaboration network to the random graphs was 0.332 (s.d. 0.003). We can definitely use this measure to

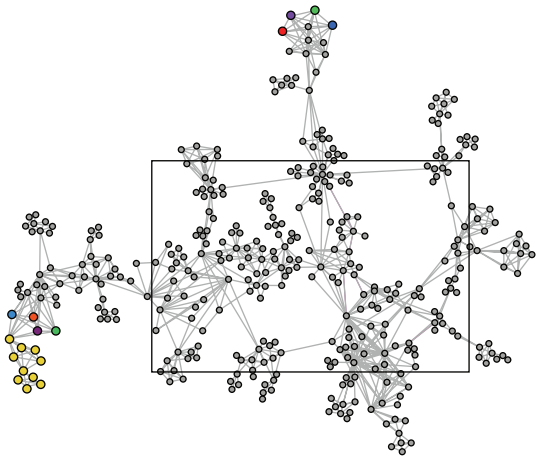


Figure 2: Largest connected component of the network science collaboration network. The enlarged nodes are the nodes that join the largest connected component at the lowest filtration value. Their colours correspond to the component they belonged to before this filtration value is reached.

distinguish the two network topologies. Even though at the start and end of the filtration these networks are very similar we can still detect a structural difference by looking at connected components during the filtration of the networks. In Figure 3(b) we have plotted the zeroth Betti numbers of ten random graphs and the zeroth Betti numbers of the networks scientist collaboration network.

Next we inspect the first Betti numbers of the clique complex associated to our network. To do so we built the 2-skeleton, which includes all vertices, edges, and triangles. Note that a filled triangle is added whenever three scientists are pairwise connected. As mentioned in the previous section, without filling in these triangles, each triple of pairwise collaborating scientists would be a loop and increase the first Betti number by one. However, we are interested in the loops in the network on a larger scale. In Figure 4 we illustrate the final stages of the graph filtration where the first Betti number is nonzero. We show the correspondence between the loops in the complex and the barcode we have computed. We find the largest loops for the lowest threshold weights, Figure 3(c). However, many of the edges that are part of these loops have high weights.

We investigated if the first Betti numbers give us further power to distinguish between the collaboration network and the random networks. We found that for random networks we obtain much higher first Betti numbers. For 1000 randomly generated networks we found an average of 520.65 (s.d. 4.39) intervals, while our structured network only has 9 intervals. The reason that this number is so much higher for random networks is that there is less clustering and thus fewer triangles that are filled in and more loops with more than three edges.

Using the first Betti numbers it is enough to only compare the final networks to distinguish between random and structured networks. We hope that the persistent homology of the whole filtration will be able to detect more subtle structural differences to distinguish networks that are more similar in structure. Notice how all of the loops that were born persisted to the end of the filtration. It would have been possible for a loop to die. For instance, if the four scientists (A. Vazquez, A. Vespignani, A. Barrat, and M. Weigt) appearing in the red loop found at $w^* = 1$ would have collaborated on a paper, there would be diagonal edges appearing at $w^* = 0.33$ which would kill the loop.

For this network all higher Betti numbers are trivial.

3.2. Physics Collaboration Networks. In this section we perform analysis on three larger collaboration networks. Again we restrict our attention to the largest connected component of each network. In Table 1 the number of nodes and edges for all of these networks are given. We computed the barcodes for the first three Betti numbers; β_0, β_1, and β_2. We found that β_0 stayed high for the largest part of the filtration and then quickly decreased to 1 at the end of the filtration in all three cases. In all cases, the smallest weight was needed to create the connected component; see Table 1. This is slightly different behaviour from the network scientist collaboration network.

We investigated if we can distinguish these collaboration networks from random networks using the persistence barcodes. We noticed that all three networks have several intervals corresponding to second Betti numbers.

Let $G(n, p)$ be an Erdös-Rényi graph with p the probability of an edge being present, that is, $p \sim 2m/n(n-1)$. Erdös and Rényi showed that if $p \gg \log n/n$ then $G(n, p)$ is almost always connected [18]. In [19], Kahle shows that there are analogous results for higher dimensional connectivity of the clique complexes of random graphs. In particular, if we define α by $p = n^{\alpha}$, Kahle shows that the kth homology group of a clique complex of a random graph is almost always zero if α is outside the interval $(-1/k, -1/(2k+1))$. In Table 2 the final values α_f for the three collaboration networks can be found.

A filtration of a random network corresponds to increasing p over time, or increasing α from $-\infty$ to α_f. For the clique complex of a random network $G(n, p)$ we expect the second Betti number to be zero for $\alpha < -0.5$. For all three networks the value of α satisfies this condition. However, we find a large number of intervals for both the condensed matter network and the astrophysics network. This clearly distinguishes these networks from random networks. Inspection of the zeroth and first Betti numbers is ongoing research.

4. Software

We used Gephi [20] for some basic graph manipulations and for graph visualisations. For the persistence computations we used javaPlex [21]. This package was developed to compute the persistent homology of point cloud data. In these computations one starts with a collection of points embedded in Euclidean space, then associates a graph filtration to these points, and finally builds a filtration of simplicial complexes of which the persistent homology is computed.

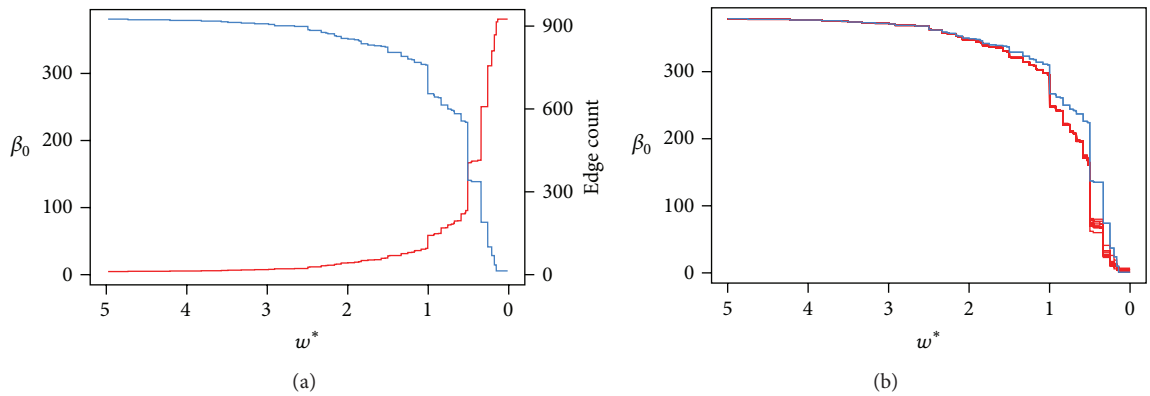
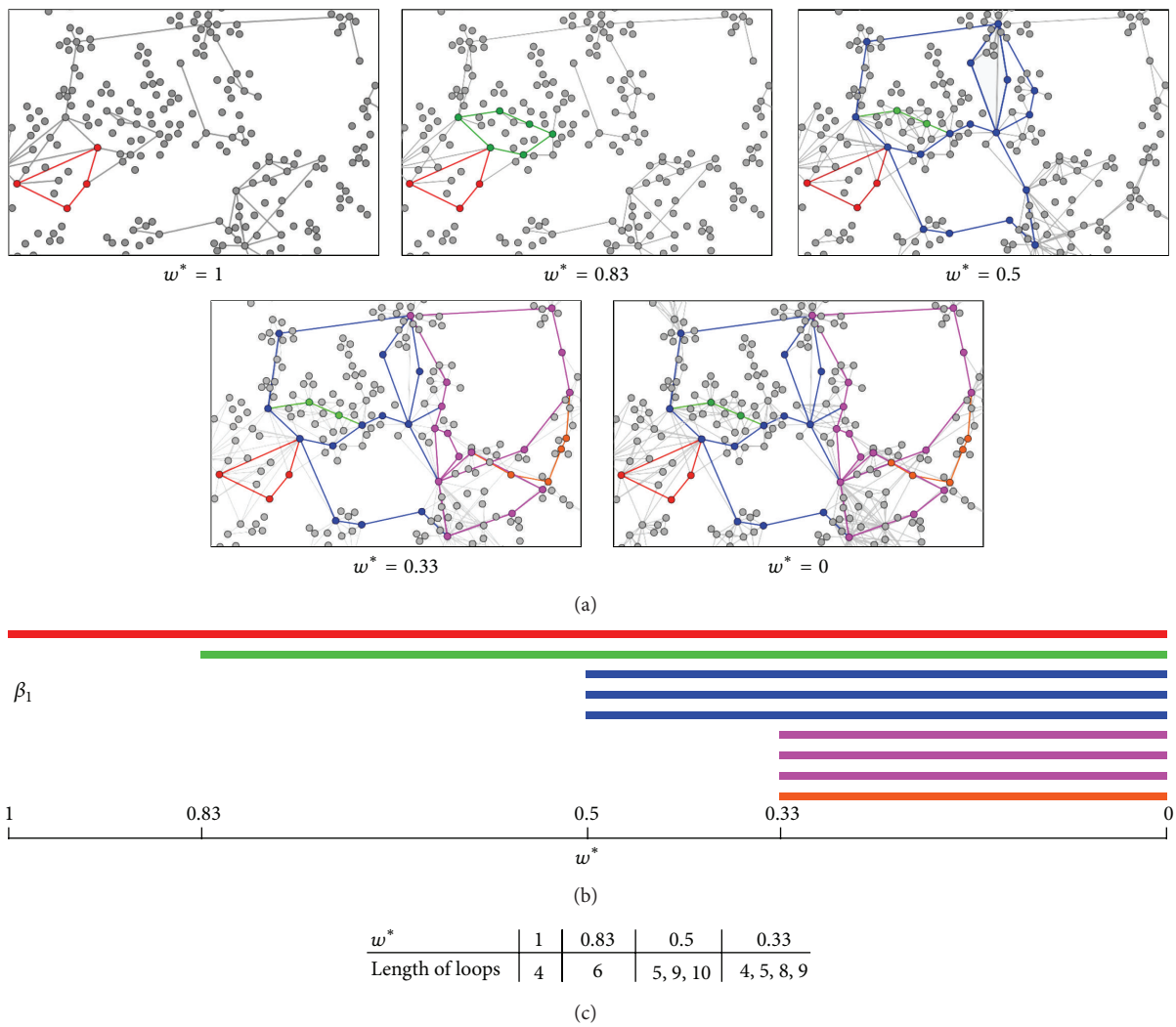


FIGURE 3: On the left (a) we plot the zeroth Betti number against the threshold w^* in blue. The total number of edges present at each stage of the filtration is plotted in red. On the right (b) we again plot the zeroth Betti number in blue. There are ten red plots, each corresponding to the sequence of zeroth Betti numbers of random graphs with the same number of vertices, edges and the same weights.



w^*	1	0.83	0.5	0.33
Length of loops	4	6	5, 9, 10	4, 5, 8, 9

(c)

FIGURE 4: We only show the central part, see Figure 2, of the network of 379 network scientists since all loops occur here (a). We find the first relatively small loop between four scientists for appearing for threshold weight 1. As we decrease the threshold weight, more loops appear. Notice how we have shaded two triangles for $w^* = 0.5$; this is to indicate that there is no loop there; there are three new blue loops added at this stage. We notice that for smaller threshold values we find larger loops. In (b) we show the barcode for the first Betti numbers of this filtration. In (c) we enlist the length of the loops that appear at each filtration value.

Table 1: Collaboration networks.

Network	No. of nodes	No. of edges	w^*	No. of edges = w^*	No. of edges < w^*
Network science	379	914	0.143	47	10
Condensed matter	36458	171735	0.034	315	0
High-energy theory	5835	13815	0.056	171	0
Astrophysics	14845	119652	0.018	357	0

Table 2: Collaboration networks.

Network	No. of intervals β_1	No. of intervals β_2	p	α
Condensed matter	11361	274	0.00026	−0.79
High-energy theory	1389	2	0.00081	−0.82
Astrophysics	4879	222	0.0011	−0.71

We wrote code in JAVA that imports a weighted edge list and converts it to a graph filtration. Subsequently we used javaPlex to build the clique complex filtration and compute the persistence intervals. The computation of the persistence intervals is the bottleneck in this computation. This took longest for the astrophysics network; 267 s (on a MacBook Pro 2.4 GHz Intel Core 2 Duo with 4 GB RAM), presumably since it is the densest network. For our current purposes these computation times are sufficient; however, if we want to apply the same computations to larger networks we need faster algorithms. This should be possible as described in Chapter 12 of [22]. To generate the random networks with n vertices and m edges we wrote code that randomly picks endpoints for m edges, avoiding double edges and loops. We used the Random Utility class from javaPlex to pick these endpoints.

5. Conclusions

By applying persistent homology to four collaboration networks of scientists we have shown that it gives us interesting information about the structure of weighted networks. We found that due to the construction of collaboration networks, weak ties form cliques and strong ties act as local bridges between those cliques. This is contrary to what has been described in other social networks. We would like to investigate this in greater detail in future work.

We used persistent homology to analyse the structure of weighted networks. The inclusion of the first and second Betti numbers gave us a richer measure to work with than in the existing literature. We were able to use persistent homology to distinguish these collaboration networks from random networks. Using the one- and two-dimensional Betti numbers of the network we did not need to take the weights into account. We are hoping that using the weights will give us the ability to distinguish networks that are more similar in structure. This is left as future work.

Acknowledgment

Research by both authors was partially supported by the Australian Department of Defence under Research Agreement 4500743680.

References

[1] M. E. J. Newman, "Analysis of weighted networks," *Physical Review E*, vol. 70, no. 5, Article ID 056131, 9 pages, 2004.

[2] T. Opsahl and P. Panzarasa, "Clustering in weighted networks," *Social Networks*, vol. 31, no. 2, pp. 155–163, 2009.

[3] M. S. Granovetter, "The strength of weak ties," *American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, 1983.

[4] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[5] V. de Silva and R. Ghrist, "Coverage in sensor networks via persistent homology," *Algebraic & Geometric Topology*, vol. 7, pp. 339–358, 2007.

[6] D. Horak, S. Maletić, and M. Rajković, "Persistent homology of complex networks," *Journal of Statistical Mechanics*, vol. 2009, no. 3, Article ID P03034, 2009.

[7] H. Lee, H. Kang, M. K. Chung, B. N. Kim, and D. S. Lee, "Persistent brain network homology from the perspective of dendrogram," *IEEE Transactions on Medical Imaging*, vol. 31, no. 12, pp. 2267–2277, 2012.

[8] H. Lee, M. K. Chung, H. Kang, B.-N. Kim, and D. S. Lee, "Discriminative persistent homology of brain networks," in *Proceedings of the 8th IEEE International Symposium on Biomedical Imaging: from Nano to Macro (ISBI '11)*, pp. 841–844, April 2011.

[9] H. Lee, M. K. Chung, H. Kang, B. N. Kim, and D. S. Lee, "Computing the shape of brain networks using graph filtration and Gromov-Hausdorff metric," in *Proceedings of the 14th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI '11)*, vol. 6891, pp. 289–296, 2011.

[10] H. Edelsbrunner and J. Harer, "Persistent homology—a survey," in *Surveys on Discrete and Computational Geometry. Twenty Years Later*, vol. 453 of *Contemporary Mathematics*, pp. 257–282, American Mathematical Society, Providence, RI, USA, 2008.

[11] G. Carlsson, "Topology and data," *Bulletin American Mathematical Society*, vol. 46, no. 2, pp. 255–308, 2009.

[12] R. Ghrist, "Barcodes: the persistent topology of data," *Bulletin American Mathematical Society*, vol. 45, no. 1, pp. 61–75, 2008.

[13] A. Hatcher, *Algebraic Topology*, Cambridge University Press, Cambridge, Mass, USA, 2002.

[14] A. Zomorodian, "Fast construction of the Vietoris-Rips complex," *Computers & Graphics*, vol. 34, no. 3, pp. 263–271, 2010.

[15] M. E. J. Newman, "The structure of scientific collaboration networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 2, pp. 404–409, 2001.

[16] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, Article ID 036104, 19 pages, 2006.

[17] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of persistence diagrams," *Discrete & Computational Geometry*, vol. 37, no. 1, pp. 103–120, 2007.

[18] P. Erdős and A. Rényi, "On random graphs. I," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

[19] M. Kahle, "Topology of random clique complexes," *Discrete Mathematics*, vol. 309, no. 6, pp. 1658–1671, 2009.

[20] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: an open source software for exploring and manipulating networks," 2009, http://gephi.org.

[21] A. Tausz, M. Vejdemo-Johansson, and H. Adams, "Javaplex: a research software package for persistent (co)homology," 2011, http://code.google.com/p/javaplex/.

[22] A. J. Zomorodian, *Topology for Computing*, vol. 16 of *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press, New York, NY, USA, 2005.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 429402, 7 pages
http://dx.doi.org/10.1155/2013/429402

Research Article

Memristive Chebyshev Neural Network and Its Applications in Function Approximation

Lidan Wang, Meitao Duan, and Shukai Duan

School of Electronic and Information Engineering, Southwest University, Chongqing 400715, China

Correspondence should be addressed to Shukai Duan; duansk@swu.edu.cn

Received 1 February 2013; Accepted 22 April 2013

Academic Editor: Chuandong Li

Copyright © 2013 Lidan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel Chebyshev neural network combined with memristors is proposed to perform the function approximation. The relationship between memristive conductance and weight update is derived, and the model of a single-input memristive Chebyshev neural network is established. Corresponding BP algorithm and deriving algorithm are introduced to the memristive Chebyshev neural networks. Their advantages include less model complexity, easy convergence of the algorithm, and easy circuit implementation. Through the MATLAB simulation results, we verify the feasibility and effectiveness of the memristive Chebyshev neural networks.

1. Introduction

Many researchers have further studied the approximation capacity of neural networks. Research indicates that there are some drawbacks in Back Propagation (BP) neural networks, such as slow convergence, getting easily into a local minimum, and difficulties in determining the numbers of hidden neurons. The Radical Basis Function (RBF) neural networks are superior to BP neural networks in approximation capacity because their learning rate is fast, but the lack of theoretical guidance in determining the central position of basis function makes us set the central position based on experience. We cannot ensure whether the performance of networks is the best or not. Especially, there is a dilemma between approximation accuracy and networks complexity when we use the aforementioned two neural networks to approximate the nonlinear function. However, the Chebyshev neural networks can solve the bottleneck; their hidden layer neuron activation functions are a group of Chebyshev orthogonal polynomials. Nowadays, Chebyshev neural networks are widely used in complex nonlinear systems [1], chaos systems [2], and discrete-time nonlinear systems [3]. Their learning rate and approximation accuracy are better than traditional neural networks, and they can quickly determine the numbers of hidden neurons. Meanwhile, Chebyshev neural networks are widely used in aerospace [4, 5] and chemistry areas [6]. The nanoscale memristor has the potential of information storage because of the nonvolatility with respect to long periods of power-down, so it can be used as synapse in the neural networks. Many researchers in the field of memristors have suggested that this device has high potential for implementing artificial synapses. Afifi et al. studied the realization of STDP learning rules based on the pulsing neuromorphic networks with memristor cross array [7]. Hu et al. built a novel chaotic neural work with memristor to implement associative memory [8]. Wang et al. proposed a PID controller based on memristive CMAC network [9]. Pershin and di Ventra realized the associative memory based on memristive neural networks [10]. Sharifi and Banadaki applied the memristors as memory units to the nonvolatile RAMs and as synapse to the artificial neural networks [11]. Chabi and Klein put forward a neural network with high fault tolerance based on a structure of crossbar switches [12]. Cantley et al. built neural networks with synapses made up from amorphous silicon thin film transistor and memristor, which realized the Hebb learning rules [13]. Gao et al. designed cellular neural network, which is applied to image denoising and edge detection [14]. Kim et al. changed the synapse weight of artificial neural networks with a pulse-based programmable memristor [15].

The memristive Chebyshev neural networks have the following advantages: (a) single-input memristive Chebyshev

neural networks can realize any nonlinear function approximation by only three layers; (b) the number of hidden neurons in memristive Chebyshev neural networks is significantly smaller than that in traditional BP neural networks; (c) we only need to adjust the weight from the hidden layer to the output layer, which can greatly help in quickening the convergence of the algorithm; (d) as memristive synapse is small and passive, the hardware circuits of the memristive Chebyshev neural networks can be implemented by the VLSI circuits easily.

In this paper, we establish the correspondence between memristor conductance and synapse weight through theoretical analysis and MATLAB simulations. The nanoscale memristor is the synapse in Chebyshev neural networks to realize function approximation by memristive BP algorithm and memristive deriving algorithm, respectively.

2. Memristive Synapse

The physical model of the memristor [16] is shown in Figure 1. $W(t)$ and D represent the thickness of the doped layer and total oxide films, respectively. R_{ON} is the value of the memristor when $W(t)$ is equal to D, and R_{OFF} is the value of the memristor when $W(t)$ is equal to zero. $M(0)$ is the initial value of the memristor.

According to the mathematical model of HP memristor, and the flux-controlled model of memristor, we can obtain the following equation [17]:

$$M(t) = \begin{cases} R_{\mathrm{OFF}}, & \varphi(t) < c_1, \\ \sqrt{2k\varphi(t) + M^2(0)}, & c_1 \le \varphi(t) < c_2, \\ R_{\mathrm{ON}}, & \varphi(t) \ge c_2, \end{cases} \tag{1}$$

where $k = (R_{\mathrm{ON}} - R_{\mathrm{OFF}})\mu_v R_{\mathrm{ON}}/D^2$, $c_1 = (R_{\mathrm{OFF}}^2 - M^2(0))/2k$, $c_2 = (R_{\mathrm{ON}}^2 - M^2(0))/2k$. μ_v is the average drift rate of oxygen vacancies, so we can obtain the following equation from (1):

$$G(t) = \frac{1}{M(t)} = \begin{cases} \dfrac{1}{R_{\mathrm{OFF}}}, & \varphi(t) < c_1, \\ \dfrac{1}{\sqrt{2k\varphi(t) + M^2(0)}}, & c_1 \le \varphi(t) < c_2, \\ \dfrac{1}{R_{\mathrm{ON}}}, & \varphi(t) \ge c_2. \end{cases} \tag{2}$$

The derivate of equation (2) is

$$\frac{dG(t)}{dt} = \begin{cases} \left[2k\varphi(t) + M^2(0)\right]^{(-3/2)} \\ \quad \times (-k) \times \dfrac{d\varphi(t)}{dt}, & c_1 \le \varphi(t) \le c_2, \\ 0, & \text{otherwise}. \end{cases} \tag{3}$$

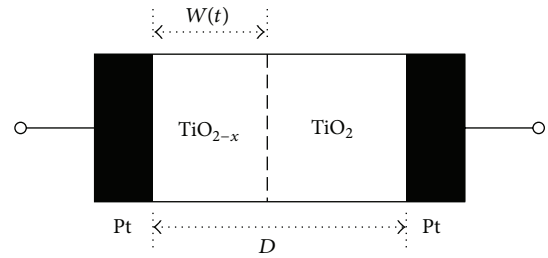


Figure 1: The physical model of the memristor of HP [16].

Using calculus, $dG(t)$ is approximately equal to ΔG when Δt approaches to zero, and then we can obtain the following equation:

$$\Delta G(t) = \begin{cases} \left[2k\varphi(t) + M^2(0)\right]^{(-3/2)} \\ \quad \times (-k) \times \dfrac{d\varphi(t)}{dt} \times \Delta t, & c_1 \le \varphi(t) \le c_2, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where $d\varphi/dt = v(t)$; the relationship curve between memristive conductance change and voltage is shown in Figure 2. It can be seen that memristive conductance change ΔG increases along with the increasing of applied voltage v. If the learning error e of neural networks is regarded as voltage v, the memristive conductance change can reasonably be described as synapse weight update whose correspondence can be built.

3. Modeling of the Memristive Chebyshev Neural Networks

The nth power orthogonal polynomials $T_n(x)$ which are related to the weight function $\rho(x) = 1/\sqrt{1-x^2}$ are called Chebyshev polynomials of the first kind [18]. Based on definition, if the polynomial system $\{T_n(x)\}$ (where $n = 0, 1, 2, \ldots$) and the weight function satisfy the following relationship:

$$\int_{-1}^{1} \rho(x) T_l(x) T_k(x)\, dx = \begin{cases} 0, & l \ne k, \\ \displaystyle\int_{-1}^{1} \rho(x) T_k^2(x) > 0, & l = k, \end{cases} \tag{5}$$

the orthogonal polynomial system $\{T_n(x)\}$ can be structured in the interval of $[-1, 1]$.

In this paper, a model (see Figure 3) of the single-input memristive Chebyshev neural networks is proposed, which consists of the input layer, hidden layer, and output layer. The weights from input layer to hidden layer are set to 1, and the memristor conductance stores the weights of hidden layer to output layer. The thresholds of all neurons are set to 0. The hidden layer has m neurons, and their transfer functions are a group of Chebyshev orthogonal polynomial functions $T_n(x)$.

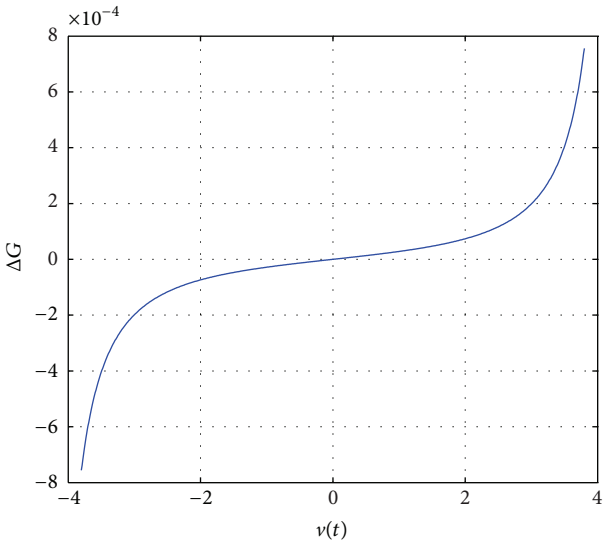


FIGURE 2: The relation curve of the change of memristive conductance and voltage.

Namely, the hidden layer of the nth neuron transfer function is $T_{n-1}(x)$, where

$$T_0(x) = 1,$$
$$T_1(x) = x, \tag{6}$$
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n = 1, 2, \ldots$$

4. Realization of Function Approximation Algorithm

Set memristive Chebyshev neural networks model as follows:

input layer: $o = x$;

input of hidden layer neuron: $\text{net}_i = o$;

output of hidden layer neuron: $o_i = T_i(\text{net}_i)$;

output layer:

$$y = \sum_{i=0}^{m-1} W_i O_i = \sum_{i=0}^{m-1} W_i T_i(x). \tag{7}$$

Samples $(x_t, f(x_t))$, $t = 1, 2, \ldots, s$ (s is the number of the samples), are assumed as inputs, and the errors of network output y and the target value $f(x)$ are $e_t = f(x_t) - y_t$.

Network training index is

$$E = \frac{1}{2}\sum_{t=1}^{s} e_t^2. \tag{8}$$

4.1. Memristive BP Algorithm. In order to establish the relationship between the memristor and the synapse, we propose the learning rule for memristive synaptic weight update. According to (4), we assume the memristive BP algorithm as follows:

$$\Delta W_i = \left[2k\Gamma(t) + M^2(0)\right]^{(-3/2)} \times (-k) \times e_t(t) \times \Delta t \times T_i(x_t),$$
$$W_i(K+1) = W_i(K) + \Delta W_i(K), \tag{9}$$

where $i = 0, 1, 2, \ldots, m-1$, $\Delta t = \eta$ is the learning rate, K is the number of learning, and Γ is the integral of the error e.

The learning algorithm is described as follows.

Step 1. Take any number of hidden neurons $n \geq 3$, choose the initial weights $W_i(0)$ as initial memristor conductance, let the learning rate $0 < \eta < 1$, give the small positive number ε, and set the training sample set $(x_t, f(x_t))$ as $E = 0$, $t = 1$, $K = 0$.

Step 2. Calculate

$$y_t(K) = \sum_{i=0}^{m-1} W_i T_i(x_t),$$
$$e_t = f(x_t) - y_t(K), \qquad E \longleftarrow E + 0.5e_t^2. \tag{10}$$

Step 3. Adjust weight,

$$W_i(K+1) = W_i(K) + \Delta W_i(K). \tag{11}$$

Step 4. $t \leftarrow t + 1$; if $t < s$, then jump back to Step 2, otherwise Step 5.

Step 5. If $E \leq \varepsilon$, then stop the learning; otherwise, $E = 0$, $t = 1$, $K \leftarrow K + 1$, and jump back to Step 2.

4.2. Memristive Deriving Algorithm. The basic principles of the memristive deriving algorithm are as follows. Take any less n hidden neurons as the initial cells. After training K times, the error e *is* no longer changed; that is, $E(K) = E(K-1) > \varepsilon$ (ε is a pretraining given precision index). The network derives automatically by adding a hidden neuron ($n \leftarrow n+1$). The previous process is repeated until $E(K) \leq \varepsilon$ when the hidden neurons stop deriving and learning. Until reaching target accuracy, the neural network topology is automatically generated.

The specific algorithm can be described as follows.

Step 1. Take number of hidden neurons $n = 2$, choose the initial weights $W_{ij}(0)$ and $C_j(0)$ as initial memristor conductance, let the learning rate $0 < \eta < 1$, give the small positive number $\varepsilon > 0$, and set the training set $\{(x_t, d_t), t = 1, 2, \ldots, s\}$ and the number of learning $K = 1$.

Step 2. Input of hidden layer neuron is

$$\text{net}_j = \sum_{i=1}^{m} W_{ij} x_i, \quad j = 0, 1, 2, \ldots, n-1. \tag{12}$$

Calculate

$$y_t(K) = \sum_{j=0}^{n} C_j(K-1) T_j(\text{net}_j), \qquad e_t = d_t - y_t. \tag{13}$$

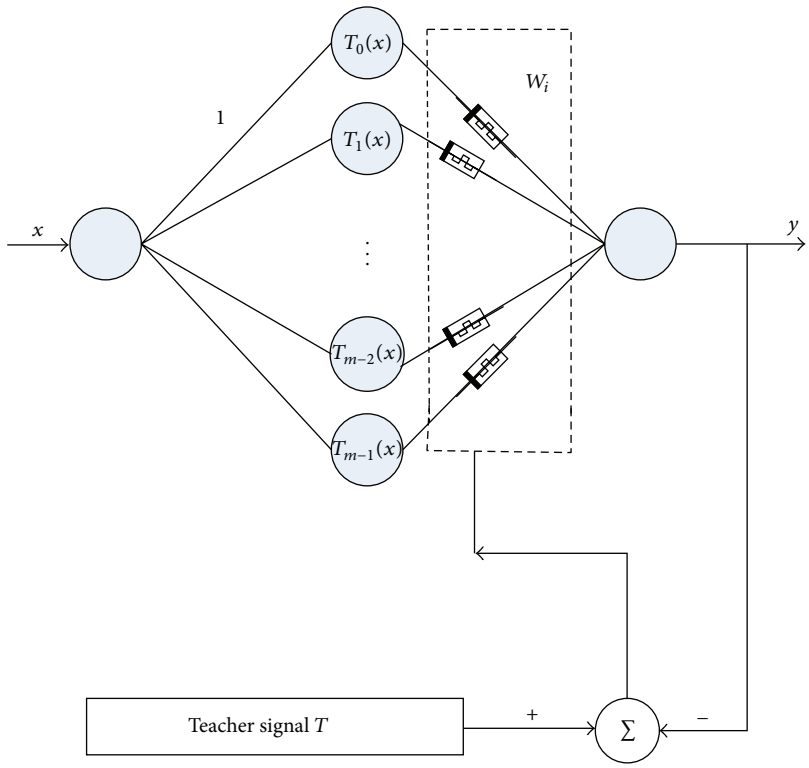


Figure 3: Model of single-input memristive Chebyshev neural networks.

Step 3. Weight update rule is

$$\begin{aligned}
\Delta C_j(K) &= \left[2k\Gamma(t) + M^2(0)\right]^{(-3/2)} \\
&\quad \times (-k) \times e_t(t) \times \Delta t \times T_i\left(\mathrm{net}_j\right), \\
C_j(K) &= C_j(K-1) + \Delta C_j(K), \\
\Delta W_{ij}(K) &= \left[2k\Gamma(t) + M^2(0)\right]^{(-3/2)} \\
&\quad \times (-k) \times e_t(t) \times C_j \Delta t \times T_j'\left(\mathrm{net}_j\right) \times x_i, \\
W_{ij}(K+1) &= W_{ij}(K-1) + \Delta W_{ij}(K).
\end{aligned} \tag{14}$$

Step 4. Calculate

$$E(K) = \frac{1}{2}\sum_{t=1}^{s} e_t^2. \tag{15}$$

If $E(K) \le \varepsilon$, stop derivative and learning which illustrate that neural network topology is automatically generated. Otherwise, go to Step 5.

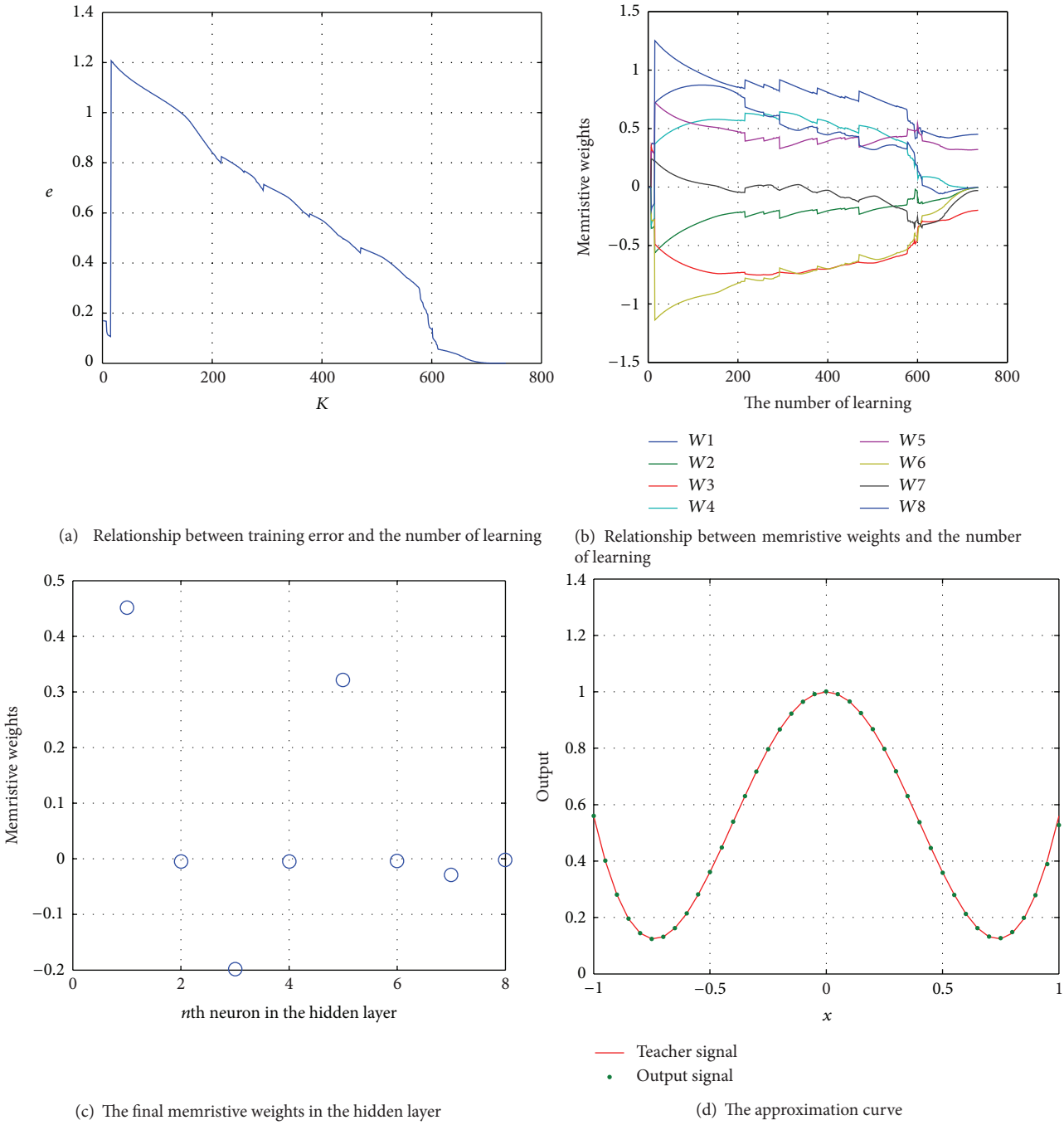
Step 5. If $E(K) = E(K-1)$, the training error of the network has no longer reduced. We should increase the number of hidden neurons in order to improve network performance. Namely, $n \leftarrow n+1$, $K \leftarrow K+1$, and jump to Step 2 to continue learning.

5. Experimental Results

We use the memristive Chebyshev neural networks to realize nonlinear function $y = x^3 \sin x + \sin^2 x + \cos(3x)$ approximation through memristive BP algorithm and deriving algorithm, respectively. We suppose that memristor model parameters are $R_{\mathrm{ON}} = 100\,\Omega$, $R_{\mathrm{OFF}} = 10000\,\Omega$, $M(0) = 1000\,\Omega$, $D = 10\,\mathrm{nm}$, $\mu_v = 10^{-14}\,\mathrm{m}^2\,\mathrm{s}^{-1}\,\mathrm{V}^{-1}$, $G(0) = 1/M(0)$.

5.1. Memristive BP Algorithm Realizes Function Approximation. The $1 \times 8 \times 1$ network structure is shown in Figure 3. The learning rate $\eta = 0.02$. The number of samples $s = 200$. The number of learning is 1000. The target mean square error $E = 1.0 \times 10^{-5}$. The experimental result is shown in Figure 4. The relationship between training error and the number of learning is shown in Figure 4(a). The number of final learning is 735 when the minimum training error meets the requirements of function approximation. The memristive synaptic weights are constantly updated in the learning process as shown in Figure 4(b). Figures 4(c) and 4(d) describe the final memristive weights and the approximation curve of actual output signal relative to the teacher signal, respectively.

5.2. Memristive Deriving Algorithm Realizes Function Approximation. We adopt the $1 \times 2 \times 1$ initial neural network, with $\varepsilon = 1.0 \times 10^{-5}$ and learning rate 0.02, and the experimental result is shown in Figure 5. The final network structure is $1 \times 25 \times 1$, and the time of learning is 420. Figure 5(a) shows the final memristive synaptic weights in hidden layer. Figure 5(b)



(a) Relationship between training error and the number of learning

(b) Relationship between memristive weights and the number of learning

(c) The final memristive weights in the hidden layer

(d) The approximation curve

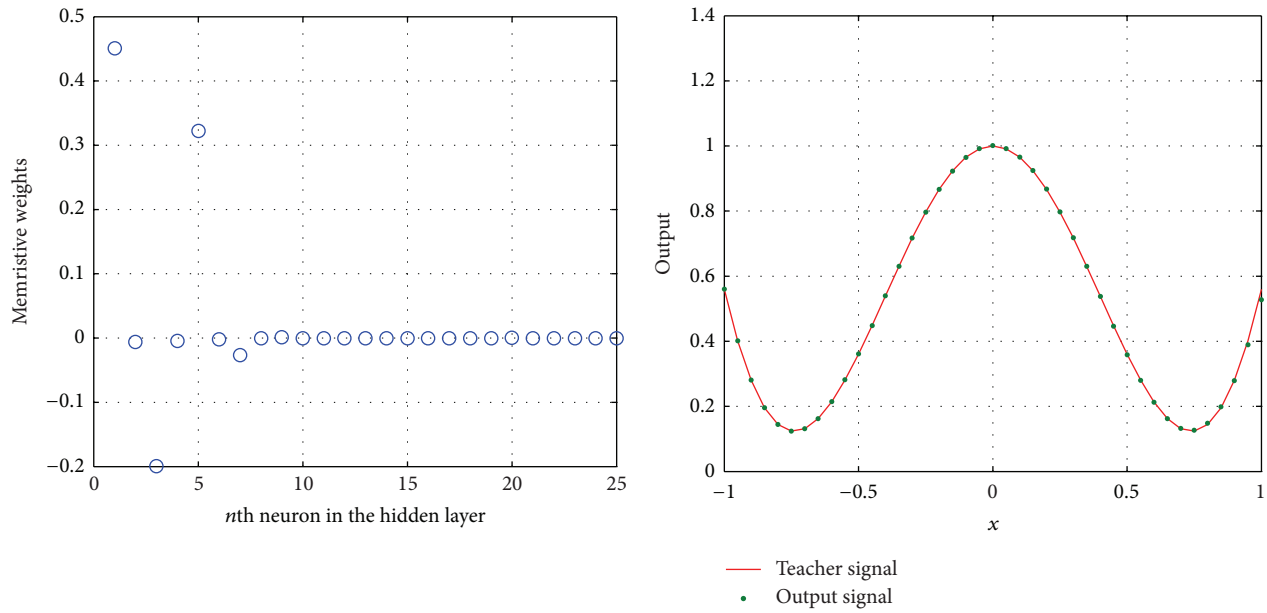
Figure 4: Simulation results for memristive BP algorithm.

shows the approximation curve of the actual output signal relative to the teacher signal.

Compared with the memristive BP algorithm, the memristive deriving algorithm has less learning times (735 − 420 = 315) for its neural network is automatically generated.

6. Conclusions

The HP memristor model is analyzed, and the correspondence between the memristive conductance change and the synapse weight update in Chebyshev neural networks is established. Two learning algorithms of memristive synapse weight update are proposed. The nonlinear function approximation is implemented by memristive BP algorithm and memristive deriving algorithm, respectively. On the one hand, the memristive BP networks have fixed structure which is easily implemented. On the other hand, the memristive derivative networks have dynamic structure reaching the optimum solution. Memristive Chebyshev neural networks are hopeful to be used in aspects of pattern recognition and data compression.



(a) The final memristive weights in the hidden layer

(b) The approximation curve

Figure 5: Simulation result for memristive deriving algorithm.

Acknowledgments

The work was supported by National Natural Science Foundation of China (Grant nos. 60972155 and 61101233), Fundamental Research Funds for the Central Universities (Grant nos. XDJK2012A007 and XDJK2013B011), University Excellent Talents Supporting Foundations of Chongqing (Grant no. 2011-65), University Key Teacher Supporting Foundations of Chongqing (Grant no. 2011-65), Technology Foundation for Selected Overseas Chinese Scholars, Ministry of Personnel in China (Grant no. 2012-186), National Science Foundation for Postdoctoral Scientists of China (Grant no. CPSF20100470116), and "Spring Sunshine Plan" Research Project of Ministry of Education of China (Grant no. z2011148).

References

[1] S. Purwar, I. N. Kar, and A. N. Jha, "On-line system identification of complex systems using Chebyshev neural networks," *Applied Soft Computing Journal*, vol. 7, no. 1, pp. 364–372, 2007.

[2] P. Akritas, I. Antoniou, and V. V. Ivanov, "Identification and prediction of discrete chaotic maps applying a Chebyshev neural network," *Chaos, Solitons and Fractals*, vol. 11, no. 1, pp. 337–344, 2000.

[3] A. K. Shrivastava and S. Purwar, "State feedback and output feedback tracking control of discrete-time nonlinear system using Chebyshev neural networks," in *Proceedings of the International Conference on Power, Control and Embedded Systems (ICPCES '10)*, pp. 1–6, Allahabad, India, December 2010.

[4] A. M. Zou, K. D. Kumar, and Z. G. Hou, "Quaternion-based adaptive output feedback attitude control of spacecraft using chebyshev neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 9, pp. 1457–1471, 2010.

[5] A. M. Zou, K. D. Kumar, Z. G. Hou, and X. Liu, "Finite-time attitude tracking control for spacecraft using terminal sliding mode and chebyshev neural network," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 41, no. 4, pp. 950–963, 2011.

[6] S. P. Yan, M. F. Peng, J. F. Lei et al., "CO_2 concentration detection based on Chebyshev neural network and best approximation theory," *Instrument Technique and Sensor*, vol. 6, pp. 107–110, 2011.

[7] A. Afifi, A. Ayatollahi, and F. Raissi, "Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nano circuits," in *Proceedings of the European Conference on Circuit Theory and Design Conference Program (ECCTD '09)*, pp. 563–566, Antalya, Turkey, August 2009.

[8] X. F. Hu, S. K. Duan, and L. D. Wang, "A novel chaotic neural network using memristors with applications in associative memory," *Abstract and Applied Analysis*, vol. 2012, Article ID 405739, 19 pages, 2012.

[9] L. D. Wang, X. Y. Fang, S. K. Duan et al., "PID controller based on memristive CMAC network," *Abstract and Applied Analysis*, vol. 2013, Article ID 510238, 2013.

[10] Y. V. Pershin and M. di Ventra, "Experimental demonstration of associative memory with memristive neural networks," *Neural Networks*, vol. 23, no. 7, pp. 881–886, 2010.

[11] M. J. Sharifi and Y. M. Banadaki, "General spice models for memristor and application to circuit simulation of memristor-based synapses and memory cells," *Journal of Circuits, Systems and Computers*, vol. 19, no. 2, pp. 407–424, 2010.

[12] D. Chabi and J. O. Klein, "Hight fault tolerance in neural crossbar," in *Proceedings of the 5th Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS '10)*, pp. 1–6, Hammamet, Tunisia, March 2010.

[13] D. K. Cantley, A. Subramaniam, H. J. Stiegler et al., "Hebbian learning in spiking neural networks with nanocrystalline silicon

TFTs and memristive synapses," *IEEE Transactions on Nanotechnology*, vol. 10, no. 5, pp. 1066–1073, 2011.

[14] S. Y. Gao, S. K. Duan, and L. D. Wang, "On memristive cellular neural network and its applications in noise removal and edge extraction," *Journal of Southwest University*, vol. 33, no. 11, pp. 63–70, 2011.

[15] H. Kim, M. P. Sah, C. J. Yang, T. Roska, and L. O. Chua, "Neural synaptic weighting with a pulse-based memristor circuit," *IEEE Transactions on Circuits and Systems I*, vol. 59, no. 1, pp. 148–158, 2012.

[16] D. B. Strukov, G. S. Snider, D. R. Stewart et al., "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 2008.

[17] L. D. Wang, E. Drakakis, S. K. Duan et al., "Memristor model and its application for chaos generation," *International Journal of Bifurcation and Chaos*, vol. 22, no. 8, Article ID 1250205, 14 pages, 2012.

[18] G. D. Mo and K. D. Liu, *Function Approximation Methods*, Science Press, Beijing, China, 2003.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 161670, 7 pages
http://dx.doi.org/10.1155/2013/161670

Research Article

A New Multiobjective Evolutionary Algorithm for Community Detection in Dynamic Complex Networks

Guoqiang Chen,[1,2] Yuping Wang,[1] and Jingxuan Wei[1]

[1] *School of Computer Science and Technology, Xidian University, Xi'an 710071, China*
[2] *School of Computer and Information Engineering, Henan University, Kaifeng, Henan 475004, China*

Correspondence should be addressed to Guoqiang Chen; xidian20120222@163.com

Received 26 March 2013; Accepted 19 May 2013

Academic Editor: Tingwen Huang

Copyright © 2013 Guoqiang Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Community detection in dynamic networks is an important research topic and has received an enormous amount of attention in recent years. Modularity is selected as a measure to quantify the quality of the community partition in previous detection methods. But, the modularity has been exposed to resolution limits. In this paper, we propose a novel multiobjective evolutionary algorithm for dynamic networks community detection based on the framework of nondominated sorting genetic algorithm. Modularity density which can address the limitations of modularity function is adopted to measure the snapshot cost, and normalized mutual information is selected to measure temporal cost, respectively. The characteristics knowledge of the problem is used in designing the genetic operators. Furthermore, a local search operator was designed, which can improve the effectiveness and efficiency of community detection. Experimental studies based on synthetic datasets show that the proposed algorithm can obtain better performance than the compared algorithms.

1. Introduction

Many real-world complex systems take the form of networks. Acquaintance networks, Internet, power grids, and neural networks are some examples. Networks could be modeled as graphs, where the individual objects are represented by nodes and the interactions among these objects are represented by edges. Community structure, that is, the vertices in networks are often found to cluster into tightly knit groups with a high density of within-group edges and a lower density of between-group edges [1], is one important property of these networks. The detection of such a community structure has great practical meaning.

Traditional analysis of community detection treats the network as a static graph, where the static graph is either derived from aggregation of data over all time or taken as a snapshot of data at a particular time. Researchers have successively proposed many effective detection methods of static network. In fact, dynamic networks capture the modifications of interconnections over time, which allow tracing the changes of network structure at different time steps. Community detection in dynamic networks is attracting increasing interest.

Recently, a framework called temporal smoothness is applied to solve community detection in dynamic networks. In this framework, it is not desirable that the significant changes of clusters structure in a short time period [2]. In order to smooth each community over time, it needs to trade off two competing objectives of snapshot quality, which is that the clustering should reflect as accurately as possible the data coming during the current time step, and temporal cost, which is that each clustering should not shift dramatically from one time step to the successive one. Using this idea, Folino and Pizzuti proposed a multiobjective approach named DYN-MOGA to discover communities in dynamic networks by employing genetic algorithms [3]. In DYN-MOGA, Community Score (CS) and Normalized Mutual Information (NMI) were selected as two objectives to be optimized simultaneously. At the end of each timestamp, *DYN-MOGA* returns a set of solutions contained in the Pareto front. They adopted modularity as a criterion to automatically select one solution with respect to another. Following this

work, Gong et al. introduced a novel multiobjective immune algorithm with local search to solve the community detection problem in dynamic networks [4]. They adopted Modularity and NMI as two objectives to optimize.

But, the modularity has been exposed to resolution limits [5–7]. Fortunato and Barthélemy [5] have recently found that modularity optimization may fail to identify modules smaller than a scale which depends on the total size of the network and on the degree of interconnectedness of the modules, even in cases where modules are unambiguously defined. To overcome the limitations of modularity function, a new measure, named by modularity density, was presented in [8].

In this paper, we present a novel multiobjective algorithm, named DNCD-MOEA, to detect community in dynamic networks. The algorithm adopts modularity density as one objective to measure how well the clustering found represents the data at the current time, and it adopts NMI as another objective to measure the distance between two clusterings at consecutive time steps. Based on the problem-specific knowledge, the new genetic operators are designed to solve the proposed model. In order to improve the quality of the solutions, a local search operator is designed.

The outline of the paper is as follows. Section 2 introduces related background. In Section 3 we present our algorithm and explain its steps. Experimental studies are presented in Section 4. Section 5 concludes this paper.

2. Background

2.1. Notation. We define a static network N_t at time t as $N_t = (V_t, E_t)$, where V_t is a set of objects, each $v_i \in V_t$ denotes a node, and E_t is a set of links, each $v_{ij} \in E_t$ represents an edge that connects two nodes v_i and v_j at time t. The dynamic network N can be defined as a sequence of static networks $N_t = (V_t, E_t)$; that is, $N = \{N_1, N_2, \ldots, N_T\}$, where each N_t represents the snapshot of nodes and edges at time t. Let $C_t = \{C_t^1, C_t^2, \ldots, C_t^k\}$ be the set of all communities in network N_t at time t where C_t^i denotes the ith community at time t.

2.2. Evolutionary Clustering. Chakrabarti et al. first introduced evolutionary clustering in [2] as the problem of clustering data coming at different time steps to produce a sequence of clustering. We adopt this framework to analyze communities and their evolutions by use of the community structure at time $t-1$ to regularize the community structure at time t. At each time step, two conflicting criteria must be simultaneously optimized to produce a new clustering. The first criterion is that the clustering at any point in time should remain faithful to the current data as much as possible. The second is that the clustering should not shift dramatically from one time step to the next. A framework called temporal smoothness is defined in order to satisfy the second property to smooth out each community over time. This framework assumes that the dramatic change of clustering in a short time is not desirable; thus the cost function consisting of two parts is defined as follows:

$$\text{cost} = \alpha \times \text{SC} + (1-\alpha) \times \text{TC}, \tag{1}$$

where SC denotes snapshot cost which measures how well a community structure C_t represents the data at time t and TC denotes temporal cost which measures how similar the community structure C_t is with the previous community structure C_{t-1}. The input parameter α is used by the user to control the level of emphasis on each part of the two objectives. When $\alpha = 1$, it returns the clustering without temporal smoothing. When $\alpha = 0$, however, the framework produces the same clustering structure with the previous time step, that is, $C_t = C_{t-1}$. Thus it can control the preference degree of each subcost by changing the value of parameter α between 0 and 1.

3. Proposed Algorithm

As [3], we adopt the same multiobjective representative frameworks for dynamic community detection which treat snapshot cost SC and temporal cost TC as two competing objectives. Modularity density and NMI are employed to denote snapshot cost SC and temporal cost TC, respectively. The main advantage of this method is that it does not need to fix the control parameter α.

The proposed algorithm DNCD-MOEA is realized under the framework of NSGA-II [9]. The details of objective functions and representation method are given as follows.

3.1. Objective Functions. To denote snapshot cost SC, we adopt modularity density as an objective function to measure the quality of the community. The modularity density is defined as follows:

$$D = \sum_{i=1}^{m} d(G_i) = \sum_{i=1}^{m} \frac{L(V_i, V_i) - L(V_i, \overline{V_i})}{|V_i|}. \tag{2}$$

In detail, given an undirected network $G = (V, E)$ consisting of the vertex set $V = \{v_1, v_2, \ldots, v_n\}$ (n is the cardinality of V) and the edge set E. Where $\{V_c\}_{c=1}^{m}$ is a partition of the vertex set V into m groups, $\overline{V_c}$ is the complement of V_c with respect to V, $L(V_c, \overline{V_c}) = \sum_{i \in V_c, j \in \overline{V_c}} A_{ij}$ ($A = (A_{ij})$ is the adjacent matrix of G), and $|V_c|$ is the cardinality of V_c.

Normalized Mutual Information (NMI) was employed to denote the second objective function, that is, temporal cost. Danon et al. have proved NMI to be a reliable similarity measure [10].

Let $A = \{A_1, A_2, \ldots, A_m\}$ and $B = \{B_1, B_2, \ldots, B_n\}$ denote two partitions of a network in communities, and C denotes the confusion matrix whose element C_{ij} is the number of nodes of the community $A_i \in A$ that are also in the community $B_j \in B$. The NMI(A, B) is defined as follows:

$$\text{NMI}(A, B) = \frac{-2\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij} \log\left(C_{ij}N/C_{i.}C_{.j}\right)}{\sum_{i=1}^{m} C_{i.} \log\left(C_{i.}/N\right) + \sum_{j=1}^{n} C_{.j} \log\left(C_{.j}/N\right)}, \tag{3}$$

where $C_{i.}$ ($C_{.j}$) is the sum of the elements of C in row i (column j) and N is the number of nodes. If $A = B$, set NMI$(A, B) = 1$. If A and B are completely different, set NMI$(A, B) = 0$. In this paper, the two objectives of $D(C_t)$ and NMI(C_{t-1}, C_t) will be maximized simultaneously.

3.2. Solution Selection. In fact, the proposed algorithm DNCD-MOEA returns Pareto front at the end of each timestamp, which contains a set of solutions. Each of these solutions corresponds to a different tradeoff between the two objectives and thus to a diverse partitioning of the network consisting of various number of clusters. It needs to establish a criterion to automatically select which solution denotes the optimal partitioning of the current network at each time step. Community score introduced in [11] has been proved to be very effective in detecting communities. In this paper, we adopt community score as a criterion to select, among the solutions found, the solution having the highest value of community score. It is defined as follows.

Let $C_t = \{C_t^1, C_t^2, \ldots, C_t^k\}$ be the set of all communities in network N_t at time t where C_t^i denotes the ith community at time t. The community score of C_t is defined as

$$\mathrm{CS}\left(C_t\right) = \sum_{i=1}^{k} \mathrm{score}\left(C_t^i\right), \tag{4}$$

where

$$\mathrm{score}\left(C_t^i\right) = \frac{\sum_{i \in C_t^i} \mu_i}{\left|C_t^i\right|} \times \sum_{i,j \in C_t^i} A_{ij}. \tag{5}$$

The parameter $\mu_i = (1/|C_t^i|) \sum_{j \in C_t^i} A_{ij}$ denotes the fraction of edges which connect each node i of C_t^i to the nodes in the same community C_t^i. The community score takes into account both the fraction of interconnections among the nodes and the number of interconnections contained in the module C_t^i. It gives a global measure of the network division in communities by summing up the local score of each module found. Thus the larger community score indicates that the community structure is stronger. So, we select the maximum community score value in the set of solutions as the best solution.

3.3. Genetic Representation. The locus-based adjacency representation proposed by Park and Song [12] is used in our community detection algorithm, similar to [11].

As a graph $G = (V, E)$ has N vertices, an arbitrary individual of the population consists of N genes, which can be represented by a number of strings as follows:

$$\left(g_1, g_2, \ldots, g_N\right), \quad g_i \in \{1, 2, \ldots, N\} \quad (i = 1, 2, \ldots, N), \tag{6}$$

where $g_i = j$ denotes that there exists a link between the nodes i and j. This means that thenodes i and j will be in the same community. It is necessary to identify all of the components of the corresponding graph in the decoding step. The nodes participating to the same component are assigned to the same community. Adopting this representation, the main advantage is that the decoding step can be done in linear time and it has been verified as an effective encoding schema for community detection as shown in [3]. Additionally, the number k of clusters is automatically determined by the number of components contained in an individual and is also determined by the decoding step [11].

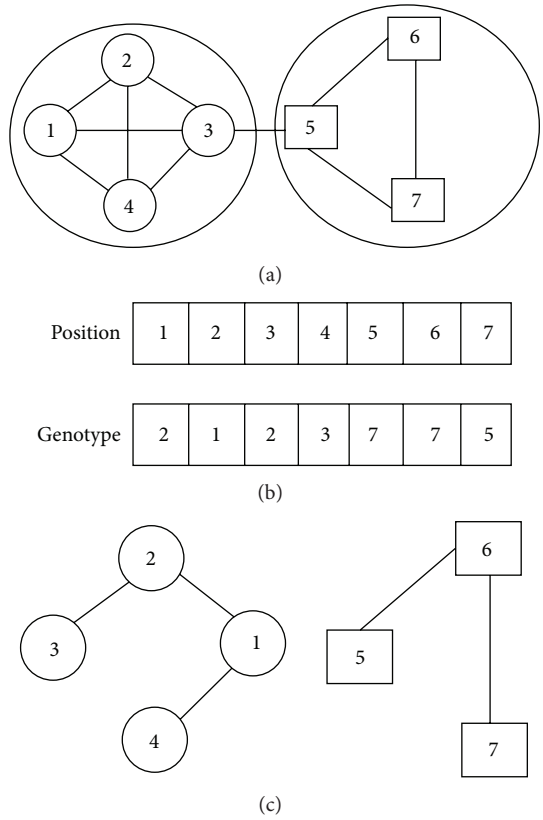


Figure 1: (a) A graph modeled as a network, (b) the representation of a possible genotype, and (c) the graph structure of the genotype to be decoded.

An example of the encoded genotype and corresponding network is shown in Figure 1. As shown in Figure 1(a), the supposed network consists of seven nodes numbered from 1 to 7. It is obvious that the network can be partitioned into two groups visualized by different shape. A possible genotype as shown in Figure 1(b), which corresponds to the optimal solution, is translated in the graph structure given in Figure 1(c). Each connected component provides a grouping of nodes that corresponds to the partitioning of the network in Figure 1(a).

3.4. Initialization. If an individual is created randomly, it may not be a feasible solution. In fact, a randomly generated individual could contain an allele value j in the ith position, but no connection exists between the two nodes i and j; that is, the edge (i, j) is not present. In order to overcome this limitation, we should check whether the individual is safe after an individual is created. When the individual is not safe, that is, a gene i contains a value j, but link (i, j) does not exist, it needs to repair to ensure whether the individual is safe or not. Safe individuals improve the convergence of the method because the space of the possible solutions is restricted.

3.5. Crossover and Mutation. As in [3], we use uniform crossover which can guarantee the maintenance of the effective connections of the nodes in the network in the child individual. Given two arbitrary safe parent individuals and that

a random binary vector is created, the genes are selected from the first parent if the vector element is 1, and the genes are selected from the second parent if the vector element is 0. Then the genes are combined to form a child. Because of the biased initialization, the child created from the two safe parents is safe also. In the child, if a position i contains a value j, then the edge (i, j) exists. The uniform crossover is shown in Figure 2.

Crossover operator is regarded as a macroscopic operation on individuals, while the mutation operator is regarded as a microcosmic operation on individuals. The mutation operator that randomly changes the value j of the ith gene causes a useless exploration of the search space. In order to guarantee the mutated child is safe as the crossover operation, the possible value of an allele after mutating is restricted to be one of the replaced gene's neighbors.

3.6. The Pseudocode of the Proposed Algorithm. The pseudocode of our DNCD-MOEA algorithm is described in Algorithm 1.

3.7. Local Search Strategy. Local search is proved to be an effective algorithm. The mutation operator is regarded as a microcosmic operation on individuals and can achieve its local search function by moving single nodes between communities. Inspired by this idea, we adopt mutation operator in our local search algorithm.

In local search algorithm, it needs to convert multiple objectives into a single objective function. In our study, we select a weighted objective as follows [3]:

$$G(x) = \sum_{i=1}^{2} r_i g_i(x), \tag{7}$$

where $g_i(x)$ is the objective function which is described in (2) and (3), respectively, and r_i is the nonnegative weights for the two objectives. The weights are calculated in a special way as follows:

$$r_i = \frac{\left(g_i(x) - g_i^{\min}\right) / \left(g_i^{\max} - g_i^{\min}\right)}{\sum_{j=1}^{2} \left(g_j(x) - g_j^{\min}\right) / \left(g_j^{\max} - g_j^{\min}\right)}, \tag{8}$$

where $g_i^{\max}$ or $g_i^{\min}$ is the maximum or minimum value of each objective function $g_i(x)$ in the obtained dominant population and $\sum_{i=1}^{2} r_i = 1$. In Algorithm 2, the detailed pseudocode of the local search algorithm is given.

3.8. Computational Complexity. The computational complexity of DNCD-MOEA algorithm is investigated in this subsection. The DNCD-MOEA algorithm contains two major components: the main program based on the NSGA-II framework and the subprogram of local search operation. The time consuming of the main program mainly consists of three parts: generating nondominated sorting, calculating crowding distance assignment, and constructing partially ordered set. Let S denote the size of population and r denote the number of objects. The computational complexity of the three parts is $O(r(2S)^2)$, $O(r(2S)\log(2S))$, and $O(2S\log(2S))$,

Parent1	2	1	2	3	6	7	5
Parent2	2	1	4	3	6	5	6
Binary mask	1	1	0	1	0	0	1
Offspring	2	1	4	3	6	5	5

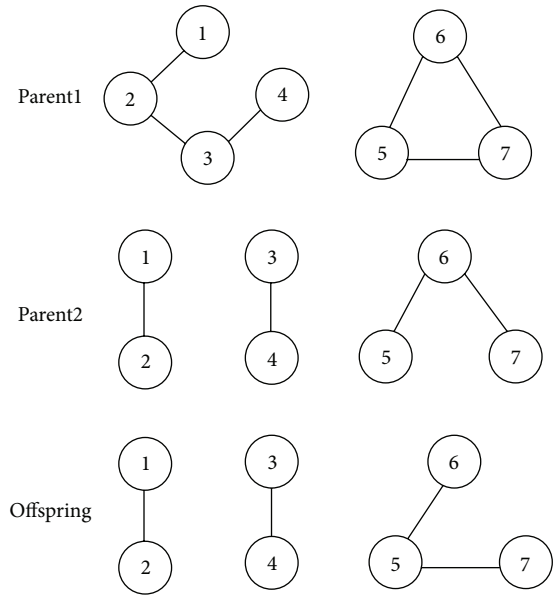


Figure 2: Illustration of the uniform crossover.

respectively. So the total time complexity of the main programming is $O(rS^2)$. The local search operation mainly contains two loops and its computational complexity is $O(MN)$. So, the total computational complexity of DNCD-MOEA algorithm is $O(rS^2) + O(MN)$. Because $M < S$, $r = 2$, and in practice $N < S$, the time complexity can be simplified to $O(S^2)$.

4. Experiments

In order to check the ability of our approach on a dynamic network, we adopt the method proposed as [3] to generate data simulating dynamic network. Firstly, we generate synthetic datasets by following the procedure suggested by Girvan and Newman [1]. The datasets are generated using the software package designed by Lancichinetti et al. [13]. The data have 128 nodes, which are divided into 4 communities of 32 vertices each. Every node has an average degree of 16 and shares a number Z_{out}, which represents the average number of edges from a node to nodes in other communities. If we increase Z_{out}, then the noise level of network is augmented. On the other hand, if we decrease Z_{out}, then the noise level in the network decreases.

A parameter μ, which represents the average ratio of external degree/total degree for each node, is used to control the noise level in the dynamic networks. If the value of μ is increased, then the network will become more noisy in the sense that the community structure becomes less obvious and hard to detect. In this study, by setting $\mu = 0.1$ and $\mu = 0.3$, the datasets under two different noise levels are generated.

Program DNCD-MOEA
Input: The number T of time steps, the sequence of dynamic network $N = \{N_1, N_2, \ldots, N_T\}$
Output: The sequence of community structure detected in the dynamic network $C = \{C_1, C_2, \ldots, C_T\}$
Begin
Step 1: Set $t = 1$. Generate the initial community structure $C_1 = \{C_1^1, C_1^2, \ldots, C_1^k\}$ of the network N_1 using GA-Net algorithm. Set $t = t + 1$.
Step 2: If $t > T$, return the sequence of community structure $C = \{C_1, C_2, \ldots, C_T\}$ as the output, algorithm stops; Else, go to Step 3.
Step 3: Set $s = 1$. Randomly generate individuals whose length equals the nodes number $n_t = |V_t|$ of network N_t as an initial population P_s;
Step 4: While termination condition is not satisfied do
Step 4.1: Create a new population Q_s of offspring by applying the variation operators on population P_s;
Step 4.2: Combine the parents P_s and offspring Q_s into a new pool R_s and;
Step 4.3: Decode each individual I of the population R_s to generate the partitioning $C_t = \{C_t^1, C_t^2, \ldots, C_t^{k_t^I}\}$ of the network N_t in k_t connected components;
Step 4.4: Evaluate the two fitness values of the translated individuals;
Step 4.5: Partition R_s into fronts, assign a rank to each individual and sort them according to nondomination rank;
Step 4.6: Select individuals based on rank and crowding length to comprise new population P_{s+1};
Step 4.7: Select the dominant individuals L_s in P_{s+1},
Step 4.8: Perform the local search algorithm on the selected individuals in L_s to generate the new dominant population L_s'. Update the dominant population L_s with L_s' in P_{s+1}.
Step 4.9: $s = s + 1$
End while
Step 5: Select the individual which has the maximum Community Score on the Pareto front. Decode the selected individual to get the community structure $C_t = \{C_t^1, C_t^2, \ldots, C_t^{k_t}\}$ of the network N_t.
Step 6: Set $t = t + 1$, go to Step 2.
End

Algorithm 1

In order to introduce dynamics into the network, we let the community structure of the network evolve in the following way. After time step 1, 5% of the nodes is randomly *choused* to leave their original community and randomly assigned to the other three communities at each time step. After the community memberships are decided, links are generated by following the parameter μ. We generate the network with community evolution in this way for 10 time steps.

Figure 3 shows the statistical average value of normalized mutual information with the ground truth over the 10 networks for the 10 timestamps when the value of $\mu = 0.1$ (Figure 3(a)) and $\mu = 0.3$ (Figure 3(b)). Both figures show that the proposed algorithm DNCD-MOEA can achieve better accuracy than the compared method. Especially, the average values of NMI at each time step obtained by DNCD-MOEA are closed to 1 when $\mu = 0.1$.

Figure 4 reports the community score obtained by the two algorithms at each time step when $\mu = 0.1$ (Figure 4(a)) and $\mu = 0.3$ (Figure 4(b)). It indicates that the corresponding network is densely connected within each subnetwork when the obtained value of community score is larger. From Figure 4, it can be found that the algorithm DNCD-MOEA outperforms

Program: Local Search Algorithm:
Input: Given a dominant population L_s before local search at the sth generation, the size M of dominant population L_s, the number N of local search.
Output: the improved population L'_s in the sth generation.
Begin
(1) for $i = 1$ to M
(2) $j = 1$; $find = false$
(3) while $j < N$ and not $find$
(4) Create a new individual applying the mutation operator on the ith individual of dominant population L_s;
(5) Calculate the objective function value of the new individual according to formula (7). If its value is greater than that before local search, add the new individual to L'_s, $find = true$;
(6) $j = j + 1$;
(7) end while
(8) if not $find$, adds the current individual to L'_s
(9) end for
END

Algorithm 2

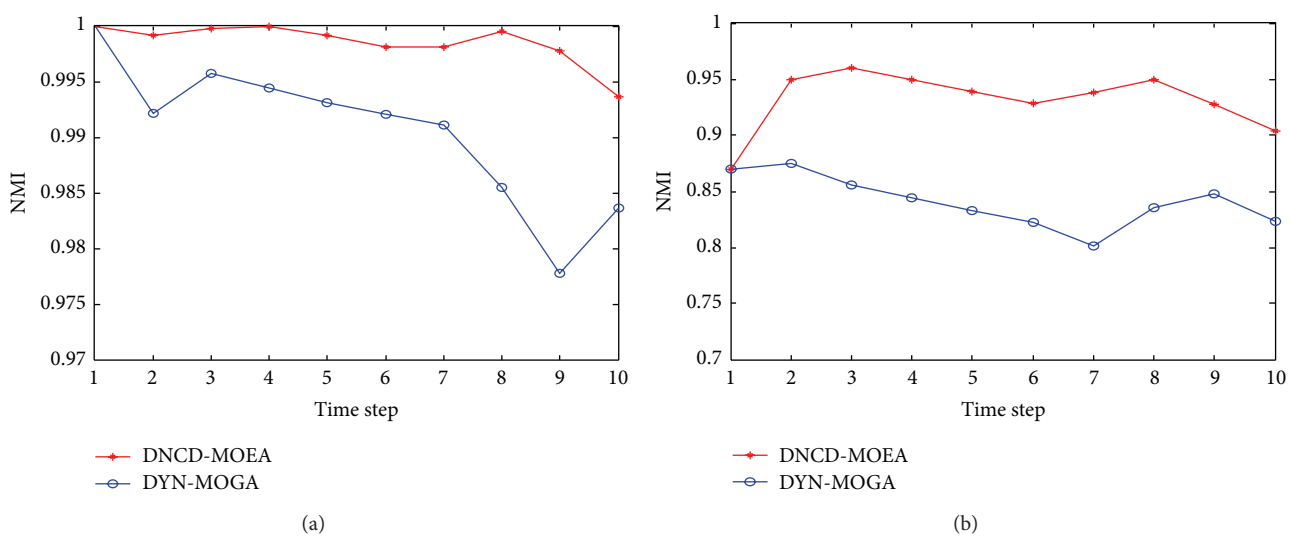


Figure 3: Normalized mutual information (NMI) of clustering results when (a) $\mu = 0.1$ and (b) $\mu = 0.3$.

the algorithm DYN-MOGA. That is to say, the community structure obtained by our algorithm is more closed to the true community structure.

5. Conclusions

In this paper, a novel multiobjective algorithm for detecting communities in dynamic networks is proposed. Based on optimizing modularity density and NMI, the algorithm automatically provides a solution representing the best tradeoff between the accuracy of the community structure obtained with respect to the data of the current time step and the deviation from one time step to the successive. To solve the proposed model, the new genetic operators are designed based on the problem-specific knowledge. A local search operator is designed in order to improve the quality of the solutions. Experimental results on synthetic datasets show that the proposed algorithm can obtain better performance than the compared algorithm. Future research will aim at improving efficiency and applying proposed algorithm to process real-life network.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grants no. 61272119 and no. 61203372.

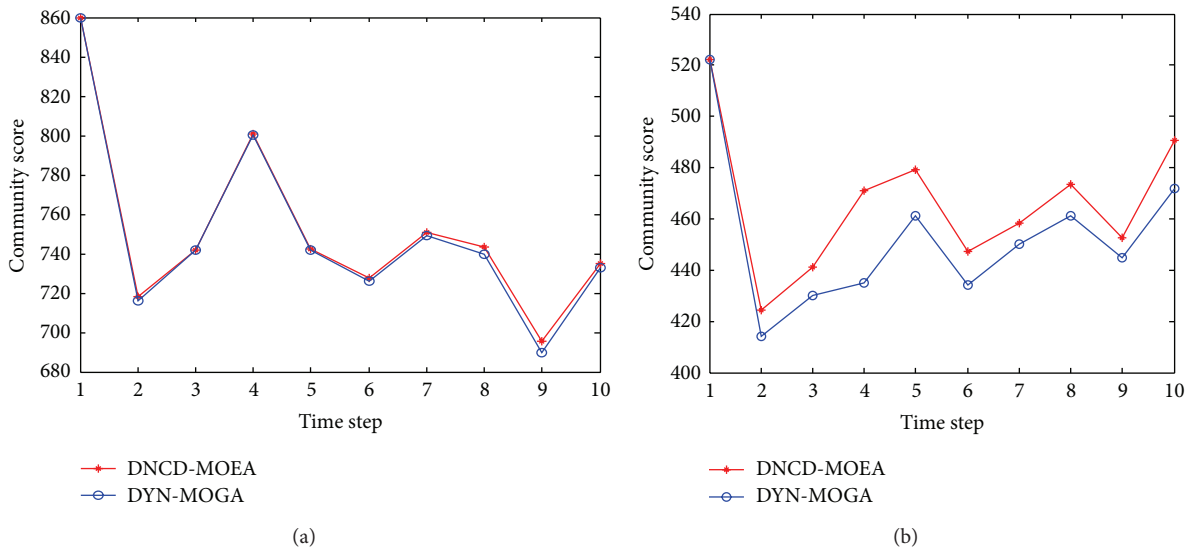


Figure 4: Community score when (a) $\mu = 0.1$ and (b) $\mu = 0.3$.

References

[1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[2] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," in *Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pp. 554–560, August 2006.

[3] F. Folino and C. Pizzuti, "A multiobjective and evolutionary clustering method for dynamic networks," in *Proceedings of the International Conference on Advances in Social Network Analysis and Mining (ASONAM '10)*, pp. 256–263, August 2010.

[4] M. G. Gong, L. J. Zhang, J. J. Ma, and L. C. Jiao, "Community detection in dynamic social networks based on multiobjective immune algorithm," *Journal of Computer Science and Technology*, vol. 27, no. 3, pp. 455–467, 2012.

[5] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 1, pp. 36–41, 2007.

[6] M. Rosvall and C. T. Bergstrom, "An information-theoretic framework for resolving community structure in complex networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 18, pp. 7327–7331, 2007.

[7] S. Muff, F. Rao, and A. Caflisch, "Local modularity measure for network clusterizations," *Physical Review E*, vol. 72, no. 5, Article ID 056107, 4 pages, 2005.

[8] Z. P. Li, S. H. Zhang, R. S. Wang, X. S. Zhang, and L. N. Chen, "Quantitative function for community detection," *Physical Review E*, vol. 77, no. 3, Article ID 036109, 9 pages, 2008.

[9] N. Srinivas and K. Deb, "Multiobjective optimization using nondominatedsorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[10] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics*, vol. 2005, no. 9, Article ID P09008, 2005.

[11] C. Pizzuti, "GA-Net: a genetic algorithm for community detection in social networks," in *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature*, pp. 1081–1090, September 2008.

[12] Y. J. Park and M. S. Song, "A genetic algorithm for clustering problems," in *Proceedings of 3rd Annual Conference on Genetic Algorithms*, pp. 2–9, 1989.

[13] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, Article ID 046110, 5 pages, 2008.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 625790, 7 pages
http://dx.doi.org/10.1155/2013/625790

Research Article

Memristive Perceptron for Combinational Logic Classification

Lidan Wang, Meitao Duan, and Shukai Duan

School of Electronic and Information Engineering, Southwest University, Chongqing 400715, China

Correspondence should be addressed to Shukai Duan; duansk@swu.edu.cn

Received 1 February 2013; Accepted 22 April 2013

Academic Editor: Chuandong Li

Copyright © 2013 Lidan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The resistance of the memristor depends upon the past history of the input current or voltage; so it can function as synapse in neural networks. In this paper, a novel perceptron combined with the memristor is proposed to implement the combinational logic classification. The relationship between the memristive conductance change and the synapse weight update is deduced, and the memristive perceptron model and its synaptic weight update rule are explored. The feasibility of the novel memristive perceptron for implementing the combinational logic classification (NAND, NOR, XOR, and NXOR) is confirmed by MATLAB simulation.

1. Introduction

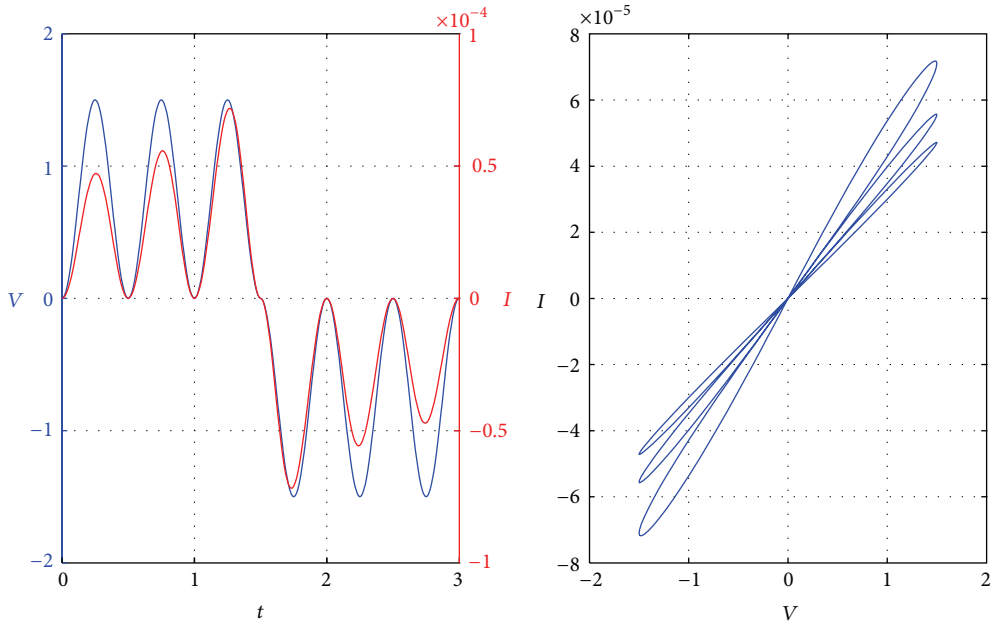
Artificial neural network is a simplification system which simulates the organization structure, the processing method, and the system function of biological neural system [1]. It draws a strong attraction to many researchers because of its performance, such as the parallel processing, distributed storage, self-learning, self-adapting, and high fault tolerance [2]. Perceptron is the simplest kind of neural network which can implement basic learning and parallel processing. It is widely studied to solve the classification problems of medical image [3], text [4], mode [5, 6], and fingerprint [7]. A large number of biological synapses respond to received signals to store a series of continuous weights; so it is hard to realize perceptron by hard circuits. Memristor, a new circuit component in electronics field, is suitably used as synapse in neural networks because of its nanoscale size, automatic information storage, and nonvolatile characteristic with respect to long periods of power-down. In recent years, more and more researchers pay attention to memristive neural networks. A novel chaotic neural network with memristor was built and applied to associative memory [8]. Long-term potentiation and long-term depression of synapses using memristor were realized [9]. Memristors as main modules of complex bionic information processing networks were utilized [10], and then memristive neural networks were gradually introduced to build memristive cellular automata and discrete-time memristive cellular neural networks [11]. The PID controller based on memristive CMAC network was built [12]. In addition, memristive crossbar arrays are used to realize pulse neural networks [13] and memristor is utilized to build synapse template of cellular neural networks [14]. These research results show that memristor devices are capable of emulating the biological synapses through building up the correspondence between memristive conductance and synapse weight. Memristor is an ideal choice of electric synapse of new neuromorphic system. In this paper, a new perceptron is built to realize combinational logic classification problems. The effectiveness of the memristive perceptron for combinational logic classification is demonstrated through MATLAB simulations.

2. Memristive Perceptron

From 2008, the memristor was under development by Hewlett-Packard team [15]. The change curve of *I-V* of memristor under sine-type input voltage ($v(t) = \pm 1.5 \sin^2(2\pi t)$) can be seen in Figure 1.

The memristive conductance change dg/dt is a function of voltage v in the voltage-controlled memristor [16]. The function is similar to sinh curve, and the correspondence between memristive conductance and synapse weight update is set as

$$\frac{dg}{dt} = \alpha \sinh(\beta v), \tag{1}$$



(a) $v(t) = \pm 1.5 \sin^2(2\pi t)$ (b) I-V hysteresis corresponding to the excitation (a)

Figure 1: Characteristic I-V curve of memristor under sine-type input voltage.

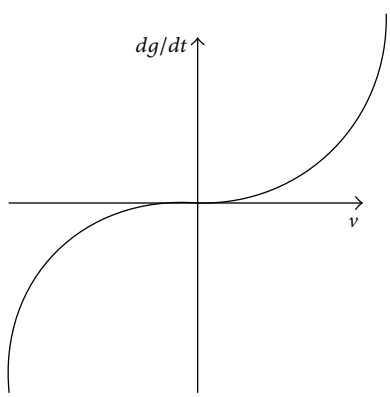


Figure 2: The relationship curve of memristive conductance change and voltage.

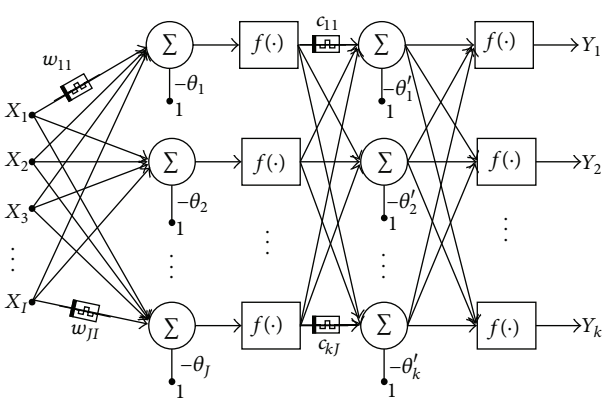


Figure 3: Memristive perceptron model.

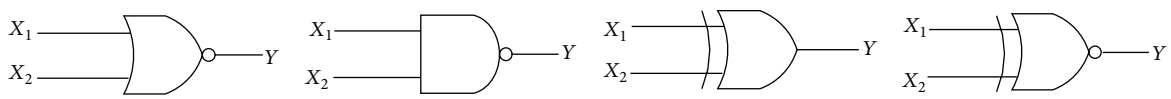
where the values of α and β depend on the kind of memristor, such as material, size, and manufacturing process. The relationship curve between the memristive conductance change and the voltage is shown in Figure 2. It can be seen that memristive conductance change dg/dt increases along with the increasing of applied voltage v. If the learning error e of perceptronis regarded as voltage v, memristive conductance change can correspondingly describe as synapse weight. The correspondence between memristive conductance change and synapse weight update can be built.

The memristor model is used as synapse of perceptron to build memristive perceptron model shown in Figure 3, whose mathematical description is as follows:

$$Y_k = f\left(\sum_{j=1}^{J} c_{kj} f\left(\sum_{i=1}^{I} w_{ji} X_i - \theta_j\right) - \theta'_k\right), \tag{2}$$

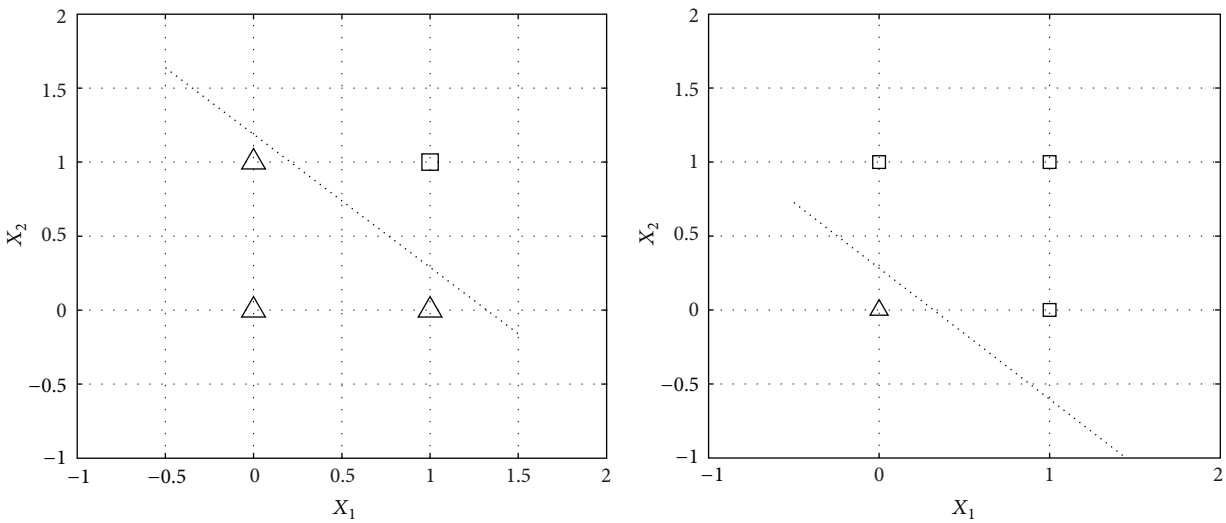
where X_i is the input of neuron i; Y_k is output of neuron k; i, j, k are the numbers of the input layer neurons, the hidden layer neurons, and the output layer neurons, respectively. w_{ji} represents the memristive synapse weights from the input layer neuron i to the hidden layer neuron j. c_{kj} represents the memristive synapse weights from the hidden layer neuron j to the output layer neuron k, respectively. θ_j and θ'_k are thresholds of the hidden layer neuron j and output layer neuron k, respectively. f is the step function, whose mathematical expression is

$$f(X) = \begin{cases} 0, & X < 0 \\ 1, & X \geq 0. \end{cases} \tag{3}$$



(a) NAND (b) NOR (c) XOR (d) NXOR

Figure 4: Graphic symbols of combinational logic.



(a) The classification result of NAND (b) The classification result of NOR

Figure 5: The classification results of NAND and NOR.

3. Weight Update Rule of Memristive Synapses

If memristive perceptron is applied to combinational logic classification, weight update rule should be set. In expression (1), when Δt is very small, dg can be replaced by Δg:

$$\Delta g = \alpha \sinh(\beta v) \times \Delta t. \tag{4}$$

Let the memristive conductance g correspond to synapse weight w of perceptron and combine BP learning rules which are used as the prototype. Then, memristive perceptron weight-update rule by following the rule through a given monitoring signal can be set:

$$\begin{gathered} \Delta w(k) = \eta \cdot \alpha \sinh\left[\beta\left(S(k) - Y(k)\right)\right] \cdot X, \\ w(k+1) = w(k) + \Delta w(k), \end{gathered} \tag{5}$$

where $\eta = \Delta t$ is learning rate, $S(k)$ is monitoring signal, $Y(k)$ is output of memristive perceptron, $S(k) - Y(k)$ is error of memristive perceptron, and X is input of memristive perceptron; let $\alpha = 4$, $\beta = 3$.

4. Memristive Perceptron Implements Combinational Logic Classification

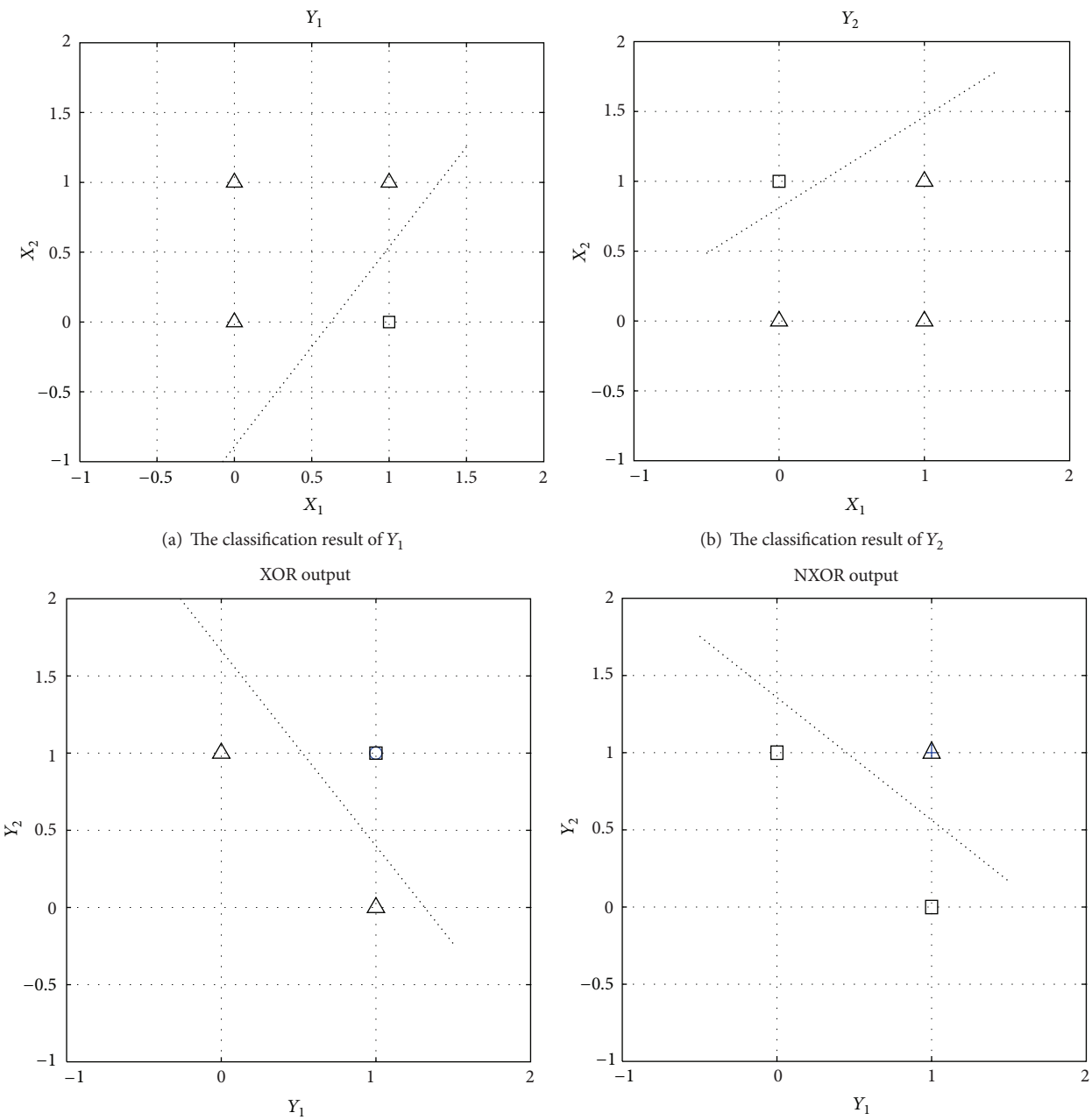
The combinational logic operators include the NAND, NOR, XOR, and NXOR operations. Their graphic symbols are shown in Figure 4 and the corresponding truth table is shown in Table 1.

Table 1: Truth table of the combinational logic operators.

Input		Output			
		Y			
X_1	X_2	NAND	NOR	XOR	NXOR
0	0	1	1	0	1
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	0	0	1

4.1. Realization of Combinational Logic NAND and NOR Classification. It can be seen from Table 1 that the combinational logic NAND and NOR classification belongs to the linear separable problems; so they can be realized by a two-input single-layer memristive perceptron model. Let the threshold $\theta = 0.8344$ and the learning rate $\eta = 0.02$. When the number of cycles is set to 30 and the initial weight is set to a random nonzero number, the experiment result is shown in Figure 5(a), where "△" denotes the output is logic "1" and "□" denotes the output is logic "0." The classification of logic NAND operator is realized efficiently. Similarly, let the threshold $\theta = 0.4121$ and the learning rate $\eta = 0.02$. The number of cycles is set to 30; so the classification of logic NOR operator is realized efficiently, whose experiment result is shown in Figure 5(b).

4.2. Realization of Combinational Logic XOR and NXOR Classification. It can be seen from Table 1 that the combinational logic XOR and NXOR classification belongs to the



(a) The classification result of Y_1

(b) The classification result of Y_2

(c) Classification result of logic XOR

(d) Classification result of logic NXOR

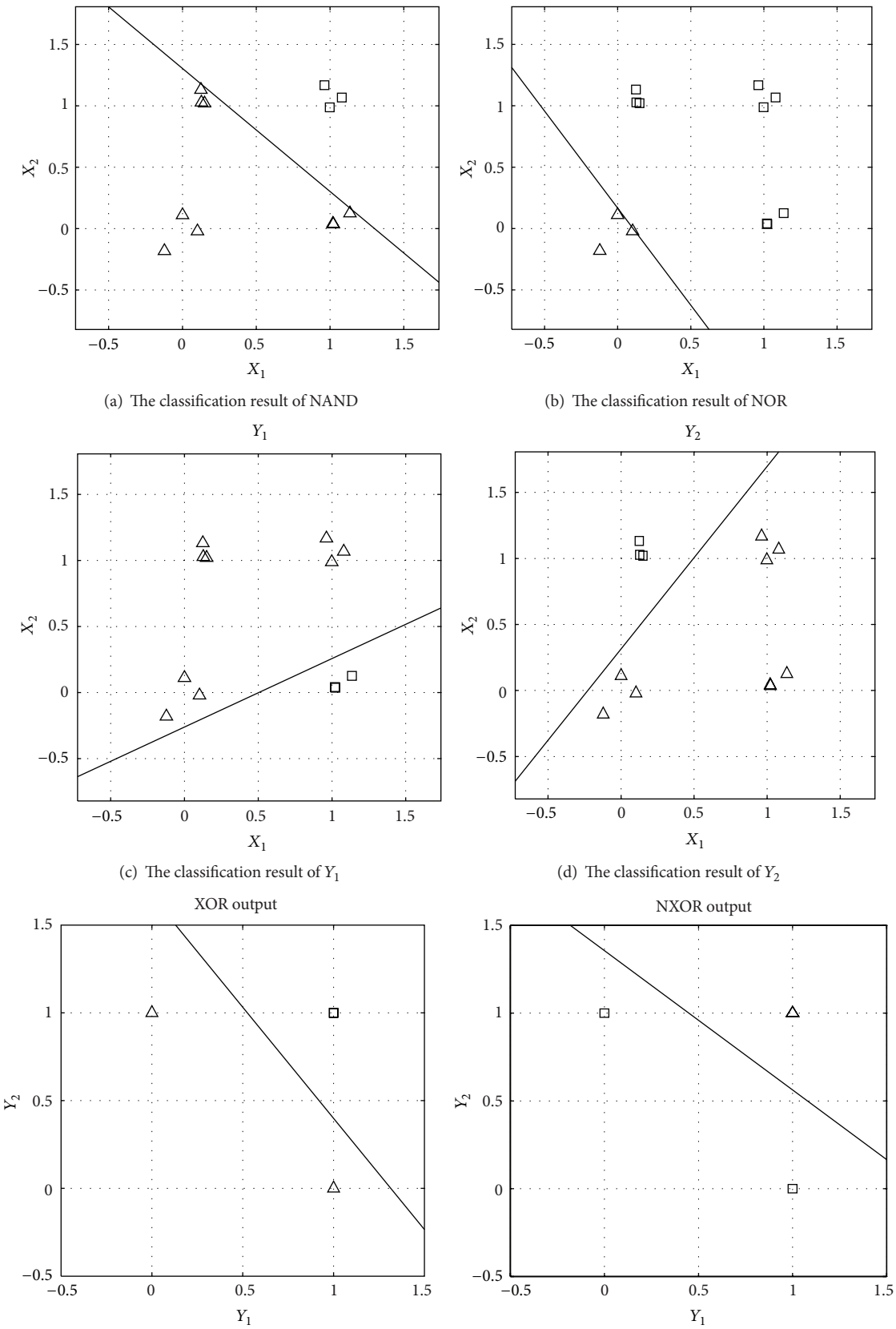
Figure 6: Classification results of the combinational logic XOR and NXOR with the monitoring signal.

linear inseparable problems. As the single-layer memristive perceptron is unable to realize the nonlinear classification problems [17], a double-layer memristive perceptron model is needed to realize logic XOR and NXOR classification, which is equivalent to the two paralleling single-layer memristive perceptrons. So, it can be used to implement logic XOR and NXOR classification problems.

Truth table of XOR and NXOR with the monitoring signal is shown in Table 2. Y_1 is the monitoring signal of the first perceptron in the hidden layer and Y_2 is the monitoring signal of the second perceptron in the hidden layer. Let $Y_1 = [1, 1, 0, 1]$, $Y_2 = [1, 0, 1, 1]$; both of them are linear separable problems. So, the memristive perceptron in the hidden layer can be used to realize classification. Let

Table 2: Truth table of XOR and NXOR with monitoring signal.

Input		Supervisor		Output	
X_1	X_2	Y_1	Y_2	XOR	NXOR
0	0	1	1	0	1
0	1	1	0	1	0
1	0	0	1	1	0
1	1	1	1	0	1



(a) The classification result of NAND

(b) The classification result of NOR

(c) The classification result of Y_1

(d) The classification result of Y_2

(e) Classification result of logic XOR

(f) Classification result of logic NXOR

Figure 7: Classification results of combinational logic with test input signals.

Table 3: The results of whole combinational logic classification.

Combinational logical operator	Initial memristive weight		Threshold	Learning times	Final memristive weight		Divider equation
NAND	0.1705	0.0994	0.8344	1	−0.6309	−0.7020	$-0.6309X_1 - 0.7020X_2 - 0.8344 = 0$
NOR	0.3110	−0.6576	0.4121	1	−1.2919	−1.4591	$-1.2919X_1 - 1.4591X_2 - 0.4121 = 0$
Y_1	−0.8728	−0.1908	0.5440	2	−0.8728	0.6106	$-0.8728X_1 + 0.6106X_2 - 0.5440 = 0$
Y_2	−0.1033	−0.2684	0.8657	3	0.6981	−1.0698	$0.6981X_1 - 1.0698X_2 - 0.8657 = 0$
XOR	0.5379	−0.2080	0.3466	3	−0.2635	−0.2080	$-0.2635Y_1 - 0.2080Y_2 - 0.3466 = 0$
NXOR	−0.3684	0.5454	−0.7397	3	0.4330	0.5454	$0.4330Y_1 + 0.5454Y_2 + 0.7397 = 0$

thresholds $\theta_1 = 0.5440, \theta_2 = 0.8657$, and learning rate $\eta = 0.02$, and let the number of cycles be 30; the initial weights are the random nonzero numbers. The experiment results are shown in Figures 6(a) and 6(b). Then Y_1 and Y_2 are looked as the inputs of memristive perceptron of output layer; so Y can be taken as the output. In MATLAB program, let threshold $\theta' = 0.3466$, learning rate $\eta = 0.02$, and the number of cycles be 30; the experiment result is shown in Figure 6(c). The XOR classification is realized. From the above analysis, the procedure of the logic NXOR is the same as the one of logic XOR. Let threshold $\theta' = -0.7397$, learning rate $\eta = 0.02$, and the number of cycles be 30; the experiment result is shown in Figure 6(d). The NXOR classification is realized.

Through the above theoretical analysis and experiment simulation, the results of whole combinational logic classification are shown in Table 3. It can be seen that NAND and NOR classifications are realized in the single-layer memristive perceptron after learning once. The XOR and NXOR classifications are realized in the double-layer memristive perceptron after learning eight times. The combinational logic classification can be realized effectively by the memristive perceptron we proposed.

5. Test Validity of Memristive Perceptron in Combinational Logic Classification

The experiments have confirmed that the memristive perceptrons can achieve the standard combinational logic classification. However, in fact, the standard inputs cannot be obtained; so the test signals with the error are introduced to test the validity of memristive perceptrons in combinational logic classification.

Let the test input signals be as follows: $X_1 = [-0.122, 0.128, 1.135, 0.998, 0.102, 0.125, 1.023, 0.962, 0, 0.15, 1.02, 1.08]$, $X_2 = [-0.181, 1.028, 0.126, 0.988, -0.021, 1.132, 0.036, 1.168, 0.111, 1.021, 0.041, 1.068]$; the memristive perceptrons in Part 4 are tested. By MATLAB simulations, the classification results are shown in Figure 7.

It can be known from the above experiments that when the test signals with the error are used as inputs, the memristive perceptrons also can effectively realize the combinational logic classification.

6. Conclusions

The relationship between the synaptic weight update in perceptron and memristive conductance change is linked by the theoretical deduction. A new synaptic weight update rule is proposed in this paper. Single-layer memristive perceptron is proposed to realize the classification of the linear separable logic NAND and NOR. Double-layer memristive perceptron is built to realize the classification of linear inseparable logic XOR and NXOR. The experiments exhibit that memristive perceptron can effectively implement the combinational logic classification. The memristive perceptron has simple structure and high integration so that it is easier to be implemented by the low-power circuit. The more complex memristive neural networks we used the more complex problems can be solved in artificial intelligence field.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (Grant nos. 60972155, 61101233), Fundamental Research Funds for the Central Universities (Grant nos. XDJK2012A007, XDJK2013B011), University Excellent Talents Supporting Foundations in of Chongqing (Grant no. 2011-65), University Key Teacher Supporting Foundations of Chongqing (Grant no. 2011-65), Technology Foundation for Selected Overseas Chinese Scholars, Ministry of Personnel in China (Grant no. 2012-186), Spring Sunshine Plan Research Project of the Ministry of Education of China (Grant no. z2011148).

References

[1] M.-Y. He, *Principles, Language, Design and Application of Neural Computing*, Press of Xidian University, Xian, China, 1992.

[2] Y. Zhang, J. Wang, and Y. Xia, "A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits," *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 658–667, 2003.

[3] A. Datta and B. Biswas, "A fuzzy multilayer perceptron network based detection and classification of lobar intra-cerebral hemorrhage from computed tomography images of brain," in *Proceedings of the International Conference on Recent Trends in Information Systems*, pp. 257–262, 2011.

[4] A. Gkanogiannis and T. Kalamboukis, "A perceptron-like linear supervised algorithm for text classification," in *Advanced Data Mining and Applications*, vol. 6440 of *Lecture Notes in Computer Science*, pp. 86–97, 2010.

[5] Y. C. Hu, "Pattern classification by multi-layer perceptron using fuzzy integral-based activation function," *Applied Soft Computing Journal*, vol. 10, no. 3, pp. 813–819, 2010.

[6] M. Fernández-Delgado, J. Ribeiro, E. Cernadas, and S. Barro, "Fast weight calculation for kernel-based perceptron in two-class classification problems," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '10)*, pp. 1–6, July 2010.

[7] I. EI-Feghi, A. Tahar, and M. Ahmadi, "Efficient features extraction for fingerprint classification with multi layer perceptron neural network," in *Proceedings of International Symposium on Signals, Circuits and Systems*, pp. 1–4, 2011.

[8] X.-F. Hu, S.-K. Duan, and L.-D. Wang, "A novel chaotic neural network using memristors with applications in associative memory," *Abstract and Applied Analysis*, vol. 2012, Article ID 405739, 19 pages, 2012.

[9] S. Choi and G. Kim, "Synaptic behaviors and modelling of a metal oxide memristive device," *Applied Physics A: Materials Science & Processing*, vol. 102, no. 4, pp. 1019–1025, 2011.

[10] A. Mittal and S. Swaminathan, "Emulating reflex actions through memristors," in *Proceedings of IEEE Conference and Exhibition (GCC '11)*, pp. 469–472, 2011.

[11] M. Itoh and L. O. Chua, "Memristor cellular automata and memristor discrete-time cellular neural networks," *International Journal of Bifurcation and Chaos*, vol. 19, no. 11, pp. 3605–3656, 2009.

[12] L.-D. Wang, X.-Y. Fang, S.-K. Duan et al., "PID controller based on memristive CMAC network," *Abstract and Applied Analysis*, vol. 2013, Article ID 510238, 6 pages, 2013.

[13] A. Afifi, A. Ayatollahi, and F. Raissi, "Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nano circuits," in *Proceedings of European Conference on Circuit Theory and Design Conference Program (ECCTD '09)*, pp. 563–566, August 2009.

[14] S.-Y. Gao, S.-K. Duan, and L.-D. Wang, "On memristive cellular neural network and its applications in noise removal and edge extraction," *Journal of Southwest University*, vol. 33, no. 11, pp. 63–70, 2011.

[15] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[16] A. Afifi, A. Ayatollahi, and F. Raissi, "STDP implementation using memristive nanodevice in CMOS-Nano neuromorphic networks," *IEICE Electronics Express*, vol. 6, no. 3, pp. 148–153, 2009.

[17] J.-G. Yang, *Artificial Neural Network Practical Guide*, Press of Zhejiang University, Hangzhou, China, 2001.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 170202, 8 pages
http://dx.doi.org/10.1155/2013/170202

Research Article

Resonant Tunneling Diodes-Based Cellular Nonlinear Networks with Fault Tolerance Analysis

Shukai Duan,[1] Xiaofang Hu,[2] Lidan Wang,[1] and Shiyong Gao[1]

[1] *School of Electronics and Information Engineering, Southwest University, Chongqing 400715, China*
[2] *Department of Mechanical and Biomedical Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong*

Correspondence should be addressed to Lidan Wang; ldwang@swu.edu.cn

Received 31 January 2013; Accepted 9 May 2013

Academic Editor: Chuandong Li

Copyright © 2013 Shukai Duan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The resonant tunneling diodes (RTD) have found numerous applications in high-speed digital and analog circuits owing to its folded-back negative differential resistance (NDR) in current-voltage (*I-V*) characteristics and nanometer size. On account of the replacement of the state resistor in standard cell by an RTD, an RTD-based cellular neural/nonlinear network (RTD-CNN) can be obtained, in which the cell requires neither self-feedback nor a nonlinear output, thereby being more compact and versatile. This paper addresses the structure of RTD-CNN in detail and investigates its fault-tolerant properties in image processing taking horizontal line detection and edge extraction, for examples. A series of computer simulations demonstrates the promising fault-tolerant abilities of the RTD-CNN.

1. Introduction

In 1988, Chua and Yang defined cellular neural/nonlinear network (CNN) based on cellular automata and neural network [1, 2]. CNN was acclaimed as a powerful back-end analog array processor and capable of accelerating various computation intensive tasks in image processing, motion detection, pattern formation and recognition, and robotics [3]. The structure of this network is easy to implement in very large integration (VLSI) technologies.

With the transistor size shrinking close to sub 100 nm, many serious issues emerge, including the difficulty in fabrication and fundamental performance defects of transistor devices. It is believed that flash memory will approach the end of scaling down in a decade. Traditional transistor-based memory encounters the development bottleneck [4]. In order to overcome the limitations of CMOS, many nanometer-scale technologies have been proposed and utilized.

The devices and circuits based on quanta tunneling shows their advantages in nanometer-scale technology. Among several proposed nanometer electronic devices, the resonant tunneling diode possesses a relatively easy fabrication process and folded-back negative differential resistance (NDR) in current-voltage (*I-V*) characteristics [5, 6], thereby exploring several applications in both digital and analog circuits [7]. Some researchers have also stated that RTD have extensive applications in Microwave oscillator circuit for terahertz and can be as the main prospect for the basic element of the new logic cell in high-speed digital integrated circuit.

By combining the advantages of RTD in NDR feature and CNN in parallel and high performance capacity in data processing, the hybrid RTD-CNN is expected to have higher speed, more compact structure, and greater capability [6–8]. However, the growing size in RTD-CNN increases the likelihood of faulting components existing in such neural networks. To some extent, we expect that RTD-CNNs with few faults may also provide acceptable results based on the inherent fault-tolerant property of artificial neural networks. It is also an essential task to evaluate the fault-tolerant ability of CNN.

This paper investigates the resonant tunneling diodes based cellular neural/nonlinear networks (RTD-CNN) with fault-tolerant properties in image processing. In Section 2, the structure of the RTD-CNN is depicted in details, including the RTD model, CNN model, and the hybrid model. Then, in Section 3, fault tolerance, an important feature of RTD-CNNs, is analyzed, and the stability of RTD-CNNs with a single stuck-at fault is proved. A series of computer simulations and numerical analysis are given in Section 4. Conclusions are drawn in Section 5.

2. Structure of the RTD-Based CNN Model

2.1. Resonant Tunneling Diodes (RTDs). Nanometer electronics make it prospective of ultralow power and ultrahigh integration density. Among the different nanometer electronic devices discovered up to now, the resonant tunneling diode stands on the center of the stage. RTDs possess extreme compactness, picoseconds switching speed, nonmonotonic voltage-current characteristics, and possible monolithic and vertical integration with GaAs FETs [5].

The basic RTD device configuration, a double barrier quantum well structure, can be measured in nanometers (as Figure 1). The structure has two contacts (called the emitter and the collector) made from a semiconductor with a small bandgap (e.g., GaAs), quantum barriers made from a semiconductor with a larger bandgap (e.g., AlGaAs or InGaAs), and a quantum well made from a smaller bandgap semiconductor. The wave nature of the electrons in such a structure leads to quantum phenomena like interference, tunneling, and energy quantization. The quantum well is so narrow (about 5 nm) that it can only contain a single energy level, the so-called resonant. Electrons wishing to travel from the emitter to the collector can only do so if they are lined up with this resonant energy level. The NDR property can be exploited to design compact bistable circuits without feedback in digital circuits.

The physical model of RTD has an effective mass approximation [8],

$$h(v) = A \ln\left[\frac{1+e^{(B-C+n_1 v)q/kT}}{1+e^{(B-C-n_1 v)q/kT}}\right] \times \left[\frac{\pi}{2} + \arctan\left(\frac{C-n_1 v}{D}\right)\right] + H\left(e^{n_2 q v/kT} - 1\right). \tag{1}$$

For the sake of simplicity, this characteristic can be approximated by a piecewise linear function as follows :

$$h(v) = \alpha v + \beta\left(\left|v+v_p\right| - \left|v-v_p\right|\right) + \gamma\left(\left|v+v_v\right| - \left|v-v_v\right|\right). \tag{2}$$

The *I-V* characteristics (NDR) of RTD is represented by a piece-wise linear model clearly. Initially, with a low voltage across the device (point A in Figure 2), the electrons are below the point of resonance, and no current can flow through the device. As the voltage increases, the emitter region is warped upwards, and the collector region is warped downwards. Eventually, the band of electrons in the emitter will line up with the resonant energy state and allows tunneling through to the collector (point B in Figure 2). With higher voltage, the electrons are pushed to pass the resonant energy level and are unable to continue tunneling, which can be observed by the drop in current to the valley at point C in Figure 2. If the voltage increases further, more and more electrons are able to flow over the top of the quantum barriers, and the current flow will rise.

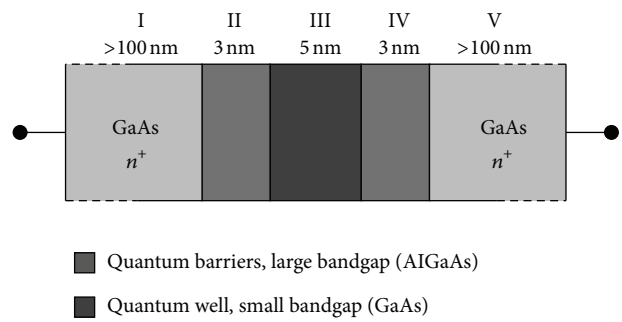


Figure 1: Resonant tunneling diode.

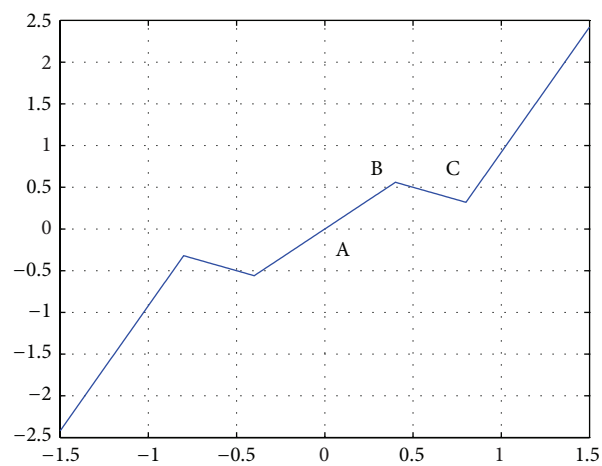


Figure 2: *I-V* characteristics of the RTD.

2.2. The Standard CNN. In recent twenty years, cellular neural networks have been greatly concerned and made significant progress in image processing and pattern recognition. The standard CNN circuit proposed by Chua and Yang is shown in Figure 3 [1]. The circuit of each cell in this network is shown in Figure 4, in which C is a linear capacitor. $I_{xu}(i,j;k,l)$ and $I_{xy}(i,j;k,l)$ are linear voltage-controlled current sources; I_{yx} is a piecewise-linear voltage-controlled current source; R_x,R_u, and R_y are linear resistors; E_{ij} is an independent voltage source.

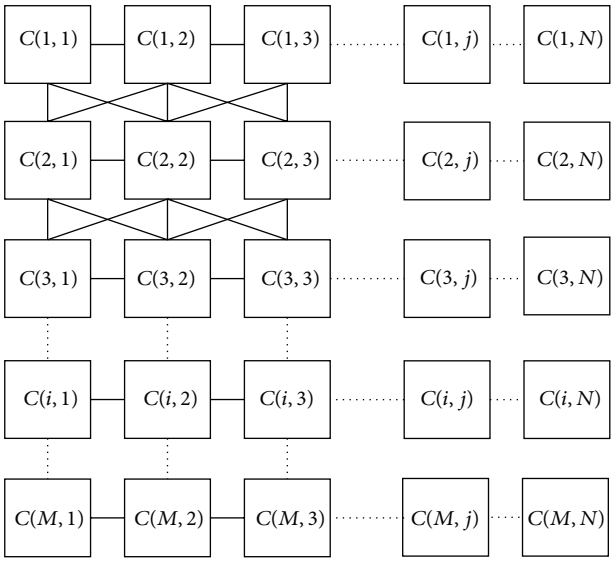


Figure 3: A 2D $M \times N$ CNN.

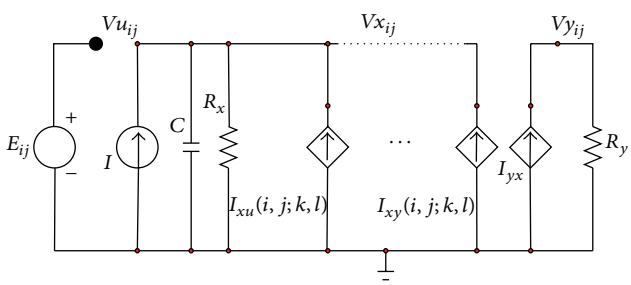


Figure 4: Schematic diagram of a CNN cell.

Applying KCL and KVL, the dynamics of a cell is derived as follows:

$$C\frac{dx_{ij}(t)}{dt} = -\frac{1}{R}x_{ij}(t) + \sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l)\, f(x_{kl}(t)) + \sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l)\, u_{kl}(t) + I, \quad 1 \le i \le M;\ 1 \le j \le N, \tag{3a}$$

$$f(x_{ij}) = \frac{1}{2}\left(\left|x_{ij}(t)+1\right| - \left|x_{ij}(t)-1\right|\right), \quad 1 \le i \le M;\ 1 \le j \le N, \tag{3b}$$

where $N_r(i, j)$ is the r-neighborhood of a cell (i, j) in a cellular neural network. $x_{ij}(t), u_{ij}$, and $f(x_{ij})$ are the state, input and output of the cell, respectively.

2.2.1. Remarks

(a) All inner cells of a cellular neural network have the same circuit structures and parameter values. An inner cell has $(2r + 1)^2$ neighbors. All other cells are called boundary cells whose neighborhood is usually set to 0. A cellular neural network is completely characterized by the set of nonlinear differential equations (3a) and (3b).

(b) The dynamics of a cellular neural network has both output feedback and input control mechanisms. The output feedback effect depends on the interactive parameter $A(i, j; k, l)$ and the input control effect depends on $B(i, j; k, l)$. Consequently, it is sometimes instructive to refer to $A(i, j; k, l)$ as a feedback operator and $B(i, j; k, l)$ as a control operator.

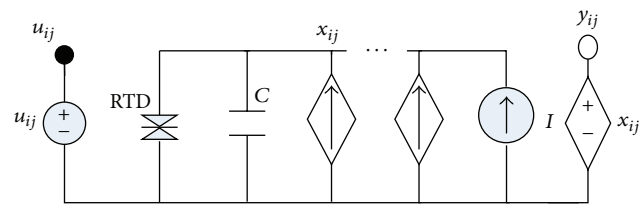


Figure 5: Schematic diagram of an RTD-CNN cell.

2.3. The RTD-CNN Model. The linear resistor in the standard CNN cell is replaced by an RTD in an RTD-CNN cell (Figure 5). Because of this, the circuit becomes nonlinear, and the nonlinear relationship between state and output is unnecessary any longer. In this RTD-CNN cell, the state and output are identical; that is, $y_{ij} = x_{ij}$.

Let $h(\cdot)$ denotes the normalized piecewise linear characteristics of the RTD (as (2)). Considering the voltage across the RTD as the state x of the cell, the dynamics of the RTD-CNN cell is governed by

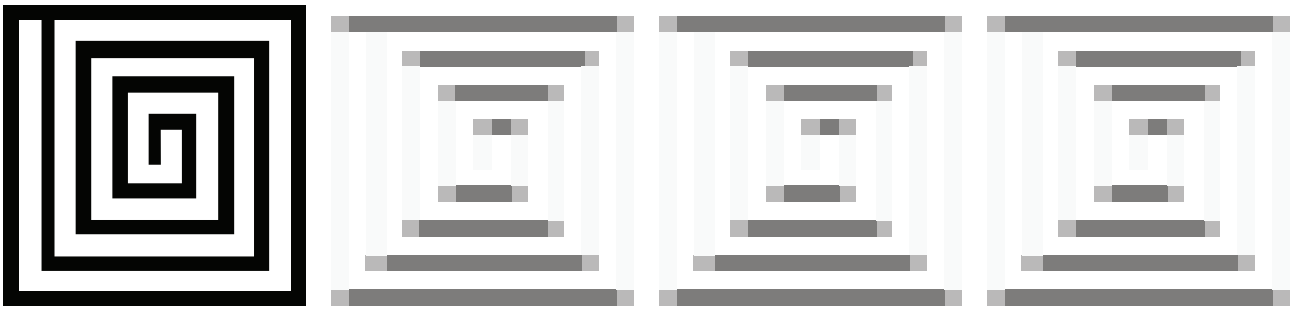
$$C\frac{dx_{ij}(t)}{dt} = -h\left(x_{ij}(t)\right) + \sum_{(k,l)}\left(a_{ij,kl}x_{kl}(t) + b_{ij,kl}u_{kl}\right) + I. \tag{4}$$

3. The Stability Analysis of RTD-CNN with Single Faulty Cell

Image processing is an important application for RTD-CNNs. It is always hoped that the network can still work smoothly even if there exists a faulty cell. For the sake of this purpose, the convergence feature of the RTD-CNN with single stuck-at fault must be guaranteed. In this section, the stability of the proposed RTD-CNN has been analyzed and proved.

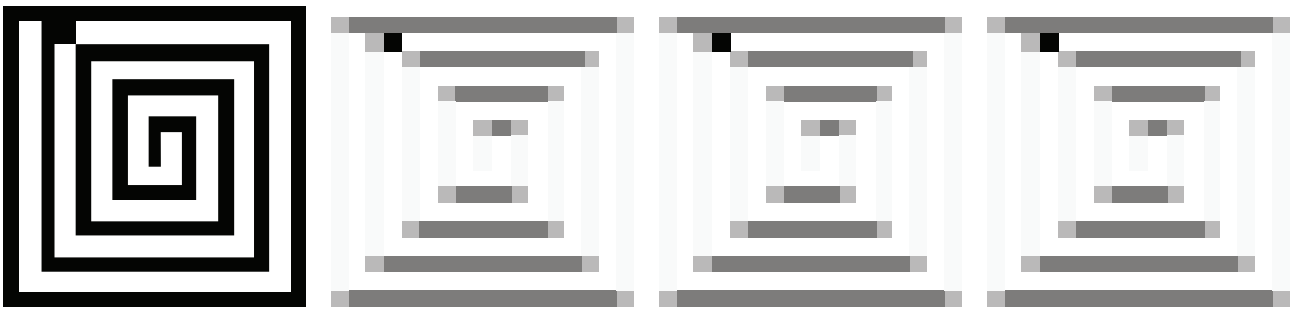
Definition 1. The stuck-at-α fault is defined that the state of a faulty cell is not changeable along with the inputs and outputs of other cells [9]:

$$F = \alpha, \quad \alpha \text{ is a constant.} \tag{5}$$



(a) (b) (c) (d)

Figure 6: Horizontal line detection with RTD-CNN. (a) Input image, (b) output image, $t = 20$, (c) output image, and $t = 30$, (d) output image, $t = 60$.



(a) (b) (c) (d)

Figure 7: Horizontal line detection with RTD-CNN with single faulty cell. (a) Input image, (b) output image, $t = 20$, (c) output image, $t = 30$, (d) output image, $t = 60$.

Then the dynamics of the RTD-CNN cell with single stuck-at fault is governed by

$$C\frac{dx_{ij}(t)}{dt} = -h\left(x_{ij}(t)\right) + \sum_{(k,l)}\left(\overline{a}_{ij,kl}x_{kl}(t) + b_{ij,kl}u_{kl}\right) + I + \alpha_f\alpha, \tag{6}$$

where a_f and $\overline{a}$ are called the fault and rest template, respectively. Then, some main results can be given.

Definition 2. A Lyapunov function $E(t)$ for the RTD-CNN cell with single stuck-at-α fault is as follows:

$$E(t) = -\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)}\overline{a}_{ij,kl}x_{kl}(t)\,x_{ij}(t) - \sum_{(i,j)}\sum_{(k,l)}b_{ij,kl}u_{kl}(t)\,x_{ij}(t) - \sum_{(i,j)}Ix_{ij}(t) + \sum_{(i,j)}\int_0^{x_{ij}}h(s)\,ds - \sum_{(i,j)}a_f\alpha x_{ij}(t). \tag{7}$$

Theorem 3. *The function $E(t)$ is monotonically decreasing; that is,*

$$\frac{dE(t)}{dt} \le 0. \tag{8}$$

Proof. If the feedback template is symmetric (i.e., $a_{ij,kl} = a_{kl,ij}$), the derivative of the energy function with respect to time t is

$$\begin{aligned}\frac{dE(t)}{dt} = &-\sum_{(i,j)}\sum_{(k,l)}\overline{a}_{ij,kl}x_{kl}(t)\frac{dx_{ij}(t)}{dt} \\ &-\sum_{(i,j)}\sum_{(k,l)}b_{ij,kl}u_{kl}(t)\frac{dx_{ij}(t)}{dt} \\ &-\sum_{(i,j)}I\frac{dx_{ij}(t)}{dt} + \sum_{(i,j)}\frac{dx_{ij}(t)}{dt}\frac{d}{dx_{ij}(t)} \\ &\times\int_0^{x_{ij}}h(s)\,ds - \sum_{(i,j)}a_f\alpha\frac{dx_{ij}(t)}{dt}.\end{aligned} \tag{9}$$

According to the definition of the cellular neural networks, we have

$$a(i,j;k,l) = 0, \qquad b(i,j;k,l) = 0, \quad \text{for } C(k,l) \notin N_r(i,j). \tag{10}$$

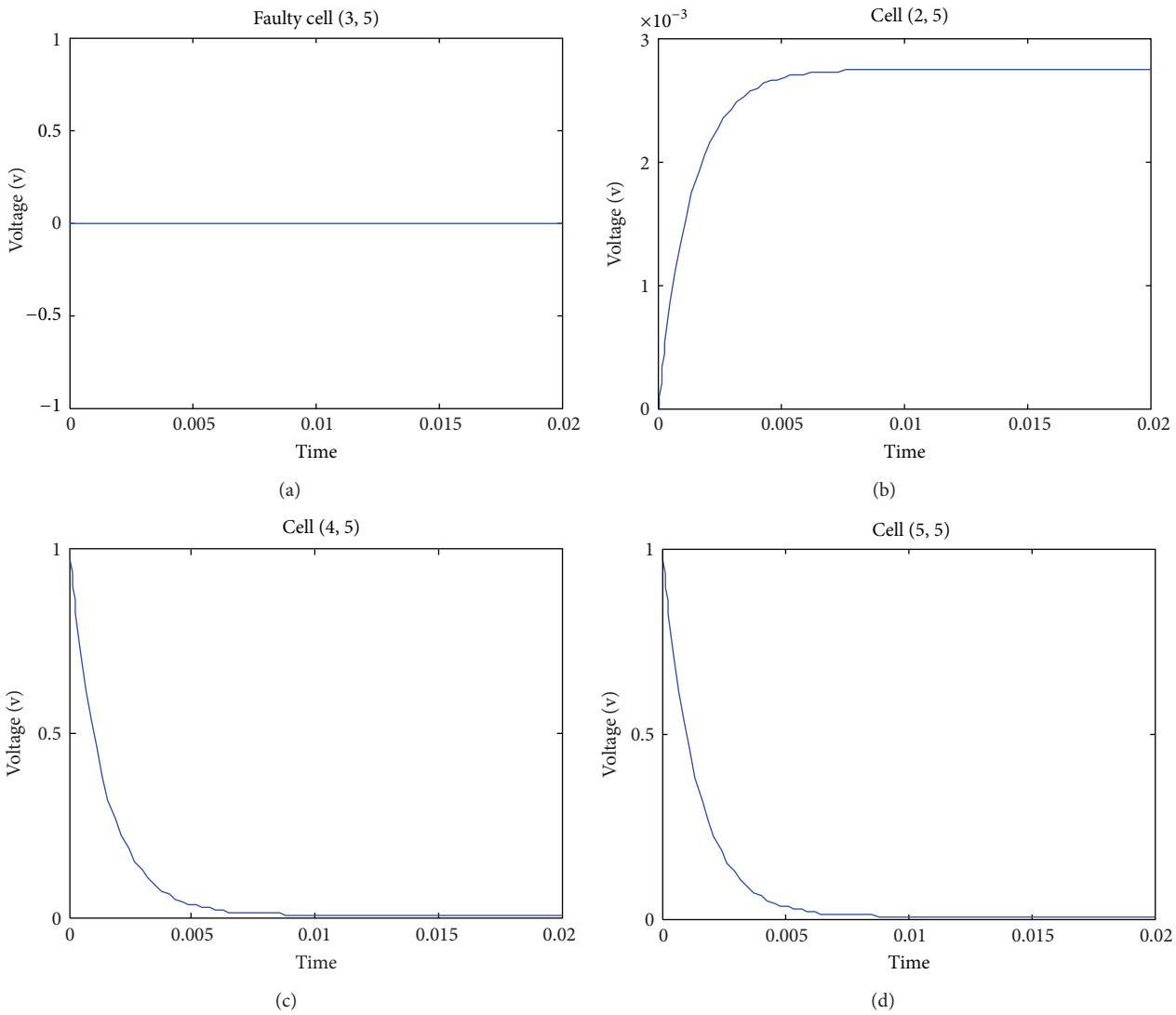


Figure 8: The dynamics of the state variable of cells in horizontal line detection of the RTD-CNN with single faulty cell.

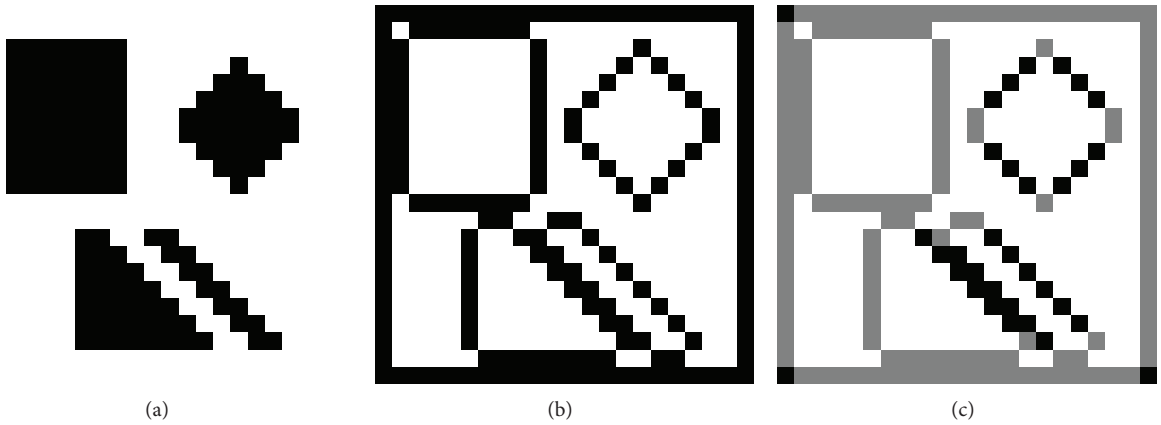
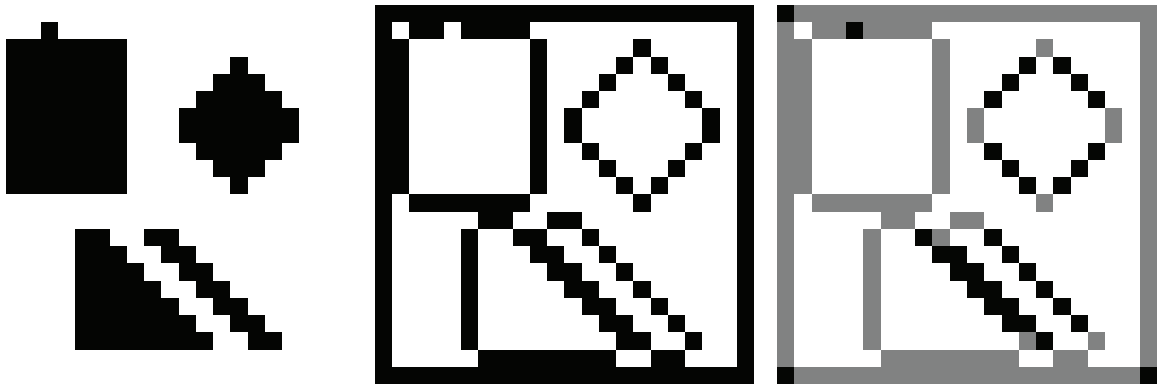


Figure 9: Edge extraction with RTD-CNN. (a) Input image, (b) output image, $q = 1$, and (c) output image, $q = 0.16$.



(a) (b) (c)

Figure 10: Edge extraction with RTD-CNN with a single faulty cell. (a) Input image, (b) output image, $q = 1$, and (c) output image, $q = 0.16$.

Then,

$$\begin{aligned}\frac{dE(t)}{dt} &= -\sum_{(i,j)} \frac{dx_{ij}(t)}{dt} \left\{ \sum_{(k,l)} \bar{a}_{ij,kl} x_{kl}(t) + \sum_{(k,l)} b_{ij,kl} u_{kl}(t) \right. \\ &\qquad \left. + I - \frac{d}{dx_{ij}(t)} \int_0^{x_{ij}} h(s)\, ds + a_f \alpha \right\} \\ &= -\sum_{(i,j)} \frac{dx_{ij}(t)}{dt} \left\{ -h\left(x_{ij}(t)\right) \right. \\ &\qquad + \sum_{(k,l)} \left(\bar{a}_{ij,kl} x_{kl}(t) + b_{ij,kl} u_{kl}(t) \right) \\ &\qquad \left. + I + \alpha_f \alpha \right\}. \end{aligned} \tag{11}$$

Substitute (4) into (11)

$$\frac{dE(t)}{dt} = -\sum_{(i,j)} C \left(\frac{dx_{ij}(t)}{dt} \right)^2 \leq 0. \tag{12}$$

Obviously, from Theorem 3, it can be found that the proposed RTD-CNN with single faulty cell is stable, which is essential in image processing. □

4. Fault Tolerance in CNN

Taking the inherent fault-tolerant property of the cell neural network into account, the RTD-CNN in the presence of faults may also provide acceptable effects. In this section, we investigate the fault-tolerant property of the RTD-CNN through two image processing examples.

4.1. Horizontal Line Detection with RTD-CNN. A 19×19 RTD-CNN is employed to implement the horizontal line detection task in image processing. The RTD-CNN template is given as:

$$a = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}, \qquad I = 2. \tag{13}$$

Figures 6 and 7 show the outputs of the horizontal line detection of the RTD-CNN with the time evolvement, which illustrate that the RTD-CNN can detect the horizontal line (at time $t = 0.02$ s) successfully. Figure 8 shows the dynamics of the state variable of cells in horizontal line detection of the RTD-CNN with single faulty cell, which can achieve steadiness in 0.01 s. A RTD-CNN with single stuck-at-α faulty cell can also accomplish horizontal line detection task satisfactorily.

4.2. Edge Extraction with RTD-CNN. A 22×22 RTD-CNN is built to verify its edge extraction capacity in image processing. The RTD-CNN template is shown in the following equation:

$$a = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 & -q & 0 \\ -q & 4q & -q \\ 0 & -q & 0 \end{bmatrix}, \qquad I = 0. \tag{14}$$

Figures 9 and 10 show the results of edge extraction of the RTD-CNN with different values of q. The RTD-CNN discriminates between edge cells (black), cells around the edge (white), and "background" cells, whose steadiness state is 0 (grey) clearly. And corner cells can apparently be separated from other edge cells with a smaller value of q. It is evident that the RTD-CNN can extract the edge (at time $t = 0.1$ s) successfully, in spite of the effect of a single faulty cell.

Figure 11 shows the dynamics of the state variable of cells in edge extraction of RTD-CNN with a single faulty cell which can achieve steadiness in 0.01 s. Similarly, A RTD-CNN with single stuck-at-α faulty cell can also accomplish edge extraction task satisfactorily.

5. Conclusions

This paper demonstrates the characteristics of the RTDs and addresses the RTD-based cellular neural/nonlinear network (RTD-CNN), which makes simple the CNN cell structure but maintains its promising capabilities. Next, the RTD-CNN with single stuck-at fault is implemented and its stability is

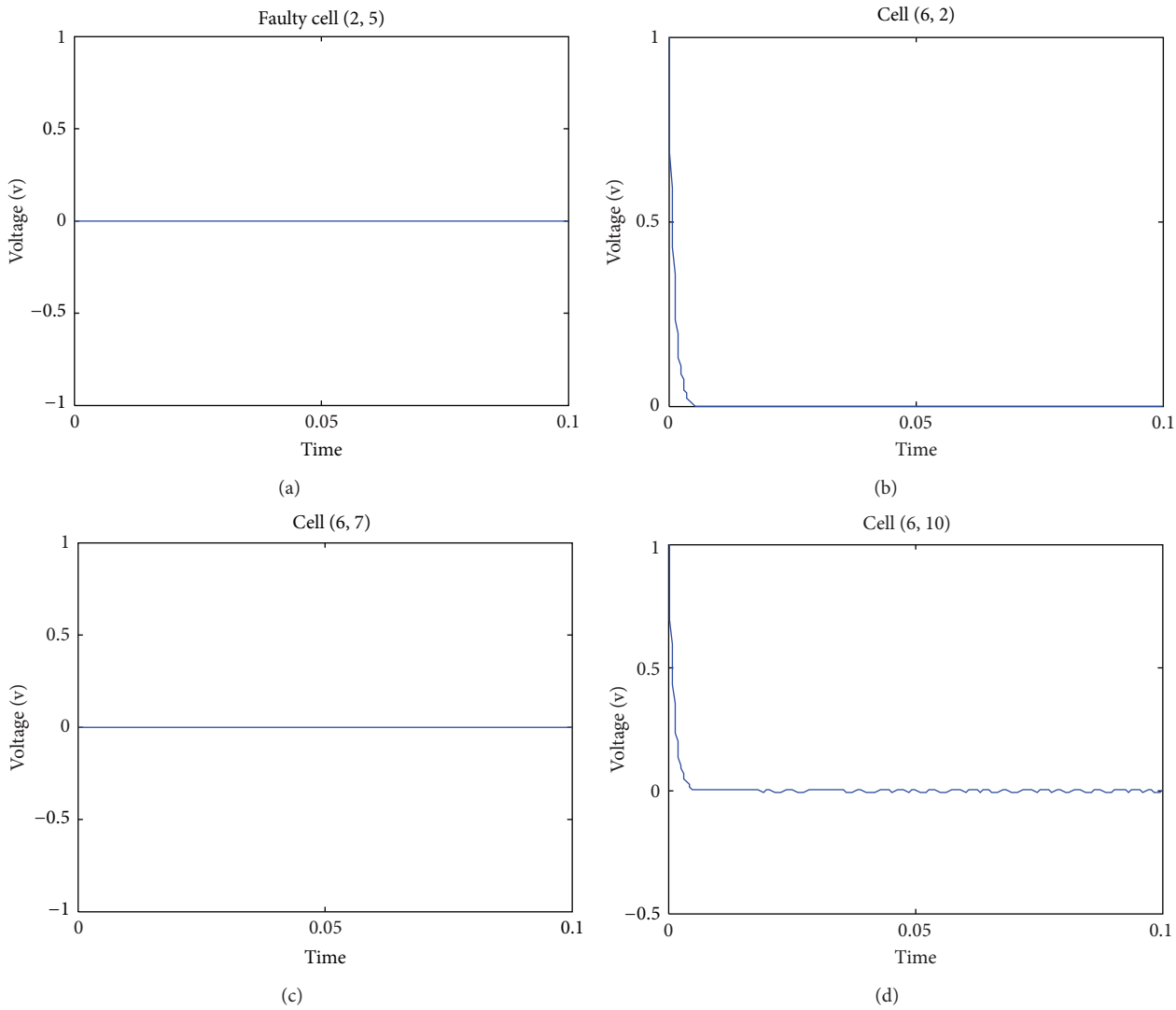


Figure 11: The dynamics of the state variable of cells in edge extraction of the RTD-CNN with a single faulty cell.

analyzed by Lyapunov stability theorem. Finally, the computer simulations of the RTD-CNN fully shows the satisfactory performance in image processing, such as horizontal line detection and edge extraction, improving the fault tolerance of RTD-CNN. Research results may provide a framework for further investigations in fault tolerant property of the RTD-CNNs and its applications in image processing.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (Grant Nos. 60972155 and 61101233), Fundamental Research Funds for the Central Universities (Grant Nos. XDJK2012A007 and XDJK2013B011), University Excellent Talents Supporting Foundations in of Chongqing (Grant No. 2011-65), University Key Teacher Supporting Foundations of Chongqing (Grant No. 2011-65), Technology Foundation for Selected Overseas Chinese Scholars, Ministry of Personnel in China (Grant No. 2012-186), and "Spring Sunshine Plan" Research Project of Ministry of Education of China (Grant No. z2011148).

References

[1] L. O. Chua and L. Yang, "Cellular neural networks: theory," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, pp. 1257–1272, 1988.

[2] L. O. Chua and T. Roska, *Cellular Neural Networks and Visual Computing*, Cambridge University Press, New York, NY, USA, 2002.

[3] L. O. Chua and T. Roska, "CNN paradigm," *IEEE Transactions on Circuits and Systems I*, vol. 40, no. 3, pp. 147–156, 1993.

[4] S. Duan, X. Hu, L. Wang, C. Li, and P. Mazumder, "Memristor-based RRAM with applications," *Science China Information Sciences*, vol. 55, no. 6, pp. 1446–1460, 2012.

[5] K. Karahaliloğlu and S. Balkir, "Nanostructure array of coupled RTDs as cellular neural networks," *International Journal of Circuit Theory and Applications*, vol. 31, no. 6, pp. 571–589, 2003.

[6] M. Hnggi and L. O. Chua, "Cellular neural networks based on resonant tunnelling diodes," *International Journal of Circuit Theory and Applications*, vol. 29, no. 5, pp. 487–504, 2001.

[7] P. Mazumder, S. Kulkarni, M. Bhattacharya, J. P. Sun, and G. I. Haddad, "Digital circuit applications of resonant tunneling devices," *Proceedings of the IEEE*, vol. 86, no. 4, pp. 664–686, 1998.

[8] S. R. Li, P. Mazumder, and L. O. Chua, "Cellular neural/nonlinear networks using resonant tunneling diode," in *Proceedings of the 4th IEEE Conference on Nanotechnology*, pp. 164–167, IEEE, Munich, Germany, August 2004.

[9] L. D. Wang, X. F. Yang, and S. K. Duan, "Analysis of fault tolerance of cellular neural networks and applications to image processing," in *Proceedings of the 3rd International Conference on Natural Computation*, J. Lei, J. T. Yao, and Q. Zhang, Eds., pp. 252–256, IEEE Computer Society, 2007.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 218682, 7 pages
http://dx.doi.org/10.1155/2013/218682

Research Article

Observer-Based Feedback Stabilization of Networked Control Systems with Random Packet Dropouts

Pengpeng Chen and Shouwan Gao

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

Correspondence should be addressed to Shouwan Gao; gaoshouwan@gmail.com

Received 27 March 2013; Accepted 13 May 2013

Academic Editor: Tingwen Huang

Copyright © 2013 P. Chen and S. Gao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper is concerned with observer-based feedback stabilization of networked control systems (NCSs) with random packet dropouts. Both sensor-to-controller (S/C) and controller-to-actuator (C/A) packet dropouts are considered, and their behavior is assumed to obey the Bernoulli random binary distribution. The hold-input strategy is adopted, in which the previous packet is used if the packet is lost. An observer-based feedback controller is designed, and sufficient conditions for stochastic stability are derived in the form of linear matrix inequalities (LMIs). A numerical example illustrates the effectiveness of the results.

1. Introduction

In the past few years, dramatic progress has been made for network analysis. Unparticular, many important results have been obtained in engineering, such as the stability analysis and controller design for networked control systems (NCSs) [1–3] and consensus for networked multiagent systems [4, 5]. While NCSs have recently been receiving increasing attention due to their advantages over classical feedback control systems, the insertion of the communication network gives rise to new challenges. Among these challenges, time delays and packet losses are two important factors that can severely degrade the performance of NCSs. In the past few years, much attention has been paid to the time delay problem of NCSs, see [6–10], to name a few. Moreover, some existing control methods for time delays [11] can be used to NCSs with time delay. On the other hand, the control problem of NCSs with packet losses has also attracted considerable research interests (see, e.g., [12, 13] and references therein). This paper focuses on the impact of packet losses on the controller design for NCSs.

An arguably popular approach for modeling the packet loss phenomenon is to view the packet loss as a binary switching sequence according to a Bernoulli process which takes on values of zero and one with certain probability. Recently, some results have been obtained on such model [15–17]. For example, in [15], the Kalman filtering problem is presented in the setting of intermittent observations and how the expected estimation error covariance depends on the tradeoff between loss probability and the system dynamics is showed. In [16], the problem of robust finite-horizon filtering is investigated for a class of uncertain systems with missing measurements. Moreover, in [17], the feedback stabilization schemes for discrete-time control systems with packet dropping network link are studied, and the feedback strategy presented is considerably simpler to implement.

It is noticed that, based on the Bernoulli distributed model, almost all the stability conditions and controller designs given in the aforementioned references are derived in terms of the assumption that the packet dropout exists only in the sensor-to-controller (S/C) side. The effect of controller-to-actuator (C/A) packet dropouts is neglected due to the complicated NCS modeling. Lately, there have appeared some research results which simultaneously consider S/C and C/A packet losses. In [14], the robust H_∞ control problem is considered for NCSs with both S/C and C/A random communication packet losses. By modeling the random packet loss as a linear function of the stochastic variable satisfying Bernoulli binary distribution, stability analysis and controller synthesis problems are investigated. In [18], the observer-based H_∞ control problem is studied for discrete-time mixed delay systems with random packet

dropouts and multiplicative noises. By modeling the packet-loss phenomenon as Bernoulli distributed white sequences, the packet losses from S/C and from C/A are simultaneously considered. Furthermore, in [19], the similar problem is investigated for a class of networked nonlinear systems with global Lipschitz nonlinearities and random communication packet losses. In the above works, the zero-input strategy is adopted, in which, without considering the disturbance, the actuator (controller) input is set to zero when the C/A (S/C) packet is lost. However, for systems whose state and input signals change little from one time step to the next, for example, process control systems, the strategy may not perform well. In these situations, the hold-input strategy, that the latest packet stored in the buffer is used when the C/A or S/C packet is lost, gives a better performance [20]. To the best of the authors' knowledge, based on the hold-input strategy, the problem of observer-based feedback stabilization for NCSs with random S/C and C/A packet dropouts has not been investigated to date, which motivates the present study.

In this paper, the observer-based feedback stabilization problem for NCSs with both random S/C and C/A packet losses is considered. If the packet is lost, the hold-input strategy is adopted. Sufficient conditions for stochastic stability are given, and corresponding controller design steps are provided. An example is finally given to show the effectiveness of the control scheme proposed.

2. Problem Formulation

Consider the following NCS with random data packet dropouts shown in Figure 1, where sensors, controllers, and actuators are clock-driven:

$$x(k+1) = Ax(k) + Bu_c(k), \tag{1}$$

$$y_c(k) = (1-\alpha_k)Cx(k) + \alpha_k y_c(k-1), \tag{2}$$

$$u_c(k) = (1-\beta_k)u(k) + \beta_k u_c(k-1), \tag{3}$$

where $x(k) \in R^n$ is the state vector, $u_c(k) \in R^m$ is the control input to the actuator, $u(k) \in R^m$ is the desired control input computed by the controller, and $y_c(k) \in R^p$ is the measurement vector with transmission missing. A, B, and C are known real constant matrices with appropriate dimensions. The stochastic variable α_k models the S/C packet loss: if the measurement packet is lost, then $\alpha_k = 1$. Otherwise $\alpha_k = 0$. Similarly, the stochastic variable β_k models the C/A packet loss: $\beta_k = 1$ implies that the control packet is lost, and $\beta_k = 0$ implies that the control packet is correctly delivered. (α_k, β_k) are independent identically distributed (i.i.d.) Bernoulli random variables with

$$\begin{aligned} \text{prob}\{\alpha_k = 1\} &= E\{\alpha_k\} = \overline{\alpha}, \\ \text{prob}\{\beta_k = 1\} &= E\{\beta_k\} = \overline{\beta}. \end{aligned} \tag{4}$$

The buffers store only one packet, and they will be updated if a new packet arrives. By such a mechanism, the buffers always store the most recent packet and are used for the purpose of packet dropout compensation. For buffer 1, if the measurement packet is correctly delivered, then $\alpha_k = 0$, that is, $y_c(k) = Cx(k)$, while if the packet is lost, $\alpha_k = 1$, that is, $y_c(k) = y_c(k-1)$. This observation model is summarized by (2). For buffer 2, if the packet is correctly delivered, then $\beta_k = 0$, that is, $u_c(k) = u(k)$. Otherwise, the actuator will employ the previous control value, that is, $\beta_k = 1$, $u_c(k) = u_c(k-1)$, as suggested in [21]. This compensation scheme is summarized by (3).

The dynamic observer-based control scheme for the system is described as follows:

$$\begin{aligned} \hat{x}(k+1) = {} & A\hat{x}(k) + Bu_c(k) \\ & + L\left[y_c(k) - (1-\overline{\alpha})C\hat{x}(k) - \overline{\alpha}y_c(k-1)\right], \\ & u(k) = K\hat{x}(k), \end{aligned} \tag{5}$$

where $\hat{x}(k) \in R^n$ is the state estimation and $L \in R^{n\times p}$ and $K \in R^{m\times n}$ are the observer and controller gains, respectively.

Let the estimation error be

$$e(k) = x(k) - \hat{x}(k), \tag{6}$$

and by substituting (2), (3), and (5) to (1) and (6), we have

$$\begin{aligned} x(k+1) = {} & \left[A + \left(1-\overline{\beta}\right)BK\right]x(k) \\ & - \left(1-\overline{\beta}\right)BKe(k) + \overline{\beta}Bu_c(k-1) \\ & - \left(\beta_k - \overline{\beta}\right)BKx(k) + \left(\beta_k - \overline{\beta}\right)BKe(k) \\ & + \left(\beta_k - \overline{\beta}\right)Bu_c(k-1), \\ e(k+1) = {} & [A - (1-\overline{\alpha})LC]e(k) \\ & + (\alpha_k - \overline{\alpha})LCx(k) - (\alpha_k - \overline{\alpha})Ly_c(k-1). \end{aligned} \tag{7}$$

Moreover, we can rewrite (2) and (3) as

$$\begin{aligned} y_c(k) = {} & (1-\overline{\alpha})Cx(k) + \overline{\alpha}y_c(k-1) \\ & - (\alpha_k - \overline{\alpha})Cx(k) + (\alpha_k - \overline{\alpha})y_c(k-1), \\ u_c(k) = {} & \left(1-\overline{\beta}\right)Kx(k) - \left(1-\overline{\beta}\right)Ke(k) + \overline{\beta}u_c(k-1) \\ & - \left(\beta_k - \overline{\beta}\right)Kx(k) + \left(\beta_k - \overline{\beta}\right)Ke(k) \\ & + \left(\beta_k - \overline{\beta}\right)u_c(k-1). \end{aligned} \tag{8}$$

By defining

$$z(k) = \left[x^T(k) \;\; e^T(k) \;\; y_c^T(k-1) \;\; u_c^T(k-1)\right]^T, \tag{9}$$

(7)-(8) can be rewritten in a compact form as follows:

$$z(k+1) = (\Phi_1 + \Phi_2)z(k), \tag{10}$$

where

$$\Phi_1 = \begin{bmatrix} A + \left(1-\overline{\beta}\right)BK & -\left(1-\overline{\beta}\right)BK & 0 & \overline{\beta}B \\ 0 & A-(1-\overline{\alpha})LC & 0 & 0 \\ (1-\overline{\alpha})C & 0 & \overline{\alpha} & 0 \\ \left(1-\overline{\beta}\right)K & -\left(1-\overline{\beta}\right)K & 0 & \overline{\beta} \end{bmatrix},$$

$$\Phi_2 = \begin{bmatrix} -\left(\beta_k-\overline{\beta}\right)BK & \left(\beta_k-\overline{\beta}\right)BK & 0 & \left(\beta_k-\overline{\beta}\right)B \\ (\alpha_k-\overline{\alpha})LC & 0 & -(\alpha_k-\overline{\alpha})L & 0 \\ -(\alpha_k-\overline{\alpha})C & 0 & \alpha_k-\overline{\alpha} & 0 \\ -\left(\beta_k-\overline{\beta}\right)K & \left(\beta_k-\overline{\beta}\right)K & 0 & \beta_k-\overline{\beta} \end{bmatrix}. \tag{11}$$

Since α_k and β_k are stochastic variables, we need to introduce the following definition before proceeding further.

Definition 1. System (1) is said to be mean square stable if for any $\varepsilon > 0$, there is a $\delta(\varepsilon) > 0$ such that $E\{|x(k)|^2\} < \varepsilon$, $k > 0$ when $E\{|x(0)|^2\} < \delta(\varepsilon)$. In addition, if $\lim_{k\to\infty} E\{|x(k)|^2\} = 0$ for any initial conditions, the system (1) is said to be globally mean-square asymptotically stable (GMSAS).

3. Main Results

In this section, we shall discuss the observer-based feedback controller design problem for (1). Without loss of generality, we make the following assumption.

Assumption 2. The matrix $B \in R^{n\times m}$ is of full-column rank; that is, $\mathrm{Rank}(B) = m$.

For the matrix $B \in R^{n\times m}$ being of full-column rank, there always exist two orthogonal matrices $U \in R^{n\times n}$ and $V \in R^{m\times m}$, such that

$$UBV = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} BV = \begin{bmatrix} \Lambda \\ 0 \end{bmatrix}, \tag{12}$$

where $U_1 \in R^{m\times n}$, $U_2 \in R^{(n-m)\times n}$ and $\Lambda = \mathrm{diag}\{\lambda_1 \ \lambda_2 \ \cdots \ \lambda_m\}$, in which λ_i $(i = 1, 2, \ldots, m)$ are nonzero singular values of B.

Furthermore, in order to derive the main result, the following lemma will be needed.

Lemma 3 (see [22]). *Let the matrix $B \in R^{n\times m}$ be of full-column rank. If matrix $P \in R^{n\times n}$ is of the following structure*

$$P = U^T \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} U = U_1^T P_1 U_1 + U_2^T P_2 U_2, \tag{13}$$

where $P_1 \in R^{m\times m} > 0$, $P_2 \in R^{(n-m)\times(n-m)} > 0$, and U_1, U_2 are defined in (12)*; then there exists a nonsingular matrix $\widetilde{P} \in R^{m\times m}$ such that $B\widetilde{P} = PB$.*

Theorem 4. *Consider the networked control problem in Figure 1. Given the dynamic observer-based control scheme* (5)*, system* (1) *is GMSAS for any S/C and C/A random data packet dropouts if there exist matrices $P > 0$, $S > 0$, $M > 0$, $N > 0$, L, and K satisfying*

$$\begin{bmatrix} \Omega_1 & \Pi_1 & \Pi_2 \\ * & \Omega_1^{-1} & 0 \\ * & * & \Omega_2^{-1} \end{bmatrix} < 0, \tag{14}$$

where

$$\begin{gathered} \Omega_1 = \mathrm{diag}\{-P \ \ -S \ \ -M \ \ -N\}, \\ \Pi_1 = \Phi_1^T, \\ \Pi_2 = \begin{bmatrix} BK & -BK & 0 & -B \\ LC & 0 & -L & 0 \\ C & 0 & -I & 0 \\ K & -K & 0 & -I \end{bmatrix}^T, \\ \Omega_2 = \mathrm{diag}\{-\alpha_1^2 P \ \ -\alpha_2^2 S \ \ -\alpha_2^2 M \ \ -\alpha_1^2 N\}, \end{gathered} \tag{15}$$

in which $\alpha_1 = [\overline{\beta}(1-\overline{\beta})]^{1/2}$, $\alpha_2 = [\overline{\alpha}(1-\overline{\alpha})]^{1/2}$.

Proof. To prove the theorem, we introduce the following functions:

$$\begin{gathered} V(z(k), k) = V_1 + V_2 + V_3 + V_4, \\ V_1 = x^T(k) P x(k), \\ V_2 = e^T(k) S e(k), \\ V_3 = y_c^T(k-1) M y_c(k-1), \\ V_4 = u_c^T(k-1) N u_c(k-1), \end{gathered} \tag{16}$$

where P, S, M, and N are positive definite matrices to be determined. Then, we have

$$\begin{aligned} &E\{\Delta V \mid z(k), \ldots, z(0)\} \\ &\quad = E\{V(z(k+1), k+1) \mid z(k), \ldots, z(0)\} \\ &\qquad - V(z(k), k) \\ &\quad = E\{\Delta V_1 \mid z(k), \ldots, z(0)\} \\ &\qquad + E\{\Delta V_2 \mid z(k), \ldots, z(0)\} \\ &\qquad + E\{\Delta V_3 \mid z(k), \ldots, z(0)\} \\ &\qquad + E\{\Delta V_4 \mid z(k), \ldots, z(0)\}, \end{aligned} \tag{17}$$

where

$$\begin{aligned}
&E\{\Delta V_1 \mid z(k),\ldots,z(0)\}\\
&\quad=\Big\{\Big[A+\left(1-\overline{\beta}\right)BK\Big]x(k)-\left(1-\overline{\beta}\right)BKe(k)\\
&\qquad+\overline{\beta}Bu_c(k-1)\Big\}^T P\\
&\qquad\times\Big\{\Big[A+\left(1-\overline{\beta}\right)BK\Big]x(k)-\left(1-\overline{\beta}\right)BKe(k)\\
&\qquad\quad+\overline{\beta}Bu_c(k-1)\Big\}\\
&\qquad+E\left\{\left(\beta_k-\overline{\beta}\right)^2\right\}\left[BKx(k)-BKe(k)-Bu_c(k-1)\right]^T P\\
&\qquad\times\left[BKx(k)-BKe(k)-Bu_c(k-1)\right]-x^T(k)Px(k),\\
&E\{\Delta V_2 \mid z(k),\ldots,z(0)\}\\
&\quad=\{[A-(1-\overline{\alpha})LC]e(k)\}^T S\{[A-(1-\overline{\alpha})LC]e(k)\}\\
&\qquad+E\left\{\left(\alpha_k-\overline{\alpha}\right)^2\right\}\left[LCx(k)-Ly_c(k-1)\right]^T\\
&\qquad\times S\left[LCx(k)-Ly_c(k-1)\right]-e^T(k)Se(k),\\
&E\{\Delta V_3 \mid z(k),\ldots,z(0)\}\\
&\quad=\left\{(1-\overline{\alpha})Cx(k)+\overline{\alpha}y_c(k-1)\right\}^T\\
&\qquad\times M\left\{(1-\overline{\alpha})Cx(k)+\overline{\alpha}y_c(k-1)\right\}\\
&\qquad+E\left\{\left(\alpha_k-\overline{\alpha}\right)^2\right\}\left[Cx(k)-y_c(k-1)\right]^T\\
&\qquad\times M\left[Cx(k)-y_c(k-1)\right]\\
&\qquad-y_c^T(k-1)My_c(k-1),\\
&E\{\Delta V_4 \mid z(k),\ldots,z(0)\}\\
&\quad=\left\{\left(1-\overline{\beta}\right)Kx(k)-\left(1-\overline{\beta}\right)Ke(k)+\overline{\beta}u_c(k-1)\right\}^T N\\
&\qquad\times\left\{\left(1-\overline{\beta}\right)Kx(k)-\left(1-\overline{\beta}\right)Ke(k)+\overline{\beta}u_c(k-1)\right\}\\
&\qquad+E\left\{\left(\beta_k-\overline{\beta}\right)^2\right\}\left[Kx(k)-Ke(k)-u_c(k-1)\right]^T N\\
&\qquad\times\left[Kx(k)-Ke(k)-u_c(k-1)\right]\\
&\qquad-u_c^T(k-1)Nu_c(k-1).
\end{aligned}\tag{18}$$

Noting that $E\{(\alpha_k-\overline{\alpha})^2\}=\overline{\alpha}(1-\overline{\alpha})$ and $E\{(\beta_k-\overline{\beta})^2\}=\overline{\beta}(1-\overline{\beta})$, we obtain

$$E\{\Delta V \mid z(k),\ldots,z(0)\}=z^T(K)\Sigma z(k),\tag{19}$$

in which

$$\Sigma=\Omega_1-\Pi_1\Omega_1\Pi_1^T-\Pi_2\Omega_2\Pi_2^T.\tag{20}$$

By the Schur complement, (14) guarantees $\Sigma<0$. Thus, for all $z(k)\neq 0$, it is easy to know that

$$E\{V(z(k+1),k+1) \mid z(k),\ldots,z(0)\}-V(z(k),k)<0.\tag{21}$$

That is, there exists $0<\gamma<1$ satisfying

$$E\{V(z(k+1),k+1) \mid z(k),\ldots,z(0)\}\leq\gamma V(z(k),k).\tag{22}$$

Using the smoothing property

$$\begin{aligned}
&E\{V(z(k),k) \mid z(k-2),\ldots,z(0)\}\\
&\quad=E\{E\{V(z(k),k) \mid z(k-1),\ldots,z(0)\} \mid\\
&\qquad z(k-2),\ldots,z(0)\}
\end{aligned}\tag{23}$$

in (22) after taking the conditional expectation, we can conclude

$$\begin{aligned}
&E\{V(z(k),k) \mid z(k-2),\ldots,z(0)\}\\
&\quad\leq\gamma E\{V(z(k-1),k-1) \mid z(k-2),\ldots,z(0)\}\\
&\quad\leq\gamma^2 V(z(k-2),k-2).
\end{aligned}\tag{24}$$

Continuing this process, we get

$$E\{V(z(k),k)\}\leq\gamma^k V(z(0),0),\tag{25}$$

which implies that

$$\begin{aligned}
E\left\{\sum_{k=0}^{N}V(z(k),k)\right\}&\leq\left(1+\gamma+\cdots+\gamma^N\right)V(z(0),0)\\
&=\frac{1-\gamma^{N+1}}{1-\gamma}V(z(0),0).
\end{aligned}\tag{26}$$

Since S, M, and N are positive definite matrices, it is easy to conclude that

$$\lim_{N\to\infty}E\left\{\sum_{k=0}^{N}x^T(k)x(k)\right\}\leq\frac{1}{(1-\gamma)\lambda_{\min}(P)}V(z(0),0),\tag{27}$$

which implies $\lim_{N\to\infty}E\{|x(k)|^2\}=0$, and the proof is completed. □

Note that the condition (14) is not an LMI, hence, cannot be solved by MATLAB LMI Toolbox. In the following, we will deal with the controller design problem and derive the explicit expression of the controller parameters in terms of LMI.

Theorem 5. *Consider the networked control problem in Figure 1. The dynamic observer-based control scheme* (5) *exists such that system* (1) *is GMSAS for any S/C and C/A random data packet dropouts if there exist matrices $P_1>0$, $P_2>0$, $S>0$, $M>0$, G, and H, satisfying*

$$\begin{bmatrix}\Omega_1 & \Xi_1 & \Xi_2\\ * & \Omega_1 & 0\\ * & * & \Omega_1\end{bmatrix}<0,\tag{28}$$

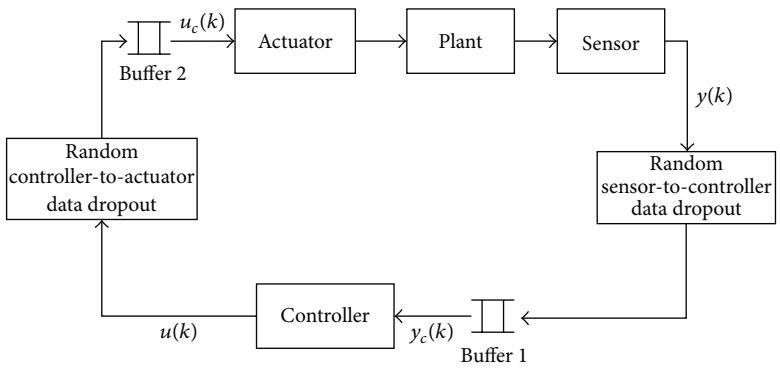


FIGURE 1: Structure of an NCS with random data packet dropouts.

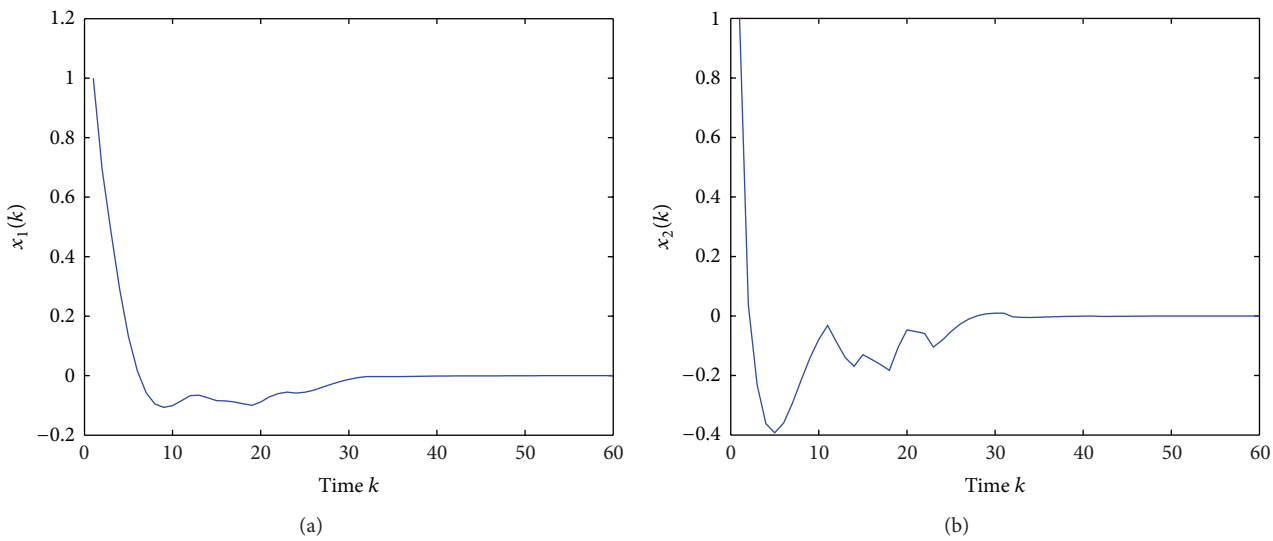


FIGURE 2: State response with controller in [14]. (a) First component of the state vector $x(k)$. (b) Second component of the state vector $x(k)$.

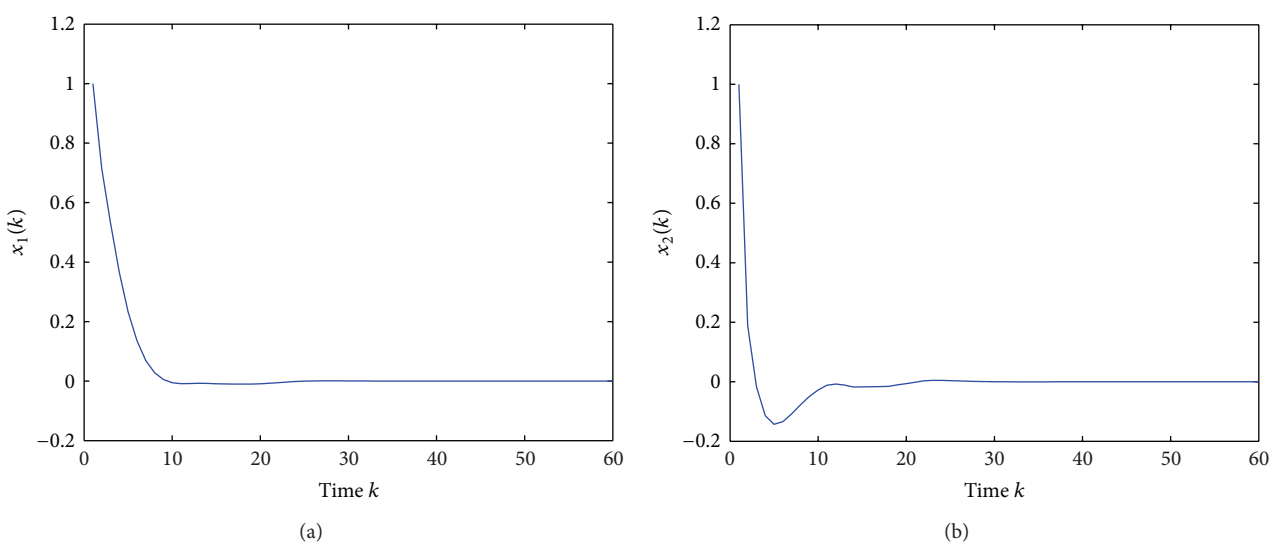


FIGURE 3: State response with controller proposed in this paper. (a) First component of the state vector $x(k)$. (b) Second component of the state vector $x(k)$.

where

$$\Xi_1 = \begin{bmatrix} PA + \left(1-\overline{\beta}\right)BG & -\left(1-\overline{\beta}\right)BG & 0 & \overline{\beta}PB \\ 0 & SA-(1-\overline{\alpha})HC & 0 & 0 \\ (1-\overline{\alpha})MC & 0 & \overline{\alpha}M & 0 \\ \left(1-\overline{\beta}\right)V\Lambda^2V^TG & -\left(1-\overline{\beta}\right)V\Lambda^2V^TG & 0 & \overline{\beta}N \end{bmatrix}^T,$$

$$\Xi_2 = \begin{bmatrix} \alpha_1 BG & -\alpha_1 BG & 0 & -\alpha_1 PB \\ \alpha_2 HC & 0 & -\alpha_2 H & 0 \\ \alpha_2 MC & 0 & -\alpha_2 M & 0 \\ \alpha_1 V\Lambda^2V^TG & -\alpha_1 V\Lambda^2V^TG & 0 & -\alpha_1 N \end{bmatrix}^T, \tag{29}$$

in which $P = U_1^TP_1U_1 + U_2^TP_2U_2$, $N = V\Lambda P_1\Lambda V^T$, *and* U_1, U_2, V, *and* Λ *come from* (12). *Moreover, the controller parameters are given by*

$$K = V\Lambda^{-1}P_1^{-1}\Lambda V^TG, \qquad L = S^{-1}H. \tag{30}$$

Proof. Since there exist $P_1 > 0$, and $P_2 > 0$, such that $P = U_1^TP_1U_1 + U_2^TP_2U_2$, where U_1, and U_2 come from (12). It follows from Lemma 3 that there exists a nonsingular matrix $\widetilde{P} = V\Lambda^{-1}P_1\Lambda V^T$ such that $B\widetilde{P} = PB$. Let $B\widetilde{P} = PB$, $G = \widetilde{P}K$, and $H = SL$., we can conclude that (28) is equivalent to (14). □

4. Numerical Example

In this section, an example is used to demonstrate that the controller designed in this paper is effective. We compare our strategy with zero-input strategy [14], which shows that, for systems whose state and input signals change little from one time step to the next, the hold-input strategy adopted in the paper has a better performance.

Consider the following discrete-time unstable system:

$$A = \begin{bmatrix} 0.7 & 0.2 \\ 0 & 1.12 \end{bmatrix}, \quad B = \begin{bmatrix} -0.03 \\ -1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{31}$$

It is assumed that $\overline{\alpha} = \overline{\beta} = 0.1$.

By zero-input strategy in [14], the controller gain and the observer gain are $K = [0.2235 \;\; 0.4312]$ and $L = [0.9064 \;\; 1.5446]^T$, respectively. Suppose that the initial conditions are $x(0) = [1 \;\; 1]^T$, $\hat{x}(0) = [1 \;\; 1]^T$, and $u_c(-1) = y_c(-1) = 0$. We have Figures 2(a) and 2(b). Then, by Theorem 5 of our paper, we obtain $K = [0.1780 \;\; 0.3916]$ and $L = [0.5770 \;\; 1.1488]^T$. With the same initial conditions, we have Figures 3(a) and 3(b). From Figures 2 and 3 we know that the system can be stabilized by both controllers; however, by comparison, it is easy to know that our controller gives a better performance.

Remark 6. From the system in this example, we can observe that the state, measurement, and input signals change slowly. Moreover, since the latest packet stored in the buffer is used when the C/A or S/C packet is lost, our controller provides a smoother sequence of inputs. These are the reason why hold-input strategy adopted in the paper gives a better performance.

5. Conclusion

In this paper, the problem of observer-based feedback stabilization is considered for NCSs with random packet dropouts. The hold-input strategy is adopted, and sufficient conditions for stochastic stability are derived in the form of linear matrix. Furthermore, the dynamic observer-based control scheme is designed. An example shows that, for systems whose state and input change little from one time step to the next, the hold-input strategy adopted performs better than the zero-input strategy.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61202478 and the Fundamental Research Funds for the Central Universities under Grant 2012QNB18.

References

[1] J. Nilsson, *Real-time control systems with delays [Ph.D. thesis]*, Lund Institute of Technology, Lund, Sweden, 1998.

[2] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, 2001.

[3] D. Yue, Q. L. Han, and C. Peng, "State feedback controller design of networked control systems," *IEEE Transactions on Circuits and Systems II*, vol. 51, no. 11, pp. 640–644, 2004.

[4] H. Wang, X. Liao, and T. Huang, "Accelerated consensus to accurate average in multi-agent networks via state prediction," *Nonlinear Dynamics*, 2013.

[5] H. Li, X. Liao, and T. Huang, "Second-order locally dynamic consensus of multiagent systems with arbitrarily fast switching directed topologies," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. 99, pp. 1–11, 2013.

[6] P. V. Zhivoglyadov and R. H. Middleton, "Networked control design for linear systems," *Automatica*, vol. 39, no. 4, pp. 743–750, 2003.

[7] D. Yue, Q.-L. Han, and J. Lam, "Network-based robust H_∞ control of systems with uncertainty," *Automatica*, vol. 41, no. 6, pp. 999–1007, 2005.

[8] L. Zhang, Y. Shi, T. Chen, and B. Huang, "A new method for stabilization of networked control systems with random delays," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1177–1181, 2005.

[9] H. Gao, T. Chen, and J. Lam, "A new delay system approach to network-based control," *Automatica*, vol. 44, no. 1, pp. 39–52, 2008.

[10] M. Liu, D. W. C. Ho, and Y. Niu, "Stabilization of Markovian jump linear system over networks with random communication delay," *Automatica*, vol. 45, no. 2, pp. 416–421, 2009.

[11] S. P. Wen, Z. Zeng, and T. Huang, "H∞ Filtering for neutral systems with mixed delays and multiplicative noises," *IEEE Transactions on Circuits and Systems Part II*, vol. 59, no. 11, pp. 820–824, 2012.

[12] J. Xiong and J. Lam, "Stabilization of linear systems over networks with bounded packet loss," *Automatica*, vol. 43, no. 1, pp. 80–87, 2007.

[13] J. Wu and T. Chen, "Design of networked control systems with packet dropouts," *IEEE Transactions on Automatic Control*, vol. 52, no. 7, pp. 1314–1319, 2007.

[14] Z. Wang, F. Yang, D. W. C. Ho, and X. Liu, "Robust H_∞ control for networked systems with random packet losses," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 37, no. 4, pp. 916–924, 2007.

[15] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[16] Z. Wang, F. Yang, D. W. C. Ho, and X. Liu, "Robust finite-horizon filtering for stochastic systems with missing measurements," *IEEE Signal Processing Letters*, vol. 12, no. 6, pp. 437–440, 2005.

[17] H. Mo and C. N. Hadjicostis, "Feedback control over packet dropping network links," in *Proceedings of the Mediterranean Conference on Control and Automation (MED '07)*, Athens, Greece, July 2007.

[18] S. P. Wen, Z. Zeng, and T. Huang, "Observer-based H_∞ control of discrete time-delay systems with random communication packet losses and multiplicative noises," *Applied Mathematics and Computation*, vol. 219, no. 12, pp. 6484–6493, 2013.

[19] J. G. Li, J. Q. Yuan, and J. G. Lu, "Observer-based H_∞ control for networked nonlinear systems with random packet losses," *ISA Transactions*, vol. 49, no. 1, pp. 39–46, 2010.

[20] L. Schenato, "To zero or to hold control inputs with lossy links?" *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1093–1099, 2009.

[21] A. V. Savkin and I. R. Petersen, "Robust filtering with missing data and a deterministic description of noise and uncertainty," *International Journal of Systems Science*, vol. 28, no. 4, pp. 373–378, 1997.

[22] D. W. C. Ho and G. Lu, "Robust stabilization for a class of discrete-time non-linear systems via output feedback: the unified LMI approach," *International Journal of Control*, vol. 76, no. 2, pp. 105–115, 2003.