

Junichi Suzuki
Tadashi Nakano (Eds.)



87

Bio-Inspired Models of Network, Information, and Computing Systems

5th International ICST Conference, BIONETICS 2010
Boston, USA, December 2010
Revised Selected Papers



 Springer

Lecture Notes of the Institute
for Computer Sciences, Social Informatics
and Telecommunications Engineering

87

Editorial Board

Ozgur Akan

Middle East Technical University, Ankara, Turkey

Paolo Bellavista

University of Bologna, Italy

Jiannong Cao

Hong Kong Polytechnic University, Hong Kong

Falko Dressler

University of Erlangen, Germany

Domenico Ferrari

Università Cattolica Piacenza, Italy

Mario Gerla

UCLA, USA

Hisashi Kobayashi

Princeton University, USA

Sergio Palazzo

University of Catania, Italy

Sartaj Sahni

University of Florida, USA

Xuemin (Sherman) Shen

University of Waterloo, Canada

Mircea Stan

University of Virginia, USA

Jia Xiaohua

City University of Hong Kong, Hong Kong

Albert Zomaya

University of Sydney, Australia

Geoffrey Coulson

Lancaster University, UK

Junichi Suzuki Tadashi Nakano (Eds.)

Bio-Inspired Models of Network, Information, and Computing Systems

5th International ICST Conference

BIONETICS 2010

Boston, USA, December 1-3, 2010

Revised Selected Papers

 Springer

Volume Editors

Junichi Suzuki
University of Massachusetts, Boston
Department of Computer Science
100 Morrissey Blvd.
Boston, MA 02125, USA
E-mail: jxs@cs.umb.edu

Tadashi Nakano
Osaka University
Graduate School of Engineering
2-1 Yamadaoka, Suita
Osaka 565-0871, Japan
E-mail: tnakano@wakate.frc.osaka-u.a.jp

ISSN 1867-8211
ISBN 978-3-642-32614-1
DOI 10.1007/978-3-642-32615-8
Springer Heidelberg Dordrecht London New York

e-ISSN 1867-822X
e-ISBN 978-3-642-32615-8

Library of Congress Control Number: 2012944046

CR Subject Classification (1998): C.2, H.4, I.2.6, D.2, I.2.9-10, F.2.2, J.2, J.3

© ICST Institute for Computer Science, Social Informatics and Telecommunications Engineering 2012
This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.
The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

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

Printed on acid-free paper

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

Preface

The 5th International ICST Conference on Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS 2010) was held at Le Meridien Cambridge Hotel, USA during December 1–3, 2010. It was jointly organized by CREATE-NET in cooperation with the ACM Special Interest Group (SIG) on Simulation and Modeling, the ACM SIG on Genetic and Evolutionary Computation and the ACM SIG on Computers and Society.

BIONETICS 2010 was a three-day conference emphasizing diverse disciplines that seek the understanding of the fundamental principles and design strategies in biological systems and leverage those understandings to build bio-inspired systems. In order to cover diverse relevant research areas, BIONETICS 2010 featured a series of focused special tracks, including (1) Artificial Intelligence and Software Engineering Track, co-chaired by Shih-Hsi Liu and Marjan Mernik; (2) Artificial Life and Bio-inspired Robotics Track, co-chaired by Jian-Qin Liu and Haruhiko Nishimura; (3) Bioinformatics Track, co-chaired by Nurit Haspel and Amarda Shehu; (4) Bio-Inspired Machine Vision Track, co-chaired by Marc Pomplun and Tyler Garaas; (5) Game Theory and its Applications Track, co-chaired by Athanasios Vasilakos and Hung-Yu Wei; (6) Network-based Computation Track, co-chaired by Hideaki Suzuki and Hiroyuki Ohsaki; and (7) State-Topology Coevolution in Adaptive Networks Track, co-chaired by Hiroki Sayama and Thilo Gross. We appreciate all the track chairs for their tremendous efforts to organize excellent special tracks.

We received 66 regular paper submissions and accepted 26 of them. In addition, three invited papers, 28 work-in-progress papers, five demo papers and six short papers were accepted. We appreciate Program Committee members for their hard work in reviewing papers carefully and rigorously. With these 68 papers in total, we believe that BIONETICS 2010 delivered a high-quality, inspiring and informative technical program.

BIONETICS 2010 had five state-of-the-art keynote speeches by internationally-renowned researchers: Radhika Nagpal (Harvard University), Karl Lieberherr (Northeastern University), Thilo Gross (Max Planck Institute for the Physics of Complex Systems), Rezarta Islamaj Dogan (National Institutes of Health), Cheng Li (Harvard School of Public Health and Dana-Farber Cancer Institute). Keynotes were well received and discussed among the participants.

Three workshops were co-located with BIONETICS 2010: the 1st International Workshop on Bio-inspired Approaches to Advanced Computing and Communications (BioAdcom), organized by Ajith Abraham, K. Chandra Sekaran, Michael Alexander and Sabu Thampi; the 1st International Workshop on Bio-inspired Models and Technologies for Ambient Information Society (BioAmbIS), organized by Naoki Wakamiya and Kenji Leibnitz; and the 1st International Workshop on Evolutionary Computation and Machine Learning in Bioinformatics (BioLearn),

organized by Kenneth De Jong, Amarda Shehu and Uday Kamath. We thank all the workshop organizers as well as the Workshop Chair, Marc Pomplun, for their accomplishments to bring out successful workshops.

The conference had over 130 participants from various parts of the world such as Japan, India, China, Germany, Portugal, Austria and Netherlands as well as the US. The logistical organization of the conference is highly commendable, and a special acknowledgment is due to the Local Arrangement Chair, Tyler Garaas.

Special thanks go to the Steering Committee and Organizing Committee for their help that made our job much easier and enjoyable. We also wish to thank our sponsor, ICST, particularly Gergely Nagy, Ildiko Rezmuves and Gabriella Magyar for their professional assistance to run the conference smoothly.

We very much look forward to another successful conference in 2011 in Europe, and in the forthcoming years.

December 2010

Junichi Suzuki
Tadashi Nakano

VIII Organization

Marjan Mernik	University of Maribor, Slovenia AI and Software Engineering
Jian-Qin Liu	National Institute of Information and Communications Technology, Japan Artificial Life and Bio-inspired Robotics
Haruhiko Nishimura	University of Hyogo, Japan Artificial Life and Bio-inspired Robotics
Hiroyuki Ohsaki	Osaka University, Japan Network-Based Computation
Marc Pomplun	University of Massachusetts, Boston, USA Bio-inspired Machine Vision
Hiroki Sayama	Binghamton University, USA State-Topology Coevolution in Adaptive Networks
Amarda Shehu	George Mason University, USA Bioinformatics
Hideaki Suzuki	National Institute of Information and Communications Technology, Japan Network-Based Computation
Athanasios Vasilakos	University of Western Macedonia, Greece Game Theory and Its Applications
Hung-Yu Wei	National Taiwan University, Taiwan Game Theory and Its Applications

Workshop Chair

Marc Pomplun	University of Massachusetts, Boston, USA
--------------	--

Publication Chair

Foad Dabiri	University of California, Los Angeles, USA
-------------	--

Publicity Chairs

Pruet Boonma	Chiang Mai University, Thailand
Michael Moore	University of California, Irvine, USA

Local Arrangements Chair

Tyler Garaas	Mitsubishi Electric Research Laboratories, USA
--------------	---

Web Chair

Chonho Lee	University of Massachusetts, Boston, USA
------------	--

Conference Coordinators

Gergely Nagy, ICST
 Gabriella Magyar, ICST

Program Committee

Ajith Abraham	Norwegian University of Science and Technology, Norway
Andrew Adamatzky	University of the West of England, UK
Ozgun Akan	Middle East Technical University, Turkey
Eitan Altman	INRIA, France
Andrea Arcuri	Simula Research Laboratory, Norway
Ebrahim Bagheri	National Research Council Canada, Canada
Alain Barrat	Centre de Physique Théorique Marseille, France
Erhardt Barth	University of Lübeck, Germany
Kevin Bassler	University of Houston, USA
Jacob Beal	BBN Technologies, USA
Paolo Bellavista	University of Bologna, Italy
David Benavides	University of Seville, Spain
Pruet Boonma	Chiang Mai University, Thailand
Rainer Breitling	University of Glasgow, UK
Neil Bruce	York University, Canada
Yu Cao	California State University, Fresno, USA
Claudine Chaouiya	Instituto Gulbenkian de Ciência, Portugal
Sharat Chikkerur	Massachusetts Institute of Technology, USA
Phan Cong-Vinh	London South Bank University, UK
David Cox	Harvard University, USA
Kenneth De-Jong	George Mason University, USA
Gianni Di Caro	IDSIA, Switzerland
Da-Qiao Ding	National Institute of Information and Communications Technology, Japan
Wei Ding	University of Massachusetts, Boston, USA
Federico Divina	Pablo de Olavide University of Seville, Spain
Rezarta Islamaj Dogan	National Institutes of Health, National Library of Medicine, USA
Marco Dorigo	Université Libre de Bruxelles, Belgium
Michael Dorr	Schepens Eye Research Institute, USA
René Doursat	ISC-PIF, France
Falko Dressler	University of Erlangen, Germany
Martin Drozda	Leibniz University of Hannover, Germany
Jun-ping Du	Beijing University of Posts and Communications, China
Andrew Eckford	York University, Canada
Wilfried Elmenreich	University of Klagenfurt, Austria

Bogdan Filipic	Jozef Stefan Institute, Slovenia
Gianluigi Folino	Institute for High Performance Computing and Networking, National Research Council, Italy
Paraskevi Fragopoulou	FORTH-ICS, Greece
Maria Ganzha	Polish Academy of Sciences, Poland
Tyler Garaas	Mitsubishi Electric Research Laboratories, USA
Dragan Gasevic	Athabasca University, Canada
Gourab Ghoshal	Northeastern University, USA
Herbert Gintis	Santa Fe Institute and Central European University, USA
Hani Hamdan	Supelec, France
Nurit Haspel	University of Massachusetts, Boston, USA
Salima Hassas	Claude Bernard University, France
Jing He	Old Dominion University, USA
Monika Heiner	Brandenburg University of Technology at Cottbus, Germany
Pedro Henriques	University of Minho, Portugal
Henry Hess	University of Florida, USA
James Hill	Indiana University-Purdue University Indianapolis, USA
Daniel Howard	QinetiQ, UK
Alex Hwang	Schepens Eye Research Institute, USA
Mark Jelasity	University of Szeged, Hungary
George Kesidis	The Pennsylvania State University, USA
Tai-hoon Kim	Hannam University, Korea
Chonho Lee	University of Massachusetts, Boston, USA
Kenji Leibnitz	Osaka University, Japan
Ming Li	Nanjing University, China
Pietro Lio	University of Cambridge, UK
Chien-Hung Liu	National Taipei University of Technology, Taiwan
Jiming Liu	Hong Kong Baptist University, Hong Kong
Ivan Lukovic	University of Novi Sad, Serbia
Eric Yu-en Lu	University of Cambridge, UK
Gang Luo	Schepens Eye Research Institute, USA
Marco Mamei	University of Modena and Reggio Emilia, Italy
Carlo Mastroianni	Institute for High Performance Computing and Networking, National Research Council, Italy
Luis Mateus	Rocha Indiana University, USA
Claudio Mattiussi	Ecole Polytechnique Fédérale de Lausanne, Switzerland
Ronaldo Menezes	Florida Institute of Technology, USA
Satoru Miyano	Tokyo University, Japan

Michae Moore	University of California, Irvine, USA
Chad Myers	University of Minnesota, USA
Akira Namatame	National Defense Academy, Japan
Chrystopher Nehaniv	University of Hertfordshire, UK
Dusit Niyato	Nanyang Technological University, Singapore
Jorge Pacheco	Universidade do Minho, Portugal
Marcin Paprzycki	Polish Academy of Sciences, Poland
Ferdinand Peper	National Institute of Information and Communications Technology, Japan
Marc Pomplun	University of Massachusetts, Boston, USA
Daniele Quercia	Massachusetts Institute of Technology, USA
Huzefa Rangwala	George Mason University, USA
Eraldo Ribeiro	Florida Institute of Technology, USA
Derek Ruths	McGill University, Canada
Adnan Salihbegovic	University of Sarajevo, Bosnia and Herzegovina
Masahiro Sasabe	Osaka University, Japan
Hiroki Sayama	Binghamton University, USA
Frank Schweitzer	ETH Zurich, Switzerland
Giovanna Di Marzo Serugendo	Birkbeck College, University of London, UK
Leah Shaw	College of William and Mary, USA
Dan Simovici	University of Massachusetts, Boston, USA
Thamar Solorio	University of Alabama at Birmingham, USA
Reiji Suzuki	Nagoya University, Japan
Ivan Tanev	Doshisha University, Japan
Jun Tanimoto	Kyusyu University, Japan
P.S. Thiagarajan	National University of Singapore, Singapore
Jon Timmis	York University, UK
Kohji Tomita	AIST, Japan
Richard Torkar	Blekinge Institute of Technology, Sweden
Duc Tran	University of Massachusetts, Boston, USA
Athanasios Vasilakos	University of Western Macedonia, Greece
Dinesh Verma	IBM T.J. Watson Research Center, USA
Hiroshi Wada	National ICT Australia, Australia
Naoki Wakamiya	Osaka University, Japan
Huijuan Wang	Delft University of Technology, The Netherlands
Jane Wang	University of British Columbia, Canada
Yufeng Wang	National Institute of Information and Communications Technology, Japan
Jules White	Vanderbilt University, USA
Di Wu	Western Kentucky University at Bowling Green, USA
Alexander Wyglinski	Worcester Polytechnic Institute, USA
Lidia Yamamoto	University of Basel, Switzerland

Kiwon Yeom

Korea Institute of Science and Technology,
Korea

Eiko Yoneki

University of Cambridge, UK

Lixia Zhang

University of California, Los Angeles, USA

Chengcui Zhang

University of Alabama at Birmingham, USA

Du Zhang

California State University, Sacramento, USA

Yuming Zhou

Nanjing University, China

Table of Contents

Conference Papers

Self Tolerance by Tuning T-Cell Activation: An Artificial Immune System for Anomaly Detection	1
<i>Mário J. Antunes and Manuel E. Correia</i>	
Information Rates of Active Propagation in Microchannel Molecular Communication	16
<i>Nariman Farsad, Andrew W. Eckford, Satoshi Hiyama, and Yuki Moritani</i>	
Organic Resilience for Tactical Environments	22
<i>Marco Carvalho, Tom Lamkin, and Carlos Perez</i>	
Interfacing Living Cells via Molecular Communication	30
<i>Tadashi Nakano, Shouhei Kobayashi, and Tokuko Haraguchi</i>	
Evolutionary and Noise-Aware Data Gathering for Wireless Sensor Networks	32
<i>Bingchun Zhu, Junichi Suzuki, and Pruet Boonma</i>	
A Checkpoint-Orientated Modelling for Cell Cycle Simulation	40
<i>Jonathan Pascalie, Hervé Luga, Valérie Lobjois, Bernard Ducommun, and Yves Duthen</i>	
SOS Cloud: Self-organizing Services in the Cloud	48
<i>Bogdan Alexandru Caprarescu, Nicolò Maria Calcavecchia, Elisabetta Di Nitto, and Daniel J. Dubois</i>	
Biomimicking the Formation of Nacre/Shell: One Step Forward	56
<i>Feisal Khoushab, Montarop Yamabhai, and Kenneth J. Haller</i>	
Algorithmically Transitive Network: A Self-organizing Data-Flow Network with Learning	59
<i>Hideaki Suzuki, Hiroyuki Ohsaki, and Hidefumi Sawai</i>	
Modelling to Contain Pandemic Influenza A (H1N1) with Stochastic Membrane Systems: A Work-in-Progress Paper	74
<i>Lei Xu</i>	
P2P-Based Scalable Execution Platform for Algorithmically Transitive Network	82
<i>Mikio Yoshida, Hideaki Suzuki, and Hidefumi Sawai</i>	

Diagnosability of Nested Intruders	92
<i>Damas P. Gruska</i>	
Dynamics and Convergence of Resource Prices in Market-Oriented Overlay Networks	100
<i>Yutaka Okaie and Tadashi Nakano</i>	
Internet as a Dataflow Computer	102
<i>Hiroyuki Ohsaki, Hideaki Suzuki, and Hidefumi Sawai</i>	
Biologically Inspired Modeling of Smart Grid for Dynamic Power-Flow Control	111
<i>Hidefumi Sawai, Hideaki Suzuki, and Hiroyuki Ohsaki</i>	
Path Heuristics Using ACO for Inter-domain Routing in Mobile Ad Hoc and Sensor Networks	128
<i>Falko Dressler, Roman Koch, and Mario Gerla</i>	
Coevolution of Game Strategies, Game Structures and Network Structures	143
<i>Reiji Suzuki and Takaya Arita</i>	
Reconstructing History of Social Network Evolution Using Web Search Engines	155
<i>Jin Akaishi, Hiroki Sayama, Shelley D. Dionne, Xiujian Chen, Alka Gupta, Chanyu Hao, Andra Serban, Benjamin James Bush, Hadassah J. Head, and Francis J. Yammarino</i>	
Learning and Generalization in Random Automata Networks	163
<i>Alireza Goudarzi, Christof Teuscher, and Natali Gulbahce</i>	
Collective Evolutionary Dynamics and Spatial Reciprocity under the N-Person Snowdrift Game	178
<i>Marta D. Santos, Francisco C. Santos, and Jorge M. Pacheco</i>	
An Evolutionary Game Theoretic Framework for Adaptive, Cooperative and Stable Network Applications	189
<i>Chonho Lee, Junichi Suzuki, and Athanasios V. Vasilakos</i>	
Evolving the Asymmetry of the Prisoner's Dilemma Game in Adaptive Social Structures	205
<i>João Moreira, Jorge M. Pacheco, and Francisco C. Santos</i>	
Feature and Kernel Evolution for Recognition of Hypersensitive Sites in DNA Sequences	213
<i>Uday Kamath, Amarda Shehu, and Kenneth A. De Jong</i>	
A Phenomic Algorithm for Inference of Gene Networks Using S-Systems and Memetic Search	229
<i>Rio G.L. D'Souza, K. Chandra Sekaran, and A. Kandasamy</i>	

Fluctuation-Driven Adaptation and Symbiosis in Cellular Dynamics	238
<i>Chikara Furusawa, Kota Ijichi, and Hiroshi Shimizu</i>	
Enhancing Sampling of the Conformational Space Near the Protein Native State	249
<i>Brian Olson, Kevin Molloy, and Amarda Shehu</i>	
Self-organized Data Aggregation among Selfish Nodes in an Isolated Cluster	264
<i>K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine</i>	
An Inter-networking Mechanism Using Stepwise Synchronization for Wireless Sensor Networks	276
<i>Hiroshi Yamamoto, Naoki Wakamiya, and Masayuki Murata</i>	
An Empirical Study of Predictive Modeling Techniques of Software Quality	288
<i>Taghi M. Khoshgoftaar, Kehan Gao, and Amri Napolitano</i>	
Sensor Based Time Series Classification of Body Movement	303
<i>Swapna Philip, Yu Cao, and Ming Li</i>	
Software Service Selection by Multi-level Matching and Reinforcement Learning	310
<i>Rajeev R. Raje, Snehasis Mukhopadhyay, Sucheta Phatak, Rashmi Shastri, and Lahiru S. Gallege</i>	
Interoperating DNA Gene Sequences and Nutrition Provisions for Personalized Wellness	325
<i>Jong P. Yoon and Joyce Yoon</i>	
Proposing a Novel Artificial Neural Network Prediction Model to Improve the Precision of Software Effort Estimation	334
<i>Iman Attarzadeh and Siew Hock Ow</i>	
Tracing Conformational Changes in Proteins Represented at a Coarse Level	343
<i>Nurit Haspel</i>	
An Event Graph Model for Discovering Trends from Text Streams	357
<i>Chengli Zhao, Xue Zhang, and Dongyun Yi</i>	
Protein Structure Alignment in Subquadratic Time	363
<i>Aleksandar Poleksic</i>	
Bio-inspired Self-organized Public Key Authentication Mechanism for Mobile Ad-hoc Networks	375
<i>Parisa Memarmoshrefi, Roman Seibel, and Dieter Hogrefe</i>	

A Method for the Detection of Meaningful and Reproducible Group Signatures from Gene Expression Profiles	387
<i>Louis Licamele and Lise Getoor</i>	
Load Balancing Using Hybrid ACO – Random Walk Approach	402
<i>Neha Bhatia, Rohan Kundra, Anurag Chaurasia, and Satish Chandra</i>	
An Ant-Colony Algorithm to Transform Jobshops into Flowshops: A Case of Shortest-Common-Supersequence Stringology Problem	413
<i>Suchithra Rajendran, Chandrasekharan Rajendran, and Hans Ziegler</i>	
On the Ambiguity and Complexity Measures of Insertion-Deletion Systems	425
<i>Kamala Krithivasan, Lakshmanan Kuppusamy, Anand Mahendran, and Khalid M.</i>	
A Multiobjective Phenomic Algorithm for Inference of Gene Networks	440
<i>Rio G.L. D’Souza, K. Chandra Sekaran, and A. Kandasamy</i>	
Breaking the Box: Simulated Protein Computing	452
<i>Christopher N. Eichelberger and Mirsad Hadzikadic</i>	
Contribution of Spatio-temporal Intensity Variation to Bottom-Up Saliency	469
<i>Eleonora Vig, Michael Dorr, and Erhardt Barth</i>	
State Clustering and Declustering of 3-Regular Graphs with Structural Rewriting	475
<i>Kohji Tomita and Haruhisa Kurokawa</i>	
Evaluating Motion Estimation Models from Behavioural and Psychophysical Data	483
<i>Émilien Tlapale, Pierre Kornprobst, Jan D. Bouecke, Heiko Neumann, and Guillaume S. Masson</i>	
An Algorithm for Automatically Discovering Dynamical Rules of Adaptive Network Evolution from Empirical Data	497
<i>Hiroki Sayama</i>	
An Evaluation of the Invariance Properties of a Biologically-Inspired System for Unconstrained Face Recognition	505
<i>Nicolas Pinto and David Cox</i>	
Epidemic Spread in Adaptive Social Networks with Community Structure	519
<i>Leah B. Shaw and Ilker Tunc</i>	

A Dynamical Mechanism for the Evolution and Breakdown of Cooperation in the Snowdrift Game in Adaptive Networks	521
<i>Gerd Zschaler, Arne Traulsen, and Thilo Gross</i>	
Evolutionary Dynamics of Cooperation under the Distributed Prisoner's Dilemma	523
<i>Flávio L. Pinheiro, Francisco C. Santos, and Jorge M. Pacheco</i>	
A Pheromone Based Mobile Agent Migration Strategy for Servicing Networked Robots	533
<i>W. Wilfred Godfrey and Shivashankar B. Nair</i>	
Knowledge Sharing in Social Network Using Game Theory	542
<i>Ping Zhu, Guiyi Wei, Athanasios V. Vasilakos, and Hung-Yu Wei</i>	
Control of Snake Type Biomimetic Structure	554
<i>Mircea Ivanescu, Nicu Bizdoaca, Hani Hamdan, Mario Eltabach, and Mihaela Florescu</i>	
A Filter for the Cooperative Kinase Network of Budding Yeast <i>Saccharomyces cerevisiae</i>	563
<i>Jian-Qin Liu and Tadashi Nakano</i>	
Application of Genetic Algorithm to Maximise Clean Energy Usage for Data Centres	565
<i>Raymond Carroll, Sasitharan Balasubramaniam, Dmitri Botvich, and William Donnelly</i>	
Safer Driving with Gaze Guidance	581
<i>Laura Pomarjanschi, Michael Dorr, Christoph Rasche, and Erhardt Barth</i>	
Infrastructure Optimization of Flight-Formation Inspired Self-organization for Address Configuration in Sensor Networks	587
<i>Rui Teng, Bing Zhang, and Jian-Qin Liu</i>	
Colour Saliency on Video	601
<i>Michael Dorr, Eleonora Vig, and Erhardt Barth</i>	
Bio-inspired Transputer Based-Fuzzy Mobile Robot	607
<i>Ebrahim Mattar, Khalid Al-Mutib, Hani Hamdan, and Mohamad Hamdan</i>	
Bio-inspired Visual Information Processing – The Neuromorphic Approach	621
<i>Woo Joon Han and Il Song Han</i>	
Firing Pattern of Default Mode Brain Network with Spiking Neuron Model	629
<i>Teruya Yamanishi, Jian-Qin Liu, and Haruhiko Nishimura</i>	

Program Equivalence Using Neural Networks	637
<i>Tiago M. Nascimento, Charles B. Prado, Davidson R. Boccardo, Luiz F.R.C. Carmo, and Raphael C.S. Machado</i>	
Representation of Spiking Neural P Systems with Anti-spikes through Petri Nets	651
<i>Venkata Padmavati Metta, Kamala Krithivasan, and Deepak Garg</i>	
Learning in a Distributed Software Architecture for Large-Scale Neural Modeling	659
<i>Jasmin Léveillé, Heather Ames, Benjamin Chandler, Anatoli Gorchetchnikov, Ennio Mingolla, Sean Patrick, and Massimiliano Versace</i>	
 Demo/Poster Papers	
Bio-inspired Robotics Hands: A Work in Progress	667
<i>Ebrahim Mattar and Khaled Al Mutib</i>	
Bio-inspired Routing and Wavelength Assignment Algorithm for Optical Mesh Networks	676
<i>Daniel Padilla, Edward Guillen, and Carlos Ramos</i>	
Artificial Immune Systems – AIS as Security Network Solution	680
<i>Edward Guillen and Rafael Paez</i>	
WebSeA: A Secure Framework for Multi-site Knowledge Representation in Software Engineering	682
<i>Muhammad Ilyas, Ahmad Ali, and Josef Kueng</i>	
SRA: A Salmon-Like Approach to MANET Routing	687
<i>Filomena de Santis and Daniele Mastrangelo</i>	
 Author Index	 693

Self Tolerance by Tuning T-Cell Activation: An Artificial Immune System for Anomaly Detection

Mário J. Antunes^{1,3} and Manuel E. Correia^{2,3}

¹ School of Technology and Management, Polytechnic Institute of Leiria,
Morro do Lena, Alto do Vieiro, 2411-901 Leiria, Portugal

`mario.antunes@ipleiria.pt`

² Department of Computer Science, Faculty of Science, University of Porto,
Rua do Campo Alegre s/n, 4169-007 Porto, Portugal

`mcc@dcc.fc.up.pt`

³ Center for Research in Advanced Computing Systems (CRACS), Portugal

Abstract. The Artificial Immune Systems (AIS) constitute an emerging and very promising area of research that historically have been falling within two main theoretical immunological schools of thought: those based on *Negative selection* (NS) or those inspired on *Danger theory* (DT). Despite their inherent strengths and well known promising results, both deployed AIS have documented difficulties on dealing with gradual dynamic changes of self behavior through time.

In this paper we propose and describe the development of an AIS framework for anomaly detection based on a rather different immunological theory, which is the Grossman's Tunable Activation Thresholds (TAT) theory for the behaviour of T-cells. The overall framework has been tested with artificially generated stochastic data sets based on a real world phenomena and the results thus obtained have been compared with a non-evolutionary Support Vector Machine (SVM) classifier, thus demonstrating TAT's performance and competitiveness for anomaly detection.

Keywords: Artificial Immune Systems, Tunable Activation Thresholds, Pattern Recognition, Signal Processing, Support Vector Machine.

1 Introduction

The Vertebrate Immune System (IS) is a complex biological system, conceptually structured into two main functional layers: *innate* and *adaptive*. The anomaly detection embodied by the IS has to cope with a highly dynamic environment where the body is constantly being exposed to external agents (*pathogens*). Thus, this interaction with the environment results in a distinction between what is benign or belong to the organism's own healthy cells and tissues (*self*) from what is harmful and may provoke harm or even motivate a disease (*non-self*). In the majority of cases, pathogens presented to the body correspond to unseen flavours of normal activity that do not represent any serious danger or are benign [\[1\]](#).

In the late 19th century, Ehrlich's started by postulating that the IS "*classifies*" pathogens as normal body antigenic components (*self antigens*) or as foreign abnormal chemical structures present in microorganisms (*non-self antigens*). After all these years, immunology continues to be a very vibrant and active open research area where no one has a definitive answer on how the IS is able to accomplish its goals in such an efficient and effective way.

One of the mainstream theories for several years, Negative Selection (NS) [2], assumes that cell activation thresholds have evolved to optimal values and constitute an intrinsic feature of each species. In the early 90's Matzinger described her controversial "Danger Theory" immunological theory, which states that the immune system is activated upon the receipt of molecular signals (danger signals) which indicate damage or stress to the host, rather than by a self-non-self distinction as previously postulated by NS [3,4].

The IS provides a very appealing metaphor for the development of innovative anomaly detection systems in the form of an Artificial Immune Systems (AIS) [5]. The research in this area is based in principles, mechanisms, models and observed functions of the IS behavior, together with engineering best practices and methodologies. The most relevant AIS developed so far for anomaly detection have been based on the NS approach [6] and on the Danger Theory (DT) [4,7]. In spite of the results achieved, the deployment of AIS based on both theories to solve real world problems [6,8] did not yet met the expectations raised by such appealing metaphors [9,7]. Firstly, the NS approaches proved to be inappropriate for large data sets and have shown to have scaling problems [9,10]. Secondly, DT approaches have not intrinsic self-tuning mechanisms and require a great deal of expert knowledge beforehand [7,8].

The well known best of breed AIS are all based on well reasoned immunological metaphors. However, the lack of research into their real biological foundations has led to well known criticism for biologically-inspired engineering approaches [11,12]. As an example, consider an *ideal* anomaly detector. It should be ready to act on a continuous changing environment and it should adapt itself throughout time to tolerate *unseen* and *untrained* forms of normal behavior, thus discriminating ongoing not yet seen anomalies. Such immune-inspired properties that mimic this self-non-self discrimination behavior are essential for the deployment of bio-inspired anomaly detectors within dynamic environments.

In this paper we present a generic AIS framework for anomaly detection, based on a simplified Tunable Activation Threshold (TAT) model, strongly inspired on Grossman's hypothesis [13]. TAT assumes that immune cells (like T-cells) tune their activation thresholds by dynamically updating the levels of two particular enzymes (Kynase and Phosphatase), whose values reflect the recent temporal history of signaling they have been receiving from the environment.

We start by defining a TAT model for T-cells (Section 2) and proceed into Section 3 by presenting the framework and its main building blocks. This is followed by showing and discussing the results obtained with stochastically generated, but real world based, data sets. A comparison is made with the non-evolutionary

classifier Support Vector Machine (SVM) (Sections 4 and 5). Finally, in Section 6 we delineate some conclusions for our work.

2 The TAT Model Adopted

The Grossman's TAT conceptual framework hypothesizes that immune cell activation depends on a dynamically adjusted threshold, which corresponds to the balance between excitation and de-excitation signalling pathways [14]. The activation process is controlled by the activity of two specific enzymes that respond to antigenic signals (S): Kinase (K) phosphorylates molecules that "excite" the cell and Phosphatase (P) that dephosphorylates them, returning the cell to a de-excitation state. The signals are delivered by a particular immune cell named Antigen Presenting Cell (APC).

It is also assumed that T-cell activation is a switch-type response that requires that K supersedes P , at least transiently. At each point in time, T-lymphocytes (T-cells) interact with the peptides presented by APC and receive a stimulus that depends on the *affinity* between its receptor and the peptide ligand, causing the cell to adapt by increasing or decreasing its activation threshold. Also, the de-excitation level is assumed to be intrinsically slow, thus allowing the outcome of a stimulus to depend mainly on the excitation index. Thus, foreign antigens will cause a very fast increase in the cells excitation level, whereas tissue-specific self-ligands will induce a much slower increase excitation level. Accordingly, since the de-excitation levels are kept above the excitation ones, it is possible to maintain tolerance to self for extended periods of time [15]. Within this model, different cells with different antigen-specificity end up having different activation thresholds as they are exposed to different stimuli.

2.1 TAT Dynamics

We have adopted a minimal mathematical model of TAT for T-cells [16], which is also derived from Grossman's hypothesis. Keeping the original Grossman's fundamental thoughts about self regulation and cellular activation, we made the following simplifications:

- both K and P are exposed to the same stimulus S ;
- P 's basal value (P_0) is higher than K 's (K_0);
- S_0 is the initial value for S ;
- K 's turnover rate (τK) is lower than P 's (τP);
- K 's slope (ϕK) is higher than ϕP 's;
- the IS's speed of response is given by a constant value (t);

We also derived the following values:

- $K_0 = S_0 \cdot \tau K$ and $P_0 = S_0 \cdot \tau P$;
- $\tau K = \tau \cdot \tau P$, with $\tau = \frac{\tau K}{\tau P}$;
- $\phi P = \phi \cdot \phi K$, with $\phi = \frac{\phi P}{\phi K}$;

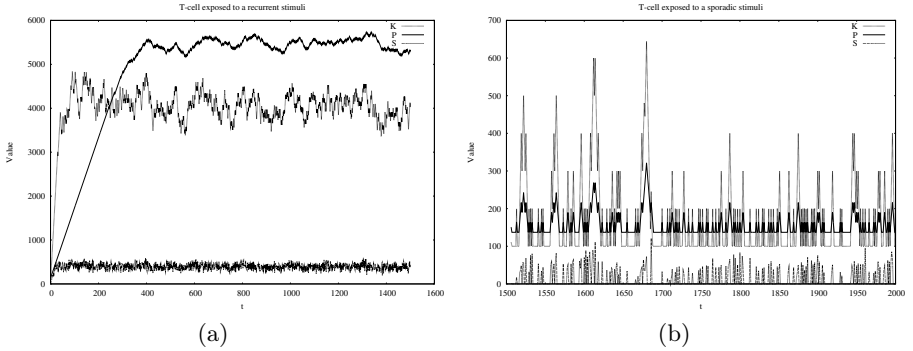


Fig. 1. TAT dynamics of two individual T-cells in the AIS repertoire. **1(a)**: the cell is exposed to a recurrent stimuli, like tissue-specific self ligands **1(b)**: the cell receives an intermittent and strong signal which leads to repeated events of activation.

Figure **1** depicts the TAT activity of two T-cells. S corresponds to a linear increase of both K and P activities until the turnover rate is reached. T-cell activation only occurs if K is higher than P . For a recurrent signal S , K will be transiently higher than P and, if the signal persists P will exceed K and the cell will become inactive. Similarly, on signaling absence, K returns to the initial level at a faster rate than P .

2.2 Signaling Model

In such a model, at each given moment in time, the stimulation history of a T-cell is reflected in the activity of K and P . The signal S sent by the APC to a T-cell is a function of the affinity between the corresponding T-cell Receptor (TCR) and the ligand times the concentration of that peptide in the APC. To give strength to the temporal meaning of the TAT dynamics, S is calculated in a *per APC in the lifespan (LS)* basis, as shown in Algorithm **1**.

The values of K and P of each cell repertoire are updated linearly based on the combined signal of all the ligands presented by each APC, as described in Algorithm **2**. The use of a linear update of such values gave simplicity to the model, being at the same time in accordance with the Grossman's derived model depicted in **16**. In short, in the presence of a signal ($S > 0$), K and P increase till reach its corresponding maximum values (τK and τP). Otherwise, they decrease gradually till the basal values ($K_0 = S_0 \cdot \tau K$ and $P_0 = S_0 \cdot \tau P$). The growth and decline rates are different for both K and P , which leads to episodes of cell activation when K becomes higher than P .

Grossman also postulated that an immunological response to an APC is usually not initiated by an individual activated cell, being instead initiated by a *population clones* of activated cells. Thus, on an APC processing, an immune response is initiated if the ratio of the population size of activated cells and

Algorithm 1. Signalling model adopted

Input: L_{pep} = List of peptides presented by an APC
Input: L_{tcell} = T-cell Repertoire
Output: S = signal sent for each t-cell in the repertoire

```

1 forall the  $tcell$  in  $L_{tcell}$  do
2   forall the peptide in APC's Lifespan (including those in  $L_{pep}$ ) do
3      $a = distance(peptide, tcell)$ 
4      $c = occurrences\ of\ peptide\ in\ the\ APC\ lifespan$ 
5      $S+ = \Sigma(c \cdot a)$ 
6   end
7 end

```

Algorithm 2. Update of K and P for a T-cell, based on a received signal

Input: S = Stimuli received by a T-cell (calculated in Algorithm 1)
Input: t = Real value corresponding to the *speed* of response of the system
Output: Updated values for K and P

```

1 if  $(S + S_0) \cdot \tau K > K$  then
2    $K \leftarrow MIN((S + S_0) \cdot \tau K, K+ = \phi K \cdot t)$ 
3 end
4 else
5    $K \leftarrow MAX((S + S_0) \cdot \tau K, K- = \phi K \cdot t)$ 
6 end
7 if  $(S + S_0) \cdot \tau P > P$  then
8    $P \leftarrow MIN((S + S_0) \cdot \tau P, P+ = \phi P \cdot t)$ 
9 end
10 else
11    $P \leftarrow MAX((S + S_0) \cdot \tau P, P- = \phi P \cdot t)$ 
12 end

```

those that were bound exceeds a threshold. This thus implies that an immune response depends always on the decision of a group of cells (*committee*), instead of an individual one [16].

3 The TAT Based Framework

A generic AIS framework can be divided into three main functional layers [5]: a data representation (Section 3.2), an affinity measure distance between the immune cells receptors and the peptide ligand (Section 3.2) and, finally, an immune-inspired algorithm that maps the system components with the relevant biological IS counterparts (Section 3.4). In what follows we describe in some detail the core building blocks of the TAT-AIS framework, depicted in Figure 2.

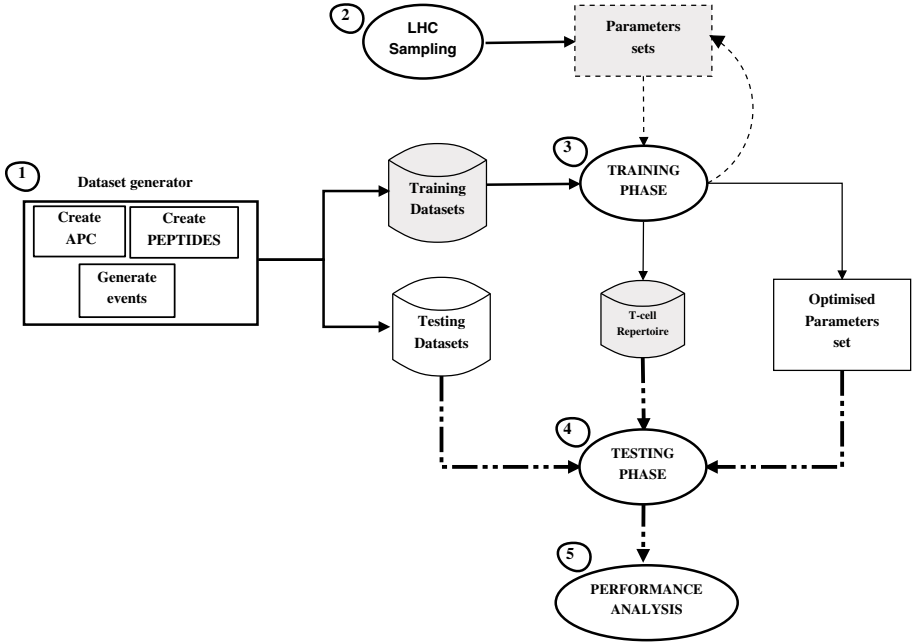


Fig. 2. Building blocks of the TAT-based AIS

Our system is composed by the following three main components: an artificial data set generator, a parameter set sensitivity analysis module and a TAT simulator that implements the TAT model previously described, for processing both training and testing data sets.

The adopted methodology is described as follows. Firstly, using the data set generator we produce the training and testing data sets according to the procedure described in Section 3.2. Then we obtain a set of parameters using an Latin Hypercube (LHC) based sampling method [17] (Section 3.3). Each one of these parameter set is then evaluated against the training data sets in order to obtain an *optimised* parameters set and a list of *trained* detectors (T-cells) that are finally confronted with a group of testing data sets, in order to analyse their behaviour and its performance compared with other competing classifiers.

3.1 The TAT Operation

The TAT simulator requires training, which is comprised by two different steps. Firstly, we have split the training data set into two sub-data sets: one that is used for training TAT with normal examples and the other containing a mixture of randomly interleaved known normal and abnormal examples. We then run the TAT simulator with the normal examples to obtain a list of *self* T-cells that converge to a “*P higher than K*” state. Finally we load the obtained T-cell

repertoire and process the second sub-set. During this phase TAT creates the T-cells responsible for the detection of abnormal patterns. The resulting repertoire is then loaded into the system for further processing the testing data set .

3.2 Data Representation

The most relevant immunological players involved in the TAT model and suitable to be represented in the artificial framework are peptide ligand, TCR and APC. Generally speaking, from a machine learning point of view, both peptide ligand and TCR are *patterns* and can be represented by strings that bound to each other by some affinity measure. APC can thus be seen as a list of characters (peptides) representing a behavior (normal or abnormal). Also, APC are ordered sequentially and tagged into two distinct classes (“Normal” or “Alert”) for evaluation purposes.

The reason behind the development and deployment of an artificial data set generator was two-fold. Firstly, we intended to generate simple and easy to understand artificial data sets with which we could test the model and compare the results with other classifiers. Secondly, although being simple, we would like the data sets to be based, as much as possible, on real world phenomena for anomaly detection purposes. We have thus decided to use the spam Enron data sets [18] and identify some of its main characteristics and use them for artificial data set generation. The Enron preprocessed data sets have six personal mailboxes made public after the Enron scandal. The ham mailboxes belong to six employees and combinations of five spam data sets were added to the ham data, coming from different sources [18, 19]. Table 1 describes the distribution of ham and spam words (average values), considering all the spam and ham messages available for each mailbox and an identical size for training and testing data sets.

Table 1. Analysis of Enron spam data sets

Data set	Words	Occurrences	Messages	Words/Msg.	Weight
Ham Training	954	74497	920	78.09	1.81
Ham Testing	140	6906	1838	49.22	1.14
Spam Training	164	7098	954	43.24	1
Spam Testing	117	6114	1907	53.34	1.21

From the Table 1 it is possible to identify the average amount of words in each example of the listed data sets (column *Words/Msg.*), as well as the proportional relation of each data set with the data set that has the lowest ratio of words per message (column *Weight*). For instance, the ham training data set has 954 different words that appears 74497 times in the 920 messages. Thus, the average of words per message in the ham training data set is about 78. Comparing to the data set with the lowest ratio (spam training), the proportional relation is 1.81. To generate the artificial data sets we defined a symbol set of length 8

for the spam training data set (the one with the lowest ratio) and built the three remaining data sets according to these proportional weights. Thus, the symbols sets lengths are 14 (corresponding approximately to 8 times 1.81), 9 (8 times 1.14) and 10 (8 times 1.21), respectively for ham training, ham testing and spam testing. The Table 2 depicts the symbols sets used to generate the artificial data sets, based on the analysis described previously.

Table 2. Alphabet used to create the peptides

Symbols	Data sets	Tag
a b c d e f g h i j k l m n	Training, Testing	Normal
A B C D E F G H I	Testing	Normal
1 2 3 4 5 6 7 8	Training, Testing	Alert
+ - . _ : ; ? ! = \$	Testing	Alert

In the data sets described above, both peptide ligand and TCR are represented by one character and an APC corresponds to a list of characters. The affinity metric adopted to measure the distance between both peptide ligand and TCR is the character match (one if equal and zero otherwise). This strategy seemed appropriate to calculate the affinity between strings of just one character, which in some way represents *words* of the email messages. Also, an APC tagged as normal has all the peptides produced with symbols picked randomly from the “Normal” alphabets symbols. On the other hand, an APC is labelled as alert if it has at least one peptide belonging to an “Alert” alphabet symbol.

The data sets generation took into count two crucial parameters: the number of peptide ligand (characters) per APC, which varies along the data set and the number and time of occurrence of events for each class (“Normal” and “Alert”). The values for these two parameters were also randomly generated for each data set. The following main features could then be observed in the artificially generated data sets:

- patterns representative of normal behavior appear recurrently, while abnormal ones appear sporadically;
- in the testing phase new unseen patterns representative of normal behavior start appearing recurrently;
- also in the testing phase, new unseen patterns representative of abnormal behavior appears sporadically;
- finally, new unseen anomalies are propagated by the conjunction of both known and unknown patterns, representative of abnormal behavior.

An example of normal and alert APC looks like the following:

apc:38:NORMAL: o d n j i m d h p k c k f l b i c l n k i a a k

apc:39:ALERT: m c 3 3 6 i g n n 6 2 h 6 b e g o j 2 8 f 1 i d 7 p h 2
7 b m a

For the experiments detailed in Section 4 we generated APC with a maximum of 200 peptides. The training data set is composed by 2000 APC sequentially ordered, being the first 75% representative of normal behavior and the remaining related to *known* abnormal examples. The testing data set is twice the size of the training (4000 APC).

3.3 Parameters set Optimisation

The T-cells' TAT dynamics is based on the parameters described on Table 3.

Table 3. Parameters set

Parameter	Range	Description	Type
τ	[0; 1]	K and P turnover rate, $\frac{\tau P}{\tau K}$	Optimised
ϕ	[0; 1]	K and P slopes rate, $\frac{\phi K}{\phi P}$	Optimised
t	[10; 100]	Speed of adaptation and response for the IS	Optimised
LS	[10; 100]	Number of APC in the lifespan	Optimised
CS	[10; 100]	T-cell maximum clonal size	Optimised
Ct	[0.00001, 0.8]	Committee threshold.	Run time
i		update factor to increase/decrease the T-cell CS	Fixed=2
S_0		Initial signal	Fixed=2
CS_0		Initial clonal size	Fixed=2
a		Affinity distance between T-cells and peptides	Fixed=1

We used a hybrid approach to choose a sufficiently good performant parameter set. Firstly, we have listed all the parameters related to the TAT dynamics (Section 3.5) and its corresponding ranges. Then, we generated an LHC sampling for the following parameters: τ , ϕ , t , LS and CS . LHC sampling is a statistical method developed to generate a distribution of collections of parameter values from a multidimensional distribution [17]. For our case we sampled the five parameters into 40 equally probable intervals.

Our model is also evolutionary. So, we defined that Ct will be updated in the testing phase by using a feedback mechanism described in Section 3.6. Ct starts with a fixed value ($Ct = 0.1$) and, as long as the system is having a too high or too low rate of alerts, this parameter increases or decreases accordingly in a gradual way.

Finally, using the combinations calculated by the LHC sampling method and the defined fixed parameters, we run a 10-fold training data set and then calculate the average of the *F1-Measure* thus obtained. This measure is a score of accuracy that considers both the precision p and the recall r , being calculated by $F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ [20].

At the end of the 10-fold training process we obtain a good LHC’s candidate multidimensional “square”, that corresponds to the parameters set with which we obtained the best performance thus far. Depending on the accuracy level, the sampling process can be recursively repeated with an even more strict hypercube, to try to obtain an even more refined parameter set of values that can produce better results.

3.4 General Algorithm

The algorithm [3](#) describes the TAT processing of each APC. It receives a list of peptides (L_{pep}) and the up to date list of T-cells (L_{tcell}) with its corresponding K and P values.

At each given moment, each T-cell will be stimulated with a signal that corresponds to the sum of signals sent by each affinity-specific peptide. If there is not a T-cell that bounds with a peptide, then a new one is created with the TCR being the string representative of the peptide ligand. After processing all the peptides of the APC, the system calculates the amount of clonal size for both bound and active T-cell. If the ratio between both values is above a threshold (Ct), then the system raise an alarm and increases the clonal size of all the activated cells. Otherwise, the clonal size is decreased (Section [3.5](#)).

3.5 Clonal Size Update Procedure

In our model the clonal size (CS) of each cell corresponds to an integer that varies between an initial value (CS_0) and a maximum (CS_{max}). The clonal size update procedure in each processing phase is depicted in Table [4](#). We also introduced the meaning of *committee* as being the clones population of cells that bound with a certain specificity the peptides ligand presented by an APC.

In each processing phase, the decision rule is thus based on the ratio between the cells of the committee that are activated and those that simply bond with the peptide ligand but remains quiescent. In the training phase, the learning procedure is supervised and the clonal size of activated T-cells ($K > P$) increases if the *committee* decided in favor to trigger a response. Otherwise, the CS decreases. The testing phase is unsupervised and the clonal size update is based on the APC classification made by the system in each moment. In general, this CS update mechanism allow that sporadically activated T-cells converge to a maximum value (CS_{max}) and the recurrently stimulated (but not activated) ones may have a clonal size near CS_0 . For the sake of simplicity the following assumptions was made: the increment factor is fixed, $i = 2$ and CS update is made linearly ($CS \pm = i$)

3.6 Feedback Control Mechanism

The evolutionary tuning of committee threshold is as follow: Ct starts with a predefined value (Table [3](#)). Then, in each APC processing, if the number of

Algorithm 3. General TAT algorithm

```

Input:  $L_{tcell}$  = T-cell Repertoire
Input:  $L_{pep}$  = List of peptides presented by an APC
Input:  $At$  = Affinity Threshold
Input:  $Ct$  = Committee Threshold
Output: Classification of the artificial APC: Normal or Alert

1  $T_{bind} = ()$ 
2  $T_{active} = ()$ 
3  $CS_{bind} = 0$ 
4  $CS_{active} = 0$ 

5 forall the  $tcell$  in  $L_{tcell}$  do
6    $S = 0$ 
7   forall the  $UNIQUE(peptide)$  in  $L_{pep}$  do
8      $a = distance(peptide, tcell)$ 
9      $c = occurrence\ of\ peptide\ ligand\ in\ the\ APC\ lifespan$ 
10    if  $a \geq At$  then
11       $S += \Sigma(c \cdot a)$ 
12       $ADD(T_{bind}, tcell)$ 
13    end
14  end
15   $UpdateTCell(t, S)$  (according to Algorithm 2)
16 end

17 forall the  $tcell$  in  $T_{bind}$  do
18   if  $K \geq P$  then
19      $ADD(T_{active}, tcell)$ 
20   end
21 end

22  $CS_{active} = \Sigma ClonalSize(T_{active})$ 
23  $CS_{bind} = \Sigma ClonalSize(T_{bind})$ 
24  $Status = Normal$ 
25 if  $CS_{active} / (CS_{bind} + CS_{active}) \geq Ct$  then
26    $Status = Anomaly$ 
27 end
28  $ReportStatus()$ 
29  $UpdateClonalSize(Status)$  according to the procedure described on Section 3.5

```

triggers observed so far exceeds a preliminary threshold, then the value for Ct is incremented. Otherwise, if there is no trigger during a long period, the Ct decreases. The number of *acceptable* triggers and the values used to increase or decrease the parameter Ct should be domain-dependent. In the experiments we defined 10% as being an acceptable value for the alerts. We imposed that Ct should vary between 0.00001 and 0.8 and its value depends on the observations made to the system in run time.

Table 4. Clonal size update process

	Normal		Known Alerts		Testing	
	K>P	P>K	K>P	P>K	K>P	P>K
Supervised						
Tag="NORMAL"	↗	↘	↘	↘		
Tag="ALERT"			↗	↘		
Unsupervised						
Trigger an alert					↗	↘
Silent mode					↘	↘

4 Experimental Evaluation and Results

Our working hypothesis is that the deployed TAT based model can be able to recognise new unseen patterns and also to further distinguish between those that are considered self from others included in APC related to abnormal activities and should be identified as non-self. In order to validate our model we compared the results obtained with a non-evolutionary SVM classifier.

For both TAT and SVM algorithms we used the following methodology. Firstly, we trained the system with a 10-fold training data set. Then we manage to process the testing phase with ten different data-sets. Finally we evaluate the performance obtained by each detection algorithm.

4.1 TAT Parameters

We run the parameters optimisation methodology described in Section 3.3 and obtained the parameters set listed on Table 5. The table shows the performance obtained during the training phase (average of the 10-fold training processing) considering the affinity as being the full match of each TCR with each peptide ligand presented in the APC.

Table 5. The parameters set with the best performance during the training phase

τ	ϕ	T	LS	CS	Ct	Accuracy	Precision	Recall	F1
0.78758	0.25646	25	99	24	0.1	0.98	1.00	0.92	0.96

4.2 Results

Table 6 depicts the results obtained with both TAT and SVM implementations. The table clearly shows the performance obtained for accuracy, precision, recall and F1 score for both TAT and SVM processing.

Table 6. Results obtained with TAT and SVM processing

<i>Dataset</i>	TAT			SVM		
	<i>Precision</i>	<i>Recall</i>	<i>F1</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
1	0.98	0.75	0.85	1.0	0.72	0.84
2	0.98	0.67	0.8	1.0	0.73	0.84
3	0.99	0.75	0.85	1.0	0.75	0.85
4	0.98	0.76	0.86	1.0	0.71	0.83
5	0.99	0.78	0.87	1.0	0.73	0.85
6	0.97	0.76	0.86	1.0	0.74	0.85
7	0.96	0.85	0.9	1.0	0.72	0.84
8	1.00	0.70	0.82	1.0	0.71	0.83
9	0.98	0.72	0.83	1.0	0.67	0.80
10	0.97	0.77	0.86	1.0	0.76	0.86
Mean	0.98	0.75	0.850	1.0	0.72	0.839

5 Discussion

We have observed that TAT-AIS has interesting properties for anomaly detection, provided the following basic generic requirements are true: *normal behavior is frequent and abnormal behavior is sporadic in time*. By *frequent* we mean a pattern that repeatedly stimulates a set of T-Cells that through time, by the TAT dynamics, stabilizes its enzymatic values ($P > K$). On the other hand, by *sporadic* we mean a pattern that stimulates intermittently a set of T-Cells with such a signal that implies its activation ($K > P$).

Our aim was to validate the appropriateness of using TAT to detect new previously unseen patterns and also to distinguish them between those that correspond to unseen “normal” and “abnormal” behaviors. We have also compared the results obtained with a non-evolutionary SVM classifier. The results show the competitiveness of TAT when comparing it with SVM. The F1-measure varies between 80% and 90% in TAT. In SVM the results are in the range of 80% and 86%. In general SVM has no false positives (Precision=100%) but not all the new alerts are correctly identified. On the other hand, TAT has some false positives (mean of Precision is 98%) but the recall is slightly higher than the SVM.

The performance advantage obtained with TAT-AIS is very tiny, when compared with the non-evolutionary SVM classifier. However, based on these empirical results, we believe that our model can compete with other approaches on the self-non-self distinction for dynamic environments that tend to change gradually throughout time their normality behavior profile. In this paper we were not aware with the performance of the speed of classification. However, in these experiments we observed similar execution times with both models.

6 Conclusions

In this paper we have presented a generic TAT-based AIS framework for anomaly detection and described its main architectural components. We have also presented some results obtained with artificially generated data sets of predefined patterns resulting from normal and abnormal behaviors. Also, we have compared the TAT performance with a non-evolutionary SVM classifier.

The results thus obtained with the AIS are very satisfactory, achieving a high rate of detection and a low level of false positives on the stochastic data sets we have produced.

We are well aware that these stochastic data sets were artificially generated and are most certainly not completely representative of real world phenomena like the data sets we could obtain with email spam collections. We have however already obtained some preliminary good results with this TAT-based framework, applied both to more complex stochastically generated data sets [21], as well as to network intrusions detection with real network traffic [22]. The research done so far give us confidence on the use of TAT based AIS framework to implement behavior based anomaly detection systems, where the temporal meaning of events is relevant, like spam and intrusion detection. The ongoing research is now on using the model and the framework presented to process the original Enron spam data sets and to compare its performance with the SVM classifier.

Acknowledgments. The authors acknowledge the facilities and research environment gracefully provided by the Center for Research in Advanced Computing Systems research unit. The authors also thank Catarina Silva (Polytechnic Institute of Leiria and CISUC - University of Coimbra) by her insights and help on the SVM data processing.

References

1. Murphy, K., Murphy, K., Travers, P., Walport, M., Janeway, C.: Janeway's immunobiology. Garland Pub. (2008)
2. Burnet, F.: The Clonal Selection Theory of Acquired Immunity. University Press Nashville, Tenn (1959)
3. Matzinger, P.: Tolerance, danger, and the extended family. *Annual Review of Immunology* 12(1), 991–1045 (1994)
4. Aickelin, U., Greensmith, J.: Sensing danger: Innate immunology for intrusion detection. *Information Security Technical Report* 12(4), 218–227 (2007)
5. de Castro, L., Timmis, J.: *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer (2002)
6. Hofmeyr, S., Forrest, S.: Architecture for an artificial immune system. *Evolutionary Computation* 8(4), 443–473 (2000)
7. Kim, J., Bentley, P., Aickelin, U., Greensmith, J., Tedesco, G., Twycross, J.: Immune system approaches to intrusion detection - a review. *Natural Computing* 6(4), 413–466 (2007)

8. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 153–167. Springer, Heidelberg (2005)
9. Stibor, T., Mohr, P., Timmis, J., Eckert, C.: Is negative selection appropriate for anomaly detection? In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 321–328 (2005)
10. Kim, J., Bentley, P.: An evaluation of negative selection in an artificial immune system for network intrusion detection. In: Genetic and Evolutionary Computation Conference, pp. 1330–1337 (2001)
11. Andrews, P., Timmis, J.: Tunable Detectors for Artificial Immune Systems: From Model to Algorithm. In: Bioinformatics for Immunomics, pp. 103–127 (2010)
12. Stepney, S., Smith, R., Timmis, J., Tyrrell, A., Neal, M., Hone, A.: Conceptual frameworks for artificial immune systems. *International Journal of Unconventional Computing* 1(3), 315–338 (2005)
13. Grossman, Z.: Cellular tolerance as a dynamic state of the adaptable lymphocyte. *Immunology Reviews* 133, 45–73 (1993)
14. Grossman, Z., Paul, W.: Self-tolerance: context dependent tuning of T cell antigen recognition. *Seminars in Immunology* 12(3), 197–203 (2000)
15. Scherer, A., Noest, A., de Boer, R.: Activation-threshold tuning in an affinity model for the T-cell repertoire. In: Proceedings: Biological Sciences, vol. 271(1539), pp. 609–616 (2004)
16. Carneiro, J., Paixão, T., Milutinovic, D., Sousa, J., Leon, K., Gardner, R., Faro, J.: Immunological self-tolerance: Lessons from mathematical modeling. *Journal of Computational and Applied Mathematics* 184(1), 77–100 (2005)
17. Helton, J., Davis, F.: Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety* 81(1), 23–69 (2003)
18. Metsis, V., Androutopoulos, I., Paliouras, G.: Spam filtering with naive bayes-which naive bayes. In: Third Conference on Email and Anti-Spam (CEAS), pp. 125–134 (2006)
19. Abi-Haidar, A., Rocha, L.: Adaptive Spam Detection Inspired by the Immune System. In: Artificial Life XI - 11th Int. Conference on the Simulation and Synthesis of Living Systems, vol. 11, pp. 1–8 (2008)
20. Silva, C., Ribeiro, B.: Inductive Inference for Large Scale Text Classification: Kernel Approaches and Techniques. Springer (2009)
21. Antunes, M., Correia, M.: Temporal Anomaly Detection: an Artificial Immune Approach Based on T-cell Activation, Clonal Size Regulation and Homeostasis. *Advances in Computational Biology - Book series* vol. 680, pp. 291–298 (2010)
22. Antunes, M., Correia, M.: TAT-NIDS: an immune-based anomaly detection architecture for network intrusion detection. In: Corchado, J.M., et al. (eds.) IWPACBB 2008. ASC, vol. 49, pp. 60–67. Springer, Heidelberg (2008)

Information Rates of Active Propagation in Microchannel Molecular Communication

Nariman Farsad¹, Andrew W. Eckford¹, Satoshi Hiyama², and Yuki Moritani²

¹ Dept. of Computer Science and Engineering, York University
4700 Keele Street, Toronto, ON, Canada M3J 1P3
nariman@cse.yorku.ca, aeckford@yorku.ca

² Research Laboratories, NTT DOCOMO Inc., Yokosuka, Kanagawa, Japan
{hiyama,moritani}@nttdocomo.co.jp

Abstract. Molecular communication is a promising technique for microchannel systems. In this paper, various microchannel molecular communication schemes are simulated and analyzed using information theory, including molecular motors and Brownian motion with drift. Results suggest Brownian motion with drift can deliver excellent performance, depending on the drift velocity.

Keywords: Molecular communication, microchannels, information theory.

1 Introduction

Molecular communication [1] is a new field of communication where instead of electric currents or electromagnetic waves, patterns of molecules are used to transfer information from a source (transmitter) to a destination (receiver). In this paper, we consider propagation media that are confined fluid environments with small dimensions in the order of micrometers, known as a *microchannel*. Molecular communication is a promising technology for passing control and information messages in microchannel and lab-on-chip systems [2].

In this paper we focus on the communication-theoretic and information-theoretic aspects of molecular communication. Notable works in this direction include a general formulation of molecular communication as a timing channel under Brownian motion [3,4], an analysis of information transfer rates using molecular motors [5,6], mathematical channel models for continuous diffusion [7], and a simple model comparing the achievable information rates of passive transport using Brownian motion to that of active transport using microtubule filaments moving over a molecular motor track [8].

Our main contribution in this paper is to compare the mutual information of molecular communication systems in microchannels. Mutual information is a key parameter of a communication system, as it is the highest rate at which information can be reliably transmitted [9]. This paper extends our previous work in [8], with the following two novel aspects. Firstly, and more importantly,

we focus primarily on *active* forms of propagation in this paper: molecular motors (which convert chemical energy directly into kinetic energy), and Brownian motion with positive drift (which requires an external driving force to maintain the average motion of the fluid medium). Drift was not considered in our previous work. Secondly, our analysis in this paper uses much more sophisticated propagation models than that in our previous work. In particular, for Brownian motion, this paper simulates three-dimensional motion (versus two-dimensional motion in [8]), and for molecular motors, we use a more realistic model for vesicle loading of microtubules. Thus, our results move closer to a complete quantitative analysis of the microchannel molecular communication system.

2 Simulation Environment and Models

Our simulation environment is similar to the one in [8,10]. We use a rectangular propagation environment (with rounded corners), consisting of a strip *loading zone* and an *unloading zone*. Regardless of the propagation model, message-bearing vesicles originate at the loading zone, and propagate until they arrive at the unloading zone. If passive transport is used, information carrying vesicles follow a Brownian motion path from the loading zone to the unloading zone. If active transport is employed, microtubule filaments moving over molecular motor tracks that cover the whole environment, pick up and transport the information carrying vesicles from the loading zone to the unloading zone. The reader is referred to [10] for detailed explanation of this process.

For Brownian motion, we perform a three-dimensional discrete-time simulation of information carrying vesicles, based on a simulation scheme from [11]. Given some initial position (x_0, y_0, z_0) at time $t = 0$, for any integer $k > 0$, the motion of the vesicle is given by the sequence of coordinates (x_i, y_i, z_i) for $i = 1, 2, \dots, k$. Each coordinate (x_i, y_i, z_i) represents the position of the particle at the end of time $t = i\Delta t$, where

$$x_i = x_{i-1} + v_{f_x} \Delta t + \Delta r \cos \theta_i \cos \phi_i, \quad (1)$$

$$y_i = y_{i-1} + v_{f_y} \Delta t + \Delta r \sin \theta_i \cos \phi_i, \quad (2)$$

$$z_i = z_{i-1} + v_{f_z} \Delta t + \Delta r \sin \phi_i, \quad (3)$$

where v_{f_x} , v_{f_y} , and v_{f_z} are flow velocity components in the x , y , and z direction. Further, over each time interval of Δt , the molecule's displacement Δr is given by

$$\Delta r = \sqrt{4D\Delta t}, \quad (4)$$

where D is the free diffusion coefficient. For a given molecule and fluid propagation environment, D is given by

$$D = \frac{k_B T}{6\pi\eta R_H}, \quad (5)$$

where $k_B = 1.38 \cdot 10^{-23}$ J/K is the Boltzman constant, T is the temperature (in K), η is the dynamic viscosity of the fluid, and R_H is the hydraulic radius

of the molecule. We assume that D is the same throughout the medium, and that collisions with the boundaries are elastic. In [11], values of D ranging from 1-10 $\mu\text{m}^2/\text{s}$ were considered realistic for signalling molecules. The angle, θ_i is an independent, identically distributed (iid) random variable for all i , uniformly distributed on $[0, 2\pi)$. The angle ϕ_i is also a iid random variable, uniformly distributed on $[-\pi/2, \pi/2)$. Figure 1 shows a sample simulation run.

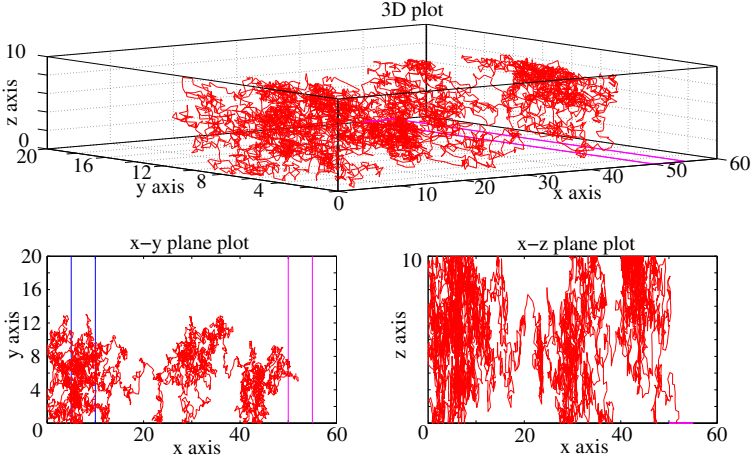


Fig. 1. A sample Brownian movement of a vesicle in three dimensions

For molecular motors, as in [10], we assume that the microchannel is lined with static kinesin motors, and that these motors cause microtubule filaments to propagate along their surface. The motion of the microtubule is largely regular, although the effects of Brownian motion cause random fluctuations. We use a simulation scheme from [12], noting that microtubules stick to the motor-lined surface, so a two-dimensional discrete-time simulation is sufficient. In this case,

$$x_i = x_{i-1} + \Delta r \cos \theta_i, \quad (6)$$

$$y_i = y_{i-1} + \Delta r \sin \theta_i. \quad (7)$$

In this case, the step size Δr is an iid Gaussian random variable with mean $v_{\text{avg}}\Delta t$ and variance $2D\Delta t$, where v_{avg} is the average velocity of the microtubule, and D is the microtubule's diffusion coefficient. The angle θ_i is no longer independent from step to step: instead, for some step-to-step angular change $\Delta\theta$, we have that

$$\theta_i = \Delta\theta + \theta_{i-1}. \quad (8)$$

Now, for each step, $\Delta\theta$ is an iid Gaussian-distributed random variable with mean $E[\Delta\theta] = 0$ and variance $\text{Var}[\Delta\theta] = v_{\text{avg}}\Delta t/L_p$, where L_p is the persistence length of the microtubule's trajectory. In [12], these values were given as $v_{\text{avg}} = 0.85 \mu\text{m}/\text{s}$, $D = 2.0 \cdot 10^{-3} \mu\text{m}^2/\text{s}$, and $L_p = 111 \mu\text{m}$. Following [12], in case of a

collision with a boundary, we assume that the microtubule sets θ_i so as to *follow the boundary*.

The loading and unloading mechanics are assumed to be the same as those proposed in [10]. Initialization, loading, and unloading have certain differences between Brownian motion and the molecular motors. We briefly outline the differences.

For Brownian motion, the vesicles start randomly and uniformly on the (x, y) plane of the loading zone, but z is set to the maximum vertical height (i.e., the molecule is “dropped” onto the microchannel). Furthermore, the vesicles start moving in Brownian motion as soon as they appear in the microchannel, so there is no need for loading. The propagation halts (“unloading”) as soon as the vesicle arrives at the unloading zone.

For molecular motors, the vesicles are attached to the loading zone and do not move until they are picked up by a microtubule filament. The starting location of the microtubule is assumed to be random and uniformly distributed across the entire propagation area. Moreover, the initial directional angle θ_0 is selected uniformly at random from the range $[0, 2\pi]$, and microtubules are assumed to be initially unloaded.

In order to capture the loading effect of active transport, in our simulations we divide the loading zone into a square grid, where the length of each square in the grid is the same as the diameter of the vesicles. We then distribute vesicle randomly and uniformly between the squares in the grid. If a microtubule enters a square which is occupied by a vesicle, we assume the microtubule loads that vesicle given it has an empty loading slot available. In general, we assume that the microtubules can load multiple vesicle and the maximum number of vesicles a microtubule can load is given by half of its length divided by the diameter of the vesicles. For unloading we assume all the loaded vesicles are unloaded as soon as a microtubule enters the unloading zone. Figure 2 shows a sample trajectory with the loading and unloading mechanism.

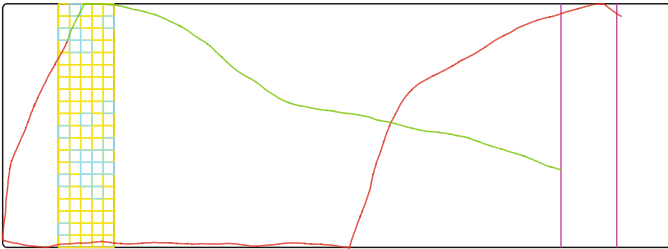


Fig. 2. A sample trajectory of active transport. The microtubule initially starts to the right of the unloading zone (box on the right side of the microchannel), and moves to the left (red trajectory). It passes through the loading zone (grid with empty yellow tiles and vesicle-bearing blue tiles) without encountering a vesicle, then passes through again, loading a vesicle. The loaded microtubule (green trajectory) then travels toward the unloading zone, where the trajectory terminates.

3 Simulation Results

We consider a microchannel with a width of $20\mu\text{m}$, a length of $60\mu\text{m}$, and height of $10\mu\text{m}$, as shown in the figure. The width of the loading strip and the unloading strip are assumed to be $5\mu\text{m}$ each. The diameter of the information carrying vesicles are assumed to be $1\mu\text{m}$ and as the results the loading strip is divided into squares of $1\mu\text{m}$ length forming a grid over the loading zone. In case of the Brownian motion, the height of the unloading zone is assumed to be 23nm since an unloading mechanism based on ssDNAs is considered.

Throughout our simulations, we use $\Delta t = 0.1\text{s}$. For Brownian motion, that the x axis is along the length of the microchannel, we set $v_y = v_z = 0$, and use various values of v_x . either $v_x = 0.3\mu\text{m/s}$, or zero (for comparison); we also

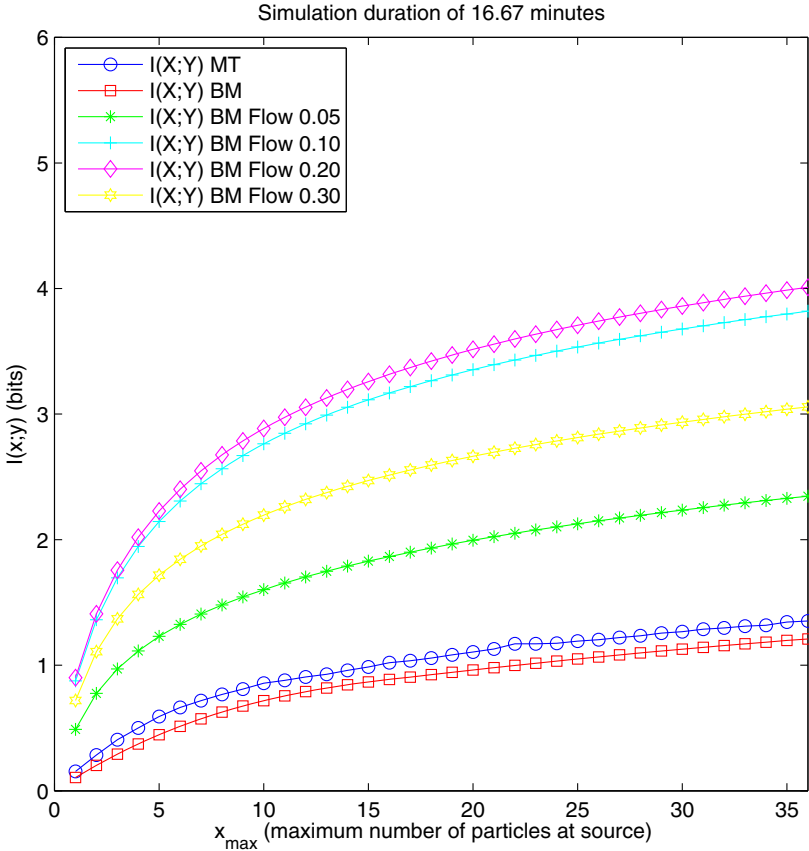


Fig. 3. Mutual information versus maximum number of vesicles at loading zone for time duration of 1000 seconds (16m 40s). Abbreviations in the legend: Microtubule (MT); Brownian motion (BM); Brownian motion with flow (BM Flow, at the given velocity).

use $D = 1 \mu\text{m}^2/\text{s}$. For molecular motors, we consider *only a single microtubule* (which may transport up to five vesicles at once), using the parameters given in Section 2. Performance should improve as the number of microtubules increases.

We use the Blahut-Arimoto algorithm to optimize the input distribution of vesicles, which is upper bounded by the parameter X_{\max} . Figure 3 shows the mutual information versus maximum number of vesicles at the loading zone for time duration of 1000 seconds. As we can see introducing a small flow increases the information rate of Brownian motion significantly. However, for higher values of flow (e.g., $v_x = 0.3$), performance declines; given our setup, this is because the vesicle speeds past the unloading zone. Overall Brownian motion with flow achieves a much higher information rate than both simple Brownian motion and active transport using a single microtubule.

References

1. Hiyama, S., Moritani, Y.: Molecular communication: Harnessing biochemical materials to engineer biomimetic communication systems. *Nano Communication Networks* 1, 20–30 (2010)
2. Moritani, Y., Hiyama, S., Suda, T.: Molecular communication for health care applications. In: *Proc. 4th IEEE Intl. Conf. on Pervasive Computing and Communications Workshops*, Pisa, Italy (2006)
3. Eckford, A.W.: Nanoscale communication with Brownian motion. In: *Proc. Conf. on Information Sciences and Systems*, Baltimore, MD, pp. 160–165 (2007)
4. Atakan, B., Akan, O.: An information theoretical approach for molecular communication. In: *Proc. 2nd Intl. Conf. on Bio-Inspired Models of Network, Information, and Computing Systems*, Budapest, Hungary (2007)
5. Moore, M.J., Suda, T., Oiwa, K.: Molecular communication: Modeling noise effects on information rate. *IEEE Trans. Nanobiosci.* 8, 169–179 (2009)
6. Eckford, A.W.: Timing information rates for active transport molecular communication. In: *Proc. 4th Intl. Conf. on Nano-Networks*, Luzern, Switzerland, pp. 24–28 (2009)
7. Pierobon, M., Akyildiz, I.F.: A physical end-to-end model for molecular communication in nanonetworks. *IEEE J. Sel. Areas in Commun.* 28(4), 602–611 (2010)
8. Eckford, A.W., et al.: Microchannel molecular communication with nanoscale carriers: Brownian motion versus active transport. In: *IEEE Intl. Conf. on Nanotechnology*, Seoul, South Korea (2010)
9. Cover, T.M., Thomas, J.A.: *Elements of Information Theory*, 2nd edn. Wiley, Hoboken (2006)
10. Hiyama, S., et al.: Biomolecular-motor-based nano- or microscale particle translocations on DNA microarrays. *Nano Lett.* 9(6), 2407–2413 (2009)
11. Berthier, J.: *Microfluidics for Biotechnology*. Artech House, Boston (2006)
12. Nitta, T., et al.: Simulating molecular shuttle movements: Towards computer-aided design of nanoscale transport systems. *Lab on a Chip* 6, 881–885 (2006)

Organic Resilience for Tactical Environments

Marco Carvalho¹, Tom Lamkin², and Carlos Perez¹

¹ Institute for Human and Machine Cognition, Ocala, FL
{macarvalho,cperez}@ihmc.us

² Air Force Research Laboratory, Dayton, OH
thomas.lamkin@wpafb.af.mil

Abstract. In this paper we present a tactical defense infrastructure for mission survivability based on three core inspirations from biological systems: multi-potentiality, feedback mechanisms, and redundancy. In tactical operational environments, these concepts may be realized through a combination of capabilities that include (1) dynamic allocation of resources for mission execution, (2) detection and identification of attacks and their effects, and (3) information sharing for system adaptation. As a proof-of-concept we introduce an extensible, multi-layer defense infrastructure inspired in the self-organization and resilience properties of biological systems. Two defense strategies are considered to validate the proposed model: a fast response consisting on rebooting a compromised system from a reference system image; and a slower response involving a process of identification of the attack, which then allows the node to change its base configuration and reboot to a state that is potentially immune to the same attack. Our experimental results show that the second strategy improves the overall resilience of the system for ongoing attacks after an initial exposure phase.

Keywords: Organic Computing, Biologically-Inspired Resilience, Tactical Networks, Resilient Systems, Distributed Control.

1 Introduction

One of the hallmark characteristics of biological systems is their capacity to adapt and evolve to environmental changes and competing pressures from peers and adversaries. Self-organization and adaption happens at all levels of scale and complexity, from intra-cellular signaling through larger scale biological systems and social interactions between individuals and groups.

In order to adapt, systems must be able to interact with the environment and respond to localized damages or perceived threats. In a general sense, this goal can only be accomplished if the system is capable to sustain some level of damage, allowing it to perceive, identify and adapt to the problem. These are capabilities that enable biological systems's intrinsic resilience to external attacks and environmental conditions. Similarly, in our view, mission critical infrastructures should be able to seamlessly absorb localized attacks with minimum impact to the ongoing tasks, while isolating and responding appropriately.

In this paper we introduce a multi-layer defense infrastructure that realize these capabilities for a mobile tactical environment.

2 An Organic Computing Approach to Mission Survivability

The concept of organic computing [1] has been recently introduced to represent and control a computational systems as a living organisms, capable to self-organize, regulate and adapt in order to survive and achieve its goals.

For tactical operational settings, mission survivability is often defined as the ability to maintain the execution of the mission, even under unexpected adverse conditions, localized failures, or attacks. A system capable to identify local failures and block external attacks in a timely manner is thought to be capable to maintain the execution of its mission-critical applications to their successful and timely completion.

The problem is further complicated in tactical environments, where the lack of a fixed coordination infrastructure, the tenuous definition of system boundaries, and the dynamic nature of the computation and communication infrastructures make it very difficult to rely on conditional approaches to improve system resilience. The security requirements for tactical operation environments are significantly different than those normally defined for infrastructure networks [2], requiring new approaches for system defense and mission assurance.

For such environments, the system's defense infrastructure must be as fluid and adaptive as the system itself. An adaptive defense infrastructure for tactical MANETs must be able to learn from successful attacks, by quickly identifying, localizing and isolating the attack to devise a defense at runtime, while ensuring that overall mission requirements continue to be met.

3 Mission Survivability in Tactical Environments

In the context of this work, a mission is defined as an ordered set of tasks that must be performed by the computational environment in a timely fashion. While there have been several approaches for mission modeling and representation, including standards such as Business Process Modeling Notation (BPMN) [3], for the sake of simplicity we will choose to describe a mission simply as an array of symbols, where each symbol represents a task and the order defines their interdependencies.

As illustrated in Figure 1, node 1 is tasked with the mission described by the sequence AFG, where each of the symbols represent a self-contained task for the mission. For a tactical service oriented architecture (SOA), the orchestration of services A, F and G must be carried out on the fly, with minimum cost and coordination overhead, and maximum performance possible. In Figure 1, three nodes collaborate to jointly execute the mission, each of them handling one of the sub-tasks A, F and G.

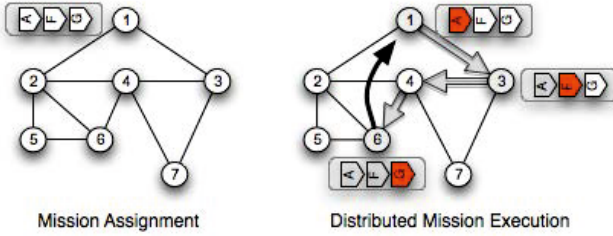


Fig. 1. Distributed mission execution through runtime distribution of tasks

The allocation of resources for task execution is handled by resource coordination and orchestration services. In tactical network environments, the allocation of services must account for the unreliability of links and topology dynamics [4]. The defense of such infrastructures, however, is not a trivial task. There are multiple attacks at the network [5], access control [6], physical [7] and also at the host level. In this environment, nodes must be able to detect, isolate and identify a problem that may affect the performance of the mission. These tasks must be carried while maintaining mission execution with minimal disruption.

4 Defense Infrastructure for Mission Survivability

Our proposed defense infrastructure is composed of three components (Figure 2). The first component manages the dynamic allocation of resources for mission execution. The second component is responsible for detection and identification of the potential attack. The third component coordinates the sharing of information about the attack, ensuring that a collective response (if appropriate) can be enforced, and that nodes that are functionally similar to the victim can be reconfigured to prevent a similar attack. A collective response to an attack may include, for instance, modifications in routing weights to disfavor the use of nodes that may have been compromised.

While simultaneously supported and coordinated, the proposed defense infrastructure must be loosely couple to prevent a cascade failure in the even that one of the components becomes temporarily impaired or permanently compromised. As conceived, the coordinated operation of all three components is necessary to enable a comprehensive response and system resilience, each component will also operate independently with limited performance gains, ensuring a graceful degradation of the survivability infrastructure itself.

Automatic resource and service re-allocation in response to localized failures is common practice in Grid environments, and has also been previously proposed for enterprise [8] and tactical [9] environments. However, in general, a change in allocation strategy happens only when degradation (or failure) has taken place and the impact on the mission has been noted, there's generally no predictive re-allocation based on increased risk of an attack or failure, learned at runtime from novel attacks. Our proposed approach leverages and extends such dynamic

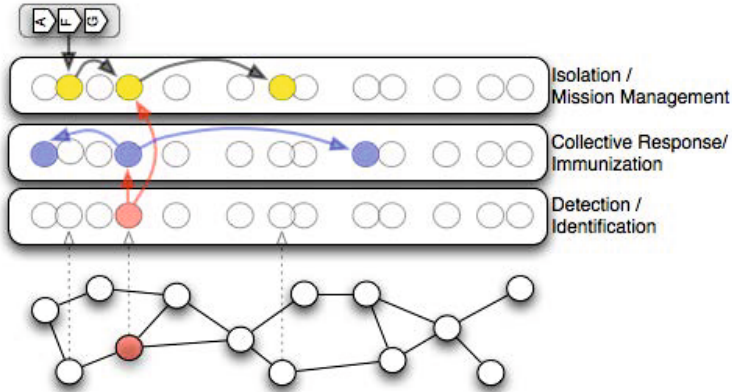


Fig. 2. Multi-layer approach to organic resilience

allocation strategies in the literature to enable proactive reallocation based on online risk estimation.

In the general case, attacks are detected indirectly, through their effects on the mission (effects-based detection), but there are related research efforts on mission mapping [10] that can also provide a better assessment of the impact of localized failures to the overall mission by matching the mission workflow with resource requirements and availability.

There are then two possible defense strategies for a node that has detected local damage. The first strategy takes significantly less time, and consists on rebooting from a reference system image, but this strategy yields no details about the attack. The node reboots just as vulnerable to the attack as it was before the detection. The second strategy involves a process of identification of the attack, which then allows the node to change its base configuration and reboot to a state that is potentially immune to the same attack.

5 Experimentation Setup and Simulation Environment

The proposed approach was implemented and evaluated in NS3. The simulations were executed over synthetic scenarios with random topologies and random mobility models and also over scenarios based on a dataset collected from a live exercise at the US Army National Training Center (NTC). The NTC dataset describes a 24-hour long exercise including approximately 480 nodes between friendlies and enemies. In addition to providing a realistic operational scenario, the NTC dataset includes the mobility information of all participating nodes, and has been used in other research efforts for mobility modeling.

Each scenario was executed with 20 different seeds and the results were averaged out across those seeds. The metric of interest for comparing results on similar scenarios is the percentage of completed missions at any given moment of

the simulation. First a baseline was computed by running all the scenarios with no attacks during the whole simulation. This baseline metric provides the upper bound for the operational limits of the system, and it is labeled as *Baseline* in the result charts. It illustrates the performance of the system (in terms of number of completed missions) under normal operations. A second baseline metric (labeled as *Baseline No Reboot*) provides the lower bound for the operational limits of the system.

In addition to a short term response (second baseline), an adaptation strategy can be used to enhance the resilience of the system to subsequent attacks. One strategy is to randomize the configuration of re-instantiated services and nodes. A second approach is to provide an immunization capability that will drive the mutations of re-instantiated servers to become resistant to previous attacks. In our experiments, we have opted for an immunization strategy.

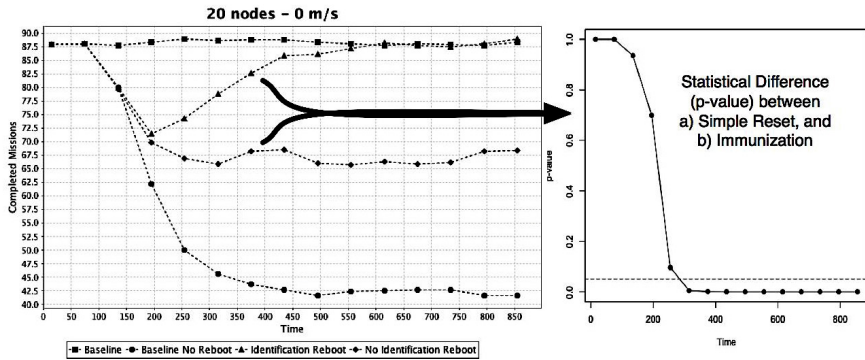


Fig. 3. Results for static scenario of 20 nodes

Figure 3 shows the results for the static scenario (no mobility). The results series labeled as *Identification Reboot* and *No Identification Reboot* represent the results for the strategies with and without immunization, respectively. In the first case, nodes detect and identify the attack and then reboot from a previously known safe state in order to maintain mission requirements. The attack detection happens indirectly (through the effects of the attack) and the identification happens by correlating the detection with the current state of the node (represented as a short string). The process takes some time, during which the services provided by the node are degraded.

In the second case (immunization), the node identifies a “mutation” strategy that is likely to make it less vulnerable to the same attack. For our simulations, the state of a node is represented by its 4-bit DNA sequence and defines how vulnerable a node is to a given attack. The immunization process involves having a node that has been attacked to announce its current DNA to its peers, which will drive “similar” nodes to mutate in order to become resistant to the attack.

A 4-bit DNA in this example represents, for instance different OS and software combinations (Web Server and libraries) of a functional node.

In the scenario illustrate in Figure 3, the strategy involving immunization starts with results close to the non-immunization strategy but then it improves until getting close to the upper operational boundaries of the system. Figure 3 shows how the p -value changes across time for a t -test of difference in percentage of completed missions for the immunization and no-immunization strategies. The dotted line in the chart represents the p -value of 0.05, for a confidence interval of 95%. So, approximately 300 seconds into the simulations (around 100 seconds after the attacks start), the difference in performance between the two defense strategies becomes statistically significant.

Figure 4 shows a different network scenario created from a subset of the National Training Center (NTC) dataset consisting of the last 2 hours of two relatively small areas with 28 nodes. The 2 hours of the scenario were compressed into 1200 seconds, to have the same number of missions as for the synthetic scenarios.

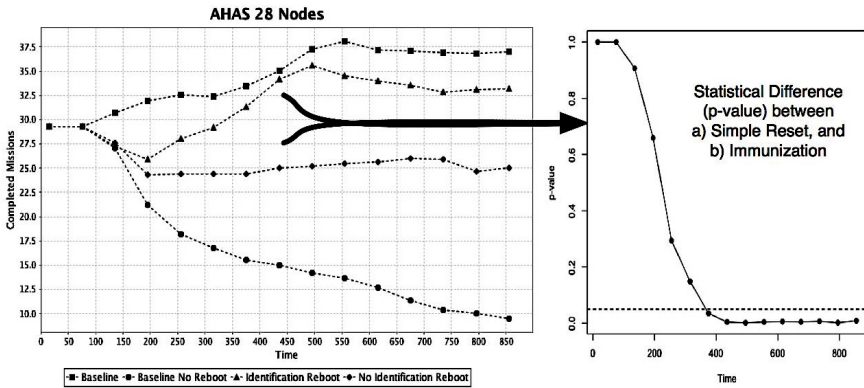


Fig. 4. Results for NTC scenario of 28 nodes

Out of the total number of nodes in the scenario, 12 nodes are selected as attackers, and 10 of the remaining nodes are issuing missions to be executed roughly every 2 seconds until completing 400 missions each. The missions and the attacks have the same characteristics as for the synthetic scenarios. Like in the previous case, the NTC scenarios were also executed with 20 different seeds and the results were averaged out across those seeds. In this scenario the system performance starts improving, but then it shows a small decrease. This is due to the fact that the topology starts with a few isolated clusters and then, due to mobility some of these clusters get connected, providing temporarily more resources to execute missions, but later a few get disconnected again, degrading the performance again.

In general, as illustrated in our preliminary results, the immunization strategy provides a gain after some initial period of time in comparison with a simply reactive approach. The results depended on the time required for the identification of the attack and definition of the adaptation strategy. For our tests we estimated an arbitrary relationship of one to four. Meaning that if the time required for detecting the damage from an attack is one second, the time required to identify the attack for immunization is four times longer. In our results, the simple response strategy *No Identification Reboot* requires only the detection, while the immunization strategy *Identification Reboot* requires the identification of the event. The objective in this example is defined as the overall number of missions executed by the system and since each mission has a time to live, it implicitly includes the time requirements for mission execution.

It is important to note that the balance between the time required to learn and disseminate an immunization strategy, the time required for damage detection and the rate of attacks will play an important role on defining the appropriate strategy (or mixture of strategies) in a more general case. While we envision simple approaches for choosing different strategies for different operational conditions, we have not yet fully evaluated these effects, which will be part of our continued work in this topic.

6 Conclusion and Future Work

In this work we presented a defense infrastructure that loosely couples a fast-response mission management capability with a threat isolation, identification and mitigation capability operating, in coordination, at a slower pace.

While the infrastructure has been designed for improved mission survivability, it is important to highlight that our approach is vulnerable to other attack strategies, which must be managed by complementary capabilities.

As future work, we would like to explore other mission allocation approaches that provide better online risk estimation. For example, other approaches using distributed reinforcement learning have been used with promising results [11].

It would be also of interest to investigate the use of distributed anomaly detection strategies for isolating malicious nodes and finding the root causes of the attacks in a coordinated fashion. This type of strategies might help mitigate some vulnerabilities of the infrastructure, like detecting a slow overall degradation of the infrastructure.

References

1. Muller-Schloer, C.: "Organic computing" on the feasibility of controlled emergence. In: CODES+ISSS 2004: Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis, pp. 2–5. IEEE Computer Society, Washington, DC (2004)
2. Carvalho, M.: Security in mobile ad hoc networks. *IEEE Security and Privacy* 6(2), 72–75 (2008)

3. Wong, P.Y., Gibbons, J.: A process semantics for BPMN. Draft (2007)
4. Carvalho, M., Winkler, R., Perez, C., Kovach, J., Choy, S.: A cross-layer predictive routing protocol for mobile ad hoc networks. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, ser. Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, vol. 6981 (May 2008)
5. Agrawal, P., Ghosh, R.K., Das, S.K.: Cooperative black and gray hole attacks in mobile ad hoc networks. In: ICUIMC 2008: Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, pp. 310–314. ACM, New York (2008)
6. Gupta, V., Krishnamurthy, S., Faloutsos, M.: Denial of service attacks at the mac layer in wireless ad hoc networks (2002)
7. Zhou, Y., Wu, D., Nettles, S.M.: On mac-layer denial of service attacks in iee 802.11 ad hoc networks: analysis and counter measures. *Int. J. Wire. Mob. Comput.* 1(3/4), 268–275 (2006)
8. Lardieri, P., Balasubramanian, J., Schmidt, D.C., Thaker, G., Gokhale, A., Damiano, T.: A multi-layered resource management framework for dynamic resource management in enterprise dre systems. *J. Syst. Softw.* 80(7), 984–996 (2007)
9. Carvalho, M., Pěchouček, M., Suri, N.: A Mobile Agent-Based Middleware for Opportunistic Resource Allocation and Communications. In: Thompson, S.G., Ghanea-Hercock, R. (eds.) DAMAS 2005. LNCS (LNAI), vol. 3890, pp. 121–134. Springer, Heidelberg (2006)
10. Musman, S., Temin, A., Tanner, M., Fox, D., Pridemore, B.: Evaluating the impact of cyber attacks on missions. In: 5th International Conference on Information Warfare and Security, April 8-9, pp. 446–456. Air Force Institute of Technology, Wright-Patterson AFB (2010)
11. Carvalho, M.: A distributed reinforcement learning approach to mission survivability in tactical manets. In: CSIIRW 2009: Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research, pp. 1–4. ACM, New York (2009)

Interfacing Living Cells via Molecular Communication

Tadashi Nakano¹, Shouhei Kobayashi², and Tokuko Haraguchi²

¹ Frontier Research Base for Global Young Researchers
Graduate School of Engineering, Osaka University, 2-1 Yamadaoka, Suita,
Osaka 565-0871, Japan

tnakano@wakate.frc.eng.osaka-u.ac.jp

² Advanced ICT Research Institute Kobe, National Institute of Information
and Communications Technology, 588-2 Iwaoka, Iwaoka-cho, Nishi-ku, Kobe 651-2492, Japan
{skobayashi, tokuko}@nict.go.jp

Abstract. Molecular communication provides a direct method for interfacing living cells via transmission of chemical signals. Here we report our recent result of developing a novel molecular communication method based on the cellular event called autophagy. The method currently under investigation uses micrometer-sized objects coated with bio-reactive molecules as information carrying molecules, which are selectively incorporated into and decoded within biological cells. This short paper briefly describes the method, experimental results, and theoretical model for the molecular communication method based on autophagy.

Keywords: Molecular communication, biological machines, autophagy.

1 Introduction

A promising material to design and engineer biological machines for molecular communication is biological cells [1]. The biological cells are by nature capable of recognizing, synthesizing, and processing chemical signals, and they can be genetically modified to implement necessary functionality for molecular communication (e.g., the distance estimation protocol [2]). The approach of using biological cells for specific applications is often employed in synthetic biology; the early work includes the establishment of sender and receiver cells using genetically modified *E. coli* (a bacterium) that collectively perform programmed pattern formation for tissue engineering [3].

Our recent efforts are focused on developing a biologically compatible interface with eukaryotic cells. A key cellular event being studied is *autophagy* in which unwanted molecular components are detected, captured, and degraded within the cells [4]. The process of autophagy can be induced with chemical signals that are externally applied (information carrying molecules in Fig. 1), and thus it can be used to provide an interface between the cell and the outside environment. After the information carrying molecule is captured in the cell, the process of autophagy starts decoding the information, which for instance leads to the formation of a membrane around a specific area in the cell.

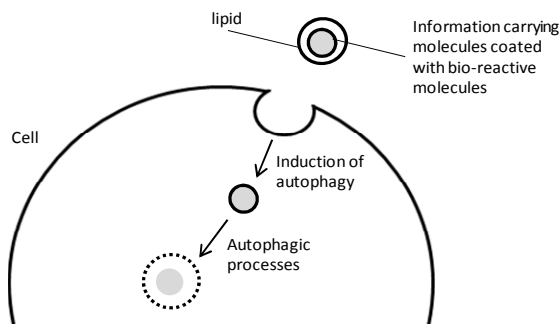


Fig. 1. Molecular communication through autophagy processes

2 Results

Our experiments using eukaryotic (and nonphagocytic) cells have identified the conditions to artificially induce autophagy processes, which have been successfully monitored by live cell imaging [4]. The molecular mechanisms of autophagy processes are now being investigated by further experiments as well as modeling and simulation.

Acknowledgements. This work was supported by KAKENHI and SCOPE, and in part carried out at the Frontier Research Base for Global Young Researchers, Osaka University, through the program Promotion of Environmental Improvement to Enhance Young Researchers' Independence, the special coordination funds for promoting Science and Technology, Japan Ministry of Education, Culture, Sports, Science and Technology.

References

1. Akyildiz, I.F., Brunetti, F., Blazquez, C.: Nanonetworks: a new communication paradigm. *Computer Networks* 52, 2260–2279 (2008)
2. Moore, M., Nakano, T., Enomoto, A., Suda, T.: Measuring distance with molecular communication feedback protocols. In: *Proc. 5th International Conference on Bio-Inspired Models of Network, Information, and Computing Systems, BIONETICS 2010* (December 2010)
3. Basu, S., Gerchman, Y., Collins, C.H., Arnold, F.H., Weiss, R.: A synthetic multicellular system for programmed pattern formation. *Nature* 434, 1130–1134 (2005)
4. Kobayashi, S., Kojidani, T., Osakada, H., Yamamoto, A., Yoshimori, T., Hiraoka, Y., Haraguchi, T.: Artificial induction of autophagy around polystyrene beads in nonphagocytic cells. *Autophagy* 6(1), 36–45 (2010)

Evolutionary and Noise-Aware Data Gathering for Wireless Sensor Networks

Bingchun Zhu¹, Junichi Suzuki¹, and Pruet Boonma²

¹ Department of Computer Science
University of Massachusetts, Boston, USA
numchun, jxs@cs.umb.edu

² Department of Computer Engineering
Chiang Mai University, Thailand
pruet@eng.cmu.ac.th

Abstract. This paper formulates a prioritized data gathering problem in noisy wireless sensor networks (WSNs) and solves the problem with a noise-aware evolutionary multiobjective optimization algorithm (EMOA). Unlike existing local search heuristics, the proposed algorithm can seek the Pareto-optimal routing structures with respect to conflicting optimization objectives. Simulation results demonstrate that the proposed algorithm outperforms a traditional EMOA in a noisy WSN.

Keywords: Wireless sensor networks, Data gathering protocol, Noisy multiobjective optimization problem, Genetic algorithm.

1 Introduction

Data gathering is one of the most fundamental tasks in wireless sensor networks (WSNs): collecting sensor data from all of, or a subset of, sensor nodes in the network on a periodic or on-demand basis [1,2]. It is known that a problem of finding the optimal data gathering solution(s) is NP-complete [2]. This paper focuses on a prioritized data gathering problem in multi-hop and semi-static WSNs. The problem is to construct the routing structures for fixed-sized query packets to visit a subset of nodes in the network, starting and ending at the base station, transmit query packets through the constructed routing structures to collect sensor data from visited nodes, and deliver collected data to the base station. In this problem, different priorities are given to different nodes; higher priorities indicate higher requirements for query packets to visit.

This paper describes a data gathering protocol, called DGP Boston (DGPB), which uses source routing to seek the optimal routing structures for prioritized data gathering in WSNs. DGPB is designed to make two key contributions. First, DGPB seeks the Pareto-optimal routing structures with respect to multiple conflicting optimization objectives such as data yield, latency and energy consumption. Since optimization objectives often conflict with each other in WSNs [3], it is important to examine the optimal trade-offs (i.e., Pareto optima) among those objectives. However, as discussed in Section 5, no existing data gathering protocols do not study that. The second contribution of DGPB is that

it considers noise in environmental and protocol parameters such as packet loss rate and node priority. No existing data gathering protocols assume noise in their parameters. (See Section 5) This paper describes and evaluates an evolutionary and noise-aware multiobjective optimization algorithm in DGPPB.

2 Problem Statement

This paper uses the following notations to state a data gathering problem in question. A WSN is considered as a graph $G(N, L)$.

- $N = \{n_0, n_1, \dots, n_m\}$ is a set of nodes in G , where n_0 is the base station. $N' = N - \{n_0\}$ is a set of m sensor nodes. This paper assumes semi-static WSNs [4, 5]; nodes are stationary and $|N|$ does not change dynamically.
- $M \subseteq N'$ is a set of nodes that query packets gather data from. It is referred as a *measurement set*. M is chosen manually or via techniques such as [6, 7].
- $L = \{(n_i, n_j) | n_i, n_j \in N; i \neq j\}$ is a set of links in G . A link is established from n_i to n_j if n_i can directly transmit a packet to n_j without intermediate nodes. Thus, nodes in G are not fully connected with L .
- r_{ij} is the packet loss rate on a link (n_i, n_j) . Due to the asymmetric nature of link quality, $r_{ij} = p_{ij} \times p_{ji}$, where p_{ij} denotes the loss rate to transmit a packet from n_i to n_j . $t_{ij} = \frac{1}{(1-p_{ij}) \times (1-p_{ji})}$ is the expected number of transmissions, including transmission retries, to successfully deliver a packet on (n_i, n_j) .
- $G_M(M, P) \subseteq G$ is a graph that consists of M . $P = \{(n_i, n_j) | n_i, n_j \in M; i \neq j\}$ is a set of paths in G_M . A path is a sequence of one or more links in L . It is established as the minimum loss rate path from n_i to n_j if n_i can transmit a packet to n_j through a single-hop or multi-hop path. Thus, nodes in G_M are fully connected with P .
- d_i is a demand (or priority) assigned to $n_i \in M$.
- g is the number of query packets used for data gathering. Each packet can carry a limited size S of data due to the limitation of packet size. This is a constraint on how many nodes a packet can collect data from.
- R_q is a route that a query packet q follows to gather data through the paths in P . It is a sequence of nodes in M , starting and ending at n_0 . l_{R_q} is the number of links in R_q . $T_{R_q} = \sum_{n, n' \in R_q} t_{nn'}$ is the expected total number of transmissions to route a packet in R_q . (n' is the next hop node of the node n in R_q .) $R = \{R_1, R_2, \dots, R_q, \dots, R_g\}$ is a route set.

A data gathering problem this paper addresses is to find a set of Pareto-optimal route sets that minimize the following objective functions:

1. *Latency*: indicates the time required for data gathering. It is computed as $\max\{T_{R_1}, \dots, T_{R_q}, \dots, T_{R_g}\}$.
2. *Per-demand energy consumption*: indicates the amount of energy consumed to collect a unit demand. It is computed as $\sum_{R_i \in R} T_{R_i} / \sum_{R_i \in R} \sum_{j \in R_i} d_j$.
3. *Per-hop packet loss rate*: indicates the average packet loss rate in a route set. It is computed as $\sum_{R_i \in R} \sum_{n, n' \in R_i} r_{n, n'} / \sum_{i=1}^g l_i$, where n' is the next hop node of the node n in R_i .

3 DGP Boston

DGPB’s algorithmic structure follows NSGA-II, an existing evolutionary multi-objective optimization algorithm [8]. Due to space limitation, this section focuses on a set of extensions that DGPB makes on NSGA-II.

3.1 Individuals

In DGPB, each individual is a variable-length representation of packet routes as shown in Fig. 1. It encodes the number of routes (i.e., the number of packets) and the order of nodes visited by each packet.

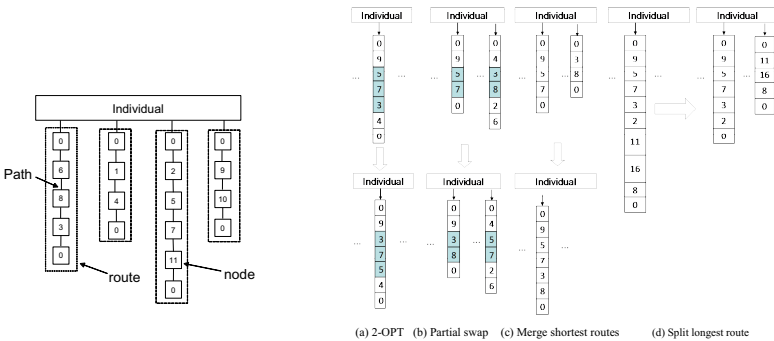


Fig. 1. An Example Individual

Fig. 2. Mutation Operators

3.2 Crossover and Mutation Operators

BGPB adopts partially-mapped crossover (PMX) [9] as its crossover operator. It also uses the following eight mutation operators. (See Fig. 2 for four of them.):

- *Add*: randomly chooses a node from the measurement set and inserts it to a randomly-selected place in a randomly-selected route. This operator ensures that the inserted node is not redundant with the other nodes in the route.
- *Delete*: removes a randomly-selected node from a randomly-selected route.
- *Exchange*: randomly chooses a node in a randomly-selected route and replaces it with a node selected from the measurement set randomly. This operator ensures that the new node is not redundant with the other nodes in the route.
- *Swap*: exchanges the positions of two randomly-selected nodes in a route, which is also selected randomly.
- *Inversion (2-OPT)*: randomly chooses two cut points in a randomly-selected route and reverses the the order of nodes between the cut points.
- *Partial swap*: randomly selects a subsequence of nodes in each of randomly-selected two routes, and swaps two subsequences between the two routes.

- *Merge the shortest routes*: identifies the two shortest routes and appends one of them to the other.
- *Split the longest route*: identifies the longest route and splits it to two routes at a randomly-chosen point.

DGPB classifies these eight mutation operators to three categories. The first category consists of *Add*, *Delete* and *Exchange*. They alter the nodes an individual visit. The second category consists of *Swap*, *Partial swap* and *2-OPT*, which alter the order of nodes that an individual visits. The remaining two operators are in the third category. They alter the number of routes that an individual have. These three categories have the same probably to be used. In each category, mutation operators are selected randomly.

3.3 The α -Dominance Operator

As discussed in Section 2, BGPB considers noise in its objective functions. The noise interferes with a dominance operator, which determines dominance relationships among individuals. For example, the operator may mistakenly judge that an inferior individual dominates an superior one. Defects in a dominance operator significantly degrades optimization performance [10].

To address this issue, DGPB replaces NSGA-II's classical dominance operator with a new noise-aware operator, called the α -dominance operator. Given objective value samples of two individuals, the α -dominance operator estimates the impacts of noise on objective values and determines whether it is statistically confident enough to judge a dominance relationship between the two individuals.

Individual A is said to α -dominate individual B (i.e., $A \succ_{\alpha} B$) iff:

1. A 's and B 's samples are classifiable with the confidence level of α , and
2. $\mathcal{C}(A, B) = 1 \wedge \mathcal{C}(B, A) < 1$.

In order to examine the first condition, the α -dominance operator classifies A 's and B 's objective value samples with Support Vector Machine, and measures a classification error (Step 1 in an example in Fig. 3). The error (e) is computed as the ratio of the number of miss-classified samples to the total number of samples. For evaluating confidence level (α) in the classification error, the α -dominance operator computes the classification error's confidence interval: $e_{int} = e \pm t_{\alpha, n-1} \sigma$ where $t_{\alpha, n-1}$ denotes a single-tail t -distribution with α confidence level and $n-1$ degrees of freedom. n denotes the total number of samples. σ is the standard deviation of e , which is approximated as $\sigma \cong \sqrt{\frac{e}{n}}$. If e_{int} is significant (i.e., if e_{int} does not span zero), the α -dominance operator cannot classify A 's and B 's samples with the confidence level of α . Thus, the operator determines that A and B do not α -dominate with each other (Step 2 in Fig. 3).

If e_{int} is not significant (i.e., if e_{int} spans zero), the α -dominance operator can classify A 's and B 's samples with the confidence level of α . Thus, the operator examines the aforementioned second condition with \mathcal{C} -metric [11]: $\mathcal{C}(A, B) = |\{b \in B \mid \exists a \in A : a \succ b\}|/|B|$ where \succ denotes a classical notion of dominance [8]. A sample $a \in A$ is said to dominate a sample $b \in B$ (i.e., $a \succ b$) iff a 's objective

values are superior than, or equal to, b 's in all objectives, and a 's objective values are superior than b 's in at least one objective. $\mathcal{C}(A, B)$ denotes the fraction of B 's samples that at least one sample of A dominates. If $\mathcal{C}(A, B) = 1$, all of B 's samples are dominated by at least one sample of A . The α -dominance operator determines $A \succ_{\alpha} B$ if $\mathcal{C}(A, B) = 1$ and $\mathcal{C}(B, A) < 1$. If $\mathcal{C}(A, B) < 1$ and $\mathcal{C}(B, A) < 1$, it determines neither $A \succ_{\alpha} B$ nor $B \succ_{\alpha} A$. See Step 2 in Fig. 3.

Fig. 3 shows an example that determines the α -dominance relationship between two individuals, A and B , with two objectives, f_1 and f_2 , to be minimized. A and B have seven samples each. The first step is to classify these 14 samples with SVM and compute e_{int} . Suppose SVM produces a classification vector as shown in Fig. 3. Two samples of B are miss-classified; $e = \frac{2}{14}$ (0.143). Thus, $\sigma \cong \sqrt{\frac{0.143}{14}} = 0.1$. Assuming $\alpha = 0.95$, $e_{int} = 0.143 \pm 1.771 * 0.1 = 0.143 \pm 0.1771$. Since e_{int} spans zero, A 's and B 's samples are classifiable with the confidence level of 95%. The second step is to compute \mathcal{C} -metric: $\mathcal{C}(A, B) = 1$ and $\mathcal{C}(B, A) = 2/14 < 1$. Therefore, the α -dominance operator determines $A \succ_{\alpha} B$.

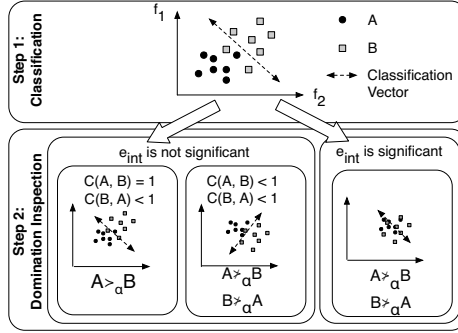


Fig. 3. An Example α -Dominance Operation

4 Preliminary Simulation Evaluation

This section evaluates DGPB, through simulations, in a WSN that have noise in packet loss rate and demand. The highest demand (100) is given to an event reporting node. Lower demands (85, 70 and 55) are given to the nodes that are 1-hop, 2-hops and 3-hops away from the event reporting node. These nodes and the event reporting node are included in the measurement set M ($|M| = 79$). Table 1 shows a set of parameter values used in simulations. The α -dominance operator performs C-support vector classification with a linear kernel in its SVM.

The noise on demands follows a uniform distribution $[-50, 50]$. The noise on packet loss rate is given based on an empirical experimental result in [4].

Table 2 compares DGPB with NSGA-II and four combinations of existing heuristics: nearest neighbor route construction (NN), 2-OPT local search and Clarke-Wright Savings algorithm (CWS). Performance metrics are the total

Table 1. Parameter Values used in Simulations

Parameter	Value	Parameter	Value
Observation area	200m×200m	Population size	100
The number of nodes	150	Max # of generations	500
Communication range	30m	Crossover rate	0.9
Base station	(0,100)	Mutation rate	0.4
Event reporting node	(100, 100)	Stopping criteria in SVM	$1e^{-3}$
Max # of data stored in a packet	10	C in SVM	1
Packet loss rate on each link	0.05	# of samples in α -dominance	20

number of data transmissions as well as the three objectives described in Section 2. The average results indicate the average of 20 simulation runs. The values in parentheses show standard deviations. Table 2 illustrates that DGPB outperforms NSGA-II except for per-hop packet loss rate. It outperforms four combinations of exiting heuristics in all performance metrics.

Table 2. Objective Values

	Average				Maximum				Minimum			
	#Trans	Obj1	Obj2	Obj3	#Trans	Obj1	Obj2	Obj3	#Trans	Obj1	Obj2	Obj3
NN	58.7(31.1)	12.71(2.01)	0.018(0.002)	0.371(0.017)	104.4	22.14	0.083	0.71	4.5	5.30	0.007	0.16
NN+2OPT	48.7(28.2)	12.19(1.09)	0.017(0.001)	0.277(0.016)	93.0	25.23	0.079	0.71	4.2	5.62	0.011	0.15
CWS+NN	64.2(32.4)	14.79(2.66)	0.031(0.002)	0.404(0.021)	108.8	24.65	0.084	0.83	10.5	4.36	0.014	0.20
CWS+NN+2OPT	52.4(27.1)	13.85(2.31)	0.020(0.001)	0.375(0.018)	97.7	24.61	0.079	0.79	9.2	4.03	0.013	0.18
NSGA-II	5.18(1.54)	4.87(1.42)	0.019(0.008)	0.016(0.006)	7.92	7.92	0.004	0.057	2.26	2.26	0.013	0.001
DGPB	4.26(1.56)	3.75(1.09)	0.017(0.008)	0.052(0.002)	7.88	7.88	0.022	0.079	3.11	3.11	0.013	0.03

Obj1: Latency; Obj2: Per-demand energy consumption; Obj3: Per-hop packet loss rate

Table 3. Quality Indicators

	Hyper Volume	Scott's Spacing	U-Measure	# of Non-dominated Individuals
NSGA-II	0.53	0.381	0.72	49.2
DGPB	0.75	0.207	0.61	100

Table 3 compares the non-dominated individuals produced by NSGA-II and DGPB. Hypervolume [11] indicates the optimality and spread of individuals in the objective space. DGPB yields 42% greater volume than NSGA-II. Scott's Spacing [12] and U-measure [13] indicate the distribution of individuals in the objective space. (A smaller value means that individuals are distributed more evenly.) DGPB evolves individuals more evenly in the objective space than NSGA-II. In 500 generations, DGPB evolves all 100 individuals to be non-dominated, while NSGA-II evolves less than half of it. This means that DGPB

maintains a stronger evolution pressure to converge individuals. As demonstrated in Table 3, DGPB outperforms NSGA-II in all comparison metrics.

5 Related Work

This work is an extension to [14], which formulates data gathering in WSNs as a VRP and solves it with an existing heuristic algorithm (CWS). As shown in Table 2, DGPB outperforms CWS-based algorithms. Moreover, DGPB considers three conflicting optimization objectives, while [14] considers energy consumption as an objective and latency as a constraint. [14] does not consider prioritized data gathering in noisy WSNs.

Meliou et al. [2] formulate data gathering as a TSP and solve it with a TSP approximation method. They consider a single optimization objective: energy consumption. Prioritized data gathering in noisy WSNs is out of their scope.

Several evolutionary multiobjective optimization algorithms (EMOAs) exist to solve VRPs. Ombuki et al. [15] study an EMOA for a VRP, and Tan et al. [16] study an EMOA for a VRP with stochastic demands. Both work assume fully-connected graphs. Since nodes in a WSN are often not fully-connected, DGPB reduces a WSN to a fully-connected graph with the notion of shortest paths. DGPB considers an extra stochastic (or noisy) parameter, packet loss rate, which Tan et al. do not.

In the field of EMOAs, several existing dominance operators consider noise in objective functions [17,18]; however, all of them assume particular noise distributions. For example, [19-21] assume normal distributions. [22] assumes a uniform distribution. [23,24] assume Poisson distributions. Given a noise distribution, each of existing noise-aware dominance operators statistically estimates each individual's objective value by collecting its samples. In contrast, the α -dominance operator assumes no noise distributions a priori because it is not realistic to predict and model them in WSNs. Instead of estimating each individual's objective values, the α -dominance operator measures the effect of noise on objective value samples and determines whether it is confident enough to compare individuals.

6 Conclusion

This paper proposes considers WSN application required to simultaneously minimize latency, energy consumption and per-hop packet loss rate in a noisy environment. A noisy-aware multiobjective evolutionary algorithm, NSGA-IIA, is proposed and applied to the problem. Evolution results show that NSGA-IIA can find better nondominated individuals in terms of convergence and diversity.

References

1. Krishnamachari, B.: Modeling Data Gathering in Wireless Sensor Networks. In: Wireless Sensor Networks and Applications, III. Signals and Communication Technology, pp. 387–399. Springer, Heidelberg (2007)

2. Meliou, A., Chu, D., Hellerstein, J., Guestrin, C., Hong, W.: Data gathering tours in sensor networks. In: Proc. of ACM/IEEE IPSN (2006)
3. Han, Q., Hakarrinen, D., Boonma, P., Suzuki, J.: Quality-aware sensor data collection. *Int'l Journal of Sensor Networks* 7(3), 127–140 (2010)
4. Woo, A., Tong, T., Culler, D.: Taming the underlying challenges of reliable multi-hop routing in sensor networks. In: Proc. SenSys (2003)
5. Zhao, J., Govindan, R.: Understanding packet delivery performance in dense wireless sensor networks. In: Proc. SenSys (2003)
6. Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J., Hong, W.: Model-driven data acquisition in sensor networks. In: Proc. VLDB (2004)
7. Wada, H., Boonma, P., Suzuki, J.: Chronus: A spatiotemporal macroprogramming language for autonomic wireless sensor networks. In: *Autonomic Network Management Principles: From Concepts to Applications*, Elsevier (in press)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6(2) (2002)
9. Goldberg, D., Lingle, R.: Alleles, loci and the traveling salesman problem. In: Proc. 1st Int. Conf. on Genetic Algorithms, pp. 154–159 (1985)
10. Bianchi, L., Dorigo, M., Gambardella, L., Gutjahr, W.: A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing* 8(2) (2009)
11. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput.* 3(4)
12. Knowles, J., Corne, D.: On metrics for comparing nondominated sets. In: Proc. World on Congress on Computational Intelligence (2002)
13. Wang, Y.-P., Wing Leung, Y., Ping Wang, Y., Ping Wang, Y.: U-measure: A quality measure for multiobjective programming. Technical Report, Hong kong Baptist University (2003)
14. Boonma, P., Han, Q., Suzuki, J.: Leveraging biologically-inspired mobile agents supporting composite needs of reliability and timeliness in sensor applications. In: Proc. IEEE FBIT (2007)
15. Ombuki, B., Ross, B.J., Hanshar, F.: Multi-objective genetic algorithms for vehicle routing problem with time windows. In: *Applied Intelligence*, vol. 24 (2006)
16. Tan, K.C., Cheong, C.Y., Goh, C.K.: Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research* 177(2) (2007)
17. Beyer, H.-G.: Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice. In: *Computer Methods in Applied Mechanics and Engineering*, vol. 186(2-4) (2000)
18. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments: a survey. *IEEE Trans. Evol. Comput.* 9(3) (2005)
19. Goh, C.K., Tan, K.C.: Noise handling in evolutionary multi-objective optimization. In: Proc. of IEEE CEC (2006)
20. Eskandari, H., Geiger, C.D., Bird, R.: Handling uncertainty in evolutionary multi-objective optimization: SPGA. In: Proc. of IEEE CEC (2007)
21. Babbar, M., Lakshmikantha, A., Goldberg, D.E.: A modified NSGA-II to solve noisy multiobjective problems. In: Proc. of ACM GECCO (2003)
22. Teich, J.: Pareto-front exploration with uncertain objectives. In: Proc. of Int'l Conf. on Evol. Multi-Criterion Optimization (2001)
23. Wormington, M., Panaccione, C., Matney, K.M., Bowen, D.K.: Characterization of structures from x-ray scattering data using genetic algorithms. *JSTOR Philosophical Transactions* 357(1761), 2827–2848 (1999)
24. Delibrasis, K., Undrill, P., Cameron, G.: Genetic algorithm implementation of stack filter design for image restoration. In: *IEE Proc. VISP*, vol. 143(3) (1996)

A Checkpoint-Orientated Modelling for Cell Cycle Simulation

Jonathan Pascalie¹, Hervé Luga¹, Valérie Lobjois²,
Bernard Ducommun², and Yves Duthen¹

¹ IRIT - UMR5505 - University of Toulouse

{jonathan.pascalie,herve.luga,yves.duthen}@irit.fr

² ITAV - CNRS - University of Toulouse

{valerie.lobjois,bernard.ducommun}@itav-recherche.fr

Abstract. In this paper we propose a new model of cell cycle simulator which will be used to analyse checkpoint response in multicellular tumor spheroids. Whereas most of the models are phase orientated, our model integrates environmental parameters and checkpoint responses that are required to control cell cycle progression. We will present in this paper our work under progress and the different experiments to be performed in order to validate our cell cycle simulator.

Keywords: artificial life, cell simulation, oncogenesis, cell cycle.

1 Introduction

In silico simulations could provide attractive and fruitful new perspectives for the investigation in biological sciences. In complementarity with or when experiments are technically difficult to address *in vivo*, virtual environments might prove to be of tremendous interest [Sauro et al., 2006]. Such simulations are generally set up thanks to a high number of parameters. We focus here on cell cycle simulation and our goal is to provide the biologist with control tools on a set of representative and tunable parameters. These tools must give to the biologist a relevant view and understanding of the simulation results. The set of parameters must also offer a generic way of building different cell profiles with specific behaviors and of setting up biologically relevant scenario.

In addition to classical 2-D monolayer cell models, our current research on cell cycle control and regulation [Boutros et al., 2007] relies on the use of an *in vivo* model called Multicellular Tumor Spheroids (MCTS) [Kunz-Schughart et al., 2004]. This model recapitulates the main properties of growth found in a tumor and reproduces most of its characteristics. These include nutrient and hypoxia gradients that generate a regionalisation of cell proliferation, with actively dividing cells found in the outer layers, quiescent (or dormant cells) in the intermediate zones and a necrotic core in the center. This model is particularly relevant for pharmacological evaluation as well as for fundamental studies on cell proliferation thanks to the recent technical advances in molecular biology and in 3-D cell and tissue imaging [Frongia et al., 2009].

The ultimate aim of our project is to develop a cell simulator which will reproduce *in silico* the spatial organisation and the proliferative behavior of cells in a MCTS 3-D model [Hohme and Drasdo, 2010], taking into account the spatial and temporal organization of growth with a special emphasis on cell cycle checkpoints. To this end we have chosen to focus on the scheduling of the cell fate and its different states. The next section will present the concept of cell cycle and checkpoint control; section 3 will present our model based on two elements, activities and checkpoints. The fourth section presents the validation procedure we want to use and section 5 will present our perspectives for the use of this simulator and discuss some of the questions it will address.

2 Cell Cycle and Checkpoints

The cell cycle is often drawn as a circular timeline with different phases starting in G1 and ending at mitosis when a cell divides into two daughter cells. The study of the cell cycle by the biologists puts major emphasis on the essential role of the checkpoints [Elledge, 1996]. The integrity of the checkpoint is essential for cell cycle progression and for the maintenance of genomic stability. By the end of the G1-phase, at the commitment point (R), the cell integrates environmental signals before proceeding towards the G1/S transition. A lack of these signals will lead the cell to enter a quiescent (G0) state. If pro-apoptotic signals are detected the cell will undergo death, called apoptosis. Alternatively differentiation signals will drive the cell out of cell cycle to a differentiation program. If the cell progresses in the cell cycle, it must duplicate accurately all its internal material (DNA, centrosome etc) and double its mass before preparing for division. Before entering into S-Phase where DNA synthesis occurs, the cell must check for the integrity of its genetic material. This is called the G1/S checkpoint. Providing that DNA synthesis is completed the cell switches to G2-phase and it finishes doubling its mass. During S-phase and G2-phase, centrosome duplication and maturation occurs thus building the two platforms that will allow the assembly of the mitotic spindle required for mitosis to occur. However, before proceeding from G2 to mitosis, the cell must ensure again the integrity of its genetic material. This is called the G2/M checkpoint. At mitosis, when cells are dividing, in order to ensure an even segregation of the genetic material in the two daughter cells, the mitotic checkpoint will prevent division until the chromosomes are perfectly aligned on the equatorial plan. Any alteration in these checkpoint mechanisms (for instance a mutation in a key regulator) leads to a genetic instability often associated with transformation and cancer. For these reasons it is essential to integrate checkpoints as artefacts of our simulation model.

3 Modelling Cell Behaviour

3.1 Activities and Checkpoints

The proposed cell cycle simulation model is composed of several activities which represent specific cell behaviours and also checkpoints which are sets of pre-conditions of advancement on the timeline. Figure 1 shows a cartography of

the cell cycle with the localisation of each activity and checkpoint. Based on the experience of our previous work on evolution of developmental systems (Evo-Devo) [Chavoya and Duthen, 2008] where we used evolution strategies to compute the sequencing of the different actions, and on the Cell2Organ model [Cussat-Blanc et al., 2010] based on an artificial regulatory network, the control relies here on a well known sequencing of different actions for cells. The differences observed between cells are function of different schedulings for each action. For instance cancer cells that are found to be defective for a specific checkpoint, such as the mitotic one, will often undergo division with uneven chromosome segregation where normal cells would stop cycling.

Activities. The different activities are composed of an initialisation time T_{min} , a time-out T_{max} and a probability of success between T_{min} and T_{max} : $P(succes(A))$ where A is one of the designed activities . If an activity is not successful between T_{min} and T_{max} the cell goes to apoptosis. These parameters allow scientists to design cells with alternate behaviours.

- **Initialisation:** this action matches the G1-Phase of the cell cycle. All cells starting their cycle observe this phase which culminates at the R restriction point. During this phase, the cells have not yet been committed to proliferation, differentiation or entry into quiescence.
- **DNA Synthesis:** this activity matches the activity observed in the-S phase. This activity starts when DNA integrity has been verified at the G1/S transition. During this action the cell replicates its DNA.
- **Growth:** this action represents the cell’s doubling of its mass. It starts at the beginning of the S-phase and ends during the G2-phase.
- **Centrosome Duplication:** this action represents the duplication of the centrosome. It occurs simultaneously with Growth across the S and G2 phases.

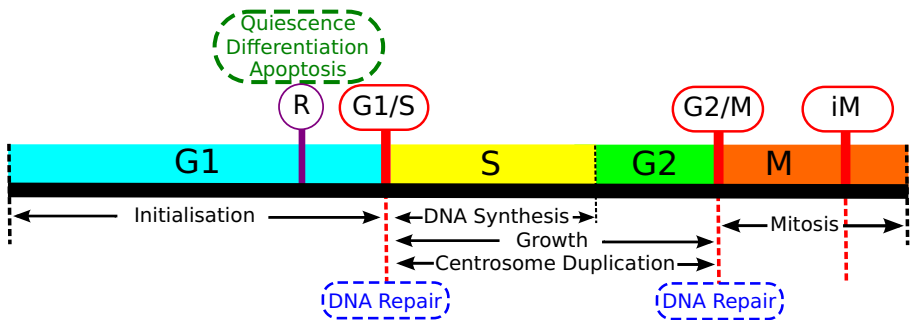


Fig. 1. Localisation of each activity and checkpoint on the cell cycle timeline. Red boxes represent checkpoints with iM being the intra-mitotic one; blue boxes are activities that could be executed during the associated checkpoint; in black with arrows are represented the different activities executed during the cell cycle; the purple circle is the commitment point and the green box represents the three exiting points.

- **Mitosis:** it is the last action of the cell cycle requiring prior checking of genomic activity at the G2/M transition. Mitosis occurs, if all pre-conditions are met, in the final stages of the cycle and ends with the beginning of the two new cycles of the daughter cells. Completion of mitosis requires chromosome alignment at the equatorial plan (mitotic checkpoint).
- **Differentiation:** it represents one of the exit points of the cell cycle. If specific conditions are available the cell will differentiate.
- **Quiescence:** also called G0-Phase, this state is an active survey loop for when environmental factors are insufficient for the cell to proliferate. The quiescent cells are able to return to the cell cycle at anytime if the conditions for growth are met.
- **Apoptosis:** it represents cellular death. Apoptosis happens if apoptotic factors or signals are delivered to the cell or if the cell spends too much time in a specific arrest situation during cell cycle. This activity has neither temporal flags nor probability of success. We assumed that apoptosis always succeeds instantaneously.
- **DNA Repair:** this action is executed if the cell detects lesions on its DNA before starting its replication or before mitosis. This activity is a recovery function allowing the cell to repair its DNA and to advance in its cycle instead of dying of damaged DNA.

Checkpoints. The other important elements of our model are the checkpoints. The checkpoints are composed of a list of activities along with the preconditions of their activation. If several activities are activated at the same time the cell executes them simultaneously. The different preconditions that we set at the checkpoints are boolean flags representing an internal state of the cell or the disponibility of environmental factors. The following list presents the different checkpoint of the cell cycle :

- The **G1/S checkpoint:** here the cell checks its DNA for lesions. If lesions are found, the cell repairs them else it starts DNA Synthesis, Growth and centrosome cycle.
- The **G2/M checkpoint:** to pass through this checkpoint the cell must have replicated its DNA, duplicated its centrosome and doubled its mass.
- The **intra-mitotic checkpoint:** to pass this checkpoint and to divide into two daughter cells, the cell needs to have aligned its chromosomes on the mitotic plan and placed centrosomes on the mitotic spindle poles.

The following list shows the different preconditions that we defined for the cell cycle and what they express when their value is true.

- $C(f)$ expresses that the environmental concentration of f factors is greater than the pre-defined threshold, f could represent glucose, oxygen, growth factors, differentiation factors or proapoptotic factors,
- F_c is the contact forces that inform the cell of its neighbouring environment,
- *Contact_Inhibition* indicates that the cell does not have enough place to divide,

- *Mass_X2* expresses that the cell has doubled its mass,
- *Centrosome_X2* expresses that the cell has duplicated its centrosome,
- *DNA_X2* informs that the cell has replicated its DNA,
- *DNA_Ok* tells that the DNA does not present lesions,
- *Mitotic_plan* expresses that the chromosomes are correctly aligned on the equatorial plan and that the centrosomes are on opposite poles.

This is an open list and some preconditions could be added to design more specific behaviours. As an example we also use a *dNTP_Carency* flag which allows an inhibition of DNA replication. Cells can be blocked in a particular action if required conditions to pass through the checkpoint are not fulfilled. If a cell stays blocked at a checkpoint until the timeout of one or several actions it dies.

3.2 The Scheduling Policy

A naturally growing population of cells presents heterogeneous characteristics. Because of the variability of the duration of each cell cycle phase, two cells *born* at the same time will not divide simultaneously even if environmental conditions are equivalent. This property is also observed in *in vitro* cultures. In these cultures, the use of pharmacological compounds allows the synchronisation of cells, blocking them at a checkpoint. The need to understand temporal behaviour of cells is at the heart of cancer research and a simulator must offer a new way of investigating this problematic. For this purpose we wish to avoid using fixed hardcoded parameters for the duration of the cell cycle. On the one hand, we want the evolution of the model to determine the duration of the cycle as function of the environment's evolution. On the other hand we need to fix some limits to avoid excessively abnormal behaviour. Keeping this goal in mind we propose the use of a set of temporal flags fixed by the designer.

The flags, as presented in the previous section, consist of a minimal time for each activity, a time out to avoid infinite processes and the probability of success between T_{min} and T_{max} . This probability of success offers the simulation guarantees of heterogeneity throughout its cell population. This heterogeneity is one of the strengths of our model. It should normally prevent the emergence of computational artifacts like synchronisation or phasing. On a biological level it should offer us the means to build cellular synchronisation thanks to virtual molecular activity. In the same way this heterogeneity offers the simulation a biological relevance by not considering cells as homogeneous agents. To increase this heterogeneity and to stay as close as possible to what is observed *in vivo*, we also add noise to each temporal flag. Applied to the population these noises are expressed as a gaussian repartition centered on the flag value for each parameter.

4 Software Architecture

Our simulator is implemented using C++ language. Figure 2 shows a simplified class diagram of the simulator. GUI and functional core are executed on

two different threads which communicate thanks to Qt 4.4 signals/slots framework. This architecture allows to launch offscreen simulation. Future extension should need more computational resources and offscreen simulations will be necessary to use parallelization on computer grid or supercomputer. Actually cells are executed sequentially with random sort between each top to simulate multithreaded execution. With its graphical interface our simulator offers to the user the possibility to change simulation parameters in real time.

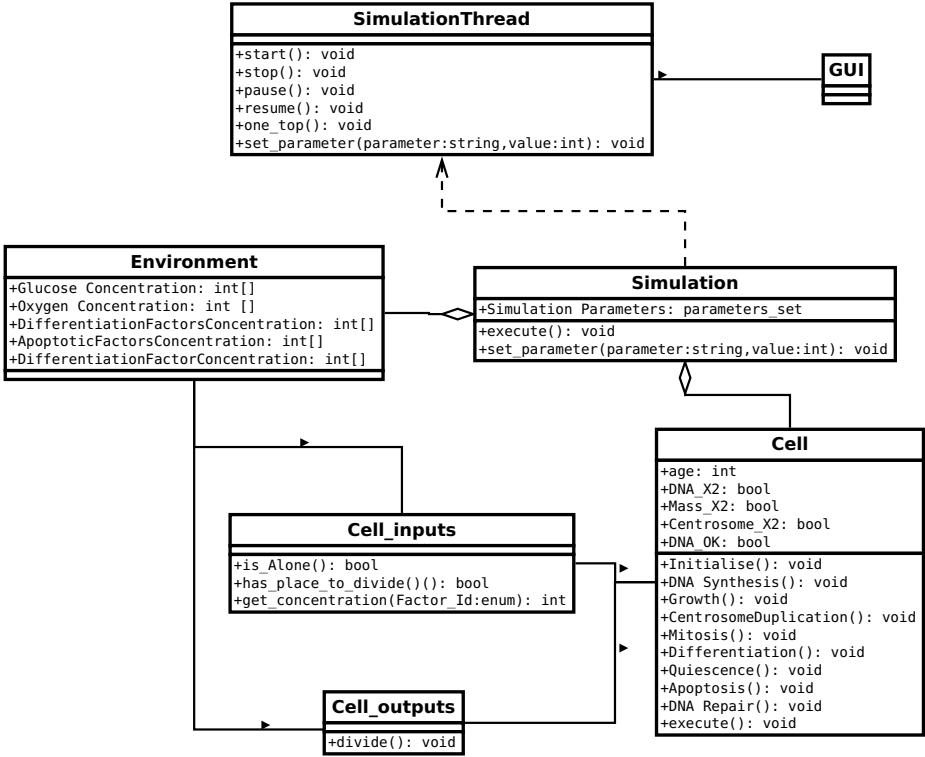


Fig. 2. Simplified class diagram of the simulator

5 Ongoing Validation Procedure

The validation of the model and of the scheduling policy will be done in a two steps process. First, we will implement a 2-D prototype to validate the cell cycle model with simple experiments. There are *in vivo* experiments in 2-D monolayer cultures that we could simulate to analyse our model response. This section shows what these experiments will be.

Our 2-D prototype will be validated by evaluating the convergence of simulation and well known biological results on different scenarii. We will validate individual and collective cell behaviour using *in silico* experiments that are reproduced *in vivo*. This validation will be done through proliferation experiments with and without environmental constraints. A good match between *in vivo* and *in silico* simulations would offer us the opportunity to build new simulations. These simulations would allow the investigation of biological theories presently unworkable in *in vivo* experimentation.

For example we will use the following validation experiments : cell cycle synchronisation through a lack of environmental factors (arrest in G0); cell cycle synchronisation using a procedure known as double thymidine block (arrest at G1/S); application of a compound targeting the assembly of the microtubules (arrest at mitosis); etc ...

6 Future Prospects

In this section we describe a set of experiments we will perform on the 3D simulator. All these experiments will be conducted at the same time on *in vivo* spheroid models to compare obtained results and increase the simulation's modalities. In a cross-talk logic between *in vivo* and *in silico* experiments, these simulations will be used to help the biologists to analyse experimental data along with the study of the efficiency of different mechanisms. The simulations may also lead us to formulate new hypothesis which would need the use of *in vivo* experiences to validate *in silico* results. To study checkpoint alteration we will introduce virtual mutation of the checkpoint machineries and analyse their consequences on the proliferative behaviour in unconstrained condition and under genotoxic stress or chemotherapy treatment. Another kind of experiment that could be performed is a coevolution between two kinds of cells : a minority of mutant cells with damaged checkpoints and a population of normal cells.

These kind of experiments will then be extended to the study of the cells' response to different kinds of environmental signals. We will also study the proliferation, in basal conditions, of the occurrence of a hypoxia gradient. One of the future extensions of our simulator will be a vascularisation module for the 3-D simulation by using the introduction of virtual neo-vessels. Finally, we will investigate the therapeutic response of cells exposed to external agent targeting the proliferation mechanisms implemented in the model. Many questions might be addressed such as the interest of a specific combination of therapeutic approaches. We will also address specific issues related to the association of a genotoxic agent and an abrogator of the G2/M checkpoint : are the consequences the same in 2-D and 3-D cultures ? Which cells are damaged ? What is their location ? What are the implications on the tumoral growth ?

7 Conclusion

We have briefly presented in this paper the design of a cell development simulator based on an accurate modeling of the cell cycle temporality. The interest of that

kind of numerical model is to provide the biologist with tools which permits to do cross-over between *in silico* and *in vivo* experiments.

Currently, our 2-D model is almost ready for the 2-D experimentation presented in section 4. We have already performed a number of experiments to test the good sequencing of the different activities and the checkpoints' responses in simple situations. These tests have been made on a single cell plugged into a virtual benchmark simulating its neighbourhood and its environmental conditions. Few elements are still under development but the 2-D simulator is near completion for proliferative simulation.

References

- [Boutros et al., 2007] Boutros, R., Lobjois, V., Ducommun, B.: CDC25 phosphatases in cancer cells: key players? Good targets? *Nature Reviews Cancer* 7(7), 495–507 (2007)
- [Chavoya and Duthen, 2008] Chavoya, A., Duthen, Y.: A cell pattern generation model based on an extended artificial regulatory network. *Biosystems* 94(1-2), 95–101 (2008)
- [Cussat-Blanc et al., 2010] Cussat-Blanc, S., Pascalie, J., Luga, H., Duthen, Y.: Morphogen positioning by the means of a hydrodynamic engine. In: *Artificial Life XII*. MIT Press, Cambridge (2010)
- [Elledge, 1996] Elledge, S.: Cell cycle checkpoints: preventing an identity crisis. *Science* 274(5293), 1664 (1996)
- [Frongia et al., 2009] Frongia, C., Lorenzo, C., Gianni, F., Prevost, G., Ducommun, B., Lobjois, V.: 3D imaging of the response to CDC25 inhibition in multicellular spheroids. *Cancer Biology & Therapy* 8(23) (2009)
- [Hoehme and Drasdo, 2010] Hoehme, S., Drasdo, D.: A cell-based simulation software for multicellular systems. *Bioinformatics* (2010)
- [Kunz-Schughart et al., 2004] Kunz-Schughart, L., Freyer, J., Hofstaedter, F., Ebner, R.: The use of 3-D cultures for high-throughput screening: the multicellular spheroid model. *Journal of Biomolecular Screening* 9(4), 273 (2004)
- [Sauro et al., 2006] Sauro, H., Harel, D., Uhrmacher, A., Hucka, M., Kwiatkowska, M., Shaffer, C., Mendes, P., Stromback, L., Tyson, J.: Challenges for modeling and simulation methods in systems biology. In: *Proceedings of the Winter Simulation Conference, WSC 2006*, pp. 1720–1730 (2006)

SOS Cloud: Self-organizing Services in the Cloud

Bogdan Alexandru Caprarescu¹, Nicolò Maria Calcavecchia²,
Elisabetta Di Nitto², and Daniel J. Dubois²

¹ West University of Timisoara

Faculty of Mathematics and Computer Science

`bcaprarescu@info.uvt.ro`

² Politecnico di Milano

Dipartimento di Elettronica e Informazione

`{calcavecchia,dinitto,dubois}@elet.polimi.it`

Abstract. Cloud computing is becoming an interesting alternative as a flexible and affordable on-demand environment for deploying custom applications in the form of services. This work proposes a bio-inspired, self-organizing solution to support the allocation and deallocation of virtual machines and the deployment of services on virtual machines in a cloud infrastructure. The goal is twofold: to meet the service level agreements and to minimize the number of required virtual machines.

Keywords: cloud computing, self-organization, autonomic computing.

1 Introduction

Cloud computing is a distributed computing paradigm with the objective of leveraging economies of scale in order to offer on-demand, flexible virtual resources. The advantage of running applications in a cloud environment is the fact that their execution is charged for the amount of resources actually used, thus reducing the cost of the initial investment and allowing applications to scale up and down in response to changing computational requirements.

A list of obstacles to the massive adoption of cloud computing was identified by [3]. The top two obstacles are particularly relevant for our work. The first one is that specific adaptation has to be provided at the application level in order to allow services to scale with the traffic demand; the second one is that services are usually bound to a single cloud provider since there are no leading interoperable standards yet. Moreover, typical solutions are centralized and thus unsuitable in contexts characterized by high dynamism, a high number of applications instances, and constraints that require fast decisions.

The SOS Cloud project aims to provide robust and scalable solutions for service deployment and resource provisioning in a cloud infrastructure. The goal is twofold: to meet the service level agreements (SLA) and to minimize the required cloud resources (i.e., virtual machines). To meet both quality and functional goals we borrow inspiration from the self-organizing systems in nature (e.g., ant

colonies, flocks of birds, school of fish) which benefit from built-in robustness and scalability and their goals emerge from the interactions among a myriad of individuals. With this idea in mind, in our approach each virtual machine will be instrumented with an autonomic layer that, through cooperation, will make decisions whether to instantiate new nodes or remove existing ones, deploy services on nodes, and route requests. Same as in biological self-organizing system, the global goals of the system are expected to emerge from local actions.

Existing bio-inspired contributions [1], [2], [6] are limited to the dynamic deployment of a set of services in a cluster with a fixed number of physical servers. Consequently, their systems cannot accommodate high traffic demands due to the limited amount of physical resources. Our work targets cloud environments and aims to provide a self-organizing algorithm that, not only allocates nodes to services, but also dynamically adjusts the number of nodes to accommodate high fluctuations in the request rate.

The remaining of this paper is organized as follows. Section 2 describes the context of this work as well as our assumptions and notations. Section 3 proposes a self-organizing solution to this problem while some preliminary simulation results are described in Section 4. Finally, Section 5 concludes the paper.

2 Context

The context of our problem involves four different actors and it is depicted in Figure 1: (i) *cloud providers* offer their computing infrastructures under the Infrastructure as a Service (IaaS) paradigm, (ii) *service providers* develop services to be deployed in the cloud and give the SLA that should be respected, (iii) the *cloud broker* deploys the services owned by many service providers on virtual machines rented from the cloud providers, finally (iv) *clients* access the services under the Software as a Service (SaaS) paradigm. Basically, the cloud broker is a company that offers Service Optimization as a Service (SOaaS) to service providers. Our long term goal is to develop a solution that allows the broker to rely on several cloud providers. However, this is outside the scope of this paper where only one cloud provider is considered.

For simplicity, we assume stateless, computation-intensive services without any form of composition; each service offers exactly one operation. Services are deployed within virtual computing environments (e.g., Amazon EC2 virtual machines). We will use the symbols s for a certain service, S for the set of deployed services and $|S|$ to denote the number of deployed services belonging to the set S . Specifically, for a service s we assume that SLA consists of two thresholds: the *maximum response time* for each request \overline{R}_s , and the *maximum rejection rate* \overline{P}_s (number of rejected requests per time unit). A service request is considered rejected whenever it is not satisfied within \overline{R}_s time units; in this case the request is discarded before the processing. The estimated processing time for service s is denoted with D_s . Either this value is given at deployment time by the service provider or it is computed at runtime. Finally, the number of requests for a service per time unit is called *request rate* and denoted with λ_s .

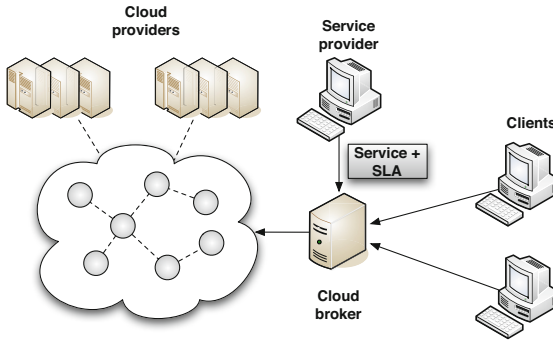


Fig. 1. Context of the SOS Cloud project

Each service may be deployed on several nodes. In this situation, we say that an instance of that service is running on each of these nodes. The cloud broker targets services with high request rates for which dozens of service instances are usually required. We assume that all nodes provided by the cloud infrastructure are homogeneous (e.g., same hardware capabilities) and that any service can run on any node. In the following, the symbol n will be used to denote a certain node. For security reasons (the services belong to different organizations) only one service is deployed on a node at a time.

3 A Self-organizing Solution

The cloud broker needs a system capable of continuously adjusting the number of nodes, the number of instances of each service, and the allocation of service instances to nodes in order to satisfy the SLA of each service and to minimize the number of nodes. Building such kind of systems that monitor and manage themselves according to certain objectives represents the goal of autonomic computing. As highlighted in our previous work [4], engineering robust and scalable autonomic systems continues to remain challenging. This is because many autonomic architectures employ a central manager that acts as a single point of failure and becomes a scalability bottleneck for large systems.

An approach to design robust and scalable large systems is to take inspiration from the self-organizing systems in nature like ant colonies. These systems are highly decentralized and their global goals emerge from the local interactions among a myriad of individuals. In our past work [5], we argued that self-organization can be successfully applied to build robust and scalable autonomic systems. We also observed that the systems composed of a high number of identical components are particularly suitable for being designed in a self-organizing manner. The broker's autonomic system exhibits this characteristic as it is composed of a large number of homogeneous nodes. Therefore, instead of having one or a few components that manage the whole system, we provide each node

with an *autonomic layer* that applies local changes. The nodes self-organize and their autonomic layers collaborate to perform local optimization. The emerging global solution is usually not the optimal one, but the high robustness and scalability of the architecture pay the trade-off. In this section, we briefly describe the architecture and the self-organizing algorithm of the broker’s autonomic system.

Like ants, our nodes have only a local view of the overall system. In other words, each node knows about and communicates with a limited number of other nodes in the system (called from now on the *neighbors* of that node). The nodes form $|S| + 1$ overlays: one overlay for each service and a system overlay which connects all nodes. Consequently, each node is simultaneously part of its service overlay and the system overlay. For example, Figure 2 depicts the overlays of a system consisting of two services (named $s1$ and $s2$) deployed on nine nodes. The number of neighbors of each node in an overlay is the same and is called the degree of the overlay. As described below, the service overlays are needed for request routing and node provisioning while the system overlay is used only for service switching (i.e., the current offered service of a node is changed with another one without restarting the virtual machine). In the following, by *neighborhood* of a node we refer to the set composed of the node itself and its neighbors in a given overlay. As a node is part of two overlays, it is also part of two neighborhoods: service neighborhood and system neighborhood.

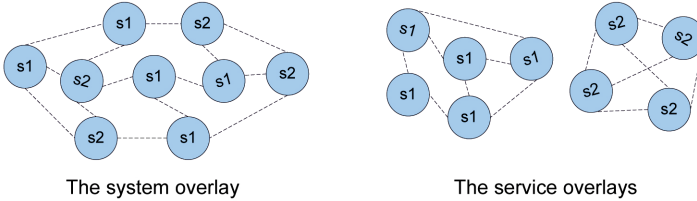


Fig. 2. Example of system and service overlays

From the architectural perspective, a node runs a service and is instrumented with an autonomic layer that executes three decentralized mechanisms: service optimization, overlays management, and request routing. Service optimization is the most important function of the autonomic layer. To fulfill it, the autonomic layer uses a feedback loop with three main activities: monitoring, analysis and decision, and execution. The autonomic layer stores performance data for the current node and for its neighbors in the service and system overlays. This data is then periodically analyzed and a decision is made whether a modification is needed. There are three possible modifications: allocate a new node in the cloud, remove the current node from the cloud, and change the service of the current node. In the following, the three feedback loop activities are described.

The autonomic layer monitors two parameters: the CPU utilization L_n and the number of rejected requests P_n . The CPU utilization is computed as the percentage of the number of requests processed by the application server in a

time frame (called monitoring time frame) out of the total number of requests that could have been processed by the application server in the monitoring time frame (computed at its turn based on the estimated service processing time D_s). The number of rejected requests is computed over the same time frame. The autonomic layer also maintains one management table for each overlay. Thus, for each neighbor in one overlay, the management table of that overlay stores an entry with the following information: data needed to communicate with that neighbor (e.g., IP address and port number), CPU utilization, number of rejected requests, and the last update time. The purpose of the management tables is twofold: to maintain the overlays topology and to store performance data of neighboring nodes. In order to keep the management tables updated we use a gossip protocol based on T-Man [7]. More specifically, we customized T-Man in two ways. First, in order to increase the stability of the topology, when updating a management table we try to keep many of the existing neighbors. Second, we make sure that a new node is inserted in the service neighborhood of the node that decided to create it.

The autonomic layer needs a way to figure out how well the service neighborhood is performing with respect to the goals of the system. For doing that, a common practice in autonomic computing is to define a utility function that realizes a trade off among goals. The first goal of our system is to respect SLA, that is to minimize the number of rejected requests. In order for this goal to be measurable we define a reward function U^{SLA} that associates a numeric reward to each node based on the maximum allowed number of rejected requests defined in the SLA and the monitored number of rejected requests.

The second goal is to minimize the number of nodes. As a rule, the more efficiently the hardware resources of each node are utilized the less nodes are required. In this paper we assume that services are computation-intensive and focus on optimizing only one resource, namely the CPU. Normally, the minimum number of nodes is achieved when the CPU of all or most of the nodes is fully utilized. However, it is not a good practice to keep the nodes overloaded because, in case of a sudden increase in the request rate, the system would reject many requests before the autonomic layer has the chance to allocate new nodes. Therefore, we define a desired CPU loading L^{des} and a reward function U^{CPU} that associates a numeric reward to a node based on how close the current loading is from the desired loading. We say that a node is lightly utilized if the current loading is lower than the desired loading. Otherwise, it is heavily utilized.

The utility function can be further defined as a weighted average of the reward functions as shown in equation (1).

$$U(n) = w^{SLA} \cdot U^{SLA}(n) + w^{CPU} \cdot U^{CPU}(n), \text{ where } w^{SLA} + w^{CPU} = 1 \quad (1)$$

The utility can be computed not only at node level, but also for a neighborhood or even for the entire system as the average utility of all nodes in that neighborhood or system, respectively. However, optimizing the utility of a large system at the global level is usually an inefficient solution that may prevent the system

from responding to traffic fluctuations in a timely manner. Therefore, we opted for a self-organizing algorithm where each node makes those decisions (i.e., add a new node, change its service or remove itself) that maximize the average utility of its service and system neighborhoods. To avoid conflicts, a dynamic and decentralized election algorithm allows only one member of a service neighborhood to execute changes at a time. In other words, while a node is effecting a change its service neighbors are forbidden to also execute changes.

Depending on the level of node utilization, a different algorithm is used for the analysis, decision, and execution activities of the autonomic layer. Thus, if the node is heavily utilized, the current average utility of the service neighborhood is compared with the predicted utility in the situation that a new node is added. If the predicted utility is higher, then a new node is allocated in the cloud. The prediction algorithm assumes that the new node will work at desired loading and will take over an equal amount of traffic from each existing neighbor.

The autonomic layer of a lightly utilized node estimates the utility of the neighborhood in the case the current node is removed. The estimation is done based on the assumption that the traffic processed by the removed node is equally distributed to its service neighbors. Here, by amount of traffic we mean the sum of processed and rejected requests. If the estimated utility is higher than the current utility, then, before removing itself, the node tries to switch to another service. The service switching mechanism works in the following way: for each service that is being run by its system neighbors, the autonomic layer estimates the utility of the system neighborhood in the event it would switch to that service. The node switches to the service that maximizes the predicted system neighborhood utility. But, if all predicted utilities are lower than the current utility, the node is deallocated from the cloud.

For request routing we use the simple algorithm proposed by [1]. First of all, we assume that the requests for a service are randomly distributed to the nodes of that service overlay. Then, once a request arrives at a node, based on the current loading and the estimated time to process the request D_s , the autonomic layer decides whether the SLA's maximum response time \overline{P}_s can be met. If it can, then the node schedules the request on its internal queue. Otherwise, the request is forwarded to the least loaded neighbor in the service overlay. Finally, a request is rejected if there is no chance to handle it in the required time even by an unloaded node because of the time lost in the routing process.

4 Preliminary Results

A custom simulator was implemented in Java. The decentralized functions of each node (service optimization, request routing, and topology maintenance) are carried out by a couple of threads. A custom, asynchronous, message-based mechanism is used for implementing the inter-node communication. The simple utility function used in the simulator is shown in equations (2), (3), and (4). In equation (3), the definition of U^{SLA} uses the constant C to provide a dominant penalty in the situation the current number of rejected requests becomes higher

than the double of the maximum allowed number of rejected requests. Other parameters of the simulator include the monitoring time frame (10 seconds) and the desired CPU loading ($L^{des} = 80\%$). The analysis is done every 0.5 seconds and both system and service overlay degrees are set to 10.

$$U(n) = \frac{U^{SLA}(n) + U^{CPU}(n)}{2} \quad (2)$$

$$U^{SLA}(n) = \begin{cases} \frac{\bar{P}_s - P_n}{\bar{P}_s} & \text{if } P_n < 2 \cdot \bar{P}_s \\ C & \text{otherwise} \end{cases} \quad (3)$$

$$U^{CPU}(n) = \begin{cases} \frac{L_n}{L^{des}} & \text{if } L_n < L^{des} \\ \frac{100 - L_n}{100 - L^{des}} & \text{otherwise} \end{cases} \quad (4)$$

Our preliminary tests addressed the provisioning of nodes as the effectiveness of a self-organizing service selection was proved by [1]. A simple test configuration of one client and one service deployed on one node was instantiated and the client was instructed to progressively send more requests. Figure 3 shows the variations of the system average utility as a response to traffic bursts. Thus, the client began by sending 500 requests per minute at a constant rate while a node at desired loading can process 96 requests per minute. The system responded by adding 4 new nodes, which raised the utility over 90. After 20 seconds, the request rate was increased to 2500 requests per minute. The number of nodes reached 25. Finally, after 60 seconds from the beginning, the request rate was set to 5000 requests per minute. As the end, 52 nodes were count.

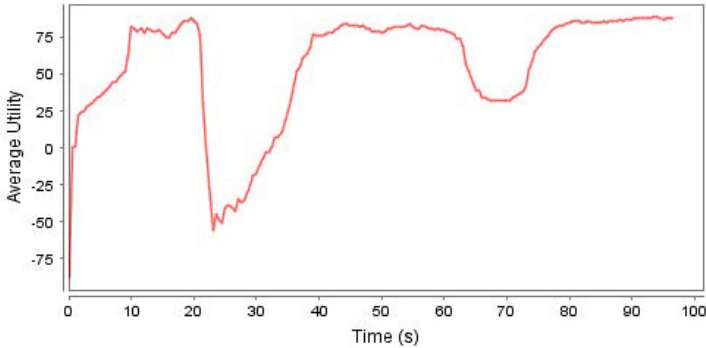


Fig. 3. System average utility (normalized between -100 and $+100$)

In conclusion, it was observed that the system is stable at constant request rates. When a traffic burst occurs, the utility initially drops, but quickly recovers after the allocation of new nodes. It is important to be aware that the decision of adding a new node can be made by any existing node. Moreover, while an existing node is adding a new node, its service neighbors are locked in the sense that they are not allowed to add other nodes.

5 Conclusion

In this paper, we have proposed an innovative self-organizing approach for service deployment and resource provisioning in a cloud infrastructure. The additional benefit of our approach, compared to other existing approaches, is the fact that we take advantage of the “elasticity” of the cloud in the sense that new resources may be allocated and deallocated to help services respect contractual SLAs. The proposed solution has been simulated and the preliminary results have shown that, after a transitory, services comply with their SLAs and the number of cloud nodes is comparable to the one obtained in the optimal solution (i.e., the solution that maximizes the utility function). Future work will include more precise and accurate simulations as well as a complete description of the architecture and algorithms of our system.

Acknowledgments. This research has been partially funded by the European Commission, Programme IDEAS-ERC, Project 227077-SMScom, and grant FP7-ICT-2009-4-246839 (SPRERS).

References

1. Adam, C., Stadler, R.: A Middleware Design for Large-scale Clusters Offering Multiple Services. *IEEE Transactions on Network and Service Management* 3(1), 1–12 (2006)
2. Andrzejak, A., Graupner, S., Kotov, V., Trinks, H.: Algorithms for self-organization and adaptive service placement in dynamic distributed systems. HP Laboratories Palo Alto, HPL-2002-259 (2002)
3. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. *Communications of the ACM* 53(4), 50–58 (2010)
4. Caprarescu, B.A.: Robustness and scalability: a dual challenge for autonomic architectures. In: *Proceedings of the Fourth European Conference on Software Architecture: Companion*, pp. 22–26 (2010)
5. Di Nitto, E., Dubois, D.J., Mirandola, R.: On exploiting decentralized bio-inspired self-organization algorithms to develop real systems. In: *Proceedings of the International Workshop on Software Engineering for Adaptive and Self-Managing Systems*, pp. 68–75 (2009)
6. Jamjoom, H., Jamin, S., Shin, K.: *Self-Organizing Network Services*. University of Michigan, CSE-TR-407-99 (1999)
7. Jelasity, M., Montesor, A., Babaoglu, O.: T-Man: Gossip-based fast overlay topology construction. *Computer Networks: The International Journal of Computer and Telecommunications Networking* 53(13), 2321–2339 (2009)

Biomimicking the Formation of Nacre/Shell: One Step Forward

Feisal Khoushab¹, Montarop Yamabhai^{1,*}, and Kenneth J. Haller²

¹ School of Biotechnology, Suranaree University of Technology, Nakhon Ratchasima, Thailand

² School of Chemistry, Suranaree University of Technology, Nakhon Ratchasima, Thailand
montarop@g.sut.ac.th, fl.khoushab@gmail.com

Abstract. Chitin is biocompatible and biodegradable, with antimicrobial activity and low immunogenicity, and the second most abundant biopolymer after cellulose. It is one of the necessary components of the formation of nacre/shell. So harnessing behavior of chitin not only would lead to mimics of these structures but also to potential applications in academic research and industry. A peptide induces a unique chitin-based gel formation. Then, in the presence of CaCO₃ a flexible structure was obtained. This structure and complex has many potential applications for different applications such as development of bioinspired instruments, or complexes, nanotechnology etc. It is early step toward mimicking the structure of nacre/shell.

Keywords: Bioinspired, nanotechnology, mimicking, nacre/shell, peptide, chitin.

1 Introduction

Chitin is biocompatible and biodegradable, with antimicrobial activity and low immunogenicity, and the second most abundant biopolymer after cellulose. It is one of the necessary components of the formation of nacre/shell [1], [3]. So harnessing behavior of chitin not only would lead to mimics of these structures but also to potential applications in academic research and industry [2].

2 Method

2.1 Phage displaying peptide that can bind specifically to chitin was obtained by affinity selection from a phage display combinatorial peptide library. The selected peptide was chemically synthesized, dissolved in water and a 0.7 mM solution was prepared. Chito-hexose was prepared at 5 mM for gel-formation analysis. Peptide and chito-hexose, 100 μ l each, were mixed and incubated at room temperature for 2 hrs before sampling for transmission electron microscopy (TEM) analysis, and at 42°C for 15 hrs for scanning electron microscopy (SEM) analysis, respectively.

2.2 Then, a 0.7 mM aqueous solution of the selected peptide was prepared. 100 μ L of 5 mg/mL CaCO₃ was added to 100 μ l of 5 mM chito-hexose followed by addition of

* Corresponding author.

100 μl of the synthesized peptide solution. This mixture was incubated at 42°C with shaking (350 rpm) for 15 hrs. The formed material was washed and used for scanning electron microscopy (SEM) photography.

3 Results

3.1 The selected 12-amino acid peptide contains charged residues. It could induce a unique pattern of gel-formation as shown in Figures 1 and 2.

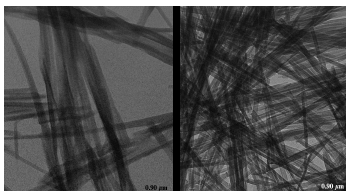


Fig. 1. TEM photographs of peptide-chitohexose vs control, after 2 hrs incubation at room temperature. Left: Control (contains only chitohexamer), right: peptide-chitohexose.

3.2 Then, a rubber-like material was obtained (Fig 3 and 4). The peptide itself is 12 residues long with charged residues at both ends and hydrophobic residues through the middle part.

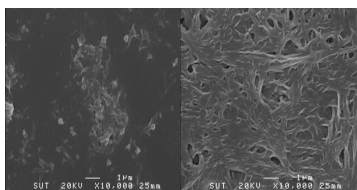


Fig. 2. SEM photographs after 15 hrs incubation at 42 °C. Left: Control contains only chitohexose vs Right: Chitohexose with the peptide.

In the presence of chitin alone CaCO_3 forms rosette-like aggregates (Figure 3 Left), a form which is lost in the presence of the peptide (Figure 3 Right).

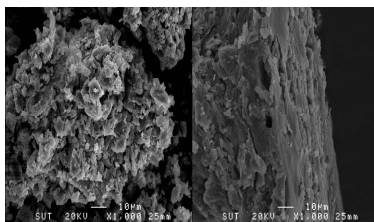


Fig. 3. SEM photographs after 15 hrs incubation at 42°C with shaking (350 rpm). Left: CaCO_3 -chitohexamer; Right: CaCO_3 -chitohexamer-peptide.

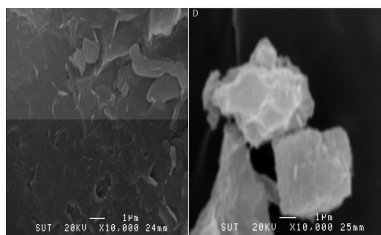


Fig. 4. SEM photographs of, Left (A), Figure 3B at higher magnification and, right (B), crystals of CaCO_3 formed in the presence of the peptide

4 Discussions and Conclusion

4.1. Results indicate that a specific chitin binding peptide obtained from applying phage display technology can induce a unique chitin-based gel formation. After incubating a 1:7 molar ratio of peptide and chitohexose at 42°C for 15 hrs, a porous structure with nano-size holes could be formed. These results suggest that a short charged peptide is able to bind to chitohexose and change its behavior toward a well-ordered shape.

4.2. This structure (Figs. 3 and 4) shows high flexibility with rubber-like characteristics. CaCO_3 forms typical rhombohedral crystals, which may indicate that the peptide interrupts the interaction between chitin and CaCO_3 crystals allowing them to grow differently. The results of this research have potential applications in several fields such as bio-nanotechnology, drug delivery, implantable devices, etc. More research needs to be done to answer the following questions; how can mineralization be harnessed in nano- and/or micro-level, or basically, how to direct the spatiotemporal deposition of macromolecules in solution to produce structures with different functions for different applications such as development of bioinspired instruments, or complexes.

References

1. Drury, J.L., Mooney, D.J.: Hydrogels for tissue engineering: Scaffold design variables and applications. *Biomaterials* 24, 4337–4351 (2003)
2. Pillai, C.K.S., Paul, W., et al.: Chitin and chitosan polymers: Chemistry, solubility and fiber formation. *Progress in Polymer Science* 34, 641–678 (2009)
3. Raabe, D., Romano, P., et al.: Microstructure and crystallographic texture of the chitin–protein network in the biological composite material of the exoskeleton of the lobster *Homarus americanus*. *Materials Science and Engineering A* 421, 143–153 (2006)

Algorithmically Transitive Network: A Self-organizing Data-Flow Network with Learning

Hideaki Suzuki¹, Hiroyuki Ohsaki², and Hidefumi Sawai¹

¹ National Institute of Information and Communications Technology
588-2, Iwaoka, Iwaoka-cho, Nishi-ku, Kobe, 651-2492, Japan
{hsuzuki, sawai}@nict.go.jp

² Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka, 565-0871, Japan
oosaki@ist.osaka-u.ac.jp

Abstract. A novel non-von Neumann computational model named “Algorithmically Transitive Network” (ATN) is presented. The ATN is a data-flow network composed of operation nodes and data edges. The calculation is propelled with node firing and token creation on the edges. After it finishes, teaching values are given to the answer nodes and an energy function is evaluated, which causes backward propagation of differential coefficients with respect to token variables or node parameters. The network’s topological alteration takes place based on these calculation/learning processes, and as a result, the algorithm of the network is refined. The basic scheme of the model is explained, and some experimental results on symbolic regression problems are presented.

Keywords: data-flow network, learning, neural network, back-propagation, artificial chemistry.

1 Introduction

For a long time, network and computation have been in a close relationship with each other in several disciplines of information sciences. Back in the 1970s to 1980s, the ‘data-flow computer’ (DFC) [18] was studied in many institutes in the world in the hope that parallel algorithms represented by the ‘data-flow network’ might remedy the ‘bottleneck’ problem which a von Neumann computer suffers from. Today, the DFC is neither a commercial-based computer nor a hot research topic, partly because the DFC’s algorithm expressed as a network is difficult for a human to design or maintain. A network is a very natural way to represent a computational algorithm, but to utilize the data-flow network, we might have to incorporate an additional function to maintain programs.

Another famous research approach on network-based computation is ‘artificial neural networks’ [3]. Unlike the data-flow network, an artificial neural network consists of nodes with (quasi-)homogeneous functions modeled by McCulloch-Pitts [11] or Hodgkin-Huxley [4]. An algorithm obtained through learning is indirectly and distributedly coded in the weight vectors for synapses. For this reason, it is hard or impossible for a human to analyze an established algorithm in the artificial neural network.

Though most models for the artificial neural network are primarily focused on the cellular activities in the brain, a natural neuron is, of course, made up of a huge

number of biomolecules from a nanoscopic point of view. These molecules govern all of the activities in a cell, which makes a cell behave dynamically on signal transmission processes. To make a more realistic model for the neurons, it might be better to implement molecular agents (agents with active functions) in artificial neurons and make them move around the neural network, giving rise to a functional change in the neurons.

One such network-based computational model with molecular agents is ‘Modified Network Artificial Chemistry’ invented by Suzuki [21]. In association with this model, Suzuki [21] proposed a paradigm of ‘program-flow computing’, wherein programs (agents) move from node to node, bringing different functions to CPUs (nodes). This is also closely related to ‘active network’ [26] proposed in the 1990s in the area of computer network. The active network enables a router (node) to have various functions by delivering packets (agents) with encapsulated programs.

Based upon these studies, very recently, the authors presented a new concept for computation and learning, named “Algorithmically Transitive Network (ATN)” [22,23,24]. The distinctive features of the ATN are summarized as:

- A program is represented by a ‘data-flow network’ whose nodes execute arithmetic/logic operations and edges transmit data, like the DFC.
- After the calculation, triggered from the teaching signals, the network transmits data backward and revises network parameters, like the artificial neural network.
- The network topology (algorithm) can be modified/improved through execution of movable agents’ programs.

The ATN’s learning process is primarily done by the famous ‘back-propagation’ (BP) learning. Since its first proposal by Werbos and Rumelhart *et al.* [15,16,27], the BP has been successfully applied to a number of real world problems and has been one of the most widely-used learning algorithms among the artificial neural network researchers [17]. Like the BP in the artificial neural network, the BP in the ATN uses the steepest descent method to revise network parameters. A similar idea was also mentioned by Kumazawa [9] who considered the possibility of using the BP to train an ‘operation network’. In both the operation network and the ATN, the node operations transmit signals from inputs to outputs so directly that the learning coefficient should be adjusted more minutely in the ATN than in the artificial neural network. The paper presents a formula for the learning coefficient, as well.

The ATN’s basic concepts [22,23] and an initial experimental result [24] were briefly given in the previous reports. Following these works, this paper presents the full description of the model’s framework and experimental results to reveal the performance of the ATN on a class of symbolic regression problems. In the following, Section 2 explains detailed design of the model, and Section 3 presents the experimental results. In Section 4, concluding remarks and discussion on the model’s meanings and future possibility are given.

2 Method

2.1 Data-Flow Network

As in the DFC, the ATN’s nodes read the input ‘tokens’ on their incoming edges, fire, and create the output tokens on their outgoing edges during calculation. This constructs

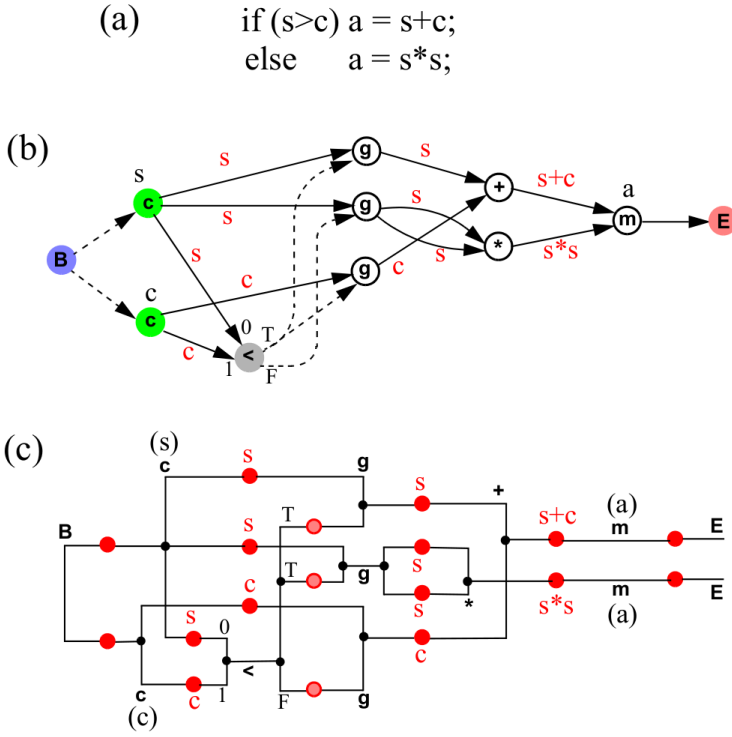


Fig. 1. (a) Higher language program of a simple conditional branch, (b) data-flow network (ATN), and (c) fire-token pedigree produced by the calculation. The variable name s represents the graph’s input (sensor) signal, and the a represents the graph’s output (answer) value. The pedigree’s top ancestor is the initial fire at the ‘B’ node, and its last descendants are firings at the ‘E’ node or nodes with no outgoing edge. In (b), arithmetic edges are expressed as the solid arrows, and regulating ones are expressed as the broken arrows.

a ‘fire-token pedigree’ whose nodes (tokens) represent variables or mathematical expressions in the original program, and whose hyper-edges (firings) represent arithmetic/logical operations used to create the tokens. An example of these relationships is shown in Fig. 1. As explained below, the pedigree is used for the BP learning. The node operations used in this paper are listed in Table 1.

2.2 Simulation Procedure

An ATN simulation proceeds as follows:

Step (1) [Initialization]. Create an initial network randomly or through the transformation of a higher language program. (More arguments on this matter are given in Section 4.)

Step (2) [Calculation]. Substitute new sensor value(s) for the s node’s $v(s)$ and make the Begin node fire. Repeat a forward propagation (FP)’s time step operation that

Table 1. Node operations

Name	Oper. code	Input num.	Arith. /Regu.	X	R
Begin	B	0	R	—	1
End	E	1(a)	R	—	—
Negative	n	1(a)	A	$-x_0$	r_0
Inverse	i	1(a)	A	$1/x_0$	r_0
Add	+	2+	A	$\sum x_i$	$\min(r_i)$
Multiply	*	2+	A	$\prod x_i$	$\min(r_i)$
Subtract	—	2	A	$x_0 - x_1$	$\min(r_0, r_1)$
Divide	/	2	A	x_0/x_1	$\min(r_0, r_1)$
Less than	<	2	R	—	$R_0 = \sigma \cdot \min(r_0, r_1)$ $R_1 = (1 - \sigma) \cdot \min(r_0, r_1)$
Greater than	>	2	R	—	$R_0 = (1 - \sigma) \cdot \min(r_0, r_1)$ $R_1 = \sigma \cdot \min(r_0, r_1)$
Equal to	==	2	R	—	$R_0 = (1 - \delta) \cdot \min(r_0, r_1)$ $R_1 = \delta \cdot \min(r_0, r_1)$
Not equal to	!=	2	R	—	$R_0 = \delta \cdot \min(r_0, r_1)$ $R_1 = (1 - \delta) \cdot \min(r_0, r_1)$
Logical AND	A	2+	R	—	$\min(r_i)$
Logical OR	O	2+	R	—	$\max(r_i)$
Logical NOT	N	1(a)	R	—	$1 - r_0$
Gate	g	2	A	x_0	$\min(r_0, r_1)$
Merge	m	1(a)	A	x	r
Arith. constant	c	1(a)	A	v	r
Regul. constant	C	1(a)	R	—	$R_0 = u, R_1 = 1 - u$

The operations are classified as asynchronous (E, n, i, N, m, c, and C) or synchronous (the others). ‘(a)’ in the third column represents ‘asynchronous’. An asynchronous node fires every time a token is created on any incoming edge, whereas a synchronous node fires only when the tokens are created on all the incoming edges. ‘2+’ in the third column means that the node can have two or more incoming edges. $\sigma \equiv \text{sig}(\kappa\beta_r(x_0 - x_1))$ where $\text{sig}(z) \equiv \frac{1}{1 + \exp(-z)}$, and $\delta \equiv \text{delta}(\kappa\beta_r(x_0 - x_1))$ where $\text{delta}(z) \equiv 4\text{sig}(z)\text{sig}(-z)$. Note that all the X and R ’s functions have differentiable formulas. The variables v (for c), u (for C), and β_r (for judging operations <, >, ==, and !=) are the node parameters adjusted by the learning. The inverse temperature coefficient κ is a predefined constant.

makes nodes fire, until there is no node able to fire. Calculate the final answer of the network.

Step (3) [Learning]. Calculate partial differential coefficients in tokens on the a ’s outgoing edges (the last descendant tokens in the pedigree). Repeat the BP’s time step operation that ‘extinguishes’ the firings, until the firing on the Begin node is extinguished. Calculate the learning coefficient η and revise node parameters.

Step (4) [Topological Reformation]. Conduct the agents’ graph modifying operations. Move agents to the next nodes/edges if required.

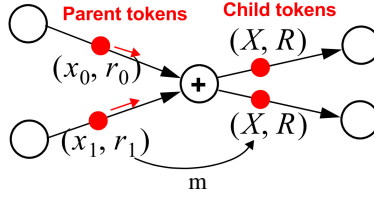


Fig. 2. The FP's unit process in the data-flow network: firing of a '+' node

Step (5) [Supplying Agents]. Randomly choose nodes and supply agents to them at some constant rates.

Step (6) [Termination or Recursion]. If a certain termination condition is satisfied, stop the simulation. Otherwise, go to Step (2).

2.3 Calculation by the Forward Propagation (FP)

In the current implementation, each token has a real value vector (x, r) delimited by $-\infty < x < \infty$ and $0 \leq r \leq 1$, where x is the arithmetic value for the calculation, and r is the regulating value that represents the probability of the token itself existing in the network. When a node fires, the output vector (X, R) is calculated from the operand vector(s) (x_i, r_i) s of the input token(s) using the functions in Table I. For example, if a '+' node fires, it calculates $X = \sum_i x_i = x_0 + x_1$ and $R = \min_i(r_i) = \min\{r_0, r_1\}$, and creates tokens with (X, R) on all the outgoing edges (Fig. 2).

To make the network learnable, the ATN's judgment nodes (such as '<') give a 'fuzzy' result. See the formulas for R_0 and R_1 in Table I. If x_0 is much smaller than x_1 in a '<' node for example, we get $R_0 = 0$ and $R_1 = 1$, hence, we create a token only on the outgoing edge with label 'T' (we 'kill' the token on the 'F' edge). If x_0 is similar to x_1 , on the other hand, we have both positive R_0 and R_1 , hence, we create tokens on both outgoing edges with labels 'T' and 'F'.

This fuzzy judgment reproduces tokens in judgment nodes, and in general, after the FP, an a node has two or more firings whose child tokens have vector (x_j, r_j) s, where j is the firing number at the a node. With these, we calculate the resultant answer value as

$$a = \frac{\sum_j x_j r_j}{\sum_j r_j}. \quad (1)$$

2.4 Learning by the Backward Propagation (BP)

The ATN's learning is conducted by propagating differential coefficients vector (D_x, D_r) from child tokens to parent tokens in the token-fire pedigree. This begins with the calculatin of the energy function

$$E = \frac{1}{2}(t - a)^2 + \mu \left(1 - \sum_j r_j\right)^2 + \nu \sum_j r_j^2 (1 - r_j)^2 \quad (2)$$

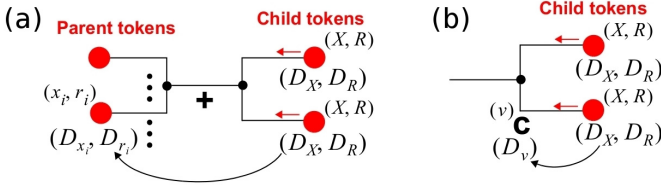


Fig. 3. The BP's unit processes in the fire-token pedigree: extinguishing of (b) a '+' fire and (c) a 'c' fire

at the **a** node. Here, t is a teaching signal given from the outside, a is a value calculated from Eq. (II), and μ and ν are predefined constants. Again, the summation for j is taken for all the firings at the **a** node. The μ - and ν -terms are the 'penalty' terms added in order to make the sum of $\{r_j\}$ be one and make each r_j close to zero or one, respectively. By partially differentiating Eq. (2), we can calculate differential coefficient vector $(D_{x_j}, D_{r_j}) = (\frac{\partial E}{\partial x_j}, \frac{\partial E}{\partial r_j})$ of the **a** node's output tokens. (Through this paper, D_{\square} signifies the energy function E 's partial differential coefficients with respect to \square .)

Once (D_X, D_R) is obtained for all the child tokens of a firing (here, (D_x, D_r) is expressed as (D_X, D_R) to indicate that it is for a child token), the firing is extinguished and the parent tokens' (D_{x_i}, D_{r_i}) is calculated with

$$D_{x_i} = \sum_{\text{children}} \left(D_X \frac{\partial X}{\partial x_i} + D_R \frac{\partial R}{\partial x_i} \right), \tag{3a}$$

$$D_{r_i} = \sum_{\text{children}} \left(D_X \frac{\partial X}{\partial r_i} + D_R \frac{\partial R}{\partial r_i} \right). \tag{3b}$$

$\frac{\partial X}{\partial x_i}$, $\frac{\partial R}{\partial x_i}$, $\frac{\partial X}{\partial r_i}$, and $\frac{\partial R}{\partial r_i}$ are calculated from Table I. The summation ' \sum_{children} ' is taken for all the child tokens of the fire (Fig. 3(a)).

Specifically, when a firing is extinguished at a node with parameter $\{v\}$, $\{u\}$, or $\{\beta_i\}$ (which are all expressed as $\{z\}$ hereafter), we also calculate the partial differential coefficient of E with respect to z using

$$D_z = \sum_{\text{children}} \left(D_X \frac{\partial X}{\partial z} + D_R \frac{\partial R}{\partial z} \right) \tag{4}$$

(Figure 3(b)). $\frac{\partial X}{\partial z}$ and $\frac{\partial R}{\partial z}$ are also calculated from Table I. The chain rule (differentiation rule for the composite function) ensures that D_z evaluated from Eq. (4) gives the partial derivatives of the original energy function E [3].

After D_z is obtained at all of the **c**, **C**, and judging firings, we finally revise the node parameters using the steepest descent method as:

$$z \rightarrow z - \eta \frac{\partial E}{\partial z} = z - \eta D_z. \tag{5}$$

Here, with a predefined constant η_{lp} , we require that the revision by a one-pass propagation (a pair of the FP and BP) make E become $1 - \eta_{lp}$ times the former value as:

$$\begin{aligned}
 (1 - \eta_{lp}) \cdot E(\{z\}) &= E\left(\left\{z - \eta \frac{\partial E}{\partial z}\right\}\right) \\
 &\simeq E(\{z\}) - \eta \sum_z \left(\frac{\partial E}{\partial z}\right)^2 \\
 \therefore \eta &= \frac{\eta_{lp} \cdot E}{\sum_z \left(\frac{\partial E}{\partial z}\right)^2}. \tag{6}
 \end{aligned}$$

The linear approximation of the Taylor expansion of $E(\{z\})$ was used. The convergence of the parameter learning by Eq. (5) is ensured by Eq. (6).

2.5 Topological Reformation

The topological reformation of an ATN is conducted by agent operations listed up in Fig. 4. A CON (constantification) agent changes the node operation into c or C (Fig. 4(a)), a DIV (division) divides a constant node into two (Fig. 4(b)), a BRG (bridge) constructs a bridge between an edge and a node (Fig. 4(c)), a MKV makes a new variable node (Fig. 4(d)), a MGT merges two adjacent addition/multiplication operations (tuples) (Fig. 4(e)), and a MGN merges two or more constant operand nodes within an addition/multiplication operation (Fig. 4(f)). Among these, CON, MGT, and MGN simplify (reduce the node/edge numbers of) the graph, whereas the others complexify the graph.

Though not shown in Fig. 4, the seventh agent WAR rewires the background ‘wa’ edges which are compared to ‘van der Waals’ bonds known as the weakest interaction between bio-molecules [19,20,21]. Some agent operations except for CON and DIV are concerned with two or more nodes. One way to do such inter-node operations is that an agent moves through the network and collects information by themselves, but in the current implementation, most agents stay at nodes and gather information through ‘wa’ edges created by the WAR agents which move around the network in lieu of them.

In what follows, we take CON, DIV, and BRG and explain their detailed operations.

A CON agent usually stays at a variable node. During the stay, it observes (x, r) of the firings created at the node, and if the value does not change for a long time, it changes the node operation into c or C depending on whether the node operation is arithmetic or regulating. At the same time, the incoming edges of the node are cut except for an edge with the minimum sum of the past r values which is conserved to make the node ‘firable’.

A DIV agent usually stays at an arithmetic/regulating constant node with two or more outgoing edges. It observes (D_x, D_r) of firings at the node for a long time, and if they contradict each other, divides the node into two. After this division, v (or u) of the divided nodes are able to learn toward different directions.

A BRG agent usually stays at an edge. It randomly chooses a neighboring node ‘a’ and constructs a bridge between the current edge and node ‘a’ as shown in Fig. 4(c).

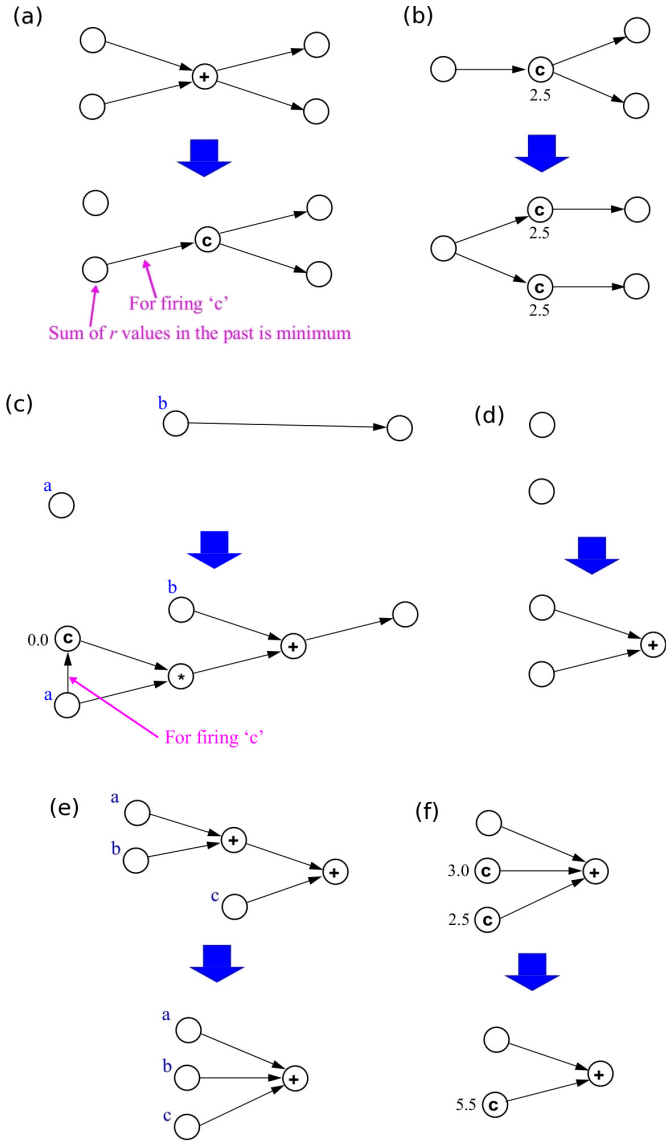


Fig. 4. Graph modifying operations by agents

Three nodes with the operations $*$, $+$, and c (with $v = 0.0$) are newly created for the bridge. Since $a * 0.0 + b = b$, this modification does not change the calculation result; however, after this operation, differential coefficients propagated to the newly created c node gradually changes v , which makes node 'a' affect the calculation result.

Through all the graph reformation processes, the current implementation prohibits a 'loop' from being created in the ATN. The future possibility of loosening this constraint is discussed in Section 4.

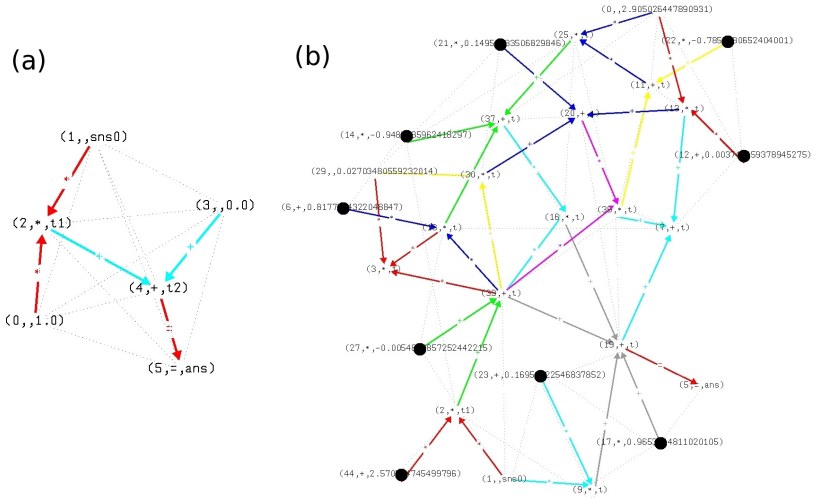


Fig. 5. (a) Initial ATN and (b) final ATN (after 80,033 time steps) obtained for a regression problem for the one-variable quadratic function (Eq. (7a)). The graphs are drawn with a commercial software named aiSee [11]. The final ATN produces the a value with a cubic function $a = 0.965 + (0.169s) + (-0.00548 + 2.57s) + ((-0.00548 + 2.57s)(-0.948 + (0.817(-0.00548 + 2.57s)) + (2.90(-0.785 + ((-0.00548 + 2.57s)(0.149 + (0.00374 * 2.90) + (0.0270(-0.00548 + 2.57s))))))$). The result was obtained from a one-minute simulation run.

3 Experiments

To demonstrate the ATN's basic capability to explore algorithms, here we apply the ATN to some symbolic regression problems. The simulation constants are taken to be: $\mu = 1.0$, $\nu = 1.0$, $\beta_{r\text{-init}} = 0.3$ (β_r 's initial value), $\kappa = 100.0$, $\eta_{ip} = 0.5$, $N_{\text{tgt}} = 300$ (target node number), $M_{\text{wa-tgt}} = 5.0$ (**wa** edge's target degree per node), $\lambda = 1.0$ (mean free path for rewiring **wa** edges; a **wa** edge is joined to a closer node with a smaller this value [20]), $T_{\text{KrUcCON}} = 20$ (number of passes until a node is judged to be constant), $r_{\text{diff-ngl}} = 1.0 \times 10^{-4}$ (threshold of r under which tokens are killed), and $T_{\text{watchStblty}} = 10$ (number of passes until DIV decides to split a node). The simulation program is implemented in Java and is run on a standard desktop computer with Intel Duo processor (1.86GHz).

3.1 Polynomial Functions

We prepare one-variable quadratic, cubic, and quartic functions

$$t = 1 - 6s + 10s^2 \quad (7a)$$

$$t = -0.5 + 11s - 35s^2 + 28s^3 \quad (7b)$$

$$t = 0.2 + 35s - 175s^2 + 300s^3 - 160s^4 \quad (7c)$$

within the domain $0 \leq s \leq 1$ as targets, and examine the ATN's learnability.

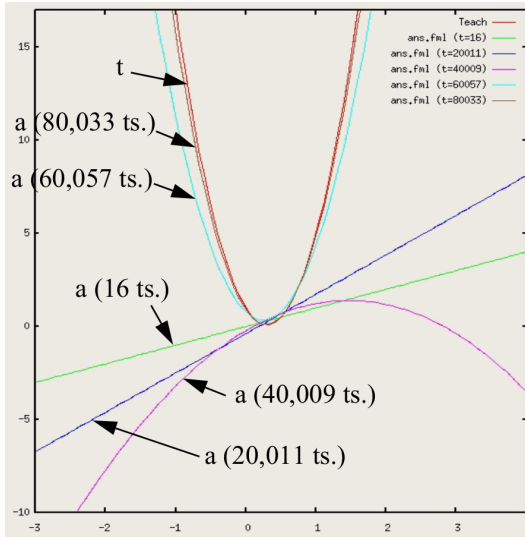


Fig. 6. The s - a (sensor-answer) and s - t (sensor-teach) plots for the run in Fig 5

Figures 5 and 6 show a representative result for the quadratic function. In this experiment, we start from a handmade 6-node ATN that represents a linear function $a = 0 - s$ (see Fig. 5(a)), then after 80,000 time steps (about 800 passes), we obtain a 39-node 158-agent network (Fig. 5(b)). Figure 6 shows the change of the s - a (sensor-answer) plot during this run. We can see from this figure that the final function of the ATN almost perfectly agrees with the target function within the domain. Results for ten different runs plotted in Fig. 7 show that nine out of ten runs succeeded in finding desirable functions for this regression problem.

Using the same method, we also tested the ATN’s learnability with the cubic and quartic functions (Eqs. (7b) and (7c)). The representative results are shown in Fig. 8. For both functions, we tested hundred runs and ten runs with different random number sequences and found that 60% and 80% runs succeeded in creating the desirable functions, respectively.

3.2 Fraction Function

Next we apply the ATN to symbolic regression of a ‘fraction’ function defined as

$$t = \frac{11 - 62s + 88s^2}{6.5 - 32s + 40s^2} \quad (8)$$

within the domain $0 \leq s \leq 1$.

We tested ten runs using different random number sequences, but all the runs failed to create solution networks that produce the desirable answer for this much more difficult function. Figure 9 shows a representative result.

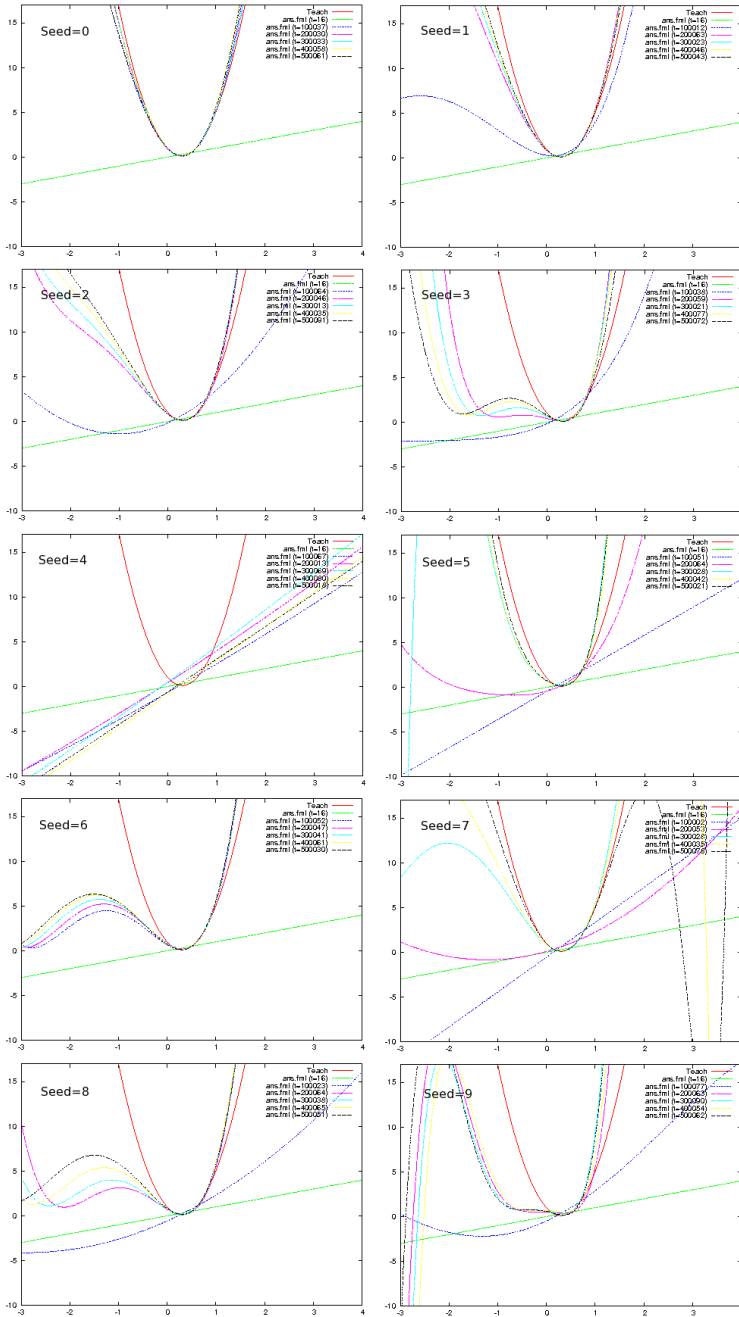


Fig. 7. *s-a* and *s-t* plots obtained for a regression problem for the one-variable quadratic function (Eq. (7a)). Out of ten different runs that use the same parameter setting, nine runs find desirable results (only the run with Seed = 4 fails). Note that the *s* is given only within the domain $0 \leq s \leq 1$.

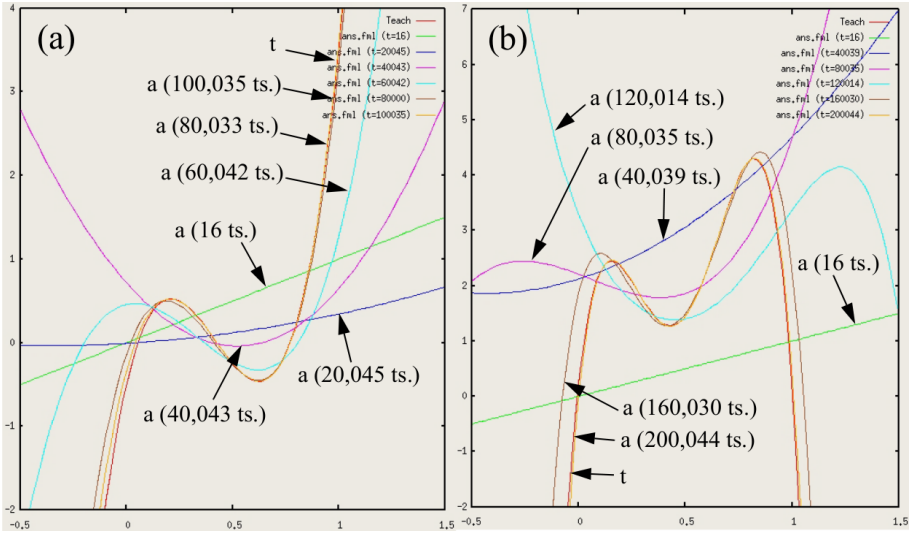


Fig. 8. The s - a and s - t plots for representative runs for the (a) cubic function (Eq. (7b)) and (b) quartic function (Eq. (7c))

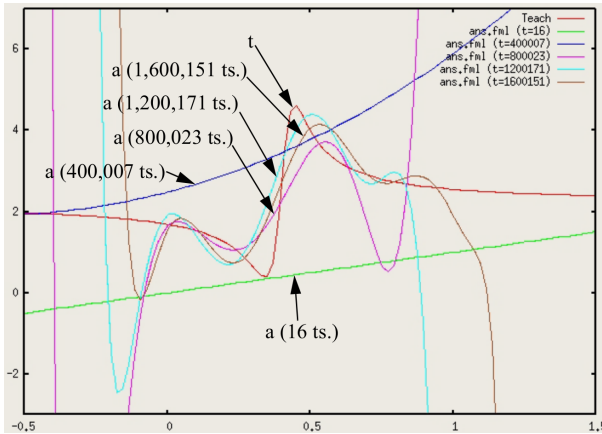


Fig. 9. The s - a and s - t plots for a representative run for the fraction function (Eq. (8))

3.3 Conditional Branch

Finally, we take a conditional branch and conduct a preliminary experiment. We set a teaching function

$$\begin{aligned} \text{if } s > 0.6, t &= 1 - s \\ \text{else } t &= 4s + 1 \end{aligned} \tag{9}$$

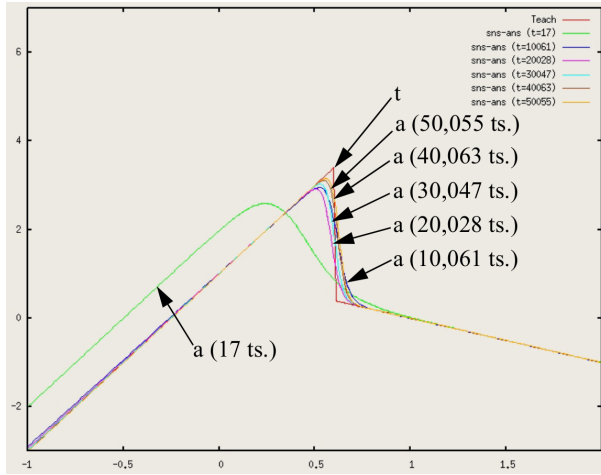


Fig. 10. s - a and s - t plots obtained for a conditional branch problem (Eq. (9))

within the domain $0 \leq s \leq 1$. We hand-design the initial ATN whose topology directly represents Eq. (9) but whose network parameters are different from the target values, and examine the ATN's capability of adjusting the parameters towards the desirable values. No agents for topological reformation are supplied.

Figure 10 shows a representative result. We can see from this figure that the ATN is able to optimize not only the threshold value but also the branch's fuzziness parameter β_r .

4 Discussion

A novel computing and learning model, algorithmically transitive network (ATN), was proposed. The ATN is represented by a data-flow network used to describe an algorithm of the data-flow computer (DFC). After the calculation by the forward propagation, the ATN propagates differential coefficients backward and adjusts node parameters with the steepest descent method. Moreover, based on the information obtained from this learning, the ATN modifies topological structure of the network through the agent operations, leading to the renovation of the network's algorithm. The model was successfully applied to a few simple symbolic regression problems, but the limitation of the learning capability was also found with more complex functions. Revising the model's framework and improving the ATN's learnability remains a future research subject.

4.1 Supervised Learning in Computation

As another auto-programming tool that represents algorithms by a network, we have Genetic Programming (GP) proposed by Koza [6,7,8]. Though the original GP used only a program graph with tree topology, but such research as PADO [25], Cartesian GP [12,13], and GNP [10] adopted networks with free topology and have extended the GP's domain. In addition, some researchers have also incorporated reinforcement

learning in the GP: node parameters were adjusted by Q-learning [5,2,10] or by the multiple linear regression analysis [14]. The ATN, on the other hand, combines the GP-like program graphs with the artificial neural network and enables more powerful supervised learning in computation.

4.2 Translation from Programming Language

Though the ATN experiments presented in this paper start from a small random/hand-made network, if we were able to develop a tool to translate a higher language program (written in C, Java, or whatever) into the ATN, the presented method would be used to improve/optimize programs designed by the humans. Of course, a usual higher language program includes a number of different control flows not tested in this paper: loop, subroutine, and so on. Translating a program with these elements into the ATN and revising it with the presented method are one of the problems to be tackled in the future.

References

1. aiSee: Commercial software for visualizing graphs with various algorithms such as rubber-band, <http://www.aisee.com/>
2. Downing, K.L.: Reinforced genetic programming. *Genetic Programming and Evolvable Machines* 2(3), 259–288 (2001)
3. Haykin, S.: *Neural networks and learning machines*. Prentice-Hall, Inc. (2009)
4. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology* 117, 500–544 (1952)
5. Iba, H.: Multi-agent reinforcement learning with genetic programming. In: Koza, J.R., et al. (eds.) *Genetic Programming 1998: Proceedings of the Third Annual Conference (GP 1998)*, pp. 167–172 (1998)
6. Koza, J.R.: *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. MIT Press, Boston (1992)
7. Koza, J.R.: *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Boston (1994)
8. Koza, J.R., Bennett III, F.H., Andre, D., Keane, M.A.: *Genetic programming III: darwinian invention and problem solving*. MIT Press, Boston (1999)
9. Kumazawa, I.: *Learning and neural network*. Morikita Publishing Company, Tokyo (1998) (Japanese)
10. Mabu, S., Hirasawa, K., Hu, J.: A graph-based evolutionary algorithm: genetic network programming (GNP) and its extension using reinforcement learning. *Evolutionary Computation* 15(3), 369–398 (2007)
11. McCulloch, W.S., Pitts, W.: A logical calculation of the ideas immanent in nervous activity. *Bullet. Math. Biophysics* 5, 115–133 (1943)
12. Miller, J.F.: An empirical study of the efficiency of learning boolean functions using a cartesian genetic programming approach. In: Banzhaf, W., et al. (eds.) *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 2, pp. 1135–1142. Morgan Kaufmann (1999)
13. Miller, J.F., Smith, S.L.: Redundancy and computational efficiency in cartesian genetic programming. *IEEE Transactions on Evolutionary Computation* 10(2), 167–174 (2006)

14. Nikolaev, N.Y., Iba, H.: Regularization Approach to Inductive Genetic Programming. *IEEE Transactions on Evolutionary Computation* 5(4), 359–375 (2001)
15. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* 323, 533–536 (1986)
16. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: McClelland, J.L., Rumelhart (eds.) *The PDP Research Group: Parallel Distributed Processing*, vol. 1. MIT Press, Cambridge (1986)
17. Sejnowski, T.J., Rosenberg, C.R.: Parallel networks that learn to pronounce English text. *Complex Systems* 1, 145–168 (1987)
18. Sharp, J.A. (ed.): *Data flow computing: Theory and practice*. Ablex Publishing Corp., Norwood (1992)
19. Suzuki, H.: Mathematical folding of node chains in a molecular network. *BioSystems* 87, 125–135 (2007)
20. Suzuki, H.: An approach toward emulating molecular interaction with a graph. *Australian Journal of Chemistry* 59, 869–873 (2006)
21. Suzuki, H.: A network cell with molecular agents that divides from centrosome signals. *BioSystems* 94, 118–125 (2008)
22. Suzuki, H., Ohsaki, H., Sawai, H.: A Network-Based Computational Model with Learning. In: Calude, C.S., Hagiya, M., Morita, K., Rozenberg, G., Timmis, J. (eds.) *UC 2010. LNCS*, vol. 6079, pp. 193–193. Springer, Heidelberg (2010)
23. Suzuki, H., Ohsaki, H., Sawai, H.: Algorithmically Transitive Network: a new computing model that combines artificial chemistry and information-communication engineering. In: *Proceedings of the 24th Annual Conference of Japanese Society for Artificial Intelligence (JSAI)*, pp. 2H1-OS4-5 (2010) (Japanese)
24. Suzuki, H., Ohsaki, H., Sawai, H.: An agent-based neural computational model with learning. *Frontiers in Neuroscience*. Conference Abstract: Neuroinformatics (2010), doi:10.3389/conf.fnins.2010.13.00021
25. Teller, A., Veloso, M.: PADO: Learning tree-structured algorithm for orchestration into an object recognition system. *Carnegie Mellon University Technical Report, CMU-CS-95-101* (1995)
26. Tennenhouse, D.L., Wetherall, D.J.: Towards an active network architecture. *ACM Computer Communication Review* 26(2), 5–18 (1996)
27. Werbos, P.J.: The roots of backpropagation: From ordered derivatives to neural networks and political forecasting. In: *Adaptive and Learning Systems for Signal Processing, Communications and Control Series*. Wiley Interscience (1994)

Modelling to Contain Pandemic Influenza A (H1N1) with Stochastic Membrane Systems: A Work-in-Progress Paper

Lei Xu

Key Laboratory of Image Processing and Intelligent Control
Department of Control Science and Engineering
Huazhong University of Science and Technology
Wuhan 430074, Hubei, China
leixuhust@gmail.com

Abstract. Pandemic influenza A (H1N1) has spread rapidly across the globe. In the event of pandemic influenza A (H1N1), decision-makers are required to act in the face of substantial uncertainties. Simulation models can be used to project the effectiveness of mitigation strategies. Since nature is very complex, the perfect model that explains it will be complex too. Membrane system (P system) can be a perfect model modelling ecological system. This paper briefly describes stochastic membrane systems for modelling spread of pandemic influenza A (H1N1) in an isolated geographical region. The model is based on a discrete and stochastic modelling framework in the area of Membrane Computing. This model can be a useful tool for the prediction of infectious diseases within predefined areas, and the evaluation of intervention strategies.

Keywords: pandemic influenza A (H1N1), pandemic spread modelling, stochastic membrane systems, membrane computing.

1 Introduction

Cases of pandemic influenza A (H1N1) were first reported in Mexico in April 2009 in [1]. Subsequently, the virus spread rapidly across the United States and Canada, and then became a global concern [2]. The influenza A (H1N1) virus is highly transmissible. Urgent implementation of measures against pandemic influenza A (H1N1) is required. Modelling the spread and control of pandemic influenza A (H1N1) can help to predict the spread tendency of pandemic influenza A (H1N1), thus help to choose the effective measures to control the pandemic.

The application of mathematical modelling to the spread of epidemics was initiated by Daniel Bernoulli's work on the effect of cowpox inoculation on the spread of smallpox. Classical epidemic modeling was built on differential equations. These models revealed the threshold nature of epidemics and explained herd immunity, where the immunity of a subpopulation can stifle epidemics' outbreaks to protect the entire herd. But such models are ill-suited to capture the direct contacts between individuals and complex virus spreading social networks.

P systems, or membrane systems, were introduced in [3] as a class of unconventional computing devices of distributed, parallel and nondeterministic type, inspired by the compartmental structure and the functioning of living cells. P systems are based on a hierarchical structure of nested membranes, inspired by the structure of living cells. Each region can contain objects, mimicking the presence of molecules, proteins, etc. in the compartments of living cells. Moreover, each region has an associated set of multiset rewriting rules. These rules are motivated by chemical reactions that occur inside the regions of living cells. We can refer to [3,13] for details on Membrane Computing.

Although most research in P systems concentrates on the computational power of the devices involved, lately they have been used to model biological phenomena within the framework of computational systems biology. In this case P systems are not used as a computing paradigm, but rather as a formalism for describing the behavior of the system to be modelled. In this respect, several P systems models have been proposed to describe oscillatory systems [4], signal transduction [5], gene regulation control [6] and quorum sensing [7]. These models differ in the type of the rewriting rules, membrane structure and the strategy applied to run the rules in the compartments defined by membranes.

Recently, P systems have been applied in modelling complex systems. In particular, P systems are used to model cellular processes and biochemical or cellular systems to analyze the complex behaviors [8,9]. They are feasible also for the investigation of biological systems at a higher scale order, such as the ecological systems, where the interactions between individuals are analyzed [10].

As P systems are inspired from the structure and functioning of the living cell, it is natural to consider them as modelling tools for biological systems, within the framework of systems biology, being an alternative to more classical approaches like ordinary differential equations (ODEs) and to some recent approaches like Petri nets and π -calculus.

Simulating natural phenomena using membrane systems can keep the most important natural factors and overcome some limitations of classical mathematical models. The complex social networks and interaction between individuals can be expressed by the complex structure and rules of membranes. There are enormous cells and chemical substances even in a limited space. Chemical substances in membrane are natural agents, and they quite differ from cyber agents where chemical reactions inside membrane systems are obedient to the nature's law. What is more, the chemical reactions in membrane systems are highly parallel. In this respect, using membrane systems to model the spread of pandemic may be more advantageous than classical mathematical modelling such as ordinary differential equations modelling.

Modelling pandemics with P systems is a novel way to use the concept of agent-based model to model pandemics spreading. P systems show their power of modelling and investigating ecological systems. Modelling pandemics with P systems can capture the direct contacts between individuals and complex virus spreading social networks. It also overcomes the shortcomings of classical mathematical modelling.

2 Objectives

We extend the stochastic modelling framework of dynamic probabilistic P (DPP) systems, initially introduced in [9]. The goal is to construct a model of dynamic probabilistic P systems to simulate and investigate the spread and control of pandemic influenza A (H1N1) in an isolated geographical region.

3 Modelling Framework

Dynamical probabilistic P systems are membrane systems where probabilities are associated with the rules, and such values vary during the evolution of the system according to a prescribed strategy. Each membrane identifies a *region*, delimited by it (if any) immediately inside it. The number of membranes in a membrane structure is called the *degree* of the P system. The number of hierarchical levels in a membrane structure is called the *depth* of the P system. The whole space outside the skin membrane is called the *environment*. Different from the DPP systems introduced in [10], the probabilities are given instead of been evaluated in our model and our construct of DPP systems is a little different from construct in [10]. More details about DPP systems and examples of simulated systems can be found in [12].

Definition 1. *A dynamical probabilistic P system of degree n is a construct*

$$\Pi = (O, \mu, M_0, \dots, M_{n-1}, R_0, \dots, R_{n-1}, I) \text{ where:}$$

- O is an alphabet, and its elements are called objects.
- μ is a membrane structure consisting of n membranes labeled with the numbers $0, \dots, n-1$. The skin membrane is labelled with 0 .
- $M_i, i = 0, \dots, n-1$, is the initial multiset over O inside membrane i .
- $R_i, i = 0, \dots, n-1$, is a finite set of evolution rules associated with membrane i . An evolution rule is of the form $r : u \xrightarrow{P} v$, where u is a multiset over O , v is a string over $O \times (\{\text{here}, \text{out}\} \cup \{\text{in}_j | 1 \leq j \leq n-1\})$ and $P \in \mathbb{R}^+$ is a constant between 0 and 1 associated with the rule.
- $I \subseteq \{0, \dots, n-1\}$ is the set of labels of the analyzed regions.

A DPP system works as follows: A fixed initial configuration of Π is given according to the simulation design, hence it consists of the multisets initially present inside the membrane structure, rules and corresponding probability of each rule.

At each step of the evolution, all applicable rules are simultaneously applied and all occurrences of the lefthand sides of the rules are consumed. We assume that the system evolves according to a universal clock, that is, all membranes and the application of all rules are synchronized. The applied rules are chosen according to the probability values assigned to them; the rules with the highest normalized probability value will be more frequently tossed. If some rules compete for objects and have the same probability values, then objects are non-deterministically assigned to them.

4 Method

Based on the framework of dynamical probabilistic P systems, a model is constructed to reveal and simulate the spread and control of pandemic influenza A (H1N1) in an isolated geographical region. It is believed that the model and parameters are based on reasonable estimates of what could happen, and these results are assisting the planning of a response.

Membranes can be agents representing different regions in our society. For example, we can make some membranes represent schools and some represent workplaces and others represent households, etc. The different substances in and out of membranes can be agents representing different individuals (susceptible, infected, recovered) in our society. The activities of human beings can be modeled by some regulated application rules of substances. The quantity of regions and people can be enormously large because of the large quantity of membranes and substances. The model can be described as Fig. 1.

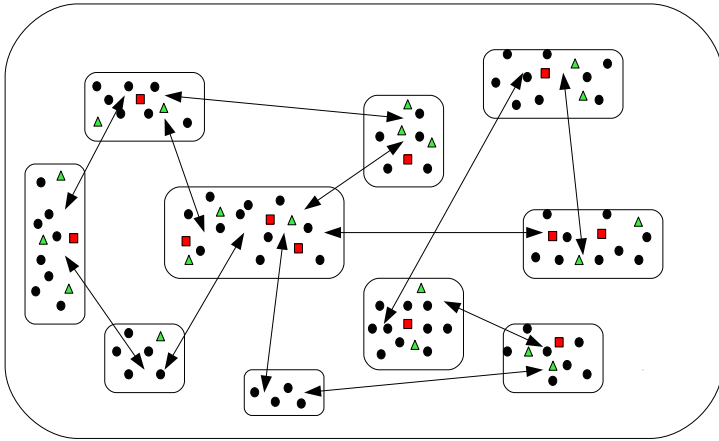


Fig. 1. Membrane system modelling spread of pandemic

The rectangle regions mean the hierarchical membranes. The shapes with different colors represent different individuals in each region: the black circles represent the susceptible individuals; the red squares represent the infected individuals; the green triangles means the recovered individuals. The arrows in the model describe the movements of individuals between different regions.

Based on [11], some assumptions about the spread of pandemic influenza A (H1N1) are made to construct the dynamical probabilistic P system model:

1. Each infected individual is equally infectious, and the infectiousness of an infected individual remains changeless during the course of symptomatic period.
2. The virulence of the pandemic influenza A (H1N1) virus remains changeless during the course of spreading.

3. The immunity and susceptibility of each individual is identical regardless of his/her age.
4. Recovered individuals get immunized and are excluded from the infection. The death incidents which are insignificant compared to the whole population in reality are ignored.

The scheme of DPP systems model for simulation of pandemic influenza A (H1N1) is:

$II = (O, \mu, M_0, M_1, M_2, M_3, \dots, M_{n-1}, R_0, R_1, R_2, R_3, \dots, R_{n-1}, I)$ where:

- $O = \{S, I, R\}$. S, I, R stand for a susceptible, a infected and a recovered individual respectively.

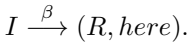
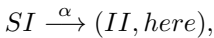
- $\mu = [[]_1 []_2 []_3 \dots []_{n-1}]_0$.

- $M_0 = \lambda$ (λ means no objects), $M_1, M_2, M_3, \dots, M_{n-1}$ contains different numbers of S, I, R respectively, which means the initial population in each region.

- $R_0 = \lambda$ (λ means no rules), $R_1, R_2, R_3, \dots, R_{n-1}$ contains different rules respectively reflecting the transmission and control of pandemic influenza A (H1N1).

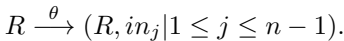
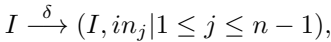
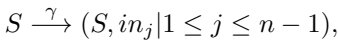
- $I = \{1, 2, 3, \dots, n - 1\}$ indicates the analyzed regions.

In modeling the spread and control of pandemic influenza A (H1N1), the membrane system model can incorporate the underlying mechanism of transmission and recovery dynamics and can be able to account for experimental data in a number of cases. In this regime, a probabilistic description must be used. Schematically, the stochastic infection and recovery dynamics is given by



The first reaction reflects the fact that an encounter of an infected individual with a susceptible results in two infecteds at a probability α ; the second indicates that an infected individual gets recovered at a probability β .

The stochastic movement dynamics is given by



The three rules reflect the fact that a susceptible, a infected and a recovered individual moves into one region at probabilities γ , δ , and θ respectively.

We assume that a global clock exists, marking the time for the whole system (for all cells of the system); that is, all application of all the substance rules are synchronized. The rules are also applied in a *maximal consistent parallelism*; that is, all those rules must be applied simultaneously in a maximal way.

5 Future Work

In the next step, the specific DPP systems modelling spread and control of pandemic influenza A (H1N1) should be constructed. The social places can be categorized into five categories: school, household, workplace, community and quarantine. The model is depicted in Fig. 2. In the model, both free movement (depicted with green arrows) and forced quarantine (depicted with red arrows) are considered.

In a future work, transmission probabilities and realistic interactions between people through which transmissions arise should be also investigated. Once we get the real sample data of the spread of pandemic influenza A (H1N1), we can build the initial model according to the sample data, and set the related rules and probabilities. Then the model is modified until it can reveal the real situation of pandemic's spread. The simulation can run thousands of times to forecast the spread of pandemic influenza A (H1N1). Moreover, different measures to be implemented can be simulated in the model so that the impact of these measures can be correctly estimated.

Before we construct the model, it is necessary to perform simulations of the model. We will perform simulations by means of a dedicated program written in Matlab language. In the simulations, the stochastic and parallel application of the rules is done by splitting each parallel step into several sequential sub-steps. Each single parallel step is separated into three stages: in the first one, a random number (between 0 and 1) generator is used to generate a probability. The probability generated is compared with the given probability of the specific rule. If the probability generated is less than the given one, the rule associated with the probability will be applied. Otherwise, the rule will not be applied in this step. So the rule is chosen to be applied according to the comparing result; in

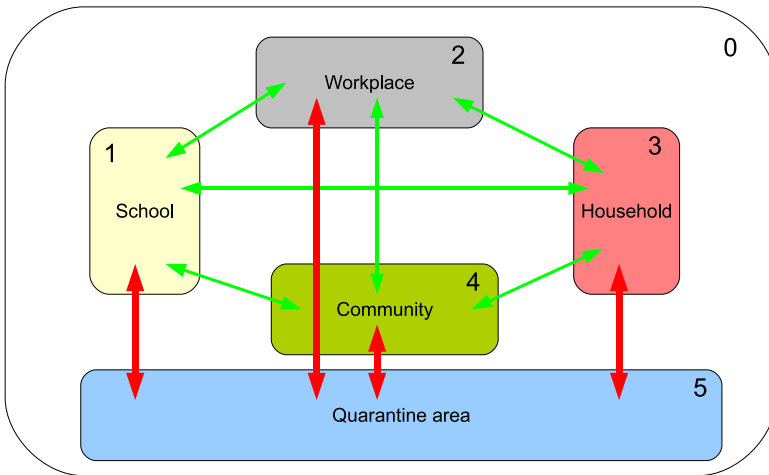


Fig. 2. Social networks of pandemic influenza A (H1N1) spreading

the second one, rule is applied to correspondingly consume the objects appearing in its left-hand side; in the third one, the multisets are updated using a stored trace of the rules previously tossed. Each stage starts only when the previous one has been applied to all the membranes in the DPP system, and the same process is repeated for all evolution steps. A detailed description of the way the simulation algorithm works and of its complexity can be found in [12].

6 Conclusions

Simulation of pandemic's spread using membrane systems is a novel way to forecast the spread of pandemic. Membrane system is a natural tool with a lot of advantages like the large quantity, the high parallelism and the natural behavior based on rules. As pandemic influenza A (H1N1) progresses, and new health challenges emerge, such novel agent-based model will show tremendous advantages in simulation. Once the model is considered to be experimentally validated, it is possible to provide hypotheses about the possible spread of pandemic influenza A (H1N1).

References

1. Centers for Disease Control and Prevention: Outbreak of Swine-origin Influenza A (H1N1) Virus Infection - Mexico. *MMWR Morb. Mortal Wkly Rep.* 58, 467–470 (2009)
2. Fraser, C., Donnelly, C.A., Cauchemez, S., Hanage, W.P., Van Kerkhove, M.D., Hollingsworth, T.D., et al.: Pandemic Potential of a Strain of Influenza A (H1N1): Early Findings. *Science* 324, 1557–1561 (2009)
3. Păun, G.: *Membrane Computing – An Introduction*. Springer (2002)
4. Fontana, F., Bianco, L., Manca, V.: P Systems and the Modeling of Biochemical Oscillations. In: Freund, R., Păun, G., Rozenberg, G., Salomaa, A. (eds.) *WMC 2005*. LNCS, vol. 3850, pp. 199–208. Springer, Heidelberg (2006)
5. Jesús Pérez-Jiménez, M., Romero-Campero, F.J.: P Systems, a New Computational Modelling Tool for Systems Biology. In: Priami, C., Plotkin, G. (eds.) *Transactions on Computational Systems Biology VI*. LNCS (LNBI), vol. 4220, pp. 176–197. Springer, Heidelberg (2006)
6. Pérez-Jiménez, M.J., Romero-Campero, F.J.: Modelling Gene Expression Control Using P Systems: The Lac Operon, A Case Study. *Biosystems* 91, 438–457 (2008)
7. Pérez-Jiménez, M.J., Romero-Campero, F.J.: A Model of the Quorum Sensing System in *Vibri Fischeri* Using P Systems. *Artificial Life* 14, 95–109 (2008)
8. Mazzaa, T., Cavalierea, M.: Cell Cycle and Tumor Growth in Membrane Systems with Peripheral Proteins. *Theoretical Computer Science* 227, 127–141 (2009)
9. Pescini, D., Besozzi, D., Zandron, C., Mauri, G.: Analysis and Simulation of Dynamics in Probabilistic P Systems. In: Carbone, A., Pierce, N.A. (eds.) *DNA 2005*. LNCS, vol. 3892, pp. 236–247. Springer, Heidelberg (2006)
10. Besozzi, D., Cazzaniga, P., Pescini, D., Mauri, G.: Modelling Metapopulations with Stochastic Membrane Systems. *Biosystems* 91, 499–514 (2008)

11. Mei, S., van de Vijver, D., Xuan, L., Zhu, Y., Sloot, P.M.A.: Quantitatively Evaluating Interventions in the Influenza A (H1N1) Epidemic on China Campus Grounded on Individual-based Simulations. *Procedia Computer Science* 1, 1669–1676 (2010)
12. Bianco, L., Pescini, D., Siepmann, P., Krasnogor, N., Romero-Campero, F.J., Gheorghe, M.: Towards a P Systems Pseudomonas Quorum Sensing Model. In: Hoogboom, H.J., Păun, G., Rozenberg, G., Salomaa, A. (eds.) *WMC 2006*. LNCS, vol. 4361, pp. 197–214. Springer, Heidelberg (2006)
13. The P System Web Page, <http://ppage.psystems.eu>

P2P-Based Scalable Execution Platform for Algorithmically Transitive Network

Mikio Yoshida¹, Hideaki Suzuki², and Hidefumi Sawai²

¹ BBR Inc.

2-1-4-206, Sonezakishinchi, Kita-ku, Osaka, 530-0002, Japan

yos@bbr.jp

² National Institute of Information and Communications Technology

588-2, Iwaoka, Iwaoka-cho, Nishi-ku, Kobe, 651-2492, Japan

{hsuzuki, sawai}@nict.go.jp

Abstract. “Algorithmically Transitive Network” (ATN) is a novel computational model based on a data-flow network, consisting of the following operations: a forward propagation propelled with node firing and token creation, a backward propagation caused by evaluating differential coefficients, and a topological alteration taken place by autonomous agents. In the research of the ATN, a simulation run on some parallel processing scheme is essential. As a flexible and powerful implementation scheme, the paper employs a P2P based distributed platform, and describes the mechanisms for simulation and P2P deployment of the ATN. The implemented platform has the following three features: flexible allocation of ATN nodes to the physical resources, unified description of communication between nodes, and several methods to realize high parallelism. The proposed scheme is also helpful to verify applicability of the employed P2P system.

Keywords: peer-to-peer network, framework system, data-flow architecture, agents, oneway RPC.

1 Introduction

“Algorithmically Transitive Network” (ATN) [1] [2] is a novel computational model based on the data-flow network [3]. The ATN consists of operations of the bi-directional propagation of ‘tokens’, a forward propagation (FP) which advances calculation ahead using ‘firings’ of nodes, and a backward propagation (BP) which evaluates calculation to the opposite direction of the flow of tokens in the FP.

As in the data-flow network, the ATN’s nodes read the input tokens on their incoming edges, fire, and create the output tokens on their outgoing edges during calculation. This constructs a ‘fire-token pedigree’ whose nodes (tokens) represent variables or mathematical expressions, and whose hyper-edges (firings) represent arithmetic/logical operations used to create the tokens. An example of these relationships is shown in Fig. 1. The BP, which is propelled on the pedigree, finally modifies parameters of the ATN at some nodes, whose statistical data is used to change the network topology by functional agents distributed in the network. The aim of the ATN research is in making the data-flow network itself learn through these operations and explore novel algorithms automatically.

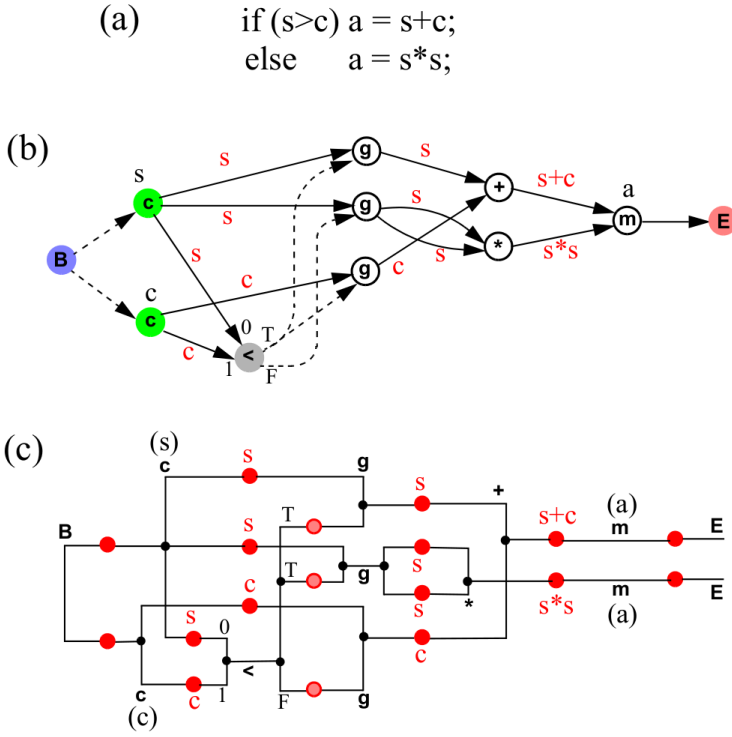


Fig. 1. (a) Higher language program of a simple conditional branch, (b) data-flow network (ATN), and (c) fire-token pedigree produced by the calculation. The variable name s represents the graph’s input (sensor) signal, and the a represents the graph’s output (answer) value. The pedigree’s top ancestor is the initial fire at the ‘B’ node, and its last descendants are firings at the ‘E’ node or nodes with no outgoing edge. In (b), arithmetic edges are expressed as the solid arrows, and regulating ones are expressed as the broken arrows.

Of course, an implementation and an experiment of the ATN can be conducted on a single core computer; however, in order to precisely evaluate the ATN’s ability to explore algorithms, implementation to parallel computers, such as a PC cluster with sufficient calculation resources, is essential. Although now in many cases, MPI [4] is used as a programming tool for a parallel computer, here we propose building the ATN on a P2P platform system named “PIAX” [5]. There are three advantages for taking this approach. First, a PIAX’s higher level library provides an efficient programming environment for software developers as well as the software’s flexibility. Second, the ATN which uses the data-flow network for the base of calculation has high affinity with a P2P network, requiring the minimal communication cost between CPU cores. This will be also enhanced by the trait of the PIAX able to use threads efficiently and utilize the core resources for the maximum. Third, the P2P library enables the ATN to be deployed to the open network environment in the world.

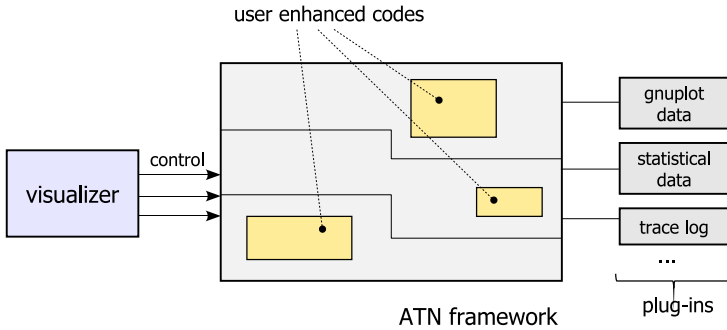


Fig. 2. Modular structure of ATN-P2P

In the research of the ATN, what kind of agents should be prepared and what kind of topology changing rules should be made are important research agenda. To inquire about these issues, we have to try many agent designs with repeating each evaluation and verification. Moreover, to confirm the ATN's high parallelism and functionality as a distributed system, it is also necessary to check about the performance of the ATN implemented onto an actual P2P network. The P2P-based platform system we have developed ("ATN-P2P") not only supports the research of the ATN from these points of view but also clarifies the outcome of or difficulty in deploying the ATN to a P2P network. Subsequent chapters describe the design and implementation of ATN-P2P, focusing on mechanisms for the ATN simulation platform and P2P deployment.

2 Basic Design of ATN-P2P

2.1 Framework and Modularity

The ATN-P2P has a form of a 'framework system' so that an ATN researcher can try and implement an idea freely. The researcher can change, add and delete the following functions without caring about the system behavior of the topology change of the ATN and the detailed flows of tokens in the FP and the BP.

- Insertion and deletion of a node, and the change of an operation defined in a node.
- Change of calculation of differential coefficients and teaching signals.
- Insertion and deletion of an agent, and the change in behavior of an agent.

In addition to implementing a new function, it should be checked how the added function is operating inside and how the variables changes. Although these are the basic functions of a simulator, change of a request occurs in several phases of research. To meet this, we took a modular design approach which separates a visualization function, etc., from the essential computation of the ATN. The modular structure of ATN-P2P is shown in the Fig. 2.

The central box named 'ATN framework' is the framework system, a core module which executes the ATN computation. The above functions, i.e., the insertion and deletion of a node or the change in behavior of an agent and so on, are implemented on this

framework. The left visualizer module is a controller of the ATN framework, which has the function of visualizing the graphical data returned from the ATN framework. The ATN framework has an API to control from outside. It is possible to control the ATN framework from not only visualizer module but also from a script and to make it deploy as a web server. On the other hand, it has a plug-in structure for gathering such information as the output in the gnuplot form of each pass state of the ATN, the variable values that nodes have for processing statistically, and fine motions in the processes of the FP and the BP, and the operations of agents. Although the visualizer module was implemented as a controller form, the data for visualization can also be collected using this plug-in function. This framework design and the modularity enhance the extendibility of the system and the independency of computation logic in the ATN, helping a researcher concentrate on the research of the mechanism of the ATN itself.

2.2 Requirement for P2P Deployment

In general, when we apply the P2P framework to some system, we have to consider the following issues which might become primary obstacles to the deployment:

1. Existence of shared resources.
2. Synchronicity over two or more nodes.
3. Consistency between nodes.

The existence of shared resources becomes a factor which disturbs the decentralization of processing. When a node is decentralized in the ATN, access from all of the nodes is focused on the shared resources, and there is a bottleneck in the performance.

In a platform including a higher-level manager that looks down at the whole network, synchronicity over two or more nodes is easily implemented; and yet, this is not the case in a distributed system. Simulation of the ATN requires repeated operations of the FP, the BP, and agents as one cycle (called pass), and for this reason, each distributed node needs to detect the completion of each pass.

The consistency between nodes becomes a problem, for example, when a change of the topology of the ATN is simultaneously made by two or more agents and the consistency of topology is no longer guaranteed. It may also happen that the a partial separation of the network occasionally occurs and the token flows of the data flow network are stopped.

3 Mechanisms for the P2P Platform Simulating the ATN

3.1 Flexible Resource Allocation

Since, through the repetition of passes, the ATN changes its topology dynamically and the number of nodes tends to increase with time, we cannot assign one node to one machine when performing the ATN on the machines of fixed number. To get around this problem, the ATN-P2P assigns the ATN nodes to the physical resources dynamically by dividing the node set into some groups and allocating them to the machines. One group is treated as a process, and the number of processes is taken to be changeable during the

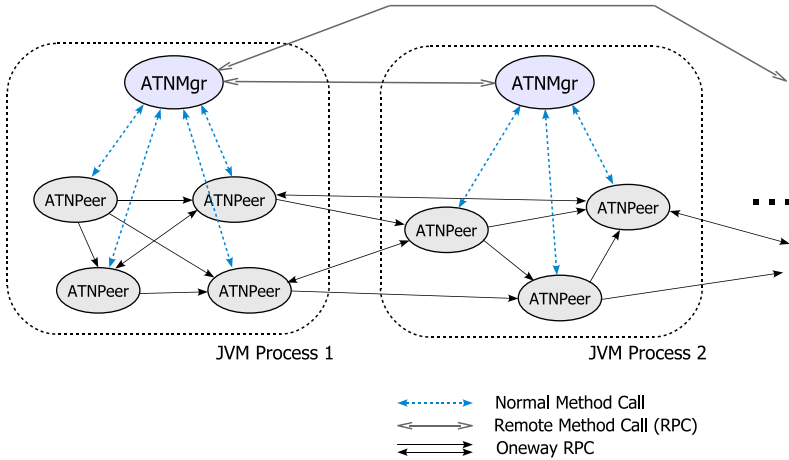


Fig. 3. Node allocations and communication patterns in the ATN-P2P

run. Furthermore, communication between processes is performed using IP addresses, irrelevant of whether it is inside or outside a machine. This makes a process boundary equivalent to a machine boundary, and assures the flexible assignment of the processes to the machines.

Basically, we can choose arbitrarily a way for the grouping of nodes in the ATN into processes on the equipped distributed machines. For example, when the whole ATN is assigned to one process without a grouping, the ATN-P2P functions as a single simulator. When nodes are grouped one by one and each process is assigned to a machine, the ATN-P2P operates as an actual P2P network. The grouping into a process and communication of nodes are more concretely shown in the Fig. 3.

Since the ATN-P2P is implemented in Java, a process here is a JVM (Java VM) process. There exists one object called 'ATNMgr' on a JVM process. The ATNMgr manages all the nodes assigned to the process by conducting the following tasks.

- Insertion and deletion of a node and management of topology information including edges.
- Cooperation with ATNMgr(s) on other processes.
- Communication with external applications and specifying its API.

An ATNPeer is a Java object which represents a node. An outgoing edge from the node is also assigned to this ATNPeer. Communication between the ATNMgr and ATNPeer(s) is performed as follows.

- Communication between ATNPeer(s) within the same process as the ATNMgr is performed by a normal method call.
- Communication of ATNMgr(s) between different processes uses a remote procedure call (RPC).
- Communication of ATNPeer(s) uses a RPC with no returning value (called 'Oneway RPC').

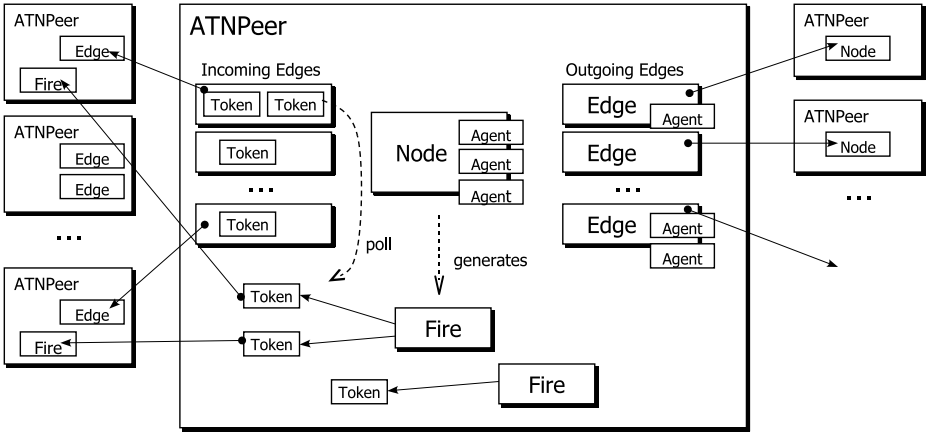


Fig. 4. Relationship of objects in ATNPeer. A 128-bit unique ID is used for the reference to Node, Edge, and Fire objects which exist in a different ATNPeer. The ATNPeer has the ID same as the Node has.

As the data-flow network used in the ATN communicate only between an adjacent node pair, so an ATNPeer exchanges such information as tokens only with its partners (adjacent ATNPeers). Moreover, in communication between ATNPeers, a communication partner's ATNPeer does not know whether the sender exists in the same JVM process. The Oneway RPC, described in the Section 4.2, is a mechanism that enables the one way message transmission in a RPC form.

The ATNMgr performs the insertion and deletion of nodes within the JVM process. It is important to decide which process of which machine the newly generated node should be assigned to. At present, the generated node is assigned to the same process as that which the parent node belongs to; however, we are able to allow an ATNPeer to migrate to another JVM process to secure good load balance between JVM processes. The problem for making rearrangement of the ATNPeers is to be solved by the cooperation of ATNMgr(s).

3.2 Object Assignment in ATNPeer

The internal structure of an ATNPeer is shown in the Fig. 4. The ATNPeer has the assigned Node object, outgoing Edge objects and Agent objects on each Node or Edge object, and Fire and Token objects generated at the time of the FP. The information on the incoming edge is also needed by the ATNPeer, but this information is made available via the reference to an Edge object which another ATNPeer holds. During the FP, if the Token objects used as the conditions for firing are prepared in the queuing elements in the incoming Edge objects, the node fires, which causes the creation of a Fire object the ATNPeer.

3.3 Unified Description for Message Sending

In order for the ATN nodes to be allocated flexibly to physical resources, it is desirable that the communication between nodes might be expressed in a program uniformly regardless of the way of the grouping of the nodes or the assignment of the processes to the machines. The guiding principle for this unified description is:

- Eliminate a bidirectional communication.
- Always use a remote communication protocol.
- Specify the destination node with the ID.

According to the data flow model which the ATN has, we basically consider one way message transmission. The scheme of communication is unified into description supposing remote communication for both within-a-process and between-processes communication. In order to improve the performance of local communication, when the communication locality is detected, a prepared optimization procedure is always conducted. For the references to objects in another ATNPeer, the IDs are used as described in Section 3.2. This avoids the disturbance of the communication after the rearrangement takes place and the communication address (IP address etc.) of the ATNPeer changes. The above-mentioned guiding principle is realized by the Oneway RPC described in Section 4.2.

3.4 Concurrency Control of ATN

The realization of effective parallelism and concurrency is a central theme in the research of the ATN. In case of the ATN-P2P, the following two step approaches are taken.

Division of the Processes to Two or More Machines. Higher parallelism is achieved by dividing a node set into the processes on multi-core machines or processes on different machines.

Event Driven Process. Unit operations in the ATN node are classified into the firing in the FP, fire extinguishing in the BP, or operations by the agents. Since all these processes are triggered by external events and none of them use a CPU stably, we do not assign a thread to a node permanently but assign a thread to the generated event. This achieves high level concurrency in a simulation.

4 Implementation

4.1 Distinctive Feature of PIAX Transport

In the implementing of ATN-P2P, the function which the transport layer of PIAX has for the realization of the unified description of communication between nodes and the concurrency using an event driven model was used.

PIAX is a platform system for the mobile agent which operates on a P2P network. It has a framework for the nodes which constitute a P2P network in order to communicate not using physical addresses like IP addresses but using global unique IDs. This is called the ‘ID/Locator separation’ and it is prepared as a communication mechanism for a P2P network (or an overlay network) in the transport layer of PIAX [6].

Regarding the high level concurrency, PIAX has a mechanism for making nodes of tens of thousands of scales perform on one JVM. This is because the transport layer of PIAX is implemented as the event driven model. The feature utilized by ATN-P2P is as follows.

- Since the thread is not assigned to the node, even if it works, the nodes of tens of thousands of scales don’t generate consumption of the thread.
- Make concurrent processing generated by event using a thread. Since many threads are assigned to CPU resources by JVM adaptively, they can utilize a multi-core for the maximum.
- Avoid the overhead of thread generation by thread pooling. Furthermore, the draining of thread assignment of the concurrent processing, which takes place explosively by giving thread pool restriction can be prevented by setting an upper limit on the thread pool.

4.2 Oneway RPC

For the unified description of the communication described in Section 3.3, a RPC for the one way call, named ‘Oneway RPC’, was implemented using the RPC function of PIAX. Although the Oneway RPC has the same call form as a RPC, since it is the one way call, neither a return value nor the exceptions which are thrown at the receiver side is returned to a caller. And since a caller does not wait for the completion of processing of the RPC, the Oneway RPC call is completed immediately.

The example of coding of the Oneway RPC is shown in the List 1.1. In this example, in an ATNPeer, in order to add outgoing edge newly, the addIncomingEdge method which adds an incoming edge of the ATNPeer is called. As preparation for treating the ATNPeer linked as a remote object, the stub of the RPC is generated by a getOnewayStub method and an addIncomingEdge method is called out as a method which the stub has. In the usual stub generation, although the IP address of a remote object is needed, the mechanism of the ID/Locator separation which PIAX has is utilized here so that the ID can be specified.

List 1.1. A sample code of Oneway RPC

```
1 public void addEdge(...) {
2     ...
3     try {
4         ATNPeerIf stub = (ATNPeerIf) getOnewayStub(dstPeerId);
5         stub.addIncomingEdge(incomingIx, getId(), edge.getId());
6     } catch (UndeclaredThrowableException e) {
7         ...
8     }
9 }
```

4.3 Mobility of Agents

An agent mobility is a function which should be supported in the ATN. Through ATN-P2P, the agent mobility is realized using the Oneway RPC. The internal state of an Agent object is sent to the ATNPeer on the destination side using the Oneway RPC, an Agent with the same ID is generated, the internal state of the original Agent is set, and then the mobility of an agent is realized. The following method is an example which moves Agent ag to the Node whose ID is dstPeerId.

```
void moveAgent(PeerId dstPeerId, Agent ag);
```

5 Conclusion

5.1 Related Work

In the research of the data-flow network, DataRush [8] which is a simulator with the framework which can be defined by a user, a language called Stella [9] aiming at the visualization of a data flow model, and so on have been developed. And in the area studying P2P networks, there is also a system like p2psim [10] for carrying out the simulation of the operation of many nodes. Thus, tools for carrying out the simulation of the network by each area of research exist. Like ATN-P2P, a function required for a simulation can be treated integratively, and as far as the authors know, a tool deployable as a distributed system like an actual P2P network does not exist.

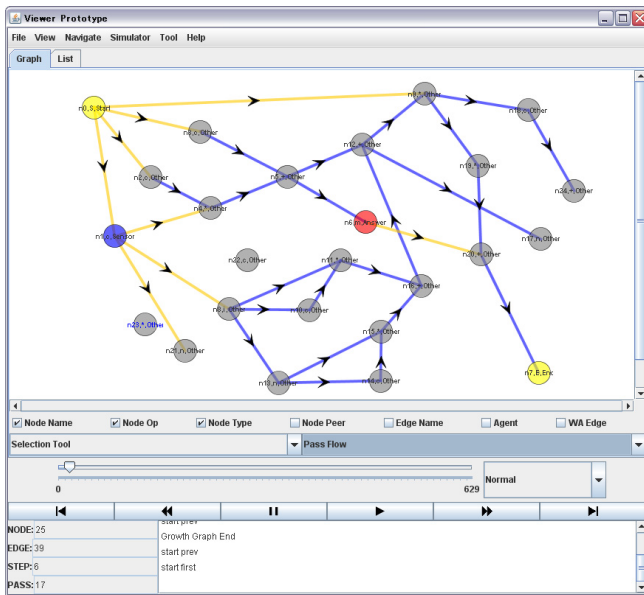


Fig. 5. Example screenshot of the visualizer of ATN

5.2 Current Status and Future Work

Currently, the basic design of ATN-P2P has been completed and ATN-P2P is in the stage where the implementation of the foundation of a framework ended, and it cooperates with a visualizer. Fig. 5 shows the screen sample of the visualizer which is actually in operation.

In the future, a brush up of the framework, fulfillment of the plug-in function, and research and development on the function which allocates nodes to physical resources adaptively using cooperation of ATNMgr(s) are subjects to be tackled.

References

1. Suzuki, H., Ohsaki, H., Sawai, H.: An agent-based neural computational model with learning. In: *Frontiers in Neuroscience, Conference Abstract: Neuroinformatics (2010)*, doi:10.3389/conf.fnins.2010.13.00021
2. Suzuki, H., Ohsaki, H., Sawai, H.: Algorithmically Transitive Network: A Self-organizing Data-flow Network with Learning. In: Suzuki, J., Nakano, T. (eds.) *BIONETICS 2010. LNICST*, vol. 87, pp. 59–73. Springer, Heidelberg (2012)
3. Sharp, J.A. (ed.): *Data flow computing: Theory and practice*. Ablex Publishing Corp., Norwood (1992)
4. The Message Passing Interface (MPI) standard, <http://www.mcs.anl.gov/research/projects/mpi/>
5. Yoshida, M., Okuda, T., Teranishi, Y., Harumoto, K., Shimojo, S.: PIAX: A P2P Platform for Integration of Multi-Overlay and Distributed Agent Mechanisms. *IPSJ Journal* 49(1), 402–413 (2008)
6. Yoshida, M., Teranishi, Y., Shimojo, S.: A Mechanism of ID/Locator Separation in Overlay Networks. *IPSJ Journal* 50(9), 2298–2311 (2009)
7. Gnuplot, <http://www.gnuplot.info/>
8. DataRush, <http://www.pervasivedatarush.com/>
9. Stella, <http://www.iseesystems.com/software/Education/StellaSoftware.aspx>
10. p2psim, <http://pdos.csail.mit.edu/p2psim/>

Diagnosability of Nested Intruders*

Damas P. Gruska

Institute of Informatics, Comenius University,
Mlynska dolina, 842 48 Bratislava, Slovakia
gruska@fmph.uniba.sk

Abstract. Formalism for analyses of biological systems specified by process algebras is proposed. Biologically motivated it combines several security notions and approaches. It allows us to formalize such properties of biological systems as diagnosability, detection ability and a presence of biological intruders and pathological changes. Resulting properties can be viewed as complementary to security ones. Moreover, these corresponding security properties are generalizations of several traditional ones and can detect security holes otherwise undetected.

Keywords: information flow, opacity, nested attackers, diagnosability.

1 Introduction

Biological systems are frequent inspirations for computational models of different nature (Neural network, P Systems, Calculus of Looping Sequences etc). Moreover, there are additional connections between biology and informatics. Inspirations and motivations from one area can be useful and fruitful in another area and vice versa. Among them an important role plays relationship of security of computational systems and such properties of biological systems as immunity, resistance, diagnosability and so on. The aim of this paper is to propose a formalism for analyses of biological systems specified by process algebras which enables us to define such properties as detection ability, diagnosability of presence of various biological intruders as viruses or pathological changes. These properties can be viewed as complementary ones to security properties. Hence, we also obtain rather general security properties which generalize several traditional ones.

The presented approach combines several ideas emerged from security theory as well as from modeling of biological systems. As regards security, we exploit an idea of an absence of information flow between public and private system's behaviour (see [GM82]). This concept has been many times exploited in various formalism. In security property called Non-Deductibility on Composition (NDC) it is assumed that system's actions are divided to private and public ones. An information flow between these two kinds of actions is expressed in the following way: a system has NDC property if for every high level user A (i.e. capable to perform only private i.e. high level actions), the low level view of the behaviour

* Work supported by the grant VEGA 1/0688/10.

(seeing only public i.e. low level actions) of P is not modified (in terms of weak trace equivalence) by the presence of A . In our approach we exploit an idea of intruders taken from NDC. Moreover we will consider several intruders which are differently nested inside a system (as it was done in [GMM10, Gru03]). This approach seems to be more suitable for investigation of biological systems.

The information flow will be formalized by opacity (see [BKR04]). Opacity again seems to be more suitable for biological systems since it can capture more complex information flow than just the flow between occurrences of private and public actions. Opacity has been also exploited for analyses of biological systems. By means of opacity a diagnosability (as a complementary concept to security) for P Systems (see [BGMM10]) has been defined. Note that opacity was already exploited for definitions of security properties for process algebras (see [Gru07]). Combining these two approaches we propose the formalism for analyses of biological systems which are specified by means of process algebras. As a side effect we obtain very general and strong security properties. We show that in general the proposed properties are undecidable but become decidable for some special cases. We consider this work as a preliminary step. Later on we plan to study some special settings and classes of systems and intruders for which the proposed properties can be checked in realistic time by software tools.

2 Context Process Algebra

In this section we define our working formalism - contexts process algebra (CPA). It is based on Milner's CCS (see [Mil89]) which is extended by placeholders to specify processes contexts. To define the language CPA, we first assume a set of atomic action symbols A not containing symbols τ , and such that for every $a \in A$ there exists $\bar{a} \in A$ and $\bar{\bar{a}} = a$. We define $Act = A \cup \{\tau\}$. We assume that a, b, \dots range over A and x, y, \dots range over Act . Assume the signature $\Sigma = \bigcup_{n \in \{0,1,2\}} \Sigma_n$, where

$$\begin{aligned} \Sigma_0 &= \{Nil\} \\ \Sigma_1 &= \{x. \mid x \in Act\} \cup \{[S] \mid S \text{ is a relabeling function}\} \\ &\quad \cup \{\backslash M \mid M \subseteq A\} \\ \Sigma_2 &= \{[, +\} \end{aligned}$$

with the agreement to write unary action operators in prefix form, the unary operators $[S], \backslash M$ in postfix form, and the rest of operators in infix form. Relabeling functions, $S : Act \rightarrow Act$ are such that $\overline{S(a)} = S(\bar{a})$ for $a \in A$, and $S(\tau) = \tau$.

The set of TPA terms over the signature Σ is defined by the following BNF notation:

$$P ::= X \mid \mathcal{A} \mid op(P_1, P_2, \dots, P_n) \mid \mu XP$$

where $X \in Var$, Var is a set of process variables, $\mathcal{A} \in PH$, PH is a set of process place holders, P, P_1, \dots, P_n are CPA terms, $\mu X-$ is the binding construct, $op \in \Sigma$. The set of CPA processes consists of closed CPA terms. The set of CCS processes consists of CPA processes without place holders.

Let P be a CPA process with (all) placeholders $\mathcal{A}_1, \dots, \mathcal{A}_n$. We will indicate this by $P[\mathcal{A}_1, \dots, \mathcal{A}_n]$. CCS process obtained from $P[\mathcal{A}_1, \dots, \mathcal{A}_n]$ by replacing placeholders \mathcal{A}_i by CCS processes A_i will be indicated by $P[A_1/A_1, \dots, A_n/A_n]$. Note that *Nil* will be often omitted from processes descriptions and hence, for example, instead of *a.b.Nil* we will write just *a.b*. A structural operational semantics for CPA terms is given by means of labeled transition systems (see [Mil89]). For $s = x_1.x_2.\dots.x_n, x_i \in Act$ we write $P \xrightarrow{s}$ instead of $P \xrightarrow{x_1} \xrightarrow{x_2} \dots \xrightarrow{x_n}$ and we say that s is a trace of P . The set of all traces of P will be denoted by $Tr(P)$. By ϵ we will denote the empty sequence of actions, by $Succ(P)$ we will denote the set of all successors of P and $Sort(P) = \{x|P \xrightarrow{s.x}$ for some $s \in Act^*$ and $x \neq \tau\}$. If the set $Succ(P)$ is finite we say that P is finite state. In the later we will use the weak trace equivalence (denoted \approx_w) and bisimulation (denoted \sim) (see [Mil89]).

Let us have a system described by CCS process P . Suppose that there are places in the system where an intruder or intruders can be put. We indicate those places by place holders and the resulting CPA process will be called its opening. The opening of process can be defined on syntactical or semantical level. For simplicity we will use the later one.

Definition 1. *Let P be a CCS process. Opening of P is any CPA process $Q[\mathcal{A}_1, \dots, \mathcal{A}_n]$ such that $P \sim Q[\mathcal{A}_1/Nil, \dots, \mathcal{A}_n/Nil]$.*

3 Diagnosable Intruders

The first inspiration for our work is the security property Non-Deducibility on Composition (NDC for short, see in [FGM03]). Suppose that all actions are divided in two groups, namely public (low level) actions L and private (high level) actions H i.e. $A = L \cup H, L \cap H = \emptyset$. Then process P has property NDC if for every high level user A , the low level view of the behaviour of P is not modified (in terms of weak trace equivalence) by the presence of A . The idea of NDC can be formulated in such a way that it is required that $(P|A) \setminus H \approx_w P \setminus H$ for every $A, Sort(A) \subseteq H \cup \{\tau\}$. Hence, in the case of NDC, only one attacker is considered and it communicates with the system on the top most level (non-nested attacker) and the system with and without the attacker are compared on level of weak traces (see Fig [1]).

Our formalism of context process algebra allows us to model several intruders which can be nested arbitrary inside the system. In style of NDC it would be required that $P \setminus H \approx_w P'[\mathcal{A}_1/A_1, \dots, \mathcal{A}_n/A_n] \setminus H$ for every opening $P'[\mathcal{A}_1, \dots, \mathcal{A}_n]$ of process P and every $A_i, Sort(A_i) \subseteq H \cup \{\tau\}, 1 \leq i \leq n$ (see Fig. [2]). Let us call such the property Nested Non-Deducibility (NND, for short).

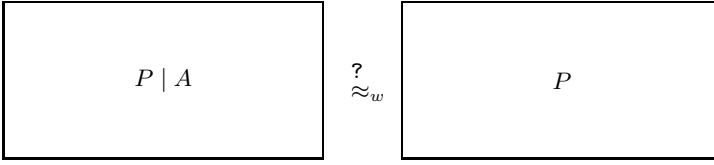


Fig. 1. Non-nested attacker

Example 1. In general we have $NND \subseteq NDC$ since clearly NDC is a special case of NND property. Let $P = l_1.Nil + (h.l_2.Nil) \setminus H$ It is easy to check that $P \in NDC$ but $P \notin NND$. Hence we have that $NND \subset NDC$.

Security property NND would be appropriate in case that an attacker can place several auxiliary processes inside the system in such a way that they can cause some information flow between private and public actions. But since for biological systems division of actions to two static groups (one type of actions cannot be observed and another one is always observed) is not appropriate. Hence instead of Non-Deducibility on Composition we will exploit more general concept opacity (see [BKR04]). First we define observation function on sequences from Act^* . The observation function is any function $\mathcal{O} : Act^* \rightarrow \Theta^*$ where Θ is a non-empty set of elements called observables. In [BKR04] observable functions are divided to static/dynamic/orwellian/m-orwellian ones. In the case of the static observation function each action is observed independently from its context. In the case of the dynamic observation function an observation of an action depends on the previous ones, in the case of the orwellian and m-orwellian observation function an observation of an action depends on the all and on m previous actions in the sequence, respectively. The static observation function is the special case of m-orwellian one for $m = 1$. Note that from the practical point of view the m-orwellian observation functions are the most interesting ones. An observation expresses what an observer - eavesdropper can see from a system behaviour and we will alternatively use both the terms (observation - observer) with the same meaning.

Now suppose that we have some security property. This might be an execution of one or more classified actions, an execution of actions in a particular classified order which should be kept hidden, etc. Suppose that this property is expressed by predicate ϕ over process traces. We would like to know whether an observer can deduce the validity of the property ϕ just by observing sequences of actions from Act^* performed by given process. The observer cannot deduce the validity of ϕ for P if for every trace w of P such that $\phi(w)$ holds, there exists trace w' such that $\neg\phi(w')$ and the traces cannot be distinguished by an observer (see Fig. 3). We formalize this concept by opacity.

Definition 2 (Opacity). *Given process P , a predicate ϕ over Act^* is opaque w.r.t. the observation function \mathcal{O} if for every sequence $w, w' \in Tr(P)$ such that $\phi(w)$ holds and $\mathcal{O}(w) \neq \epsilon$, there exists a sequence $w', w' \in Tr(P)$ such that*

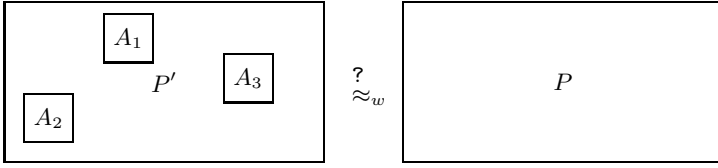


Fig. 2. Nested attacker

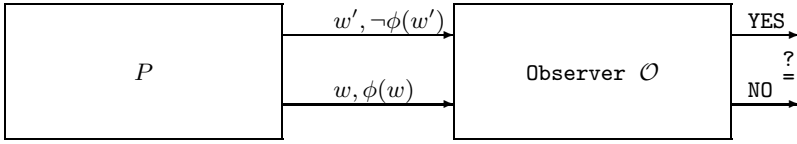


Fig. 3. Opacity observer

$\neg\phi(w')$ holds and $\mathcal{O}(w) = \mathcal{O}(w')$. The set of processes for which the predicate ϕ is opaque with respect to \mathcal{O} will be denoted by $Op_{\mathcal{O}}^{\phi}$.

Now we are ready to define diagnosability of several nested intruders. In a sense it is complementary property with respect to opacity.

Definition 3 (Diagnosable intruders). Given CPA process $P[A_1, \dots, A_n]$ and a set $V, V = \{A_1, \dots, A_n\}$ of CCS processes called intruders. We say that the intruders V are diagnosable by a predicate ϕ over Act^* and by the observation function \mathcal{O} if $P[A_1/A_1, \dots, A_n/A_n] \notin Op_{\mathcal{O}}^{\phi}$.

Diagnosability of intruders assumes that we know possible holes (place holders in our formalism) for the intruders in a system specification (as CPA term) and a set of intruders. This is not always the case and hence we define diagnosability for CCS processes.

Definition 4 (Strongly diagnosable intruders). Given CCS process P and a set $V, V = \{A_1, \dots, A_n\}$ of CCS processes called intruders. We say that the intruders V are strongly diagnosable by a predicate ϕ over Act^* and by the observation function \mathcal{O} if for every opening every P' which is opening of P it holds $P'[A_1/A_1, \dots, A_n/A_n] \notin Op_{\mathcal{O}}^{\phi}$.

Strong diagnosability of intruders assumes that we know a set of intruders what is again not always the case. Hence we define diagnosability for unknown set of intruders.

Definition 5 (Strong diagnosability for processes). Given CCS process P and the observation function \mathcal{O} . We say that P is strongly diagnosable by a

predicate ϕ over Act^* if there exists a set V such that V is strongly diagnosable by ϕ and \mathcal{O} .

The above mentioned properties have different strengths as regards diagnosability as well as their complements have different strengths as security properties. The later is expressed formally in the following theorem but a similar theorem would hold also for diagnosability properties.

Theorem 1. *Let SDP^C denotes the subset of CCS processes which have not strong diagnosability property, NND and NDC denote process with Nested Non-Deducibility and Non-Deductibility on Composition property, respectively. Then the following holds:*

$$SDP^C \subset NND \subset NDC$$

Let SDI^C and DI^C denote CCS processes which have not Strongly diagnosable intruders and Diagnosable intruders properties, respectively. Then the following holds:

$$SDP^C \subset SDI^C \subset DI^C.$$

Proof. Sketch. Let as consider static observation function which maps all private actions and τ action to ϵ . For such observation function we get an observer corresponding to NDC observer. To check the rest of inclusion is quit straightforward as well is to show that they are proper.

The previous theorem can be illustrated by simple Venn diagram (see Fig. 4). As regards decidability even the weakest of the diagnosable properties is in general undecidable.

Theorem 2. *Intruders diagnosability is undecidable.*

Proof. Sketch. The proof is based on the fact that opacity is undecidable for CCS processes (see [Gru07]). We can find process P , its opening, observation function and a set of intruders in such a way that diagnosability opacity is undecidable.

Clearly from Theorem 2 we have the following corollary. Similar property can be formulated also for NDC a NND .

Corollary. Strong diagnosability for intruders and for processes is undecidable.

Now we will examine situations when the above mentioned properties are decidable. One possibility is to limit strength of a corresponding predicate.

Definition 6. *Let as define predicate ϕ over traces to be set defined if there exists a set $D, D \subset Act$ such that $\phi(w)$ holds if w contains an element from D .*

In fact, set defined predicates can detect an occurrence of private action what is the main concern of traditional security properties. For such predicates strong diagnosability of intruders becomes decidable under some special conditions.

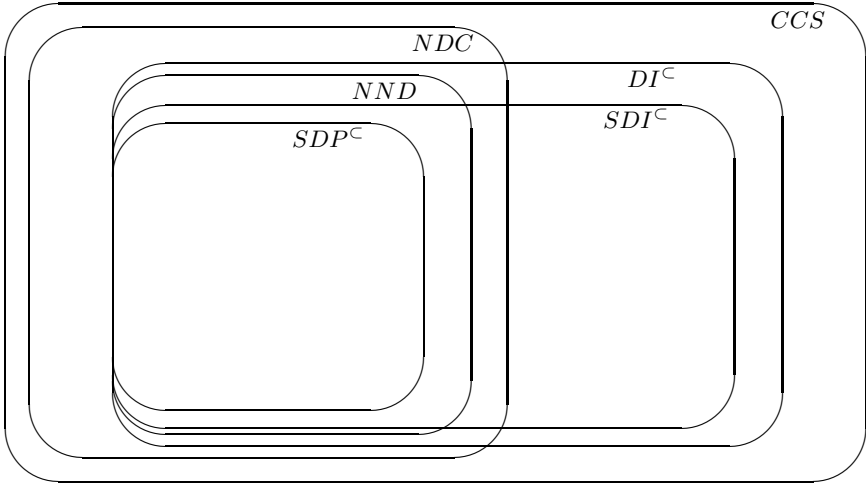


Fig. 4. Properties Hierarchy

Theorem 3. *Strong diagnosability of intruders is decidable for finite state process and observation functions \mathcal{O} such that $\mathcal{O}(x) \neq \epsilon$ and $\mathcal{O}(x_1 \dots x_n) = \mathcal{O}(x_1) \dots \mathcal{O}(x_n)$ for every $x, x_i \in \text{Act}$ (i.e. static observation function which cannot hide anything completely) and for set defined predicates.*

Proof. Main idea. There is only a finite number of non-bisimilar openings and since the predicate is set defined and observations cannot hide any action completely we can try all possible traces.

4 Conclusions

We have defined the formalism for analyzes of biological systems specified by process algebras. Properties as (strongly) diagnosable intruders and strong diagnosability for processes can be also seen as complementary properties to security ones. In fact, to all of them correspond some (either specific or rather general) security properties. Moreover, many already known and studied security properties can be seen as their special cases (for example NDC and NND).

All these properties assume attacks (changes of behaviour) based on a nested presence of cooperating or non-cooperating intruders (viruses, degenerated or mutated parts and so on). This would naturally correspond to malicious software components (software viruses, Trojan horses and so on) embedded to systems.

As regards decidability properties, as one way how to extend the result from Theorem 3, we could consider the most powerful attackers (see [FGM03]) or a technique of *Generalized Unwinding* (see [BFPR03]). To get decidability properties one can also limit power of diagnoser/attacker by restricting observation

function. Moreover, another direction of research might be to study different behaviour of $P[\mathcal{A}_1/A_1, \dots, \mathcal{A}_n/A_n]$ with respect to $P[\mathcal{A}_1/Nil, \dots, \mathcal{A}_n/Nil]$. This approach would be closer to NND but it can be further developed by opacity techniques.

References

- [BGMM10] Barbuti, R., Gruska, D.P., Maggiolo-Schettini, A., Milazzo, P.: A notion of biological diagnosability inspired by the notion of opacity in systems security. *Fundamenta Informaticae* 102(1), 19–34 (2010)
- [BFPR03] Bossi, A., Focardi, R., Piazza, C., Rossi, S.: Refinement Operators and Information Flow Security. In: Proc. of SEFM 2003. IEEE Computer Society Press (2003)
- [BKR04] Bryans, J., Koutny, M., Ryan, P.: Modelling non-deducibility using Petri Nets. In: Proc. of the 2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP 2004) (2004)
- [BKMR06] Bryans, J., Koutny, M., Mazaré, L., Ryan, P.Y.A.: Opacity Generalised to Transition Systems. In: Dimitrakos, T., Martinelli, F., Ryan, P.Y.A., Schneider, S. (eds.) FAST 2005. LNCS, vol. 3866, pp. 81–95. Springer, Heidelberg (2006)
- [FGM03] Focardi, R., Gorrieri, R., Martinelli, F.: Real-Time information flow analysis. *IEEE Journal on Selected Areas in Communications* 21 (2003)
- [GMM10] Gorrieri, R., Martinelli, F., Matteucci, I.: Specification and Analysis of Information Flow Properties for Distributed Systems (2010) (submitted for publications)
- [Gru03] Gruska, D.P., Maggiolo-Schettini, A.: Nested Timing Attacks. In: Proceedings of FAST 2003, Pisa, pp. 147–161 (2003)
- [Gru07] Gruska, D.P.: Observation Based System Security. *Fundamenta Informaticae* 79(3-4), 335–346 (2007)
- [GM82] Goguen, J.A., Meseguer, J.: Security policies and security models. In: Proc. of IEEE Symposium on Security and Privacy (1982)
- [Mil89] Milner, R.: Communication and concurrency. Prentice-Hall International, New York (1989)

Dynamics and Convergence of Resource Prices in Market-Oriented Overlay Networks

Yutaka Okaie and Tadashi Nakano

Frontier Research Base for Global Young Researchers
Frontier Research Center, Graduate School of Engineering,
Osaka University, 2-1 Yamadaoka, Suita,
Osaka 565-0871, Japan
{yokaie,tnakano}@wakate.frc.eng.osaka-u.ac.jp

Abstract. Overlay networks use computational powers available at the edges of the Internet to provide large-scale computing and networking environments. In this paper, we consider market-oriented overlay networks in which peers trade resources and services through a common currency. From the perspective of market economics, the dynamics and convergence of resource prices in such networks is investigated by simulation and mathematical modeling, and the results are briefly reported in this short paper.

Keywords: Overlay networks, resource markets, mobile agents, peer-to-peer networks.

1 Introduction

Our model of market-oriented overlay networks consists of three types of peers: (1) service agents that implement specific network services, (2) resource platforms that provide execution environments and supporting facilities for service agents, and (3) users that use services provided by service agents. In this model, peers trade resources and services by exchanging a common currency. For example, a peer (a platform) provides its resources to a service agent and receives the price from the service agent; the peer later uses the earnings to buy services available in the network.

When rationally designed, resource or service providing peers set prices for own resources/services to maximize their monetary income. As in financial markets, high rates of inflation or deflation of resources and services could disrupt the market and are not preferable, whereas it is in general a challenging task to maintain stable prices without central authority. This paper reports our recent results obtained from simulation and analytical studies to understand price dynamics in such networks.

2 Results

A variant of our previously published models [1][2] was used to simulate price dynamics in peer-to-peer resource markets. The conditions satisfied at Nash

Equilibria that lead to long-term price stability were mathematically analyzed and identified. Figure 1 shows representative results from our simulation and analytical studies, demonstrating that the theoretical results correspond well with the simulation results at steady-states. Further simulation and analysis are currently being conducted to understand the price dynamics in dynamic network environments.

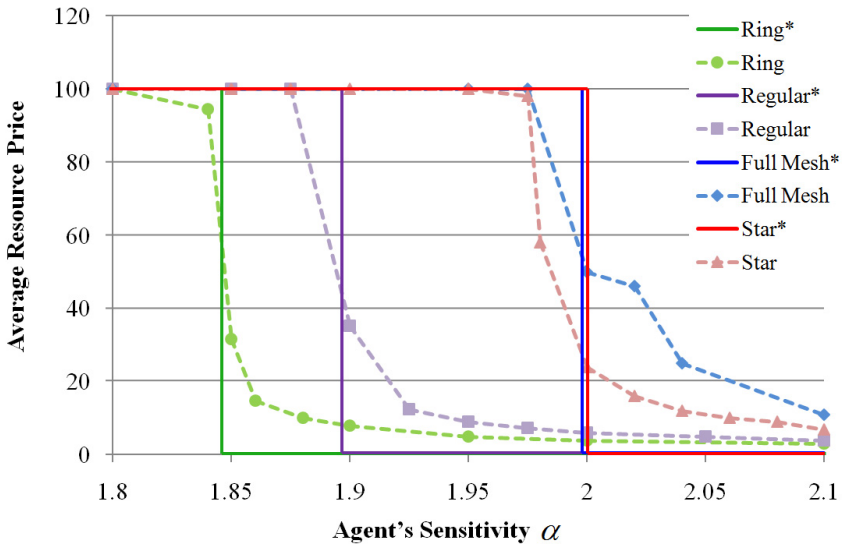


Fig. 1. Resource price dynamics investigated through simulation and theoretical studies. The vertical axis represents the average resource prices in the network of platforms, and the horizontal axis a model parameter that determines the agent behavior (see [2]). Typical network topologies are assumed, including ring, regular, full mesh, and star topologies. For each network, the dashed line represents the simulation results, and the solid line the theoretical ones.

References

1. Nakano, T., Okaie, Y.: Cooperative resource pricing in service overlay networks for mobile agents. *IEICE Transactions on Communications E93-B(7)*, 1927–1930 (2010)
2. Okaie, Y., Nakano, T.: Modeling and simulation of market-oriented service overlay networks. In: *Proceedings of 13th International Conference on Network-Based Information Systems (NBiS)*, pp. 125–132 (2010)

Internet as a Dataflow Computer

Hiroyuki Ohsaki¹, Hideaki Suzuki², and Hidefumi Sawai²

¹ Graduate School of Information Science and Technology, Osaka University, Japan
oosaki@ist.osaka-u.ac.jp

² Kobe Advanced ICT Research Center (KARC), National Institute of Information and
Communications Technology (NICT), Japan
{suzuki, sawai}@nict.go.jp

Abstract. This paper proposes a novel dataflow architecture called *DFNET* (*DataFlow architecture on the interNET*), which realizes a scalable dynamic dataflow computer on a packet switching network. A vast amount of research on dataflow computers has been extensively performed during, in particular, 1970s and 1980s as a promising approach for realizing very high-speed computers. In spite of the large amount of expectations, success of dataflow computer researches is quite limited. The objective of this paper is to present the concept of a scalable and extensible dataflow architecture on a packet switching network, which utilizes the abundant resources of a large-scale computer network. In this paper, we introduce the concept of DFNET (DataFlow architecture on the interNET). DFNET is composed of configuration and control methodologies of routers in a packet switching network. The key of DFNET is that a packet switching network is utilized not for *end-to-end communication* but for *dataflow computing*. Because of desirable properties of a packet switching network, DFNET has high scalability in terms of the dataflow program size and high robustness against failures.

1 Introduction

Dataflow computer is computer architecture, which is significantly different from the conventional von Neumann architecture. In dataflow computers, computing is driven not by the program counter but by data themselves [12]. Dataflow computer is expected to solve the performance bottleneck of the von Neumann architecture with massively parallel program execution. A vast amount of research on dataflow computers has been extensively performed during, in particular, 1970s and 1980s as a promising approach for realizing very high-speed computers [12]. In von Neumann computers, which are also referred as *control flow* systems, every instruction pointed by the program counter is sequentially fetched by a processor for execution. On the contrary, dataflow computers have no program counter. In dataflow computers, operations are *fired* immediately when all required data (i.e., operands) are available.

In what follows, a classical data-driven dataflow model proposed by Dennis *et al.* [3] is briefly explained. In dataflow computers, a program to be executed is represented as a directed graph (*dataflow program*), which is composed of nodes (i.e., data and functions) and arcs (i.e., data flows) (Fig. 1). Several tokens, each of which holds datum to be processed, are transferred on arcs connecting nodes. Data in those tokens are updated

at every node. A node without input arcs generates a datum; i.e., a token containing the datum is created and it is then transferred to the downstream node. A node with one or more input arcs receives tokens from upstream nodes. Once all tokens are received, the node performs a predefined operation for data contained in tokens. A token containing the result is created and it is then transferred to the downstream node. There are two types of arcs: *data arc* and *control arc*. The data arc carries tokens with arbitrary datum whereas the control arc does token with a Boolean value.

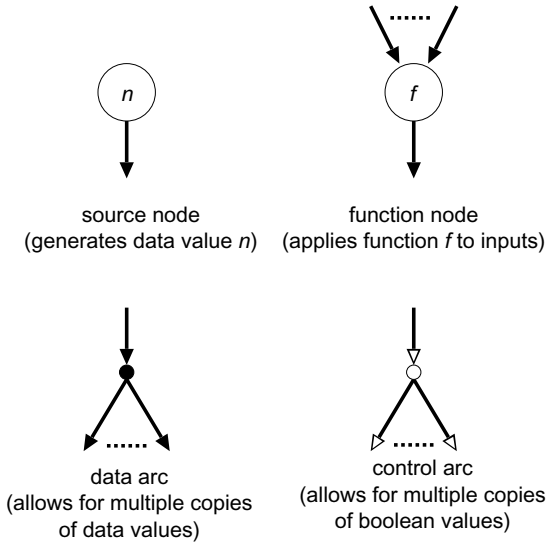


Fig. 1. Primitive nodes in Dennis's notation [3]

In 1970s and 1980s, the main concern of dataflow computer researchers was primarily to build a much faster computer than the conventional von Neumann computers. A number of dataflow architectures and prototype systems have been published (see [1] and references therein). The key for realizing a very fast dataflow computer is in efficient parallel processing of program execution. In the literature, there exist several approaches for accelerating parallel processing of program execution. The classical dataflow architecture proposed by Dennis *et al.* is classified as *static dataflow architecture* or equivalently *single-token-per-arc* architecture [12]. In static dataflow architecture, only a single token can be transferred through an arc. Static dataflow architecture therefore limits the parallelism of program execution; i.e., loops and recursions must be performed sequentially. Inefficient parallelism of static dataflow architecture results in significantly slow program execution. *Dynamic dataflow architecture* extends static dataflow architecture to allow multiple tokens on an arc by, for example, adding a tag to every token or replicating a subgraph of the dataflow program [12]. Dynamic dataflow architecture therefore enables simultaneous execution of loops and recursions, which significantly accelerates the program execution. Several dynamic dataflow systems have

been proposed and studied in the literature [12], in which either the tag format or the replication method of a dataflow subgraph is different.

In spite of the large amount of expectations, success of dataflow computer researches is quite limited. Dataflow computers were originally expected as a new computer architecture, which hopefully excelled the limitation of the conventional von Neumann architecture. The von Neumann architecture, however, has been continuously improved with several technological innovations such as the advancement and enhancement of semiconductor technologies. Researches on the von Neumann architecture have successfully caught up with the growing demand of computing resources. On the other hand, researches on dataflow computers have faced a difficulty. For instance, software development environment for dataflow programs has not been fully matured. Programming languages for dataflow programs are generally designed based on a *single static assignment paradigm*, which considerably limits the freedom of software development. Also parallel compilers for dataflow programs is not easy to develop, so that users of dataflow computers are little benefited from massively parallel program execution.

In this paper, we propose a novel dataflow architecture called *DFNET (DataFlow architecture on the interNET)*, which realizes a scalable dynamic dataflow computer on a packet switching network. Note that the objective of this paper is not to present yet another architecture for realizing a *high-speed* dataflow computer. Instead, this paper aims to present the concept of a *scalable and extensible* dataflow architecture on a very large-scale packet switching network. A packet switching networks such as the Internet is *best-effort*. Namely, QoS (Quality of Service) such as the speed and the delay of data transfer is not guaranteed. Conversely, packet switching network achieves efficient utilization of networking resources by scarifying the communication quality. It is therefore not desirable to build a dataflow computer on a packet switching network if one needs a very high-speed dataflow computer. We believe that a packet switching network such as the Internet, which has been exponentially expanding both in speed and size, should be a viable platform for realizing a (not so fast but) very large-scale dataflow computer.

DFNET is composed of configuration and control methodologies of routers in a packet switching network. A packet switching network has high scalability in terms of the network size (i.e., the number of routers and links) as well as high robustness against router and/or link failures. The key of DFNET is that a packet switching network is utilized not for *end-to-end communication* but for *dataflow computing*. Because of desirable properties of a packet switching network, DFNET has high scalability in terms of the dataflow program size and high robustness against failures. Also, utilization of the dynamic routing mechanism and network virtualization technologies realize a highly flexible and extensible dataflow computer.

Note that this paper only covers the concept and high-level view on building blocks of DFNET. In other words, several practical issues such as implementation, deployment, security, and software development environment are not covered. The core of DFNET assumptions is that the underlying network is a packet switching network. Although detailed discussion on DFNET implementation is beyond the scope of this paper, it should be worth considering possible DFNET implementations. DFNET might be

implemented, for instance, either as an extension to IP routers [4], an application-level overlay network [5], or an application for active networking architecture [6].

The organization of this paper is as follows. Section 2 presents the overview of DFNET (DataFlow architecture on the interNET). We also explain four building blocks of DFNET: *token communication mechanism*, *token synchronization mechanism*, *data processing mechanism*, and *token routing mechanism*. Finally, Section 3 concludes this paper and discusses future works.

2 DFNET (DataFlow architecture on the interNET)

2.1 Overview

We first introduce the overview of DFNET (DataFlow architecture on the interNET), which is a novel architecture for realizing a scalable and dynamic dataflow computer on a packet switching network.

DFNET is essentially one of token-based dynamic dataflow architectures. DFNET builds a dataflow computer on a packet switching network, which is composed of several routers and links.

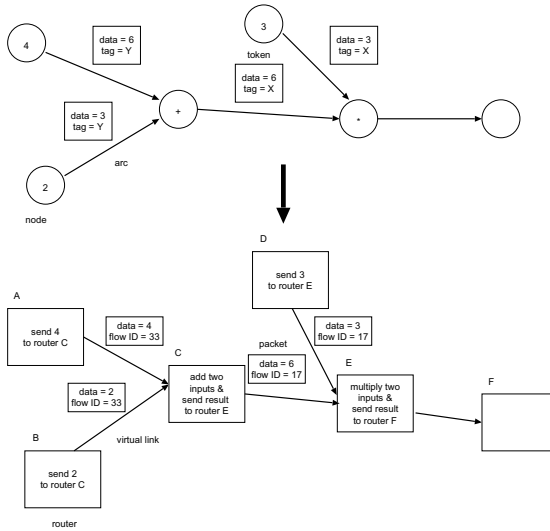


Fig. 2. DFNET (DataFlow architecture on the interNET) overview; in DFNET, a dataflow program is directly mapped to a packet switching network

In DFNET, a dataflow program is directly mapped to a packet switching network. Tokens, nodes, and arcs in a dataflow program are mapped to *packets*, *routers*, and *virtual links* in a packet switching network. A set of nodes in a dataflow program are assigned to a router in a packet switching network. A token in a dataflow program is

encapsulated in a packet, and packets containing tokens are transferred between routers. The key of DFNET is that a packet switching network is utilized not for *end-to-end communication* but for *dataflow computing*.

In conventional dataflow architecture, such as MIT’s Dynamic Architecture [7] and Manchester Architecture [8], separate PEs (Processing Elements) are in charge of data processing. Multiple PEs are therefore interconnected through a communication network. On the contrary, routers in DFNET performs both data processing and communication. In other words, PEs of a dataflow computer are embedded in the router. Integration of data processing and communication in a router greatly simplifies the architecture. It also enables direct mapping of a dataflow program onto a packet switching network. In DFNET, a dataflow program to be executed is split into multiple subgraphs, each of which is assigned to a router. In the followings, we will explain the case of one-to-one mapping of a node in the dataflow program to a router in the packet switching network. The case of many-to-one mapping can be easily realized by utilizing loopback interfaces in a router (i.e., by transferring a token within the router).

DFNET has four building blocks: *token communication mechanism*, *token synchronization mechanism*, *data processing mechanism*, and *token routing mechanism*.

2.2 Token Communication Mechanism

In DFNET, token communication from an upstream node to a downstream node in a dataflow program is realized by encapsulating a token in a packet and transferring the token between routers (see Fig. 3). Specifically, the address of a router, to which the downstream node is assigned, is written in the destination address field of the packet header. Also the token containing datum and tag is stored in the payload of the packet.

In a packet switching network, the destination address field of the packet header usually stores the address of a destination host for realizing end-to-end communication. On the contrary, DFNET stores the address of the next router in the destination address field of the packet header. Namely, the packet switching network is utilized not for end-to-end communication between hosts but for hop-by-hop communication among routers.

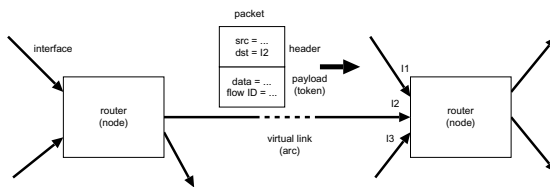


Fig. 3. DFNET token communication mechanism; token communication from an upstream node to a downstream node in a dataflow program is realized by encapsulating a token in a packet and transferring the token between routers

2.3 Token Synchronization Mechanism

In dataflow computers, a node is *fired* immediately when tokens from all input arcs are available. DFNET uses a *flow table* at a router for token synchronization (see Fig. 4); i.e., every token is assigned a globally unique identifier called *flow ID*, and a router maintains availability of tokens by updating records corresponding to their flow ID's.

For token synchronization at a router, every token is assigned a unique flow ID. A record of the flow table corresponding to the flow ID represents *token count* (i.e., the number of tokens received) and *firing condition* (i.e., the number of token required for firing). The flow ID, token count, and firing condition are determined from the dataflow program.

When a router receives a packet, the router searches the flow table for the flow ID specified in the payload of the packet. The router then increments the token count of the record in the flow table. If the token count is equal to the firing condition of the record, the router is *fired*; i.e., the data processing mechanism is invoked and the token count is set to zero.

2.4 Data Processing Mechanism

In DFNET, capability of PEs are embedded in a router; i.e., data processing is performed at the router. Data processing at the router can be realized either by utilizing internal computing resource in the router or providing external computing resource outside of the router (see Fig. 5).

In the case of internal computing resource, an operation for tokens is determined by the synchronization mechanism. One of internal PEs performs the specified operation, and the result is sent to the token routing mechanism, which will be explained below.

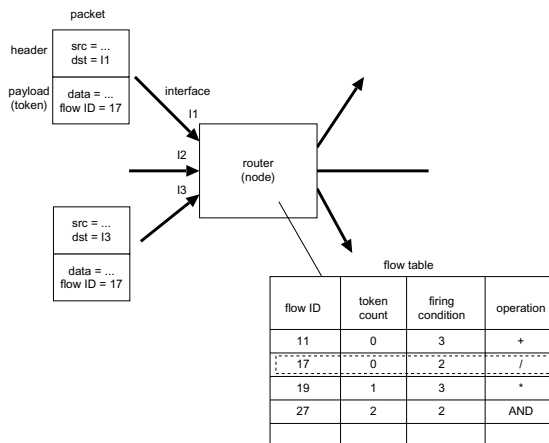


Fig. 4. DFNET token synchronization mechanism; every token is assigned a globally unique identifier called *flow ID*, and a router maintains availability of tokens by updating records corresponding to their flow ID's

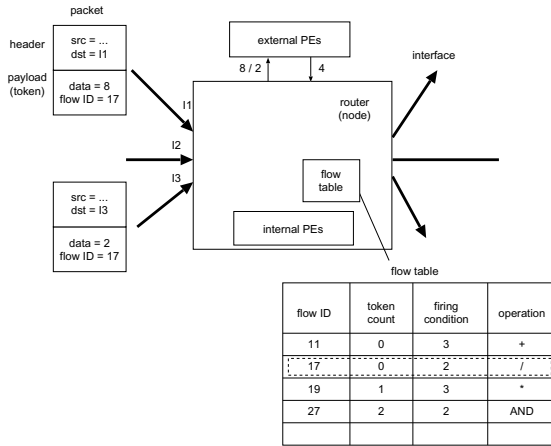


Fig. 5. DFNET data processing mechanism; data processing at the router can be realized either by utilizing internal computing resource (internal PEs) or providing external computing resource (external PEs)

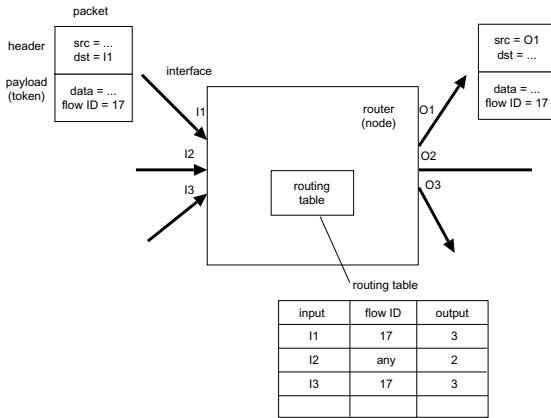


Fig. 6. DFNET token routing mechanism; DFNET utilizes a routing table in the router for determining the next-hop router, to which the downstream node in the dataflow program is assigned

Also in the case of external computing resource, an operation for tokens is determined by the synchronization mechanism. One of external PEs receives data and type of operations, and the result is sent to the token routing mechanism.

Note that both internal and external PEs can be used according to the type of data processing. For instance, for simple and/or realtime operations, internal PEs would be appropriate. On the other hand, for complex and non-realtime operations, external PEs would be appropriate. Introduction of external PEs slightly complicates the router architecture while the flexibility and expandability of the dataflow computer can be realized.

2.5 Token Routing Mechanism

DFNET utilizes a routing table in the router for determining the next-hop router, to which the downstream node in the dataflow program is assigned (see Fig. 6).

Each record of the routing table is a triplet of input interface, flow ID, and output interface. Based on the input interface of an arriving token and the flow ID written in the token, the router determines the output interface from the corresponding record of the routing table. If multiple output interfaces are found, the router replicates the token, and sends those tokens through all output interfaces.

Each record of the routing table is directly determined from the dataflow program. Note that change of the dataflow program during program execution can be easily realized by dynamically modifying the routing table.

3 Conclusion

In this paper, we have proposed a novel dataflow architecture called *DFNET (DataFlow architecture on the interNET)*, which realized a scalable dynamic dataflow computer on a packet switching network. The objective of this paper was not to present yet another architecture for realizing a *high-speed* dataflow computer. Instead, this paper aimed to present the concept of a *scalable and extensible* dataflow architecture on a very large-scale packet switching network. DFNET is composed of configuration and control methodologies for routers in a packet switching network. In this paper, we have explained four building blocks of DFNET: token communication mechanism, token synchronization mechanism, data processing mechanism, and token routing mechanism. Because of desirable properties of a packet switching network, DFNET has high scalability in terms of the dataflow program size and high robustness against failures.

Our future work includes architectural comparison of DFNET with other dynamic dataflow architectures, designing router architecture for DFNET, qualitative and quantitative performance analysis of DFNET, and experiments with prototype implementation of DFNET.

Moreover, extension of DFNET to support several types of different computing models than dynamic dataflow architecture would be of great importance. In [9,10,11], we have proposed ATN (Algorithmically Transitive Network), which is a self-organizing dynamic dataflow network with a learning mechanism. ATN adopts a BP (Back Propagation) learning mechanism similar to that of neural networks. DFNET has been initially designed as an execution environment for ATN. We should note, in particular, that such a BP-based learning mechanism can be easily implemented with the token routing mechanism in DFNET.

References

1. Sharp, J.A. (ed.): Dataflow computing: theory and practice. Ablex Publishing Corporation (1992)
2. Silč, J., Robič, B., Ungerer, T.: Processor architecture: from dataflow to superscalar and beyond. Springer, Heidelberg (1999)

3. Dennis, J.B., Misunas, D.P.: A preliminary architecture for a basic dataflow processor. In: Proceedings of the 2nd Annual Symposium on Computer Architecture, pp. 126–132 (1975)
4. Baker, F.: Requirements for IP version 4 routers. Request for Comments (RFC) 1812 (June 1995)
5. Lua, E.K., Crowcroft, J., Pias, M., Sharma, R., Lim, S.: A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials* 7, 72–93 (2005)
6. Bhattacharjee, S., Calvert, K.L., Zegura, E.W.: An architecture for active networking. Tech. Rep., College of Computing, Georgia Institute of Technology, Atlanta, Georgia (1996)
7. Gostelow, A.K.P., Plouffle, W.: An asynchronous programming language and computing machine. Tech. Rep. 114a, Department of Information and Computer Science, University of California, Irvine, CA (1978)
8. Gurd, J.R., Watson, I.: A multilayered dataflow computer architecture. In: Proceedings of 7th International Conference on Parallel Processing (August 1977)
9. Suzuki, H., Ohsaki, H., Sawai, H.: A Network-Based Computational Model with Learning. In: Calude, C.S., Hagiya, M., Morita, K., Rozenberg, G., Timmis, J. (eds.) UC 2010. LNCS, vol. 6079, pp. 193–193. Springer, Heidelberg (2010)
10. Suzuki, H., Ohsaki, H., Sawai, H.: Algorithmically Transitive Network: a new computing model that combines artificial chemistry and information-communication engineering. In: Proceedings of the 24th Annual Conference of Japanese Society for Artificial Intelligence (JSAI), pp. 2H1–OS4–5 (2010)
11. Suzuki, H., Ohsaki, H., Sawai, H.: An agent-based neural computational model with learning. In: Conference Abstract of the 3rd INCF Congress of Neuroinformatics (Neuroinformatics 2010) (2010), doi:10.3389/conf.fnins.2010.13.00021

Biologically Inspired Modeling of Smart Grid for Dynamic Power-Flow Control

Hidefumi Sawai¹, Hideaki Suzuki¹, and Hiroyuki Ohsaki²

¹ National Institute of Information and Communications Technology, Japan
{sawai,hsuzuki}@nict.go.jp

² Graduate School of Osaka University, Suita 565-0871 Japan
oosaki@ist.osaka-u.ac.jp

Abstract. Smart grid is an electric power network which enables an effective use of electric power in a highly parallel distributed manner. We have first formulated the basic equations for the smart grid by inspiring from the mechanisms in biological organism, and controlled the power-flow dynamically in the smart grid by monitoring an objective function, which reflects the power-flow and the constraint imposing on the power nodes. To validate the operation of the smart grid, we performed several simulation experiments: which include the operations of a conventional power network, a microgrid (comprises eight power nodes), and a smart grid (comprises three microgrids integrated into the conventional power network) both in *synchronous and asynchronous manners* for the operation of power nodes. Furthermore, even for the case of power failure such as outage, power recovery can be automatically achieved through bypass connections similar to synaptic interconnections in a dynamic function of brain. This kind of flexible function in the smart grid makes it possible to promote the introduction of renewable energy, such as solar energy, wind energy, and biomass energy rather than fossil energy.

Keywords: smart grid, dynamic control, highly distributed asynchronous system, synaptic connection, brain function.

1 Introduction

Smart grid is an intelligent power network which recently attracted a lot of attention because it enables many things to positively affect the environment on the Earth [3]-[13]. Fig. 1 shows the evolution from a conventional power network (a) to a smart grid with microgrids (b) [5]. The conventional power network is generally a commercial network which includes power plants, power generators, and power networks connected to demand sites. This network is very reliable and robust against a small accident in a local area. In reality, there were no experiences of outage except for a large-scale disaster in Japan. On the other hand, from the viewpoint of an efficient use of energy on the demand side, as a simultaneous use of electric and thermal energy increases an integrated power-efficiency, i.e, reducing energy costs for users on the demand side, many on-site

distributed power supplies have already been introduced in the neighborhood of the demand sites. Furthermore, very recently, from the viewpoint of prevention against the global warming, the power plant systems based on renewable energy such as solar energy, wind energy, and biomass energy have been introduced. This situation is shown in Fig. 1(b). In this figure, many power supplies including batteries or storages, and demand sides are distributed in the microgrid power networks and integrated into the conventional power network through the transformer and distribution substations as a whole power system. There are many *pro-and-cons* on the introduction of smart grid: for examples, it is not observable nor controllable from the conventional power networks, an increase of demand on the power efficiency on the demand sides, and the balance between them should be adjusted. However, we should plan to complementarily promote the symbiosis between the smart grid and the conventional power network for the future of the Earth.

The microgrid basically comprises (a)power (both electric and thermal) supplies, (b)power storage equipments, (c)power network, (d) a control system based on IT (Information Technology), and (e)power demand. The big difference between the commercial power network and smart grid is the application of thermal energy, and the existence of power storages, especially batteries. As the number of demand sites in the microgrid is small, and the fluctuation in power supply becomes large, the power-flow control should be adequately operated in real-time. Also, as the power supply itself is unstable due to the climate change in the case of natural energy such as solar energy and wind energy, it is always necessary to improve the energy supply. The previous studies, e.g., the paper [10] overviews several areas of computational intelligence techniques in the smart grid of the future. The paper [12] claimed computational intelligence for smart grid without showing any concrete methods, and the paper [13] focused on the automatic electric load detection. The paper [11] proposed a domestic energy management methodology for optimizing efficiency in smart grid. In this paper, we will show a dynamic control method of power-flow in microgrids and then integrate them into a public (infrastructural) power network using simulation experiments including a power failure such as outage. Even in such an unusual situation, we will show that automatic recovery of power supply will be achieved through bypass connections.

These kinds of situations for the smart grid to dynamically adapt to a changing environment would remind us of the corresponding biological functions in several hierarchical levels, which include: (1)a gene regulatory network for the metabolism of a cell [1] in the micro-level for responding to the demand in micro-grid. (2)each power node could be regarded as each neuron in a brain network [2], and the change of synaptic plasticity between neurons would correspond to the switching of power supply between the power nodes. (3)If we regard each micro-grid as each cell in a biological creature, some microgrids would form a multi-cell creature, and the smart grid would form an eco-system in the macro-level whose network architecture would always adapt to a dynamically changing environment. (4)the evolution of life would correspond to the evolution of smart grid [5],

where each power node (usually *alive* and sometimes *dead*) seems to behave like each agent or individual in a macroscopic eco-system.

The above mentioned analogical inspiration would lead us to design the architectures and analyse the functions of a smart grid in the later sections.

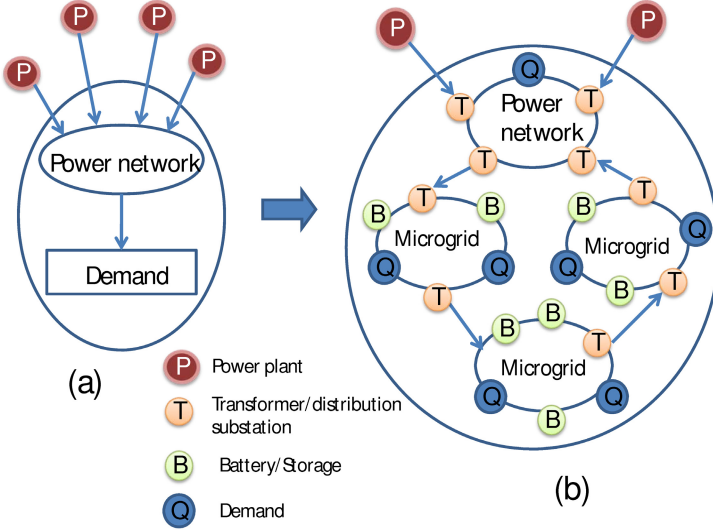


Fig. 1. Evolution of smart grid power network [5]

2 Microgrid Power Network

2.1 Basic Power-Flow Equations

The basic equations for the microgrid power network shown in Fig. 2 are defined as follows. This network architecture can adapt to any situation in the microgrid without loss of generality.

$$B_j(t) = \sum_i w_{ij}(t)P_i(t) + \sum_{j'} w_{j'j}(t)B_{j'}(t), \quad (1)$$

$$B_{j'}(t) = \sum_i w_{ij'}(t)P_i(t), \quad (2)$$

$$Q_k(t) = \sum_j w_{jk}(t)B_j(t) + \sum_i w_{ik}(t)P_i(t) + \sum_{j'} w_{j'k}(t)B_{j'}(t), \quad (3)$$

where, $B_j(t)$ is the power in the j^{th} battery or storage node, and $B_{j'}(t)$ is the power in the j'^{th} battery or storage node which interconnects with $B_j(t)$. $P_i(t)$ is

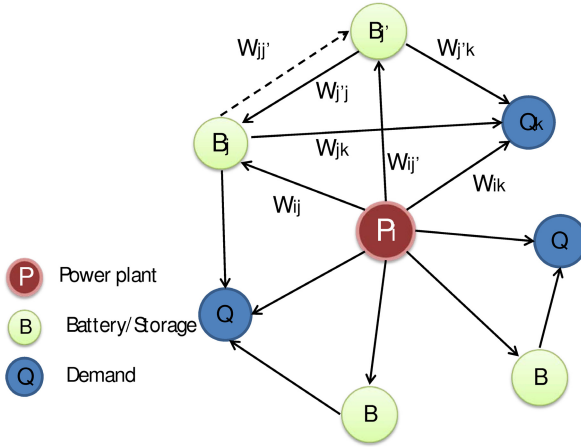


Fig. 2. Model of microgrid power network

the power in the i^{th} power plant. The interconnection $w_{ij}(t)$ between the nodes $P_i(t)$ and $B_j(t)$ represents the power-flow ratio between 0 and 1 for the two nodes. $Q_k(t)$ is the power at the k^{th} demand node. The interconnection $w_{jk}(t)$ represents the power-flow ratio for nodes $B_j(t)$ and $Q_k(t)$.

2.2 Objective Function

The *nonlinear* and *time-varying* objective function $R(t)$ to be minimized can be defined as follows:

$$R(t) \equiv P_0(t) + \lambda H_p^*(t) + \mu H_v^*(t) + \nu H_e^*(t) + \xi H_s^*(t), \tag{4}$$

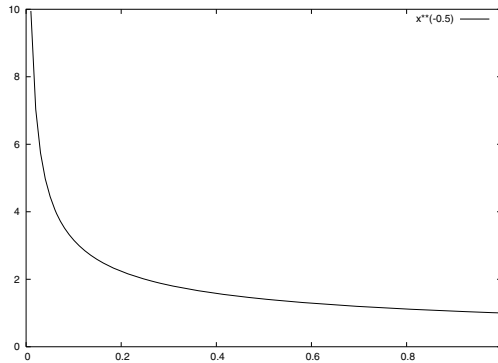


Fig. 3. Constraint function $H_p(t)$ for power plant

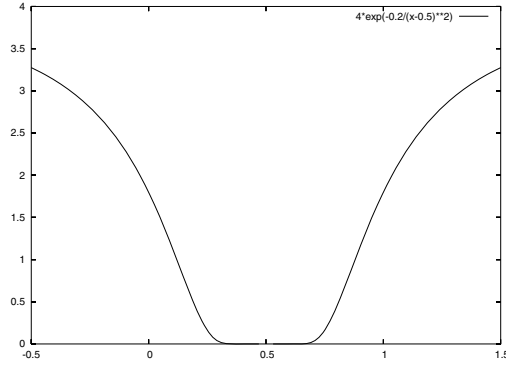


Fig. 4. Constraint functions $H_B(t)$ and $H_Q(t)$ common to transformer and distribution substations and demand nodes

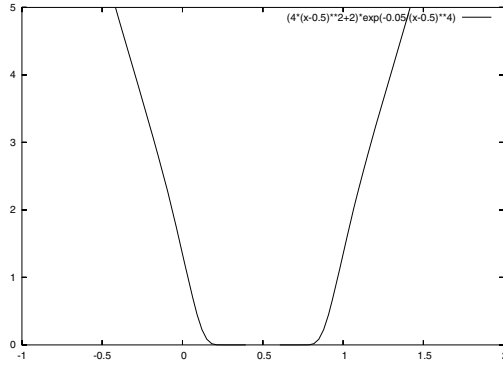


Fig. 5. Constraint function $H_e(t)$ for power supply

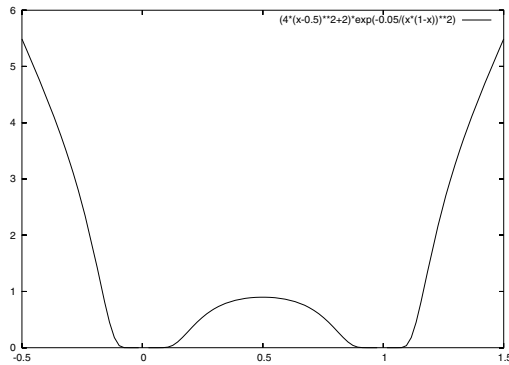


Fig. 6. Constraint function $H_s(t)$ for switching

where λ, μ, ν , and ξ are constants, and $P_0(t)$ is the total of each fuel cost $f(P_i(t))$ in each power plant, approximated as a quadratic function of $P_i(t)$:

$$P_0(t) \equiv \sum_i f(P_i(t)) = \sum_i (a_i + b_i P_i(t) + c_i P_i^2(t)),$$

where, a_i, b_i, c_i are positive constants. $H_p^*(t)$ is defined as a function concerning the power plants P , where $H_p(P_i(t))$ is a function that should satisfy the positive power generation constraint at the power plant, as shown in Fig 3.

$$H_p^*(t) \equiv \sum_i H_p(P_i(t)) = \sum_i P_i(t)^{-\alpha_p}, \alpha_p = \text{const.}$$

$H_v^*(t)$ is the combinational function of the constraint functions $H_B(t)$, $H_{B'}(t)$ and $H_Q(t)$. Functions $H_B(t)$, $H_{B'}(t)$ and $H_Q(t)$ should satisfy the capacity and demand, respectively, in the battery or storage B and demand Q .

$$\begin{aligned} H_v^*(t) \equiv & \sum_j H_B(B_j(t); B_j^{max}, B_j^{min}) + \sum_j H_{B'}(B_{j'}(t); B_{j'}^{max}, B_{j'}^{min}) \\ & + \sum_k H_Q(Q_k(t); Q_k^{max}, Q_k^{min}). \end{aligned}$$

These functions are *well-shaped* functions, as shown in Fig 4, and they determine the upper and lower limits of power capacity as shown below:

$$\begin{aligned} H_B(t) & \equiv \alpha_B \exp\{-\beta_B / (B_j(t) - \gamma_B)^2\}, \\ \text{where, } \alpha_B & = \text{const.}, \beta_B = 3(B_j^{max} - B_j^{min})^2 / 8, \gamma_B = (B_j^{min} + B_j^{max}) / 2, \\ H_Q(t) & \equiv \alpha_Q \exp\{-\beta_B / (Q_k(t) - \gamma_B)^2\}, \end{aligned}$$

where, $\alpha_Q = \text{constant}$. β_B and γ_B are determined if two flection points in the above-mentioned functions correspond to the minimum B_j^{min} and maximum B_j^{max} values of $B_j(t)$.

Next, we define an edge-constraint function $H_e^*(t)$ as follows:

$$H_e^*(t) \equiv \sum_i H_e(W_i(t)) + \sum_j H_e(W_j(t)) + \sum_{j'} H_e(W_{j'}(t)),$$

where,

$$0 \leq W_i(t) = \sum_j w_{ij}(t) \leq 1, 0 \leq W_j(t) = \sum_k w_{jk}(t) \leq 1, 0 \leq W_{j'}(t) = \sum_k w_{j'k}(t) \leq 1,$$

$$0 \leq w_{ij}(t) = c_{ij} x_{ij}(t) \leq 1, 0 \leq w_{jk}(t) = c_{jk} x_{jk}(t) \leq 1, 0 \leq w_{j'k}(t) = c_{j'k} x_{j'k}(t) \leq 1,$$

$$0 \leq c_{ij}, c_{jk}, c_{j'k} \leq 1, 0 \leq x_{ij}(t), x_{jk}(t), x_{j'k}(t) \leq 1,$$

where, the edge constraint function $H_e(W_i)$ is defined as a well-shaped function as shown in Fig.5; whose mathematical forms are as follows:

$$\begin{aligned} H_e(W_i(t)) &\equiv \alpha_e(W_i(t)) \exp\{-\beta_e/(W_i(t) - \gamma_e)^4\}, \\ \alpha_e(x(t)) &= 4(x(t) - 0.5)^2 + 2, \beta_e = 0.05, \gamma_e = 0.5. \end{aligned}$$

$H_e(W_i(t))$ is the function used to satisfy the condition that the total power limits the total flow produced by the power plant $P_i(t)$, as shown in Fig.5. Similarly, $H_e(W_j(t))$ is used to satisfy the condition that the total power limits the total flow produced by the battery or storage $B_j(t)$, and so on.

Next, we define the switching function $H_s^*(t)$ as shown in Fig.6, which represents the tendency toward two different switching modes, i.e., ON(0) and OFF(1), as follows:

$$\begin{aligned} H_s^*(t) &\equiv \sum_i \sum_j \alpha_s(x_{ij}(t)) \exp[-\beta_s/\{x_{ij}^2(t)(1 - x_{ij}(t))^2\}] \\ &+ \sum_j \sum_k \alpha_s(x_{jk}(t)) \exp[-\beta_s/\{x_{jk}^2(t)(1 - x_{jk}(t))^2\}] \\ &+ \sum_i \sum_k \alpha_s(x_{ik}(t)) \exp[-\beta_s/\{x_{ik}^2(t)(1 - x_{ik}(t))^2\}] \\ &+ \sum_i \sum_{j'} \alpha_s(x_{ij'}(t)) \exp[-\beta_s/\{x_{ij'}^2(t)(1 - x_{ij'}(t))^2\}] \\ &+ \sum_{j'} \sum_j \alpha_s(x_{j'j}(t)) \exp[-\beta_s/\{x_{j'j}^2(t)(1 - x_{j'j}(t))^2\}] \\ &+ \sum_{j'} \sum_k \alpha_s(x_{j'k}(t)) \exp[-\beta_s/\{x_{j'k}^2(t)(1 - x_{j'k}(t))^2\}], \end{aligned}$$

where, $\alpha_s(x) = 4(x - 0.5)^2 + 2$, $\beta_s = \text{const.}$,

$$0 \leq w_{ij}(t) = c_{ij}x_{ij}(t) \leq 1, 0 \leq w_{jk}(t) = c_{jk}x_{jk}(t) \leq 1,$$

$$0 < c_{ij}, c_{jk} \leq 1, 0 \leq x_{ij}(t), x_{jk}(t) \leq 1, \text{ etc.},$$

where c_{ij} is the coefficient of power-flow from node i to node j , and $x_{ij}(t)$ represents the switching value between the power nodes i and j .

The change in power $\Delta P_i(t)$ is calculated by

$$\Delta P_i(t) = -\eta \frac{\partial R(t)}{\partial P_i(t)} = -\eta \left\{ \frac{\partial P_0(t)}{\partial P_i(t)} + \lambda \frac{\partial H_p^*(t)}{\partial P_i(t)} + \mu \frac{\partial H_v^*(t)}{\partial P_i(t)} \right\}, \quad (5)$$

where,

$$\begin{aligned}
\frac{\partial P_0(t)}{\partial P_i(t)} &= b_i + 2c_i P_i(t), \quad \frac{\partial H_p^*(t)}{\partial P_i(t)} = -\alpha_p P_i(t)^{-\alpha_p - 1}, \quad \alpha_p = \text{const.}, \\
\frac{\partial H_v^*(t)}{\partial P_i(t)} &= \frac{\partial H_B^*(t)}{\partial P_i(t)} + \frac{\partial H_{B'}^*(t)}{\partial P_i(t)} + \frac{\partial H_Q^*(t)}{\partial P_i(t)}, \\
\frac{\partial H_B^*(t)}{\partial P_i(t)} &= \sum_j \frac{\partial H_B(t)}{\partial B_j(t)} \left\{ \frac{\partial B_j(t)}{\partial P_i(t)} + \sum_{j'} \frac{\partial B_j(t)}{\partial B_{j'}(t)} \frac{\partial B_{j'}(t)}{\partial P_i(t)} \right\} \\
&= \delta_j(t) (w_{ij}(t) + \sum_{j'} w_{j'j}(t) w_{ij'}(t)), \quad \delta_j(t) \equiv \frac{\partial H_B(t)}{\partial B_j(t)}, \\
\frac{\partial H_Q^*(t)}{\partial P_i(t)} &= \sum_k \frac{\partial H_Q(t)}{\partial Q_k(t)} \left\{ \frac{\partial Q_k(t)}{\partial P_i(t)} + \sum_j \frac{\partial Q_k(t)}{\partial B_j(t)} \left(\frac{\partial B_j(t)}{\partial P_i(t)} + \sum_{j'} \frac{\partial B_j(t)}{\partial B_{j'}(t)} \frac{\partial B_{j'}(t)}{\partial P_i(t)} \right) \right. \\
&\quad \left. + \sum_{j'} \frac{\partial Q_k(t)}{\partial B_{j'}(t)} \frac{\partial B_{j'}(t)}{\partial P_i(t)} \right\} \\
&= \sum_k \delta_k(t) \{ w_{ik}(t) + \sum_j w_{jk}(t) (w_{ij}(t) + \sum_{j'} w_{j'j}(t) w_{ij'}(t)) + \sum_{j'} w_{j'k}(t) w_{ij'}(t) \}, \\
\text{where, } \delta_k(t) &\equiv \frac{\partial H_Q(t)}{\partial Q_k(t)}.
\end{aligned}$$

Next, the change in the switching values $\Delta x_{jk}(t)$ can be calculated as follows:

$$\begin{aligned}
\Delta x_{jk}(t) &= -\eta \frac{\partial R(t)}{\partial x_{jk}(t)} = -\eta \left\{ \mu \frac{\partial H_Q(t)}{\partial Q_k(t)} B_j(t) c_{jk} + \nu \frac{\partial H_e(W_j(t))}{\partial W_j(t)} c_{jk} + \xi \frac{\partial H_s^*(t)}{\partial x_{jk}(t)} \right\} \\
&= -\eta \left\{ \mu \delta_k B_j(t) c_{jk} + \nu \frac{\partial H_e(W_j(t))}{\partial W_j(t)} c_{jk} + \xi \frac{\partial H_s^*(t)}{\partial x_{jk}(t)} \right\}, \\
\Delta x_{ik}(t) &= -\eta \frac{\partial R(t)}{\partial x_{ik}(t)} = -\eta \left\{ \mu \delta_k P_i(t) c_{ik} + \nu \frac{\partial H_e(W_i(t))}{\partial W_i(t)} c_{ik} + \xi \frac{\partial H_s^*(t)}{\partial x_{ik}(t)} \right\}, \\
\Delta x_{ij}(t) &= -\eta \frac{\partial R(t)}{\partial x_{ij}(t)} = -\eta \left\{ \mu \delta_j^* P_i(t) c_{ij} + \nu \frac{\partial H_e(W_i(t))}{\partial W_i(t)} c_{ij} + \xi \frac{\partial H_s^*(t)}{\partial x_{ij}(t)} \right\},
\end{aligned} \tag{6}$$

where,

$$\delta_j^*(t) \equiv \delta_j(t) + \sum_k \delta_k(t) w_{jk}(t), \quad \delta_j(t) \equiv \frac{\partial H_B(t)}{\partial B_j(t)}, \quad \delta_k(t) \equiv \frac{\partial H_Q(t)}{\partial Q_k(t)}.$$

2.3 Dynamic Operation of Power-Flow Control

These power nodes and the switching values can be operated in a *synchronous or asynchronous manner*, where the synchronous operation means all power nodes ($P_i(t)$, $B_j(t)$, $B_{j'}(t)$ and $Q_k(t)$) and the switching values $x_{ij}(t)$ etc. are operated

at the same time, and on the other hand, the asynchronous operation means these power nodes and the switching values are operated at any time.

The procedure for the *synchronous* updating of the values $P_i(t)$, $x_{jk}(t)$ and $x_{ij}(t)$ as follows:

- (1) Randomly set the initial values of $\forall^{i,j} x_{ij}(t)$ between 0 and 1.
- (2) Calculate the value of the objective function $R(t)$ using Eq.(4) during the forward propagation of power.
- (3) Using Eq.(6), calculate the values of $\forall^{i,j} \Delta x_{ij}(t)$. Then, update the values of $x_{ij}(t)$ using the following equations:

$$x_{ij}(t+1) = x_{ij}(t) + \Delta x_{ij}(t).$$

- (4) Using Eq.(5), update $\forall^i P_i(t)$ according to the following equation:

$$P_i(t+1) = P_i(t) + \Delta P_i(t).$$

- (5) Check the termination criterion.

When the changes in power $\forall^i \Delta P_i(t)$ and the switching values $\forall^{i,j} \Delta x_{ij}(t)$ become stable, stop the calculation. Else, go to (2).

In the smart grid power network described in this section, it is critical that the changes in power $\Delta P_i(t)$ and switching values $\Delta x_{ij}(t)$ be updated in an *asynchronous* manner. This is because the smart grid is a *highly decentralized, parallel, distributed, autonomous* communication and processing system in which all functions of the power nodes ($P_i(t)$, $B_j(t)$, $Q_k(t)$) and interconnections ($x_{ij}(t)$) should be operated in an *asynchronous* manner.

2.4 Smart Grid Integrated into Conventional Power Network

We constructed a smart grid model which integrates some microgrids shown in Fig.2 into a conventional power network. This architecture is shown in Fig.7, where the conventional power network serves power energy from the power plant (shown as No.1) through several transformer substations (shown as No.2-6) to some distribution power stations (shown as No.7,8,9). In this model, each microgrid has eight power nodes which include one distribution station, four batteries or storages, and three demand nodes. The distribution station distributes power from the conventional power network to the batteries/storages and demand nodes through the interconnections between the nodes. This architecture is hierarchical and somewhat robust because even if some interconnections are broken, the other interconnections could serve power energy to the power nodes with a bypass connection. We will show some computer-simulation experimental results supposing this kinds of situations.

3 Simulation Experiments

3.1 Experiments for MicroGrid

We performed simulation experiments both in a *synchronous* and an *asynchronous* manner under the following initial conditions: setting parameters λ

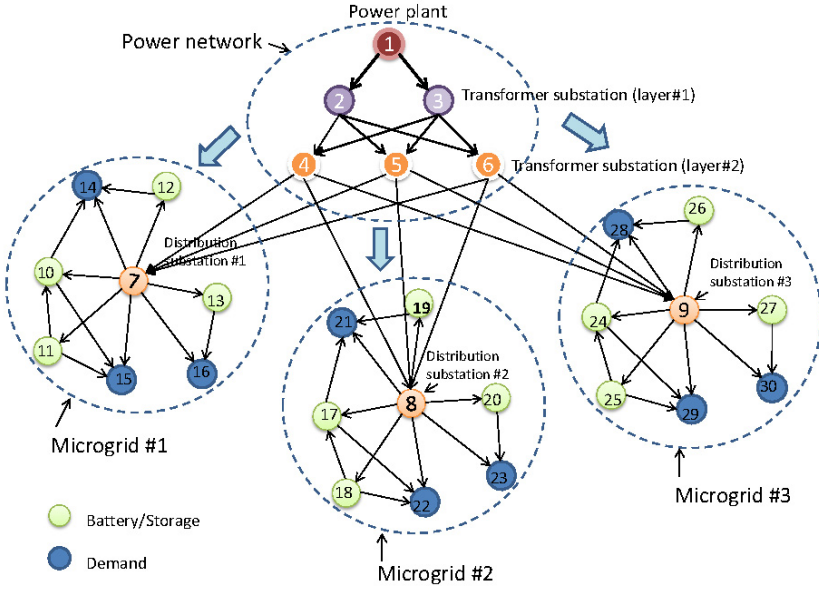


Fig. 7. Model of smart grid integrated into power network

$\mu = \nu = \xi = 1$ for the weight of penalty, and $a_i = b_i = c_i = 1$ for simplicity in the objective function $R(t)$. $B_{min} = 0.2$, $B_{max} = 0.8$, $Q_{min} = 0.2$, and $Q_{max} = 0.8$. The power values for the transformer or distribution substation (or battery) $B_j(t)$ and the demand $Q_k(t)$ are *normalized*, and their values, in principle, are between 0 and 1. Other miscellaneous parameters are set as follows: $\alpha_p = 0.5$, $\alpha_B = \alpha_Q = 4.0$, $\beta_e = 0.05$, $\gamma_e = 0.5$, $\eta_0 = 0.01$, and $c_{ij} = c_{jk} = c_{ik} = c_{j'j} = c_{j'k} = c_{ij'} = 1.0$.

The architecture of the microgrid power network is shown in Fig. 2 as an example, one power plant $P_i(t)$, four batteries $B_j(t)$, and three sites of demand $Q_k(t)$ are shown in this figure. We only show the results in an *asynchronous manner* below due to the limited space.

When an accident occurred at the 2500th time-step at the interconnection $w_{ik}(t)$ in the microgrid, and power could not be supplied from the power plant node $P_i(t)$ directly, as shown in Fig. 10, the power supply for the demand node $Q_k(t)$ decreased instantaneously at the 2500th time-step, as shown in Fig. 9, and the power supply $P_i(t)$ fluctuated slightly, as shown in Fig. 8. However, except $w_{ik}(t)$, the weights increased to recover the power supply through bypass connections such as $w_{ij}(t)$, $w_{jk}(t)$, $w_{ij'}(t)$, and $w_{j'k}(t)$, as shown in Fig. 10. Even in this case, the objective function $R(t)$ showed a small fluctuation, and then, converged to a stable state, as shown in Fig. 11.

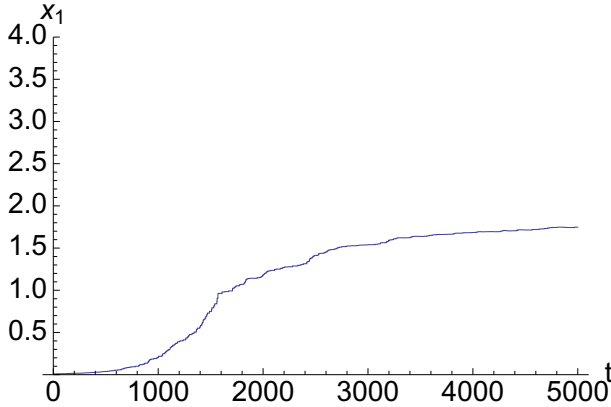


Fig. 8. Change in power $P_i(t)$ in the microgrid during a failure in an *asynchronous* manner

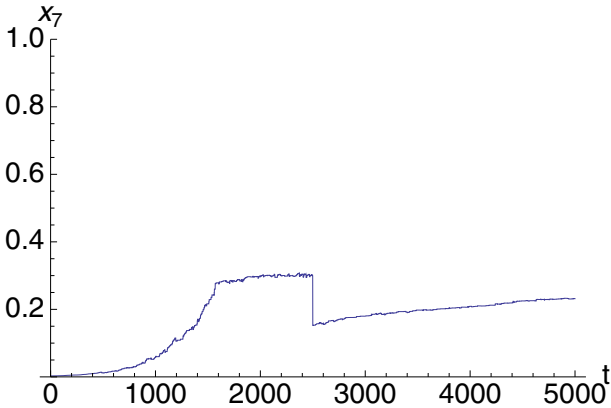


Fig. 9. Change in power at demand node $Q_k(t)$ in the microgrid during a failure in an *asynchronous* manner

3.2 Experiments for Smart Grid Integrated into Conventional Power Network

We performed simulation experiments for the architecture shown in Fig. 7 both in a *synchronous* and an *asynchronous* manner under the following initial conditions: setting parameters $\lambda = \mu = \nu = 1.4$, $\xi = 1.0$ for the weights of penalty, and $a_i = b_i = c_i = 1$ for simplicity in the objective function $R(t)$. $B_{min} = 0.2$, $B_{max} = 0.8$, $Q_{min} = 0.2$, $Q_{max} = 0.8$. The power values for the transformer or distribution substation (or battery) $B_j(t)$ and the demand $Q_k(t)$ are *normalized*, and their values, in principle, are between 0 and 1. Other miscellaneous parameters are set as follows: $\alpha_p = 0.22$, $\alpha_B = \alpha_Q = 4.0$, $\beta_e = 0.05$, $\gamma_e = 0.5$, $\beta_s = 0.15$, $\eta_0 = 0.01$. We will only show the computer-simulation experimental results in an

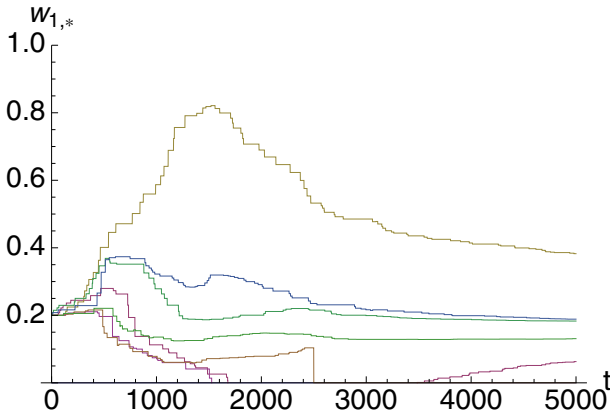


Fig. 10. Change of transmission weights $w_{ij}(t)$ in the microgrid with a failure of w_{ik} in an *asynchronous* manner

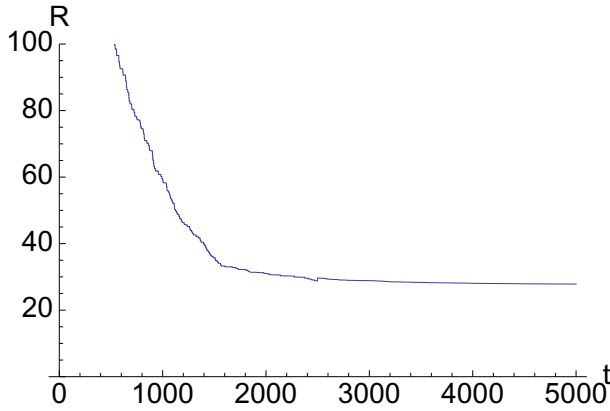


Fig. 11. Change in objective function $R(t)$ in the microgrid during a failure in an *asynchronous* manner

asynchronous manner because this manner reflects a real-world situation than the synchronous manner does.

Change in power $P_i(t)$ at power plant node, the battery node $B_j(t)$ (e.g., No.17 node), the transmission weights $w_{ij}(t)$, and the objective function $R(t)$ are shown in Figs. 12, 13, 14 and 15, respectively.

The power $P_i(t)$ increased rapidly and reached a stable value of 3.8 at the 900th time-step. The power of the battery node $B_j(t)$ (for example, No.17 node) increased gradually to satisfy the minimum demand value of $B_j(t)$ ($=B_j^{min}$) mostly. All the transmission weights $w_{ij}(t)$, $w_{jk}(t)$, and $w_{ik}(t)$ between the power

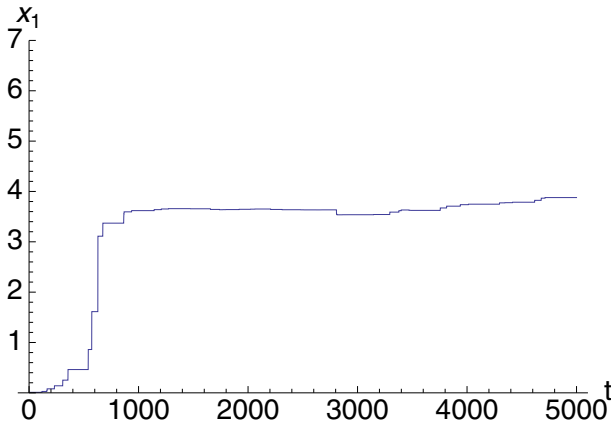


Fig. 12. Change in power $P_i(t)$ during a failure in an *asynchronous* manner

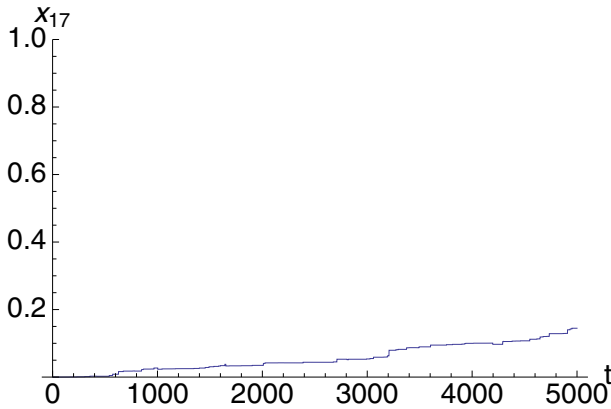


Fig. 13. Change in power at battery node $B_j(t)$ (e.g., No.17) during a failure in an *asynchronous* manner

nodes reached stable values in the range 0.0-0.25. The objective function value $R(t)$ decreased smoothly to a stable value of 150. Similarly, as in the previous subsection, when an accident occurred at the 3000th time-step at the interconnection $w_{ij}(t)$ (the interconnection between No.8 and No.17), which is shown in Fig.14, and power could not be supplied from the power node directly. However, power could be supplied through bypass connections such as No.8 to No.18 and No.18 to No.17 nodes. Even in this case, the power did not fluctuate so much as shown in Fig.13, and the objective function $R(t)$ decreased as shown in Fig.15. This situation shows a *robust* feature of the smart grid shown in Fig.7.

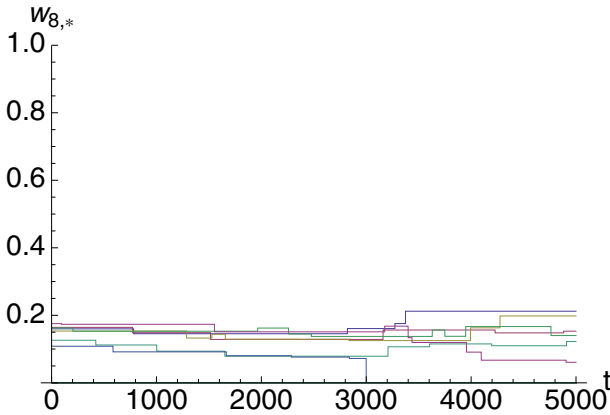


Fig. 14. Change in transmission weights $w_{ij}(t)$ during a failure of w_{ik} in an *asynchronous* manner

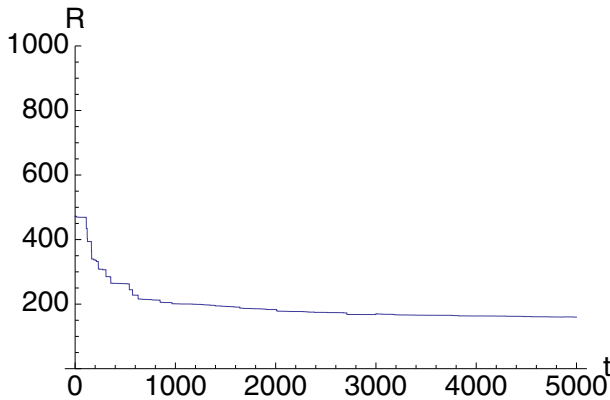


Fig. 15. Change in objective function $R(t)$ during a failure in an *asynchronous* manner

4 Discussions

We considered a smart grid and formulated the power-flow control equations when several constraints were imposed on the power plants, transformers or distribution substations, batteries/storages, and the demands. In general, the objective function is *nonlinear and time-varying*; The first term of the objective function $R(t)$ is a total fuel cost $P_0(t)$ which is a *nonlinear* term with the increase of power. The second term is $H_p^*(t)$ which is a penalty term for not decreasing the power $P_i(t)$ to zero. The third term is a constraint function which satisfies the capacity and demand in the transformer or distribution substation, and the demand, respectively. These constraint functions can be basically defined

as “well-shaped” functions shown in Figs. 4 and 5. The switching function can be defined as the function shown in Fig. 6 where the switching value tends to 0(OFF) or 1(ON) as the minimum value of the function.

The minimum demand always changes with time because the demand becomes high during daytime and less at midnight. Under these conditions, the demand $D_k(t)$ is approximately represented by

$$D_k(t) = a_k \sin(\omega_k t + \theta_k) + b_k, a_k = (\epsilon Q_k^{max} - Q_k^{min})/2, \\ b_k = (\epsilon Q_k^{max} + Q_k^{min})/2, \epsilon = 0.8,$$

where a_k is the amplitude, θ_k is the phase of the k^{th} demand node, b_k is an offset with an appropriate allowance of 20% for the maximum value Q_k^{max} . When the demand $D_k(t)$ changes over time, the power-flow will change so that the objective function $R(t)$ is minimized. The smart grid can adapt to this changing environment by changing the power $P_i(t)$ as well as the switching values $x_{ij}(t)$ between all i and j nodes operating in a *highly dicentralized, parallel distributed, asynchronous and autonomous manner*. This is one of the great advantageous features for the smart grid because even if the number of power nodes increases or changes for adapting the increasing demand or the changing environment, it is unnecessary to increase the computational power for the dynamic control due to this remarkable feature.

If the battery node $B_j(t)$ belongs to the power-plant site $P_i(t)$ and the battery node $B_{j'}(t)$ belongs to the demand site $Q_k(t)$, as shown in Fig. 2, the connection $w_{j',j}(t)$ can be regarded as the power-supply cable connecting $B_{j'}(t)$ to $B_j(t)$ when $B_j(t)$ has to be charged.

If the battery or demand node lacks sufficient capacity to receive power from the power plants and other batteries, the maximum capacity B_j^{max} of the battery node and Q_k^{max} of the demand node can be changed using the following equation:

$$\Delta B_j^{max} = -\eta \frac{\partial R(t)}{\partial B_j^{max}} = -\eta \left(\mu \frac{\partial H_B(t)}{\partial B_j^{max}} \right),$$

where,

$$\frac{\partial H_B(t)}{\partial B_j^{max}} = -\alpha_{Bexp} \{ -\beta_B / (B_j(t) - \gamma_B)^2 \} \\ \times 3(B_j^{max} - B_j^{min}) / \{ 4(B_j(t) - \gamma_B)^2 \} \\ \times [1 + (B_j^{max} - B_j^{min}) / \{ 2(B_j(t) - \gamma_B) \}],$$

Therefore, the maximum capacity B_j^{max} can be changed as follows:

$$B_j^{max} = B_j^{max} + \Delta B_j^{max},$$

Similarly, the value of Q_k^{max} in the demand node can be changed as,

$$Q_k^{max} = Q_k^{max} + \Delta Q_k^{max},$$

where,

$$\begin{aligned}\Delta Q_k^{max} &= -\eta \frac{\partial R(t)}{\partial Q_k^{max}} = -\eta \left(\mu \frac{\partial H_Q(t)}{\partial Q_k^{max}} \right) \\ \frac{\partial H_Q(t)}{\partial Q_k^{max}} &= -\alpha_Q \exp\{-\beta/(Q_k(t) - \gamma_Q)^2\} \\ &\quad \times 3(Q_k^{max} - D_k) / \{4(Q_k(t) - \gamma_Q)^2\} \\ &\quad \times [1 + (Q_k^{max} - D_k) / \{2(Q_k(t) - \gamma_Q)\}].\end{aligned}$$

One possible way of responding to an increase in demand is to increase the capacity of power plants; however, it is very expensive to construct another power plant to meet the increased demand. On the other hand, increasing the maximum capacity of the battery and demand nodes and increasing the number of battery nodes in the smart grid are cost-effective ways to satisfy the increased demand. Furthermore, in the smart grid, the *topology of the network* can be changed by changing the switching values $x_{ij}(t)$, $x_{ik}(t)$ and $x_{jk}(t)$ to 0(OFF) or 1(ON) corresponding to the cables $w_{ij}(t)$, $w_{ik}(t)$ and $w_{jk}(t)$, respectively. This flexible function enables rapid adaptation to sudden changes in environment, such as outages due to power failures and disruptions during a disaster; this adaptation enables recovery by switching and connecting a *dead* node to an *alive* node. This is a *robust* feature of the smart grid that can adapt to a changing environment and promote the use of renewable energy rather than fossil energy. For this purpose, the formulation presented in this paper can be used to monitor the changing power balance in the smart grid when new power nodes are added by necessity.

The smart grid, which is a *highly decentralized, distributed, parallel communication and processing system*, will be realized in the near future. In this grid, a multi-agent system in which the information on switching values $x_{ij}(t)$, $x_{ik}(t)$ and $x_{jk}(t)$ will be embedded in communication packets will be important; these communication packets will serve as mobile agents and can be exchanged between the nodes in an *asynchronous* manner during communication. This situation will be realized by extending the smart grid from a local area to a societal, a national, an international, and a global network step-by-step as in the case of the Internet.

Furthermore, we can also extend this dynamic flow-control method in the smart grid for other fields such as logistics, dynamic resource (other than power) assignment and management, and network-flow control; for example, to extend it to logistics, the power nodes $P_i(t)$, $B_j(t)$ and $Q_k(t)$ can be regarded as suppliers, depots or substations, and consumers, respectively; and the interconnections $w_{ij}(t)$ can be regarded as transportation efficiency (i.e., inverse of cost).

5 Conclusions

Smart grid is a promising, highly distributed power network for promoting an effective use of natural energy by integrating it into the commercial power

networks. Inspiring from the dynamic features of biological creatures (e.g., a gene-regulatory network in the micro-level, function of brain in the meso-level, and behavior of agents in an ecosystem, etc.) we modeled an architecture of the smart grid by integrating a conventional power network and some microgrids into it, and then, defined the basic equations and operated the dynamic power-flow control through the defined objective function (Note that this is a *nonlinear and time-varying* function in general) imposing constraint (represented by the *well-shaped* constraint functions) on each power capacity.

To validate the model and behavior of the smart grid, we performed several simulation experiments including a power failure such as outage, both in a *synchronous and asynchronous manner* of operation. As a consequence, we found that the dynamic power-flow control was successfully achieved even for the case of the power failure both in the two manners.

Furthermore, the dynamic control method is applicable to other control problems such as dynamic resource assignment and management except power, logistics, network-flow control, dynamic control of traffic-flow such as railways and communication packets using the similar objective and constraint functions defined in this paper because there is inevitable constraint imposing on limited resources.

References

1. Alberts, B., et al.: *Molecular Biology of the Cell*, 4th edn. Garland Science (2002)
2. Bear, M.F., Connors, B.W., Paradiso, M.A.: *Neuroscience: Exploring the Brain*, 3rd edn. Lippincott Williams & Wilkins Inc. (2007)
3. Clark, W., Gellings: *The Smart Grid: Enabling Energy Efficiency and Demand Response*. The Fairmont Press, Inc. (2009)
4. Strzelecki, R., Benysek, G. (eds.): *Power Electronics in Smart Electrical Energy Network*. Springer (2008)
5. Aida, et al.: *The Micro Grid: For the Symbiosis of Distributed Power Supply and Power Networks*, p. 29. Denki Shimbun Books (2004) (in Japanese)
6. Massoud Amin, S.: *For the Good of the Grid*, pp. 48–59. *IEEE Power and Energy Magazine* (November/December 2008)
7. Massoud Amin, S., Stringer, J.: *The Electric Power Grid: Today and Tomorrow*. *MRS Bulletin* 33, 399–407 (2008)
8. Massoud Amin, S., Schewe, P.F.: *Preventing Blackouts*. *Scientific American*, 60–67 (May 2007)
9. Massoud Amin, S.: *Powering the 21st Century*. *IEEE Power and Energy Magazine*, 96, 93, 94 (March/April 2005)
10. Jiang, Z.: *Computational Intelligence Techniques for a Smart Electric Grid of the Future*. In: Yu, W., He, H., Zhang, N. (eds.) *LNCS*, vol. 5551, pp. 1191–1201. Springer, Heidelberg (2009)
11. Molderink, A., Bakker, V., Bosman, M.G.C., Hurink, J.L., Smit, G.J.M.: *Domestic Energy Management Methodology for Optimizing Efficiency in Smart Grids*. In: *Proceedings of IEEE Conf. on Power Technology*, June 28–July 2 (2009)
12. Ma, Y., Zhou, L., Tse, N., Osman, A., Lai, L.L.: *An initial study of computational intelligence for smart grid*. In: *Proc. of the Eighth Int. Conf. on Machine Learning and Cybernetics*, Baoding, pp. 3425–3429. *IEEE* (2009)
13. Kupzog, F., Zia, T., Zaidi, A.A.: *Automatic electric load identification in self-configuring microgrids*. In: *Proc. of IEEE AFRICON*, pp. 1–5 (September 2009)

Path Heuristics Using ACO for Inter-domain Routing in Mobile Ad Hoc and Sensor Networks

Falko Dressler¹, Roman Koch¹, and Mario Gerla²

¹ Dept. of Computer Science, University of Erlangen, Germany
dressler@cs.fau.de

² Dept. of Computer Science, University of California, Los Angeles
gerla@cs.ucla.edu

Abstract. We investigate the use of biologically inspired routing heuristics in the field of inter-domain routing in sensor networks. Instead of relying on classical topology control techniques for routing in sensor networks, the use of geographical coordinates has been investigated for self-organized and fully distributed message forwarding. However, the identification of the nodes' positions is either expensive in terms of necessary equipment or message exchange. Therefore, the use of virtual coordinates has been investigated in this domain. The key advantage is that these virtual identifiers can also be used for data management similar as in a Distributed Hash Table (DHT). It is, however, extremely challenging to provide routing functionality between multiple independent networks or network domains. In previous work, we developed the Virtual Cord Protocol (VCP) that provides all the means for creating and maintaining such virtual identifiers and that is even able to route between neighboring network domains. This paper extends VCP by providing a generalized inter-domain routing framework using Ant Colony Optimization (ACO) for optimizing routes between multiple network domains. In extensive simulations, we evaluated this routing bio-inspired heuristic. The obtained results clearly demonstrate that ACO is very efficient even in highly mobile scenarios.

Keywords: Inter-domain routing, virtual cord protocol, ant colony optimization, sensor networks.

1 Introduction

Several classes of different routing techniques have been investigated in the field of sensor networks. The key objective is to cope with heterogeneity of nodes, dynamics of the environment, and, most importantly, the limited available energy resources [1]. Early approaches mainly focused on establishing routing tables similar to routing protocols studied in the field of Mobile Ad Hoc Networks (MANETs). However, it turned out that the inherent protocol overhead for topology control is not adequate to operate sensor networks for a longer lifetime [8]. Therefore, stateless approaches have been investigated such as georouting, where the content is represented by geographic coordinates of the destination. In this case, all nodes have geographic position identifiers (learned for

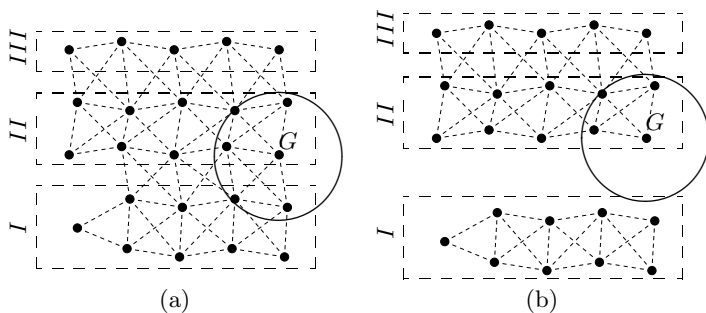


Fig. 1. Schematic representation of partially connected sensor network domains

example from GPS). Such position-based routing solutions inherently improve the situation as simple greedy routing towards the destination can be used. However, such approaches only work well if the network is dense, as routing holes cause geographic routing to rely on inefficient face routing methods. Additionally, GPS is required for all the nodes have to be able to precisely obtain their geographic locations, and a Geo Location Service (GLS) is necessary to find the destination coordinates [16]. Recently, a number of improvements to overcome geo-routing holes have been proposed. One idea is to “re-arrange” the nodes’ positions appropriately to prevent routing holes [15].

A conceptually more innovative concept is to rely on virtual coordinates only and to create an overlay network that connects the nodes and guides the search. Protocols like Virtual Cord Protocol (VCP) [3] and Virtual Ring Routing (VRR) [4] build their own coordinate system, which is completely independent of the geographic node positions. Furthermore, the virtual node positions can be used as IDs in a Distributed Hash Table (DHT) to efficiently store and retrieve data. Current work on virtual coordinate based approaches focuses on two aspects: The provided quality of service, which is mainly an issue of optimizing the delivery ratio or even providing guarantees [18,19], and the reliability of the system as a whole, using data replication and other redundancy increasing techniques [2]. Such solutions are inherently self-organizing and scale extremely well even for large-scale networks [10].

Many scenarios can be envisioned in which multiple (virtual coordinate based) networks have to be established and maintained separately, yet with a strong demand to support routing among these different networks in case of connectivity. The problem is illustrated in Figure 1. As shown in Figure 1a, multiple network domains may be operated by protocols such as VCP, even though communication between the domains also becomes necessary. As depicted in Figure 1b, the connectivity between such domains might not be constantly available, e.g. domains move according to a group mobility model [22]. However, we assume that the network integrity (in terms of an ordered overlay) for a single domain is almost always ensured. Whenever two networks get into each other’s physical radio communication range, data can be exchanged between the domains. A key

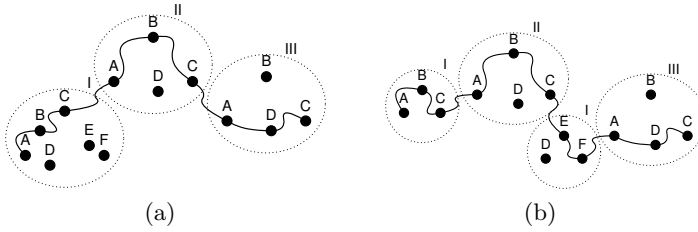


Fig. 2. Topology changes at domain level: domain I splits and gets re-located

challenge is to provide inter-domain routing between different local networks because the virtual coordinates are usually managed locally in each domain.

Inter-domain routing in MANETs has been first discussed in [5]. Four challenging issues have been identified: addressing, membership management, handling domain-level topology changes, and routing between the networks. As Internet-based protocols have been considered, the addressing and membership management basically targeted the IP address assignment procedure and the resulting routing problems. A cluster-based solution for inter-domain routing in MANETs has been described in [21]. Here, especially the issue of domain-level topology changes has been addressed. Using bloom filters, the effort for necessary topology updates was greatly reduced. Figure 2 outlines some of the most typical problems. At a macroscopic level, domain management techniques must be developed taking care of splitting and merging domains, and of domain-wide topology changes. On a microscopic level, different nodes will have to provide gateway functionality as soon as physical connection is available. The inter-domain routing is responsible for establishing adequate paths.

Motivated by this work, we investigated the issue of inter-domain routing for virtual coordinate based routing protocols, in particular focusing on our VCP approach [12,13]. We were able to show that inter-domain routing in virtual coordinate environments can be established exploiting available DHT-based data management operations. Inter-domain routing between neighboring domains becomes feasible with only marginal overhead.

In this paper, we extend this previous work introducing a generalized routing framework for inter-domain routing in virtual coordinate based networks using bio-inspired techniques. We show how to establish topology information on a *macroscopic* domain level as well as on a *microscopic* gateway level. Furthermore, we used a routing heuristic based on Ant Colony Optimization (ACO) [9] to optimize both the macroscopic and the microscopic behavior even in very dynamic environments. The development of such self-organizing algorithms strongly depends on an optimal calibration of the system parameter [11]. Thus, we first investigated the configuration of the ACO algorithm using empirical studies. Using these results, we performed a detailed performance analysis of the developed ACO heuristics based inter-domain routing scheme. The results clearly indicate that the developed algorithm is extremely stable and robust to topology changes.

2 Virtual Cord Protocol

The Virtual Cord Protocol (VCP) has been developed for efficient routing and data management in sensor networks. In previous work, we demonstrated that VCP outperforms MANET-based solutions as well as other virtual coordinate-based protocols such as VRR [2,3]. We continued this research by studying inter-domain routing between multiple VCP domains [2,13]. In the following, after briefly outlining the concepts of VCP, we present our generalized inter-domain routing framework.

2.1 VCP Cord Management

The main idea is to arrange all the nodes in the network in form of a virtual cord. The topology of this cord must not be “optimal” in any sense, because routing is organized by exploiting information about the physical neighbors for greedy forwarding. Nevertheless, the cord ensures the availability of at least one path between any two nodes in the network for guaranteed delivery. The cord is established using periodic HELLO messages. Besides the assigned virtual address, these messages carry all relevant information including the physical and the virtual neighbors. Based on received HELLO messages (at least one is required) in the last HELLO interval, a new node can determine its position in the cord. A cord is formed according to a number of simple rules. Basically, new nodes either join at one end of the cord, or get integrated if at least two other nodes that are virtual neighbors in the cord are detected. A special rule is applied if the node has connectivity to a non-end node but not to its virtual neighbors. Then, a *virtual position* is generated at the discovered potential neighbor that is close to its virtual coordinate. This address allows the new to join between the real and the virtual position in the cord, i.e. to extend the cord without disrupting it. An application-dependent hash function is used for associating data items to nodes; thus, both pushing to a node and pulling data from a node are supported. The same mechanism can also be used for service discovery.

2.2 Basic Inter-domain Routing

The basic inter-domain routing solution for VCP relies on unique domain identifiers. This can be performed in VCP during the cord setup phase by assigning this ID to the start node. The periodically exchanged HELLO messages also contain the domain ID, thus, joining nodes also received this domain identifier. Furthermore, if two networks are getting into each other’s communication range, a node receiving HELLO messages from another domain automatically becomes a gateway node. It then stores this information into the local DHT by hashing the identifier of the router towards the neighboring network. If the gateway no longer receives HELLO messages from the detected neighbor, it removes the gateway information from the DHT. The basic procedure is depicted in Figure 3 (G and R denoting gateway and router nodes, respectively).

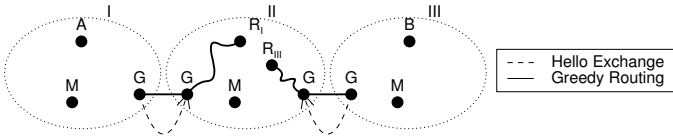


Fig. 3. Exchange and processing of received HELLO messages

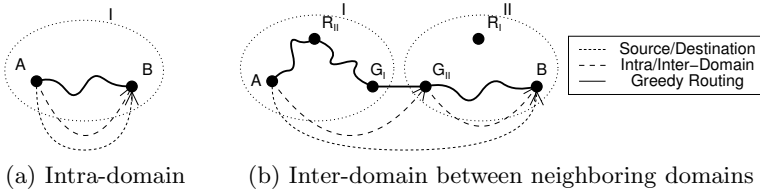


Fig. 4. VCP routing within a domain and between neighboring domains

Direct communication between two nodes in arbitrary domains requires global topology information, i.e. the gateway information needs to be distributed into all VCP domains. VCP’s greedy routing is only used within a domain (Figure 4a).

Inter-domain routing can be supported using the available router nodes, i.e. nodes storing information about neighboring domains [12]. Figure 4b outlines such a scenario. An indirection to the router node is used together with source routing on domain level. However, no transit domains are supported yet.

2.3 Extended Inter-domain Routing Framework

In order to develop a generalized inter-domain routing framework VCP, we had to define several roles, which need to be executed by the network nodes:

- *Gateway* nodes are responsible for detecting neighboring domains, storing this information in the local DHT, and to provide forwarding capabilities to remote domains.
- *Router* nodes represent a virtual function storing all available gateways to a particular domain. They basically provide all the inter-domain routing functionality using indirections as known from peer-to-peer routing.
- *Moderator* nodes maintain, update, and exchange domain tables with moderators in remote domains. Thus, they are responsible for creating all the relevant domain-level topology information.

Figure 5 depicts the setup of routing information. After detecting neighboring domains using the HELLO mechanism, the gateway node forwards this information to a local router node responsible for the detected domain, i.e. a node storing information for the associated hash entry. If the local routing table changes, this information is further forwarded to the moderator node, and, via the basic VCP inter-domain routing also to moderators in the neighboring domains.

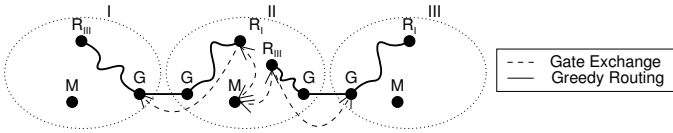


Fig. 5. Routing of EXCHANGE_DOMAIN_SET messages

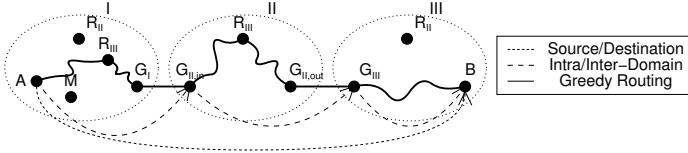


Fig. 6. Inter-domain routing between arbitrary domains

Based on the established routing information, messages can be routed within a domain using the standard VCP greedy routing techniques, between neighboring domains relying on the indirection via the router node, and now, even between arbitrary domains exploiting the knowledge provided by the moderator nodes. Figure 6 outlines the message forwarding over a transit domain. In the source domain, a router node (R_{III}) has been created by the moderator. Thus, a message towards domain III is first routed to R_{III} . The indirection points towards an adequate transit domain (here, domain II), to which the message is forwarded using an appropriate gateway node. From within domain II, the message is forwarded as described for inter-domain routing between neighboring domains.

3 ACO-Based Routing Heuristic

Organizing inter-domain routing between arbitrary domains in an optimized way has a high complexity: First, the routing tables, i.e. the inter-domain network topology needs to be updated and maintained in order to ensure stable topology information and loop-free routes. This requires an extremely high amount of network traffic for topology control if dynamics and mobile nodes are considered. Secondly, the complexity of the routing tables and the paths that need to be calculated might be too high for use on embedded sensor nodes. Therefore, classical routing algorithms cannot be used, even on domain level. Basically, two problems need to be solved as illustrated in Figure 7:

- First, the inter-domain routing needs to be organized, i.e. the path between source and destination domains. This represents a *macroscopic* view to the routing problem. For example, as shown in Figure 7a, two possible routes exist between domains I and VII.
- Secondly, *microscopic* level problems need to be solved, i.e. which particular gateway node should be used for routing between two connected domains. This problem is outlined in Figure 7b.

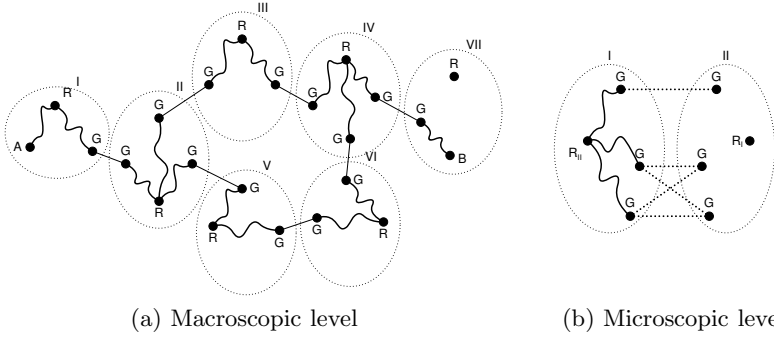


Fig. 7. Routing problems at macroscopic (domain) and microscopic (gateway) level associated with inter-domain routing

In this paper, we propose a routing heuristic based on ACO that is able to cope with these two problems while ensuring a high degree of robustness to topology changes. Also, the selected solution is rather accurate, i.e. selected routes are close the theoretical shortest path. In the following, we outline the ACO based routing heuristic.

3.1 ACO

Ant Colony Optimization (ACO) is a biologically inspired technique simulating the foraging process of social insects [9, 14]. ACO uses a graph $G(N, E)$, N denoting the nodes and E undirected edges, respectively. Two nodes $i, j \in N$ are neighbors if $(i, j) \in E$. Each edge E_{ij} is annotated with some cost. A path is a sequence of nodes and edges between a source and a destination node. The objective of ACO is to find a path between source and destination with minimal costs.

During initialization, each edge $(i, j) \in E$ in the graph G is associated with some initial pheromone level (weight) τ_{ij} :

$$\tau_{ij} \leftarrow \tau_0, \forall (i, j) \in E \tag{1}$$

A complete iteration of the ACO algorithm consists of three steps:

1. *Stepwise probabilistic solution*

Setting up a path is based on stepwise estimation for each edge (i, j) according to Equation 2. Here, \mathcal{N}_i^k depicts the neighborhood of the k -th ant at node i .

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha}{\sum_{l \in \mathcal{N}_i^k} \tau_{il}^\alpha} & j \in \mathcal{N}_i^k \\ 0 & j \notin \mathcal{N}_i^k \end{cases} \tag{2}$$

2. Deterministic pheromone update

After finding a solution, the ant returns. On this path, loops are eliminated by checking whether a path includes the same node twice. Furthermore, the returning ant updates the pheromone level for all edges (i, j) on the path. The new pheromone concentration is calculated according to Equation 3.

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau^k \quad (3)$$

3. Pheromone evaporation

In order to make the algorithm robust even in case of high dynamics in the topology, the pheromone needs to be evaporated over time for all the edges. Basically, the pheromone level is decremented over time by some value $\rho \in (0, 1]$ as shown in Equation 4.

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}, \forall (ij) \in E \quad (4)$$

The algorithm converges if a solution reaches some certain quality level or if no more changes are performed.

ACO has already been successfully applied to several problems in networking. Most importantly, early approaches to routing need to be named such as the AntNet [6] proposal. Here, ACO has been used to set up probabilistic routing tables for standard Internet routing. This work has been directly used in the AntHocNet [7] algorithm, which has been designed for use in MANET environments, thus, in very dynamic networks with rapidly changing network topologies. It turned out that ACO was perfectly able to handle these dynamics.

Hierarchical solutions relying on a combination of ACO and table-driven routing on a higher layer have been investigated for example in the HopNet approach [20]. In this paper, we use a similar scheme but using ACO on the higher (domain) level. As a further step, even combined routing and task allocation in mobile sensor networks has been investigated [17]. In this work, not only routing in mobile networks has been considered but also the distribution of multiple tasks to sensor nodes generating network traffic with different profiles (bursty, constant but high traffic volume, etc.). Obviously, ACO seems to be a perfect candidate for handling dynamics in the network topology with low overhead.

3.2 Optimized Inter-domain Routing

In order to apply ACO to the problem of inter-domain routing in virtual coordinate based networks, we need to construct a graph, define a solution for the pheromone update, and find appropriate parameters for this update. We interpret the entire network as graph G and each VCP domain as a node $k \in G$. For each available gateway between two domains, we draw an edge $(i, j) \in E$ connecting the domains i and j . Thus, different to the classical ACO, we allow multiple edges between nodes. The resulting algorithm follows the ACO principles:

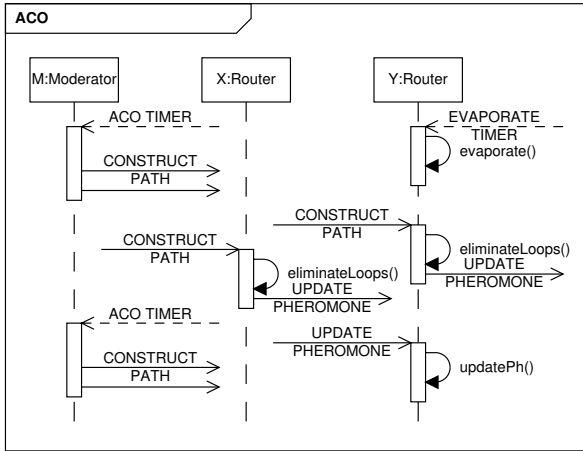


Fig. 8. Estimation of ACO-based routing heuristics

1. Set up a probabilistic route, estimate resulting costs;
2. Prune routing loops, update the costs for each gateway such that the probability to chose the best gateway reflects the estimated costs;
3. Update the costs while the algorithm is running to incorporate dynamic topology changes.

ACO is thus used to weight the used gateways in order to find a shortest (cost minimal) path between two domains. In the scope of this paper, we use the hop count as a routing metric to derive the costs for each gateway. Topology control and minimal cost routing is performed using artificial ants transmitted between the VCP routing domains. This technique is very robust to changes in the network topology. Domains that become connected to the network can be quickly integrated using such explorer ants. Removing a domain because no more gateways are available leads to a short period of inconsistent routing (which is typical for ACO-based heuristics). However, as the costs decrease quickly, this has no influence on active parts of the network.

Figure 8 shows a sequence diagram outlining the stepwise creation of a solution as well as the update of the costs for the gateway nodes. After receiving the ACO_TIMER signal, the moderator of a domain initiates the setup of paths to each reachable domain. Thus, it sends CONSTRUCT_PATH to the router nodes responsible for the respective destination domains. In order to cope with the dynamic topology at domain level, the CONSTRUCT_PATH messages can limit the maximum costs and the maximum number of hops.

When a message arrives at the destination router, the path is cleaned up and an UPDATE_PHEROMONE message is sent back to the destination using source routing along the stored nodes. The pheromone value τ_G represents the cost of each gateway entry in the routing table. After receiving an UPDATE_PHEROMONE,

Table 1. ACO parameter selection

Parameter Values	
t_{ACO}	1
ρ	0.05, 0.1, 0.5 , 0.75
$t_{Evaporate}$	0.2, 0.5, 1.0
$\Delta\tau$	0.2, 0.5, 1.0

the cost value of the gateway is updated according to Equation 5. The initial cost value is a small value τ_0 , K represents the path costs.

$$\tau_G \leftarrow \tau_G + \Delta\tau, \text{ with } \Delta\tau = \frac{C}{1+K}, C = \text{const} \quad (5)$$

The use of path costs in the pheromone update as some nice properties [9]: The quality of a solution is increased, a good solution can already be found using only a few explorer ants, and the quality of the solution becomes almost independent of the parameter α in Equation 2. Thus, the cost of a gateway in the routing table is directly proportional to the length of the entire path to the destination domain.

The evaporation process runs in parallel with the cost update. The parameter ρ influences the speed and quality of the routing convergence. For $\rho = 0$, no convergence is to be expected and for large ρ , the algorithm quickly converges to suboptimal solutions. Furthermore, the degree of mobility needs to be considered for identifying an optimal value for ρ .

We analyzed all the ACO parameters using some initial simulations. More details on the simulation framework and the used parameters are discussed in Section 4. In this initial simulation, we used a network consisting of four domains of nine nodes. We calculated the optimal paths offline using the Dijkstra shortest path algorithm. All the analyzed parameters are listed in Table 1.

The parameter t_{ACO} depicts the time between the periodic evaluations of the routing table. The evaporation factor ρ describes the evaporation speed. The delay between two evaporations is defined by $t_{Evaporate}$. We further analyzed the impact of C indirectly represented by $\Delta\tau$.

As selected quality metrics, we analyzed the success rate, path length, and the difference between the discovered paths to the shortest path. Table 1 also shows the best parameters for our scenario (printed in bold) w.r.t. the selected quality metrics. We used these parameters for the performance evaluation in Section 4. Please note that if completely different scenarios are to analyzed, the parameter selection needs to be repeated.

4 Simulation Results

We investigated the feasibility and the performance of the inter-domain routing concept for VCP in several simulation scenarios. We used our implementation of

Table 2. Simulation parameters

Input Parameter	Value
Number of Nodes	10+10, 100+100, 40
Speed fixed,	1 m s ⁻¹ , 3 m s ⁻¹ , 6 m s ⁻¹
Query period	1 s ⁻¹
Initialization time	100 s
mac.bitrate	2 Mbit s ⁻¹
mac.broadcastBackoff	31 slots
mac.maxQueueSize	14 packets
mac.rtsCts	false

VCP for the simulation tool OMNeT++ to analyze the behavior of the dynamic gateway configuration and the performance of the inter-domain routing using indirections. The basic simulation parameters are summarized in Table 2. We simulated VCP over IEEE 802.11b wireless LAN.

In a first set of simulation experiments, we validated and compared the enhanced inter-domain routing algorithm for VCP to previous results obtained for inter-domain routing between neighboring domains [12]. We first used two networks consisting of 10 nodes each. We allow an initial setup time of 100 s to establish two VCP networks, one for each group. Within this time, a node in the mobile network creates and inserts data items in this VCP domain. After the initialization, the mobile group moves towards the stationary group. After some time, the first nodes get into the radio range of the other group and they start to set up gateway information, and to exchange data packets. The simulation time has been chosen such that for the slow 1 m s⁻¹ scenario the mobile domain completely passes the stationary domain. At higher speeds, multiple of such connections occur. In the second scenario, we used network 100 instead of 10 nodes in both networks to evaluate the impact of a larger number of gateway nodes and longer communication paths.

We evaluated a number of measures such as the available gateways, success rate, communication delays, and path length. All the results clearly show the feasibility of ACO to quickly find adequate routes in this simple setup. Exemplary, we analyzed the path length between source and destination nodes located in opposite domains. Figure 9 shows the simulation results for a sample run in form of a time series plot (we selected speeds of 1 m s⁻¹, 3 m s⁻¹, and 6 m s⁻¹ according to the experimental setup in [21]). As can be seen, after both domains are getting into radio communication range, more and more gateways become available, resulting in decreasing path lengths. When both domains start to depart again, the path lengths are increasing due to the reduced number of gateway nodes. All the other metrics behave as expected from the earlier experiments without ACO optimization. All the other results (data not shown) validate the simple routing behavior.

In a second set of experiments, we evaluated the capabilities of the extended inter-domain routing framework using ACO for inter-domain routing including

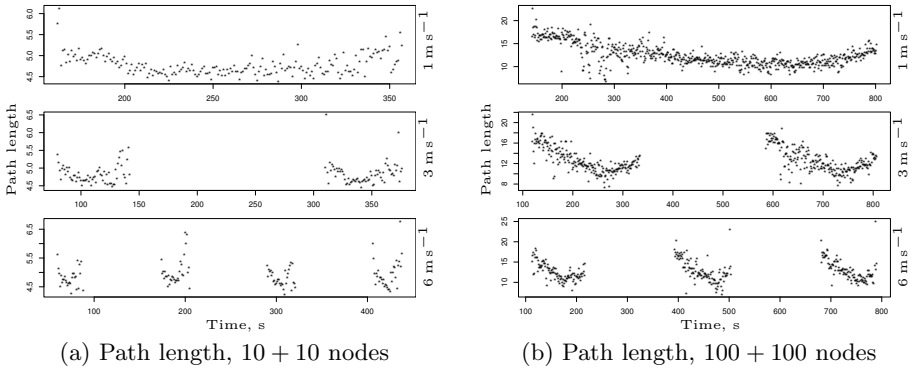


Fig. 9. Distribution of the path length for the two domain scenario

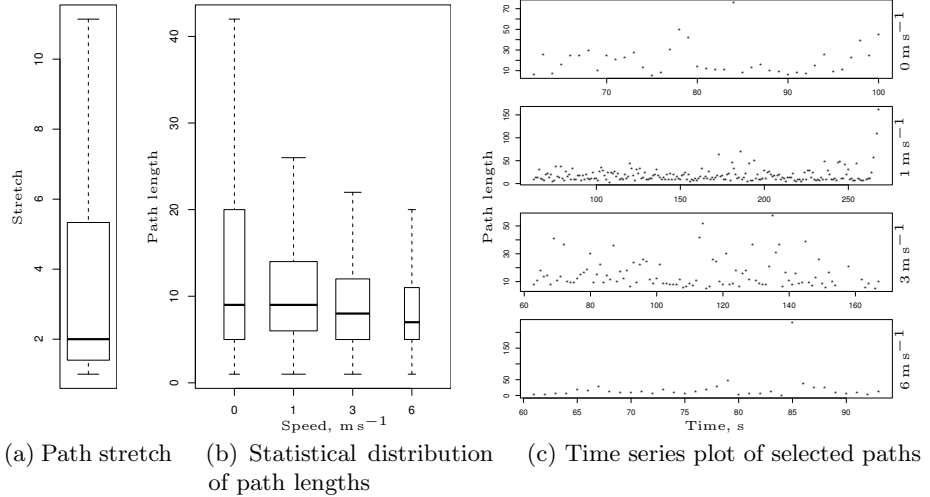


Fig. 10. Distribution of the path length in the multi-network scenario

transit domains. We therefore created a setup including 10 VCP domains, 9 being stationary arranged in form of a rectangle and one network domain being mobile, moving close to the border of the rectangle. Thus, the inter-domain routing framework had to keep track with a rather high degree of system dynamics, i.e. topology changes on domain level due to mobility.

As can be seen in Figure 10, the discovered paths have been quite stable even though the network topology continuously changed on domain level. ACO very quickly reacted on these changes and enforced the use of alternate paths. Figure 10a shows the path stretch, i.e. the distance of discovered paths from the theoretical shortest path. On average, roughly a factor two has to be considered, which is extremely promising for fully self-organizing routing protocols in highly

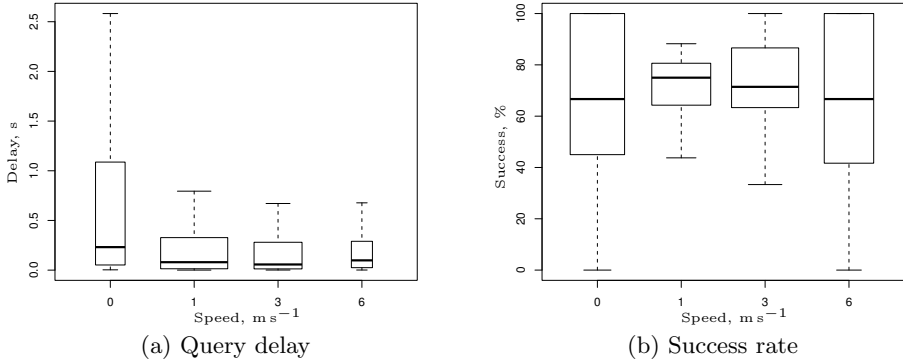


Fig. 11. Query delay and success rate in the multi-network scenario

dynamic environments. Figures 10b and 10c show further statistics (statistical distribution, selected time series plots) of the path length for different speeds of the mobile domain.

We finally evaluated the query delay and the success rate for the multi-network scenario. As shown in Figure 11a, the transmission delay decreases with increasing dynamics, i.e. higher speeds of the mobile network domain. This effect was expected as the ACO algorithm needs some time to reinforce the use of alternate paths by updating the associated pheromone level. The setup for 6 m s⁻¹ shows reduced delays, which is due to the scenario setting. The used MAC protocol tried to resent packets and the mobile domain returned to its previous position before the MAC layer gave up. The same effect can be seen in the plot of the success rate in Figure 11b. The success rate drops for increasing speed from a median of about 90 % in the stationary case to 80 % for the mobile 3 m s⁻¹ case. However, for 6 m s⁻¹ example, the success rate increases again to more than 90 %.

5 Conclusion and Further Challenging Issues

We studied the capabilities of a bio-inspired routing heuristic, the Ant Colony Optimization approach, for inter-domain routing in virtual coordinate-based network environments. We first established a generalized routing framework that is able to maintain information about inter-connected domains. In particular, the framework provides a microscopic view on gateways directly connecting neighboring domains and a macroscopic view on the high-level domain topology. Using indirect directions, routes can be established between arbitrary nodes in any domain. The ACO-based routing heuristic provides means for routing well suited even in networks with significant dynamics, i.e. established and broken connections between multiple domains due to node movements. According to our simulation results, the involved overhead is rather small and the obtained route information are close to shortest path solutions (which could only be calculated theoretically,

because complete topology information cannot be centrally collected in such dynamic environments).

Open issues to be studied include the scalability of the approach w.r.t. the supported number of connected network domains. Furthermore, domain splitting is not yet considered. Split detection can be supported based on the neighborhood management. The main problem is to distinguish between a node failure and a domain split.

Acknowledgments. This work was partially supported by BaCaTec (project “support for inter-domain routing and data replication in virtual coordinate based networks”).

References

1. Akkaya, K., Younis, M.: A Survey of Routing Protocols in Wireless Sensor Networks. *Elsevier Ad Hoc Networks* 3(3), 325–349 (2005)
2. Awad, A., Shi, L.R., German, R., Dressler, F.: Advantages of Virtual Addressing for Efficient and Failure Tolerant Routing in Sensor Networks. In: 6th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2009), Snowbird, UT, pp. 111–118. IEEE (2009)
3. Awad, A., Sommer, C., German, R., Dressler, F.: Virtual Cord Protocol (VCP): A Flexible DHT-like Routing Service for Sensor Networks. In: 5th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2008), Atlanta, GA, pp. 133–142. IEEE (2008)
4. Caesar, M., Castro, M., Nightingale, E.B., O’Shea, G., Rowstron, A.: Virtual Ring Routing: Network routing inspired by DHTs. In: ACM SIGCOMM 2006, Pisa, Italy. ACM (2006)
5. Chau, C.-K., Crowcroft, J., Lee, K.-W., Wong, S.H.Y.: Inter-Domain Routing Protocol for Mobile Ad Hoc Networks. In: ACM SIGCOMM 2008, 3rd ACM International Workshop on Mobility in the Evolving Internet Architecture (ACM MobiArch 2008), Seattle, WA, pp. 61–66. ACM (2008)
6. Di Caro, G., Dorigo, M.: AntNet: Distributed Stigmergetic Control for Communication Networks. *Journal of Artificial Intelligence Research* 9, 317–365 (1998)
7. Di Caro, G., Ducatelle, F., Gambardella, L.M.: AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications, Special Issue on Self-organization in Mobile Networking* 16, 443–455 (2005)
8. Dietrich, I., Dressler, F.: On the Lifetime of Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)* 5(1), 1–39 (2009)
9. Dorigo, M., Stützle, T.: *Ant Colony Optimization*. MIT Press/Bradford Books (2004)
10. Dressler, F.: *Self-Organization in Sensor and Actor Networks*. John Wiley & Sons (December 2007)
11. Dressler, F., Akan, O.B.: A Survey on Bio-inspired Networking. *Elsevier Computer Networks* 54(6), 881–900 (2010)
12. Dressler, F., Awad, A., Gerla, M.: Inter-Domain Routing and Data Replication in Virtual Coordinate Based Networks. In: IEEE International Conference on Communications (ICC 2010), Cape Town, South Africa. IEEE (2010)

13. Dressler, F., Awad, A., German, R., Gerla, M.: Enabling Inter-Domain Routing in Virtual Coordinate Based Ad Hoc and Sensor Networks. In: 15th ACM International Conference on Mobile Computing and Networking (MobiCom 2009), Poster Session, Beijing, China, ACM (2009)
14. Ducatelle, F., Di Caro, G.A., Gambardella, L.M.: Principles and applications of swarm intelligence for adaptive routing in telecommunications networks. *Swarm Intelligence* 4(3), 173–198 (2010)
15. Flury, R., Pemmaraju, S.V., Wattenhofer, R.: Greedy Routing with Bounded Stretch. In: 28th IEEE Conference on Computer Communications (IEEE INFOCOM 2009), Rio de Janeiro, Brazil. IEEE (2009)
16. Kim, Y.-J., Govindan, R., Karp, B., Shenker, S.: Geographic Routing Made Practical. In: *USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2005)*, San Francisco, CA. USENIX (2005)
17. Labella, T.H., Dressler, F.: A Bio-Inspired Architecture for Division of Labour in SANETs. In: Dressler, F., Carreras, I. (eds.) *Advances in Biologically Inspired Information Systems - Models, Methods, and Tools. SCI*, vol. 69, pp. 209–228. Springer, Heidelberg (2007)
18. Lin, C.-H., Liu, B.-H., Yang, H.-Y., Kao, C.-Y., Tasi, M.-J.: Virtual-Coordinate-Based Delivery-Guaranteed Routing Protocol in Wireless Sensor Networks with Unidirectional Links. In: 27th IEEE Conference on Computer Communications (IEEE INFOCOM 2008), Phoenix, AZ. IEEE (2008)
19. Tsai, M.-J., Wang, F.-R., Yang, H.-Y., Cheng, Y.-P.: VirtualFace: An Algorithm to Guarantee Packet Delivery of Virtual-Coordinate-Based Routing Protocols in Wireless Sensor Networks. In: 28th IEEE Conference on Computer Communications (IEEE INFOCOM 2009), Rio de Janeiro, Brazil, IEEE (2009)
20. Wang, J., Osagie, E., Thulasiraman, P., Thulasiram, R.K.: HOPNET: A Hybrid ant colony OPTimization routing algorithm for Mobile ad hoc NETwork. *Elsevier Ad Hoc Networks* 7(4), 690–705 (2009)
21. Zhou, B., Cao, Z., Gerla, M.: Cluster-based Inter-domain Routing (CIDR) Protocol for MANETs. In: 6th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2009), Snowbird, UT, pp. 19–26. IEEE (2009)
22. Zhou, B., Xu, K., Gerla, M.: Group and Swarm Mobility Models for Ad Hoc Network Scenarios Using Virtual Tracks. In: *MILCOM 2004*, Monterey, CA (2004)

Coevolution of Game Strategies, Game Structures and Network Structures

Reiji Suzuki and Takaya Arita

Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
{reiji,arita}@nagoya-u.jp

Abstract. This paper aims at understanding the coevolutionary dynamics of game strategies, game structures and network structures of interactions. As a first approach, we constructed a coevolutionary model of game strategies and network modification strategies, in which individuals can evolve the game structure by developing new strategies that expand the existing payoff properties of Prisoner’s Dilemma (PD) and Symmetric Coordination (SC). Evolutionary experiments showed that the dynamically evolving network brought about the emergence of an adaptive and mutually coordinating network from an isolated and defective population through a shift from a PD to a SC-type game structure, which bootstrapped the subsequent occurrence of adaptive coevolutionary cycles based mainly on a PD-type game structure.

Keywords: Coevolution, evolutionary game, network structures, Prisoner’s Dilemma, Symmetric Coordination, artificial life.

1 Introduction

Dynamics of mutual interactions between network topologies and states of the nodes are attracting much attention in various scientific and engineering fields [1,2]. In evolutionary game theory, the spatial locality of interaction and reproduction has been regarded as a key factor for evolution of cooperation [4-6]. The evolution of strategies in evolving network structures is currently being investigated extensively [3]. This is because cooperative behaviors and network structures can coevolve by affecting their evolutionary dynamics mutually in real world situations – both physical and biological. This attention is also due to the recent interest in complex network structures in social relationships [8,9].

Zimmerman and Eguiluz constructed a model in which the neighboring network structure of an individual can be changed according to the results of games with neighbors, in addition to the evolution of the Prisoner’s Dilemma (PD) strategies [10]. They adopted a simple rule that the links between mutually defected individuals were rewired with other randomly selected individuals. They found that the emergence of the cooperative leader who had the largest payoff in the cluster of cooperative agents brought about the global cooperation. Pacheco *et al.* assumed a situation termed “active linking processes,” in which

there are different birth and death rates of links based on a combination of the strategies [11]. An important finding is that the effect of rapid evolution of the network structure could be interpreted as a transformation of the payoff matrix in an existing game. They also discussed it in the context of repeated games [12]. Tanimoto recently discussed a relationship between assortativity by degree of the evolved network and emerging cooperation in PD, showing that the weak (or strong) dilemma makes the assortativity of emerging networks positive (or negative) [13].

These studies were all based on the strong assumption that while every agent may have its own strategy for modifying its neighboring network, they all adopt the same fixed rule for rewiring. From this viewpoint, we constructed an evolutionary model in which each individual not only has a strategy for PD to play with its neighboring members on the network, but also has its own strategy for changing its neighboring structure of the network [14]. The behavior of this system was complex. We observed coevolutionary cycles of cooperating behaviors and the network structures, reflecting the dynamic aspect of the emergence and collapse of cooperative networks in a real world (see [14] for detailed analyses).

In this study, we focus on the evolution of game structures as another essential property of a real human society. Previous studies mainly discussed the evolution of strategies in the context of a unique 2×2 game such as PD or Snow Drift. However, in addition to choosing a strategy from the existing ones, it is also possible to evolve or expand the game structure by developing another new strategy of which the relationship with the existing ones reflects the existing game structure or constraint. For example, in a standard PD, agents may devise new strategies which bring about more beneficial cooperation, but they may also be exploited more heavily by existing strategies, due to some environmental constraints.

Our purpose is to clarify the coevolutionary dynamics of game structures and network structures of interactions. As a first approach, we expanded an evolutionary model of our previous work [14] to a version of a game, in which individuals can evolve the game structure by developing new strategies that expand the existing properties of Prisoner's Dilemma (PD) and Symmetric Coordination (SC). By starting experiments from the initial population of the minimal set of strategies, we discuss whether and how game strategies, game structures, and their network structures can coevolve by comparing the cases with and without evolution of network structures.

2 Expandable PD/SC Game

We assumed a game, termed an Expandable PD/SC game, in which individuals can evolve the game structure by developing new strategies that expand the existing properties of Prisoner's Dilemma (PD) and Symmetric Coordination (SC).

In this model, we start the population with a minimal set of strategies. Let us call them strategy 1 and 2. There is a PD-type game structure between them.

A new strategy can be developed from its adjacent strategies. For example, a new strategy, let us call it strategy 3, could be developed from the strategy 2 in the initial population. This expands the size of the payoff matrix, and another PD-type structure with expanded payoffs appears between the new strategy 3 and its adjacent strategy 2. At the same time, a SC-type structure appears between the new strategy 3 and its distant strategy 0. Similarly, a new PD structure appears between the strategy x and $x + 1$ if a strategy $x + 1$ is developed from existing strategies, and there appear SC structures between these new strategies and other distant strategies. Through this process, individuals can increase the size of the matrix indefinitely by developing an indefinite number of integer strategies. However, for simplicity, we limit its maximum value to M , which is large enough for discussion.

This situation could be interpreted as a kind of innovative evolution of decision making processes, technologies and so on, if we regard the difference in the strategy number also reflects the qualitative difference between such options or methods. Strategy $x + 1$ can be interpreted as an improvement of the existing strategy x , and may have a conflict (PD) with it because they tend to share some social or physical resources. The succeeding strategy $x + 2$ is a further improvement but more different from the strategy x , and tends to have less conflict (SC) with strategy x . However, it can be exploited by the existing option $x + 1$.

Specifically, let us assume that the maximum strategy value in the current population is m . In this case, each individual can take an integer value $(1, 2, \dots, m)$. If individuals A and B take strategies s_A and s_B respectively, the set of payoff values for these individuals p_A and p_B are determined by the following equation:

$$(p_A, p_B) = \begin{cases} (s_A, s_A) = (s_B, s_B) & \text{if } s_A = s_B, \\ (-S \times s_B, T \times s_A) & \text{if } s_A - s_B = 1, \\ (T \times s_A, -S \times s_B) & \text{if } s_B - s_A = 1, \\ (0, 0) & \text{otherwise.} \end{cases} \tag{1}$$

Table II shows an example payoff matrix of Expandable PD/SC game when $T = 1.1$, $S = 0.3$ and $M = 5$. The properties of this matrix are summarized as follows:

- There is a structure of the PD game between two adjacent strategies. The strategy with a higher number can be interpreted as a cooperator, and the lower one can be interpreted as a defector in the context of a standard two-person prisoner’s dilemma game. The parameter T determines the relative benefit of successful defection to that of mutual cooperation that is the same as the higher strategy value. The parameter S determines the relative cost of being exploited to the benefit of mutual defection that is the same as the lower strategy value.
- There is a structure of the SC game between two distant strategies. The individuals can obtain the payoff that is the same as their own strategy value only when they have the same strategy. Otherwise, their payoff becomes 0.

Table 1. An example payoff matrix of Expandable PD/SC game when $M=5$. Note that this payoff matrix is a snapshot during evolution of game structure when the maximum strategy value is 5. This matrix evolves its size gradually according to the invention of a new strategy.

opponent	1	2	3	4	5
player 1	(1, 1)	(2.2, -0.3)	(0, 0)	(0, 0)	(0, 0)
2	(-0.3, 2.2)	(2, 2)	(3.3, -0.6)	(0, 0)	(0, 0)
3	(0, 0)	(-0.6, 3.3)	(3, 3)	(4.4, -0.9)	(0, 0)
4	(0, 0)	(0, 0)	(-0.9, 4.4)	(4, 4)	(5.5, -1.2)
5	(0, 0)	(0, 0)	(0, 0)	(-1.2, 5.5)	(5, 5)

(player's score, opponent's score)
 $T > 1, S > 0$

- It is beneficial for the whole population to share the higher numbered strategy. At the same time, the degree of these payoff properties increases as the strategy values increase because of the increase in the absolute values of payoffs.

By using this expandable matrix, we can discuss how the evolution of game structure can affect the coevolution of game strategies and network structures.

3 Model

We constructed a coevolutionary model of strategies for the Expandable PD/SC game and network structures described above. This is basically based on our previous work on coevolution of strategies for Prisoner's Dilemma and network structures [14].

A population of N individuals are represented as nodes in the network and each (non directional) link between the two nodes represents that a mixed game will be conducted between the two individuals. Each individual has the information of four genes: g_{gs} , g_{na} , g_{ns} and g_{nd} . $g_{gs}(= 1, 2, \dots M)$ directly encodes a strategy for the game. A set of g_{na} , g_{ns} and g_{nd} (0 or 1 respectively) represents the strategy for modifying its neighboring network structure as illustrated in Table 2.

Each step consists of the three phases defined as follows:

1. Each individual plays Expandable PD/SC games using g_{gs} against all neighboring (directly connected) individuals respectively, and obtains payoffs. At the same time, for each game, a fixed value σ is subtracted from the obtained payoff. σ is a constant which decides the relative difference in the payoff between the individual who played a game and the individual who did not play. The total payoff obtained in all participating games is taken as the fitness of each individual.

Table 2. A set of genetic information for network modification

gene \ value	0	1
g_{na}	does nothing	creates a new link with a randomly selected individual
g_{ns}	does nothing	removes all links with the individuals with the same action
g_{nd}	does nothing	removes all links with the individuals with the different action

2. For each individual, if there are any neighboring individuals whose fitness is higher than that of the individual itself, the genetic information of the focal individual is replaced by that of the neighboring individual with the highest fitness. If there are more than one individual whose fitness tie as the highest among neighbors, an individual is selected randomly from them. Then, for each gene in all individuals, mutation occur with a small probability. E.g., for g_{gs} , a mutation occurs with probability p_{mg} . Such a mutation adds a randomly selected value from $\{-1, 1\}$ to the current value of g_{gs} . If a generated value exceeds its domain, another mutation is operated on the original value again. As for g_{na}, g_{ns} and g_{nd} , a mutation may occur with probability p_{mn} , and inverts the corresponding genetic value. Note that the strategy of an isolated individual cannot be replaced by other strategies, but a mutation can occur. All updates of the genetic information occur at the same time.
3. Each individual modifies its neighboring network structure by using the results of games in phase 1 and its current network-modifying strategies. If $g_{ns} = 1$, the individual removes all links with the individuals whose action is the same as that of the individual itself. If $g_{nd} = 1$, the individual removes all links with the individuals whose action is different from that of the individual itself. In addition, if $g_{na} = 1$, the individual creates a new link with a randomly selected individual who was not connected with itself in phase 1.

This is repeated for G times.

4 Results

We conducted experiments using the following parameters: $N = 1000$, $M = 20$, $p_{mg} = p_{mn} = 0.02$ and $G=5000$. We adopted the condition for the payoff $T = 1.1$, $S = 0.3$ and $\sigma = 1.5$. The initial population was generated with initial values of g_{gs} that were randomly decided from 1 and 2; the genetic values of g_{na} , g_{ns} and g_{nd} were randomly assigned from $\{0, 1\}$. We adopted this initial condition to see whether and how dynamic evolutionary process can emerge from a simple and the least adaptive situation of Prisoner’s Dilemma through the coevolution of game strategies, game structures and network structures.

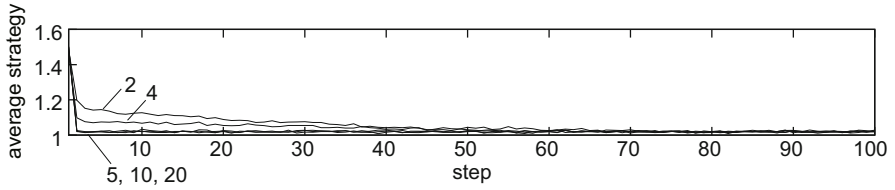


Fig. 1. The evolution of the average value of game strategies during initial 100 steps on regular networks with the degree $D=2, 4, 10, 20$ or 50

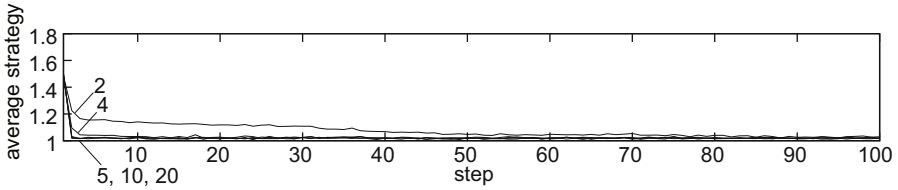


Fig. 2. The evolution of the average value of game strategies during initial 100 steps on random networks with the average degree $D=2, 4, 10, 20$ or 50

4.1 Experiments with Fixed Network Structures

First, we conducted experiments with fixed network structures. In these cases, we omitted the phase 2 in each step, so that the network structure was not modified by the individuals' network modification strategies through time steps. We adopted the following fixed structures: 1) one-dimensional regular networks in which each individual was connected with D ($= 2, 4, 10, 20$ or 50) neighboring individuals, 2) a random network with the average degree D ($= 2, 4, 10, 20$ or 50).

Fig. 1 shows typical examples of the evolution of the average value of game strategy through the initial 100 steps in cases of regular networks. The horizontal axis represents the step, and each line shows the average value of game strategy with the corresponding value of D . We see from this figure that the average strategy decreased very quickly from around 1.5, and converged to 1.0 in all cases. This means that adaptive populations did not emerge from the initial population of a PD-type game structure on regular networks. This seems to be due to the relatively high temptation to defect and small “sucker’s payoff”. We also see that the smaller D was, the slower the game strategy decreased. This is expected because that the higher spatial locality of small D values tended to retard the invasion by the defect-like strategy 1.

Random fixed network evolved similar to the cases of regular networks; the average strategy quickly decreased to 1.0 within 100 steps in the all cases of random networks, as shown in Fig. 2.

As a whole, we can say that the population with fixed networks could not evolve adaptive relationship at all.

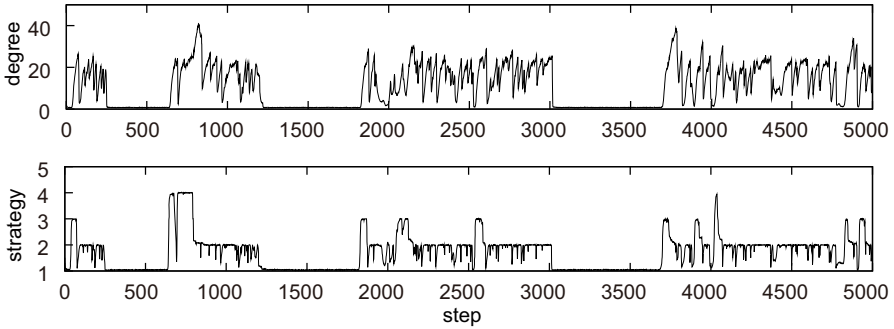


Fig. 3. An example evolutionary process of the average value of game strategies and the average degree over 5000 steps

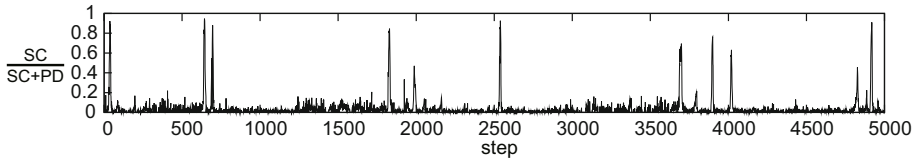


Fig. 4. An example evolutionary process of the proportion of the game structure

4.2 Experiments with Coevolving Game Strategies and Network Structures

We conducted experiments with coevolving game strategies and network structures. We adopted the same parameters as the ones used in the previous section, and also adopted a random network with the average degree 2 as an initial network. Fig. 3 shows an example evolution of the average value of game strategies and the average degree over $G = 5000$ steps. Note that we observed the qualitatively similar phenomenon in every trial with this experimental condition. Contrary to the previous experiments, the average game strategy sometimes rapidly increased from around 1.0 and kept high value for several hundred steps, with a high average degree. Fig. 4 shows the evolution of the proportion of the game structures played between different game strategies in this experiment. The distribution of the game structures (PD and SC) between different strategies was calculated for each step. The value in Fig. 4 shows the proportion of games with SC structures among all games between different strategies. We see that it was basically small, which means that the games between different strategies were basically PD games. At the same time, we can also see that the proportion of SC games did increased rapidly for several times, together with the rapid increase in the average strategy shown in Fig. 3. Fig. 5 shows the evolution of the average values of network modification strategies in the same experiment. Adaptive population was seen to emerge when both network modification strategies and game strategies could evolve. These results clearly show that the evolution of the game and network structures contributed to the emergence of adaptive relationships between individuals.

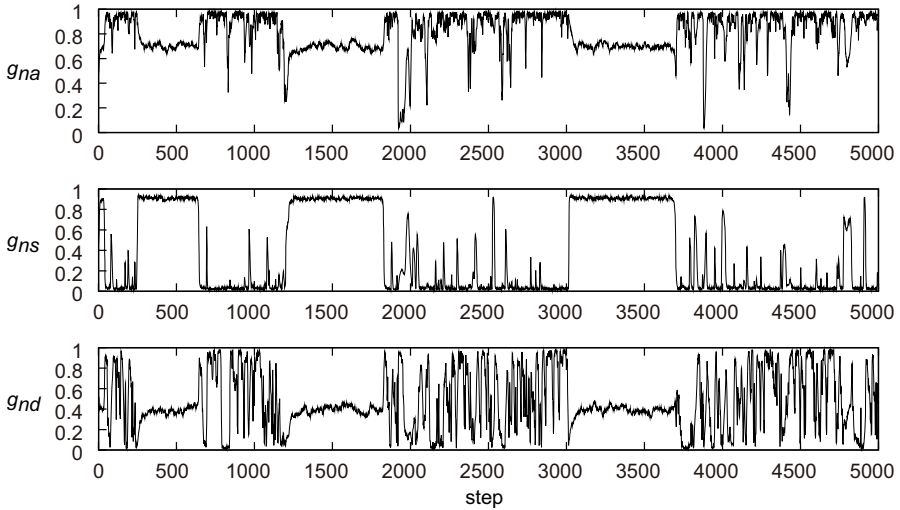


Fig. 5. An example evolutionary process of the average values of network modification strategies

Detailed exploration of one example will help to understand the dynamics of evolution by focusing on the trajectory of the population on the two indices in Fig. 3, as is shown in Fig. 6. The cyclic evolutionary process of these indices is evident; each cycle was traversed basically in a counterclockwise manner. Fig. 7 illustrates a schematic image of evolutionary trajectory, although there were large fluctuations and exceptional moves in the trajectory. There were 6 evolutionary transitions in each cycle, in which can be summarized as follows:

- (i) As observed around the 500th step in Fig. 3, the population sometimes stayed on the stable state in which both indices kept nearly the smallest. In this state, the population consisted of a large number of individuals with genes $(g_{gs}g_{na}g_{ns}g_{nd}) = (1110)$. They removed links with the same game strategy and connected to a randomly selected individual at each step. Because most of the individuals shared the same game strategy 1, they were repeating the rewiring process with randomly selected conspecifics every step. Thus, the network was sparsely and dynamically connected in this state. This state was stable in the sense that a mutant individual with the strategy 2 could not grow their clusters because of the high “sucker’s payoff” and the high participation cost σ .
- (ii) The dynamic property of the network in (i) sometimes created tiny clusters of individuals such as (3101) or (4101), and enabled them to invade into the population as observed at around the 650th step. Because their game strategy 3 or 4 played SC-type games with the game strategy 1, they could not be exploited by the dominant individuals. In addition, they maintained their links with the same strategies while removed the links with different strategies, and they also created a link with a random individual. This

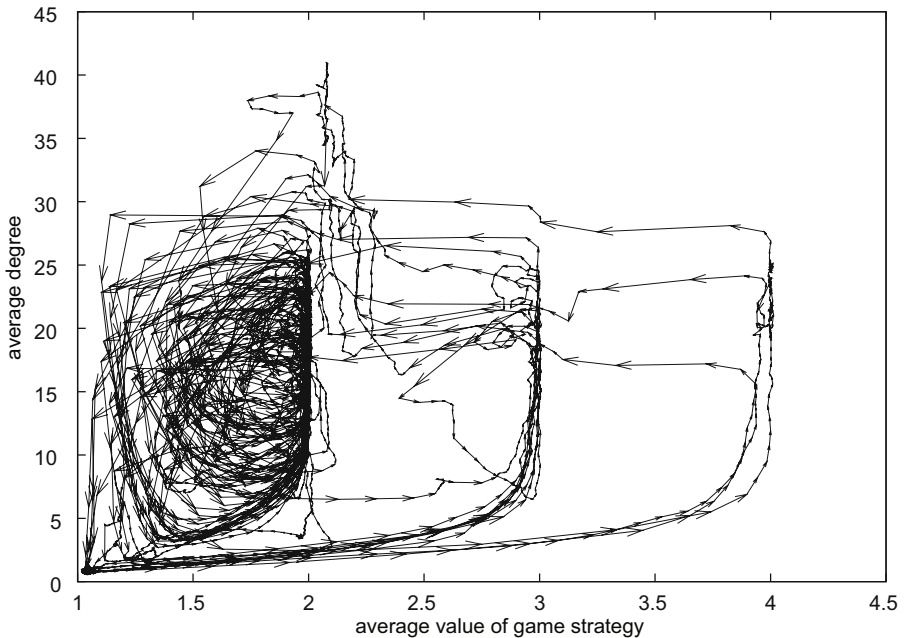


Fig. 6. An example evolutionary trajectory of the average value of game strategies and the average degree

network modifying process enabled them to coordinate with each other and not to be exploited by the existing strategies. As a result, they could occupy the population quickly due to the mutual benefit of the change in the game structure and their adaptive network modification.

- (iii) After such individuals occupied the population, they continued to increase their fitness by increasing their degree because of their high benefit of successful coordination. This gradually made the whole population highly connected. We also observed that g_{nd} gradually evolved from 1 to 0, which means that the individuals came to keep the links with different strategies. It reflects that keeping and increasing the degree was adaptive during this process. It is interesting that the population could keep both high average value of game strategies and high average degree, which is beneficial condition for the whole population but never observed in the cases without network evolution.
- (iv) However, the increase in the average degree changed the global property of population. When the average degree became sufficiently large, the smaller numbered game strategy than that of the dominant strategy by 1 quickly occupied the population, as observed at around the 800th step. They did not change the highly connected network structure, because they could successfully exploit the dominant strategy by using PD-type game structures in a relatively well-mixed population. This process occurred several times until the most population was occupied by the game strategy 1.

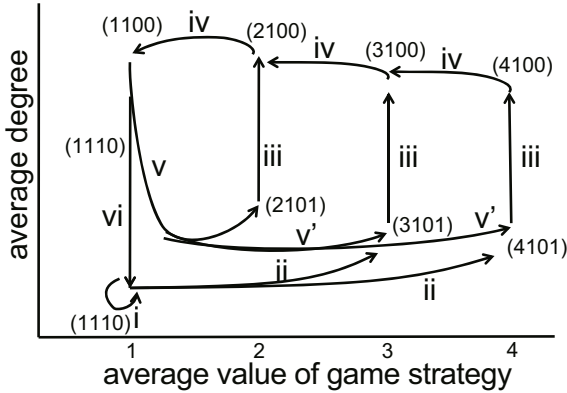


Fig. 7. A schematic image of evolutionary trajectory. Each arrow with the number corresponds to each evolutionary transition explained in the text. Each 4-length gene string ($g_{gs}g_{na}g_{ns}g_{nd}$) also represents the dominant genetic information at the corresponding state of the population on this graph.

(v) When the most population was occupied by the game strategy 1, it became adaptive not to play a game. This is because that their net payoff from a game with each other was $-0.5 (= 1.0 - \sigma)$. It is smaller than the one when there was no game between individuals (0.0). Thus, the individuals (1110) that removed the links with the same dominant strategy but kept the ones with the different strategy began to occupy the population. This caused the rapid decrease in the average degree, which enabled the remaining small clusters of the individuals (2101) that killed the links with different strategies to grow again due to their cooperative benefit of a PD-type game structure. As a result, the population often repeated cycles composed of transitions (iii)-(iv)-(v), as observed at the 800-1200th step. This mechanism was basically similar to the ones observed in [14], but each cycle occurred in shorter time scale, which seems to be due to the relatively high mutation rates.

In addition, we sometimes observed the evolution of the higher numbered game strategy 3 or 4 through this process, as observed at around the 2100th or 2500th step (v' in Fig. 7). Although this seems to be due to the similar mechanism to the one in the transition (ii), it is interesting that the cyclic evolution facilitated the occurrence of these adaptive evolutionary transitions through the shift from a PD-type game structure to SC-types.

Clearly, these cyclic behaviors brought about more adaptive benefit for the whole population than the one in the least adaptive state observed in cases of fixed networks.

(vi) However, the whole population was sometimes completely occupied by the game strategy 1 during (iv) or (v), which made the population converge to (i).

As a whole, we can say that an emergence of a mutually coordinating network from an isolated and defective population through a shift from a PD to a SC-type game structure brought about the subsequent maintenance of adaptive coevolutionary cycles mainly based on a PD-type game structure.

5 Conclusion

We discussed whether and how game strategies and structures can coevolve with their network structures of interactions. As a first step, we conducted evolutionary experiments in which individuals can evolve the game structure by developing new strategies that expand the existing properties of Prisoner's Dilemma (PD) and Symmetric Coordination (SC), in addition to be able to modify their neighboring structure of network.

Evolutionary experiments based on an individual-based model in which the game strategy and the network modification strategy coevolve clearly showed that the dynamically evolving network brought about the emergence of an adaptive and mutually coordinating network from an isolated and defective population through a shift from a PD to a SC-type game structure, which bootstrapped the subsequent occurrence of adaptive coevolutionary cycles based mainly on a PD-type game structure. It should be emphasized that this process was not observed at all in experiments with several fixed networks, which implies that this kind of complex interactions among game strategies, game structures and network structures could be an important factor for maintenance of adaptive behaviors in a real human society.

Future work includes detailed analyses on the effects of parameters on the behaviors of the population, and experiments based on different kinds of game-theoretical situations.

References

1. Gross, T., Blasius, B.: Adaptive Coevolutionary Networks: A Review. *Journal of Royal Society Interface* 5(20), 259–271 (2008)
2. Gross, T., Sayama, H. (eds.): *Adaptive Networks: Theory, Models and Applications*. Springer (2009)
3. Perc, M., Szoinoki, A.: Coevolutionary Games - A Mini Review. *Biosystems* 99(2), 109–125 (2010)
4. Axelrod, R.: *The Evolution of Cooperation*. Basic Books, New York (1984)
5. Nowak, M.A., May, R.M.: Evolutionary Games and Spatial Chaos. *Nature* 359, 826–829 (1992)
6. Suzuki, R., Arita, T.: Evolution of Cooperation on Different Combinations of Interaction and Replacement Networks with Various Intensity of Selection. *International Journal of Bio-Inspired Computation* 3(3), 151–158 (2011)
7. Watts, D.J., Strogatz, S.H.: Collective Dynamics of 'Small-world' Networks. *Nature* 393, 440–442 (1998)
8. Barabási, A.L., Albert, R.: Emergence of Scaling in Random Networks. *Science* 286, 509–512 (1999)

9. Watts, D.J.: *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, Princeton (1999)
10. Zimmermann, M.G., Eguiluz, V.M.: Cooperation, Social Networks, and The Emergence of Leadership in A Prisoner's Dilemma with Adaptive Local Interactions. *Physical Review E* 72, 056118 (2005)
11. Pacheco, J.M., Traulsen, A., Nowak, M.A.: Coevolution of Strategy and Structure in Complex Networks with Dynamical Linking. *Physical Review Letter* 97, 258103 (2006)
12. Pacheco, J.M., Traulsen, A., Ohtsuki, H., Nowak, M.A.: Repeated Games and Direct Reciprocity under Active Linking. *Journal of Theoretical Biology* 250, 723–731 (2008)
13. Tanimoto, J.: The Effect of Assortativity by Degree on Emerging Cooperation in A 2×2 Dilemma Game Played on An Evolutionary Network. *Physica A* 389(16), 3325–3335 (2010)
14. Suzuki, R., Kato, M., Arita, T.: Cyclic Coevolution of Cooperative Behaviors and Network Structures. *Physical Review E* 77(2), 021911 (2008)

Reconstructing History of Social Network Evolution Using Web Search Engines

Jin Akaishi^{1,6}, Hiroki Sayama^{1,2,3,4}, Shelley D. Dionne^{1,4,5}, Xiujian Chen^{1,5},
Alka Gupta^{1,4,5}, Chanyu Hao^{1,4,5}, Andra Serban^{1,4,5},
Benjamin James Bush^{1,3}, Hadassah J. Head^{1,3}, and Francis J. Yammarino^{1,4,5}

¹ Collective Dynamics of Complex Systems Research Group

² Department of Bioengineering

³ Department of Systems Science and Industrial Engineering

⁴ Center for Leadership Studies

⁵ School of Management

Binghamton University, State University of New York

P.O. Box 6000, Binghamton, NY 13902-6000, USA

⁶ Kumamoto National College of Technology

2627 Hirayamashinmachi, Yatsushiro, Kumamoto, Japan

{jakaishi, sayama, sdionne, xichen, agupta1, chao2, aserban1,
bbush2, hhead1, fjyammo}@binghamton.edu

Abstract. We propose a simple web search engine based method for collecting approximated historical data of temporally changing social adaptive networks, which are rather difficult to obtain experimentally in conventional research methods. In the proposed method, a search query string is combined with additional keywords that specify inclusion/exclusion of specific years to limit the search results to a particular time point. Using the proposed method, we reconstructed the temporal evolution of a social network from 2005 to 2009 of 93 individuals who are important in the US economy. We measured centralities of those individuals for every year and found several illustrative cases where the temporal change of centrality of an individual correctly captured the actual events that are related to him/her over this time period. These results indicate the effectiveness of the proposed method. Limitations and future directions of research are discussed.

Keywords: Social networks, adaptive networks, network evolution, centrality, data collection, web search engines.

1 Introduction

The importance of temporal dynamics of network topology and their coupling with node/link state dynamics has been increasingly recognized in network science communities [1,2]. However, it is generally difficult to experimentally obtain large-scale data of real-world social network evolution over time [3,4]. The exceptions to this are some well-studied electronic data sets, such as citations in scientific publications and friendship networks in social media (e.g., Facebook and YouTube),

which have been causing a concentration of network analysis research on these limited sets of networks.

Recently, Lee et al. [5] proposed a new web search engine based data collection method by which a researcher can easily reconstruct social networks of any kind by simply using the number of Google search results (hits) for two names as the link weight between them. They demonstrated the effectiveness of this method by applying it to the social network reconstruction for the 109th US Senate members. They also considered the temporal change of this network over several months in late 2006. However, the network “snapshot” data had to be acquired during the time period under investigation, so that the entire data collection process required that data be taken over the course of several months. A remaining open question is how one could use web searches to reconstruct the history of social network evolution *retrospectively*.

We propose a similar web search engine based method for collecting approximated historical data of temporarily changing social adaptive networks by adding to a search query string other keywords that specify the inclusion/exclusion of specific years to limit the search results to a particular time point. For example, one can selectively search for results that are most likely relevant only to year 2006 by adding “2006”, “-2007”, “-2008”, “-2009” and “-2010” to the search query string. We implemented a prototype of this data collection method using the Google AJAX Search API, and applied it to the reconstruction of network history from 2005 to 2009 for 93 individuals who are important in the US economy and industries. In this Work-In-Progress paper, we report the procedure of the proposed method and preliminary results obtained using it, as well as its limitations and future directions of research.

2 Method

The data collection method we propose takes as an input a list of keywords to be searched. Keywords can be of any kind, but in what follows, we focus on the names of the people we wish to include in our social network reconstruction. To each name we add some additional personally indefinable information (in this case, the person’s affiliation). This helps reduce some of the errors associated with the fact that many people often share the same name.

Search queries are generated from this list as follows:

1. Two entries (corresponding to 2 people) are chosen from the list described above.
2. A year is chosen (e.g., 2007). The search query will be designed to examine the nature of the relationship that existed between the two people during the chosen year.
3. To eliminate the influence of search results corresponding to documents created in years after the chosen year, we compose a series of partial search queries which exclude the unwanted years. This is done in Google by placing

a negative symbol directly to the left of the unwanted year. That is, if the chosen year is 2007, then the partial search queries -2008, -2009, and -2010 must be included in the search query. Since documents often discuss events which occurred in previous years, we did not add keywords to exclude years prior to the chosen year.

4. Our completed search query is composed by putting together the elements described in steps 1, 2 and 3. For example, to examine the relationship that existed between Warren Buffet and Alan Greenspan in 2007, we used the following search query:

“Alan Greenspan” “Federal Reserve” “Warren Buffett” “Berkshire Hathaway” “2007” -2008 -2009 -2010

Search queries were generated for all possible pairs from the list of people under study for every year. Each search query was then used to perform a search using the Google AJAX Search API. The code was implemented in Java. The number of search results obtained from each search was recorded and used as the weight, w , of the corresponding link of the social network in that year. Note that link weights in this social network are symmetric. To capture the potential asymmetry in social relationships, we calculate the asymmetric weight, \hat{w} , of each directed link between keywords i and j at year t as follows:

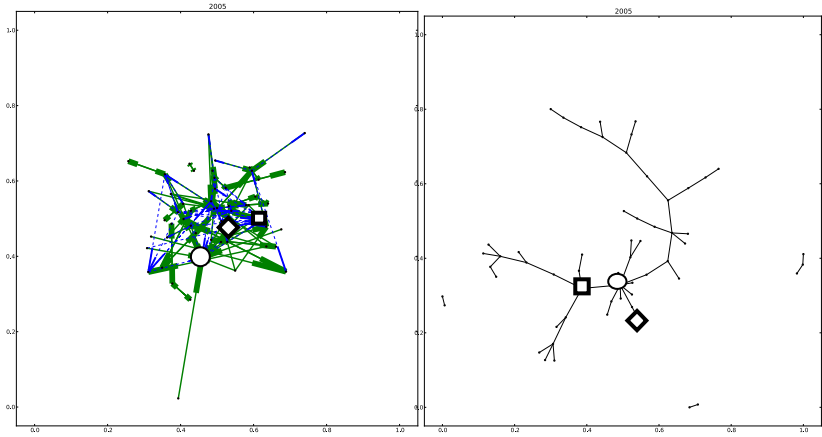
$$\hat{w}(i, j, t) = \frac{w(i, j, t)}{\sum_{k \neq i} w(i, k, t)}$$

For simplicity, we will henceforth refer to “directed networks,” “directed links,” and “directed weights” simply as “networks,” “links,” and “weights,” respectively.

3 Preliminary Results

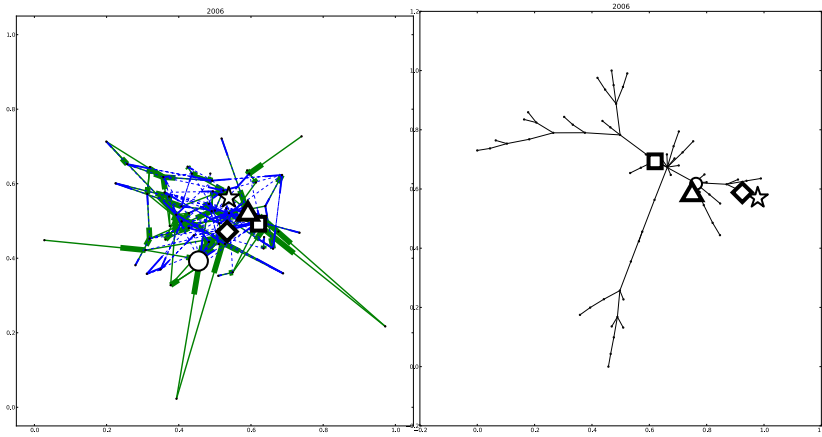
We conducted a preliminary experiment to test the proposed method. We made a list of 93 people who have played major roles in the US economy recently. Using this list, we constructed a social network for each year from 2005 to 2009 as described in section 2. We chose these years in order to observe relational changes surrounding the 2008 economic crisis. Data collection was done using a single laptop over 15 hours.

These 93 individuals are high profile business leaders that were either leading Fortune 100 companies or under intense media scrutiny during the 2008-09 recession. The companies covered in our paper span several major industry sectors essential for the US economy, such as financial services, insurance, banking, investment, retails, technology, and defense contracting. All sectors were heavily affected by the slumping economy, primarily the financial sector in the middle of the housing market downturn and stock market collapse.



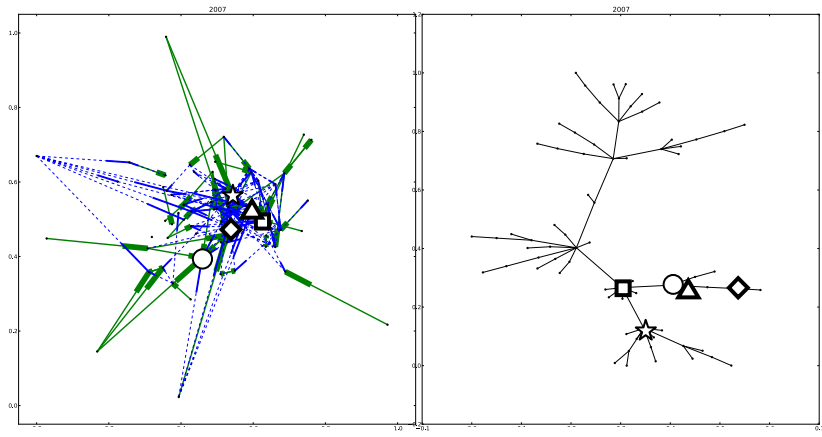
(a) 2005

(b) 2005



(c) 2006

(d) 2006



(e) 2007

(f) 2007

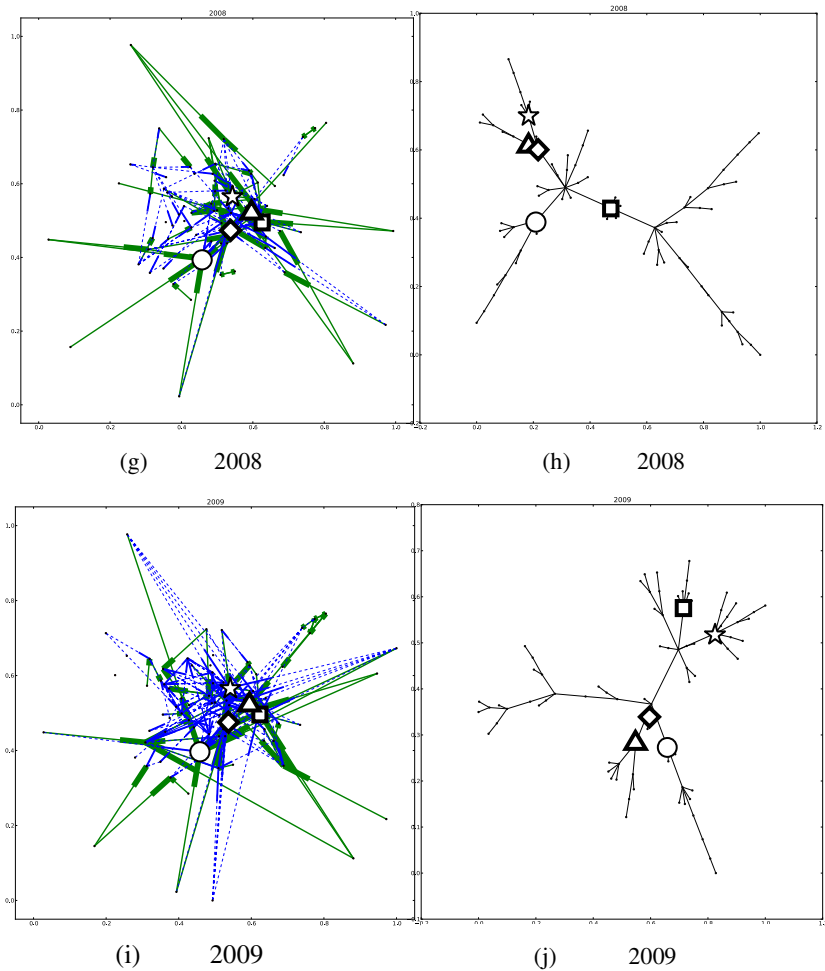


Fig. 1. (a), (c), (e), (g) and (i): Network visualizations corresponding to the data obtained for the years 2005, 2006, 2007, 2008 and 2009, respectively. Dotted lines indicate weak connections. Solid lines indicate strong connections. All the network visualizations use the same node positions. (b), (d), (f), (h) and (j): Maximum spanning trees which are created based on the same data as digraphs. Symbols on the each graph show the positions of 5 individuals of interest: Squares, triangles, circles, stars and dimonds indicate Warren Buffett, Timothy Geithner, Alan Greenspan, Lloyd Blankfein and Henry Paulson, respectively.

Figures 1 and 2 illustrate the results of the experiment. Figure 1 (a), (c), (e), (g) and (i) are the network visualizations corresponding to the data obtained for the years 2005, 2006, 2007, 2008 and 2009, respectively. Dotted lines indicate weak connections, e.g. links whose weights are between 0.1 and 0.2. Solid lines indicate strong connections, e.g. links whose weights are greater than 0.2. The same node positions are used in all network visualizations so that the changes in link structure will be visible. Figure 1 (b), (d), (f), (h) and (j) are maximum spanning trees which

are created based on the same data as the network visualizations. Symbols in the figures show the positions of 5 individuals of interest: Squares, triangles, circles, stars and dimonds indicate Warren Buffett, Timothy Geithner, Alan Greenspan, Lloyd Blankfein and Henry Paulson, respectively.

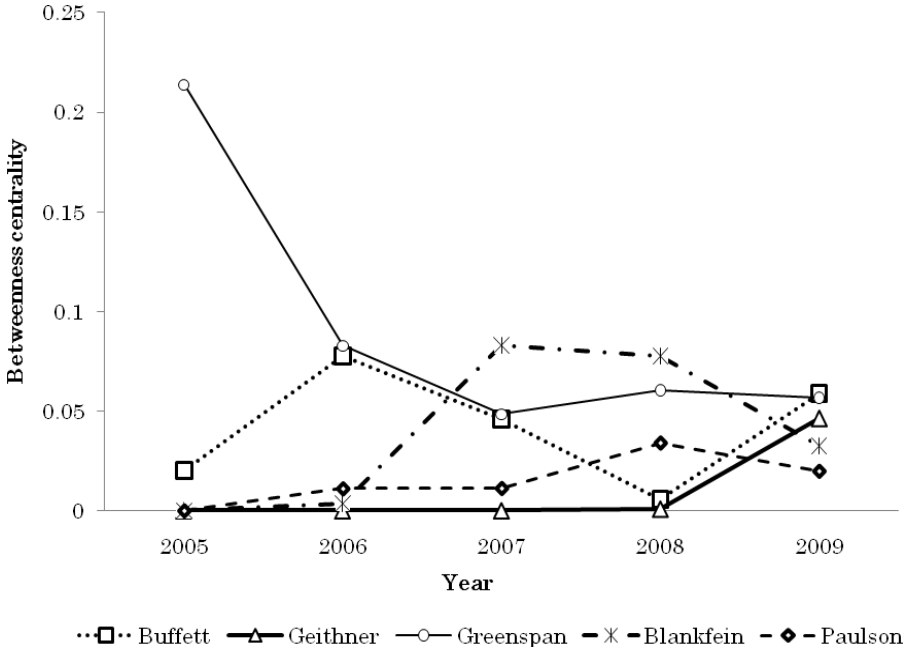


Fig. 2. Betweenness centralities of 5 individuals of interest from 2005 to 2009

Figure 2 shows temporal changes in betweenness centrality of the 5 well known individuals from 2005 to 2009. The betweenness centrality in Fig. 2 is normalized for comparison so that the sum of the betweenness centralities of all nodes is 1. These plots correctly reflect several actual events that happened in the US economy. A straightforward example is that the centrality of Alan Greenspan, who was the chairman of the Federal Reserve until 2006, drastically decreased over time. In contrast, the centrality of Timothy Geithner, who became the US Treasury Secretary in 2009, increased in 2009.

An interesting observation on the relationship between Lloyd Blankfein, CEO of Goldman Sachs, and Henry Paulson, US Treasury Secretary and former CEO of Goldman Sachs, is that their closeness in this social network varies from time to time. Their closeness in 2006 (Fig. 1 (d)) is apparently because Blankfein took Paulson’s position as Goldman Sachs CEO in that year. They were apart in the following year (Fig. 1 (f)), but again came close to each other in 2008 when the economic crisis occurred (Fig. 1(h)). Although it was not widely known, Goldman Sachs was AIG’s largest trading partner in 2008 [6]. When Blankfein noticed that AIG was having

severe liquidity problems, he called Paulson for help [7]. The U.K. Guardian reported this as follows [8]:

“Paulson’s office calendar at the Treasury, obtained by the New York Times through a Freedom of Information request, revealed that he spoke to Blankfein two dozen times during the September week when the Treasury bailed out AIG. That was “far more frequently” than Paulson talked to any other Wall Street executive.”

Our network visualization for 2008 correctly captures this strong relationship between Blankfein and Paulson.

These correspondences between the history of those individuals and the changes observed in the figures suggest that our method is capable, to some extent, of illuminating changes that have occurred in real-world social networks.

4 Conclusion

In this paper, we have proposed a new method for acquiring historical social network data retrospectively using web search engines. Data were collected by recording the number of Google search results yielded by carefully crafted search queries. To test our proposed method, we conducted a preliminary experiment in which we reconstructed a social network of 93 important figures in the US economy. Five annual snapshots of the network evolution were generated for years ranging from 2005 to 2009. Temporal changes in network topology and node centrality measure reflected several real-world events, such as shifts of power/influence and temporary formation of strong relationships. These results demonstrate the potential of our proposed method for examining changes that have occurred in real-world social networks.

Our method has several limitations. First, our method relies solely on simple Google searches using heuristically crafted search queries, and therefore its validity is yet to be fully examined and established. Additional filtering using more contextual information may help improve the quality of search results. Our method also has a computational limitation, since the number of search queries required to construct a social network grows quadratically as the number of keywords increases. Yet another limitation is the error in search results. Multiple attempts of an identical Google search query sometimes produce very different results: the number of search hits might vary in orders of magnitude. In order to minimize this type of error, we need to collect data multiple times (e.g., using multiple PCs) and aggregate them.

In future studies, we plan to conduct a more rigorous validation of our data collection method and to develop more advanced temporal filtering techniques using textual contents of the documents returned by web search engines.

References

1. Gross, T., Sayama, H. (eds.): Adaptive Networks. Springer, Heidelberg (2009)
2. Braha, D., Bar-yam, Y.: From Centrality to Temporary Fame: Dynamic Centrality in Complex Networks. Complexity 12(2), 59–63 (2006)

3. Doreian, P., Stokman, F.N. (eds.): *Evolution of Social Networks*. Gordon and Breach, New York (1997)
4. Wasserman, S., Faust, K.: *Social Network Analysis*. Cambridge University Press, Cambridge (1999)
5. Lee, S.-H., Kim, P.-J., Ahn, Y.-Y., Jeong, H.: Googling social interactions: Web search engine based social network construction. *PLoS ONE* e11233 (2010)
6. Morgenson, G.: Behind Insurer's Crisis, Blind Eye to a Web of Risk, *New York Times*, September 27, A1 (2008),
<http://www.nytimes.com/2008/09/28/business/28melt.html?fta=y>
7. Morgenson, G., Natta, D.V.: Paulson's Calls to Goldman Tested Ethics, *New York Times*, August 9 (2009),
<http://www.nytimes.com/2009/08/09/business/09paulson.html>
8. Clark, A.: How close are Goldman Sachs's connections with the US treasury? *The U.K. Guardian*, August 10 (2009),
<http://www.guardian.co.uk/business/andrew-clark-on-america/2009/aug/10/goldman-sachs-aig-treasury>

Learning and Generalization in Random Automata Networks

Alireza Goudarzi¹, Christof Teuscher², and Natali Gulbahce³

¹ Computer Science and Systems Science Department
Portland State University (PSU), Portland, OR, USA
alirezag@csecs.pdx.edu

² Department of Electrical and Computer Engineering
Portland State University (PSU), Portland, OR, USA
teuscher@pdx.edu

³ Department of Cellular and Molecular Pharmacology
University of California, San Francisco (UCSF), CA, USA
natali.gulbahce@ucsf.edu

Abstract. It has been shown [7,16] that feedforward Boolean networks can learn to perform specific simple tasks and generalize well if only a subset of the learning examples is provided for learning. Here, we extend this body of work and show experimentally that random Boolean networks (RBNs), where both the interconnections and the Boolean transfer functions are chosen at random initially, can be evolved by using a state-topology evolution to solve simple tasks. We measure the learning and generalization performance, investigate the influence of the average node connectivity K , the system size N , and introduce a new measure that allows to better describe the network's learning and generalization behavior. Our results show that networks with higher average connectivity K (supercritical) achieve higher memorization and partial generalization. However, near critical connectivity, the networks show a higher perfect generalization on the even-odd task.

1 Introduction

Pattern recognition is a task primates are generally very good at while machines are not so much. Examples are the recognition of human faces or the recognition of handwritten characters. The scientific disciplines of machine learning and computational learning theory have taken on the challenge of pattern recognition since the early days of modern computer science. A wide variety of very sophisticated and powerful algorithms and tools currently exist [5]. In this paper we are going back to some of the roots and address the challenge of learning with networks of simple Boolean logic gates. To the best of our knowledge, Alan Turing was the first person to explore the possibility of learning with simple NAND gates in his long forgotten 1948 paper, which was published much later [21,25]. One of the earliest attempts to classify patterns by machine came from Olivier Selfridge [19,20] in 1958. Later, many have explored random logical nets made up

from Boolean or threshold (McCulloch-Pitts) neurons: [1-4, 18]. Martland [15] showed that it is possible to predict the activity of a boolean network with randomly connected inputs, if the characteristics of the boolean neurons can be described probabilistically. In a second paper [14], Martland illustrated how the boolean networks are used to store and retrieve patterns and even pattern sequences auto-associatively. Seminal contributions on random Boolean networks came from Stuart Kauffman [9-11] and Weisbuch [26, 27].

In 1987, Patarnello and Carnevali [6, 16] used *simulated annealing* and in 1989 also *genetic algorithms* [17] as a global stochastic optimization technique to train feedforward Boolean networks to solve computational tasks. They showed that such networks can indeed be trained to recognize and generalize patterns. Broeck and Kawai [7] also investigated the learning process in feedforward Boolean networks and discovered their amazing ability to generalize.

In 2007, Teuscher *et al.* [22] presented preliminary results that true RBNs, i.e., Boolean networks with recurrent connections, can also be trained to learn and generalize computational tasks. They further hypothesized that the performance is best around the critical connectivity $K = 2$.

In the current paper, we extend and generalize Patarnello and Carnevali's results to random Boolean networks (RBNs) and use genetic algorithms to evolve both the network topology and the node transfer functions to solve a simple task. Our work is mainly motivated by the application of RBNs in the context of emerging nanoscale electronics [23]. Such networks are particularly appealing for that application because of their simplicity. However, what is lacking is a solid approach that allows to train such systems for performing specific operations. Similar ideas have been explored with none-RBN building blocks by Tour *et al.* [24] and by Lawson and Wolpert [12]. One of the broader goals we have is to systematically explore the relationship between generalization and learning (or memorization) as a function of the system size, the connectivity K , the size of the input space, the size of the training sample, and the type of the problem to be solved. In the current paper, we restrict ourselves to look at the influence of the system size N and of connectivity K on the learning and generalization capabilities. In the case of emerging electronics, such as for example self-assembled nanowire networks used to compute simple functions, we are interested to find the smallest network with the lowest connectivity that can learn how to solve the task with the least number of patterns presented.

2 Random Boolean Networks

A *random Boolean network* (RBN) [9-11] is a discrete dynamical system composed of N nodes, also called *automata*, *elements* or *cells*. Each automaton is a Boolean variable with two possible states: $\{0, 1\}$, and the dynamics is such that

$$\mathbf{F} : \{0, 1\}^N \mapsto \{0, 1\}^N, \quad (1)$$

where $\mathbf{F} = (f_1, \dots, f_i, \dots, f_N)$, and each f_i is represented by a look-up table of K_i inputs randomly chosen from the set of N nodes. Initially, K_i neighbors and

a look-up table are assigned to each node at random. Note that K_i (i.e., the fan-in) can refer to the *exact* or to the *average* number of incoming connections per node. In this paper we use K to refer to the average connectivity.

A node state $\sigma_i^t \in \{0, 1\}$ is updated using its corresponding Boolean function:

$$\sigma_i^{t+1} = f_i(\sigma_{i_1}^t, \sigma_{i_2}^t, \dots, \sigma_{i_{K_i}}^t). \tag{2}$$

These Boolean functions are commonly represented by *look-up tables* (LUTs), which associate a 1-bit output (the node’s future state) to each possible K -bit input configuration. The table’s out-column is called the *rule* of the node. Note that even though the LUTs of a RBN map well on an FPGA or other memory-based architectures, the random interconnect in general does not.

We randomly initialize the states of the nodes (initial condition of the RBN). The nodes are updated synchronously using their corresponding Boolean functions. Other updating schemes exist, see for example [8] for an overview. Synchronous random Boolean networks as introduced by Kauffman are commonly called NK networks or models.

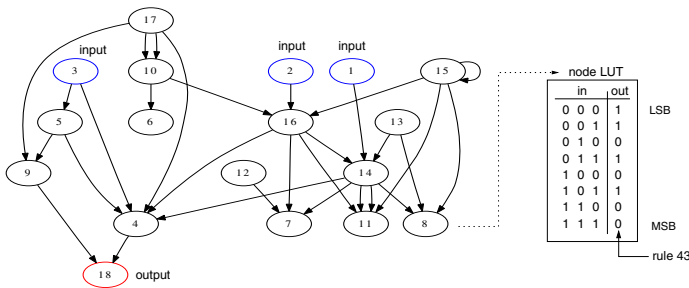


Fig. 1. Illustration of an 18-node RBN with 3 input nodes (node IDs 1, 2, and 3, colored in blue) and 1 output node (node ID 18, colored in red). The average connectivity is $K = 2.5$. The node rules are commonly represented by *lookup-tables* (LUTs), which associate a 1-bit output (the node’s future state) to each possible K -bit input configuration. The table’s out-column is commonly called the *rule* of the node.

The “classical” RBN is a closed system without explicit inputs and outputs. In order to solve tasks that involve inputs and outputs, we modify the classical model and add I input nodes and designate O nodes as output nodes. The input nodes have no logical function and simply serve to distribute the input signals to any number of nodes in the network. On the other hand, the output nodes are just like any other network node, i.e., with a Boolean transfer function, except that their state can be read from outside the network. Figure [1] shows an 18-node RBN with 3 input nodes and 1 output node.

3 Experimental Setup

We use *genetic algorithms* (GAs) to train the RBNs to solve the even-odd, the mapping task, and the bitwise AND task. The *even-odd task* consists of determining if an l -bit input has an even or an odd number of 1s in the input. If the number of 1s is an odd number, the output of the network must be 1, 0 otherwise. This task is admittedly rather trivial if one allows for counting the number of 1s. Also, if enough links are assigned to a single RBN node, the task can be solved with a single node since all the combinations can be enumerated in the look-up table. However, we are not interested to find such trivial solutions, instead, we look for networks that are able to generalize well if only a subset of the input patterns is presented during the training phase. In Section 5 we also use the *bitwise AND task*, which does exactly what its name suggests, i.e., form the logical AND operation bit by bit with two l -bit inputs and one l -bit output. The *mapping task* is used in Section 6 and consists of a l -bit input and an l -bit output. The output must have the same number of l -bits as the input, but not necessarily in the same order. Throughout the rest of the paper, we use I to refer to the total number of input bits to the network. For example, the bitwise AND for two 3-bit inputs is a problem with $I = 6$ inputs.

To apply GAs, we encode the network into a bit-stream that consists of both the network's adjacency matrix and the Boolean transfer functions for each node. The genetic operators consist of the standard mutation and one-point crossover operators that are applied to the genotypes in the network population. We further define a fitness function f and a generalization function g . For an input space M' of size m' and an input sample M of size m we write: $E_M = \frac{1}{m} \sum_{j \in M} d(j)$ with $f = 1 - E_M$, where $d(j)$ is the Hamming distance between the network output for the j^{th} input in the random sample from the input space and the expected network output for that input. Similarly, we write: $E_{M'} = \frac{1}{n} \sum_{j \in M'} d(j)$ with $g = 1 - E_{M'}$, where $d(i)$ is the Hamming distance between the network output for the i^{th} input from the entire input space and the expected network output for that input.

The simple genetic algorithm we use is as following:

1. Create a random initial population of S networks.
2. Evaluate the performance of the networks on a random sample of the input space.
3. Apply the genetic operators to obtain a new population.
4. Continue with steps 2 and 3 until at least one of the networks achieves a perfect fitness or after G_{max} generations are reached.

To optimize feedforward networks (see Section 5), we have to make sure that the mutation and crossover operators do not violate the feedforward topology of the network. We add an order attribute to each node on the network and the nodes accept connections only from lower order nodes.

Since RBNs have recurrent connections, their rich dynamics need to be taken into account when solving tasks, and in particular interpreting output signals.

Their finite and deterministic behavior guarantees that a network will fall into a (periodic or fixed point) attractor after a finite number of steps. The transient length depends on the network's average connectivity K and the network size N [11]. For our simulations, we run the networks long enough until they reach an attractor. Based on [11], we run our networks (with $k < 5$) for $2N$ time steps to reach an attractor. However, due to potentially ambiguous outputs on periodic attractors, we further calculate the average activation of the output nodes over a number of time steps equal to the size N of the network and consider the activity level as 1 if at least half of the time the output is 1, otherwise the activity will be 0. A similar technique was used successfully in [21].

4 Training and Network Performance Definitions

Patarnello and Carnevali [16] introduced the notion of *learning probability* as a way of describing the learning and generalization capability of their feedforward networks. They defined the learning probability as the probability of the training process yielding a network with perfect generalization, given that the training achieves perfect fitness on a sample of the input space.

The learning probability is expressed as a function of the fraction of the input space, $s = \frac{m}{m'}$, used during the training. To calculate this measure in a robust way, we run the training process r times and store both the fitness f and the generalization g values. We define the learning probability as a function of s , $\delta(s) = Pr(g = 1 | f = 1) = \frac{\alpha'(s)}{\alpha(s)}$, where $\alpha(s) = Pr(f = 1)$ is the probability of achieving a perfect fitness after training, i.e., $f = 1$, and where $\alpha'(s) = Pr(g = 1)$ is the probability of obtaining a perfect fitness in generalization, $g = 1$. In the following sections, we will define new measures to evaluate the network performance more effectively.

One can say that the probabilistic measures, such as the learning probability described above, only focus on the perfect cases and hence describe the performance of the training process rather than the effect of the training on the network performance. Thus, we define the *mean training score* as $\beta(s) = \frac{1}{r} \sum_r f_{final}$ and the *mean generalization score* as $\beta'(s) = \frac{1}{r} \sum_r g_{final}$, where f_{final} and g_{final} are the training fitness and the generalization fitness of the best networks respectively at the end of training.

To compare the overall network performance for different training sample sizes, we introduce a *cumulative measure* for all four measures as defined above. The cumulative measure is obtained by a simple trapezoidal integration [28] to calculate the area under the curve for the learning probability, the perfect training likelihood, the mean generalization score, and mean training score.

5 Learning in Feedforward Boolean Networks

The goal of this first experiment was to simply replicate Patarnello and Carnevali's [17] results with feedforward Boolean networks. Figure 2 shows the

learning probability of such networks on the even-odd (RIGHT) and the bitwise AND task (LEFT) for $K = 2$ networks. We observe that as the size I of the input space increases, the training process requires a smaller number of training examples to achieve a perfect learning probability. For $I = 3$, some of the networks can solve a significant number of patterns without training because the task is too easy. We have initially determined the GA parameters (see figure legends), such as the mutation rate and the maximum number of generations experimentally, depending on how quickly we achieved perfect fitness on average. We have found the GA to be very robust against parameter variations for our tasks. These results shown in Figure 2 directly confirm Patarnello and Carnevali’s [17] experiments.

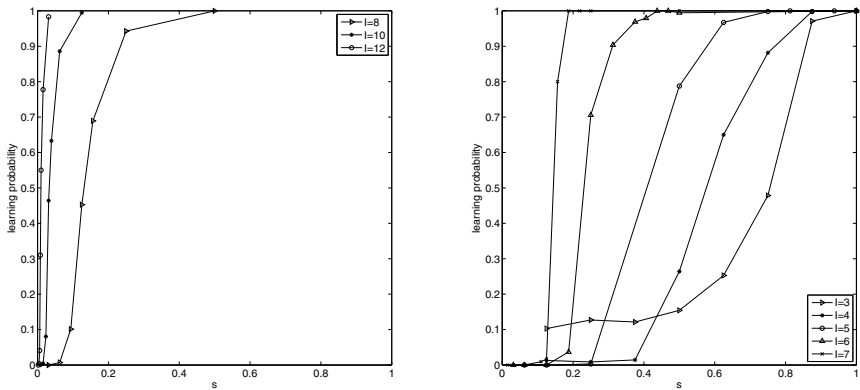


Fig. 2. LEFT: The learning probability of feedforward networks on the bitwise AND task for different input sizes I . $s = \frac{m}{m'}$ is the fraction of the input space used in training. As I increases, the learning process requires a smaller fraction of the input space during the training to achieve a perfect learning probability. RIGHT: The learning probability of feedforward networks on the even-odd task for various input sizes I . As I increases, the learning process requires a smaller fraction of the input space during the training to achieve a perfect learning probability. For $I = 3$, some of the networks can correctly classify a significant number of patterns without training because the task is too easy. For both plots: $N = 50$, $K = 2$, $G_{max} = 3000$, initial population size = 50, crossover rate = 0.6, mutation rate = 0.3. The GA was repeated over 700 runs.

6 Learning in RBNs

Next, we trained recurrent RBNs for the even-odd and the mapping tasks. Figure 3 (LEFT) shows the learning probability of the networks on the even-odd task with different input sizes I . While the problem size increases exponentially with I , we observe that despite this state-space explosion, a higher number of inputs I requires a smaller fraction of the input space for training the networks to achieve a high learning probability. Figure 3 (RIGHT) shows the same behavior for the mapping task, however, since the task is more difficult, we observe a worse

generalization behavior. Also, compared to Figure 2, we observe in both cases that the generalization for recurrent networks is not as good as for feedforward Boolean networks. In fact, for the studied input sizes, none of the networks reaches a learning probability of 1 without training it on all the patterns.

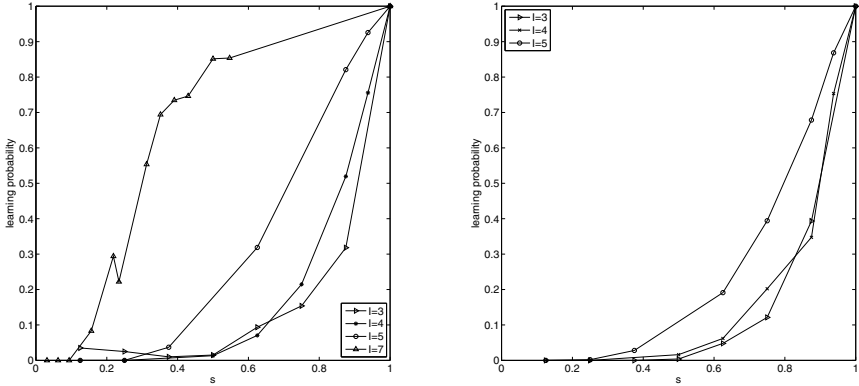


Fig. 3. LEFT: The learning probability of RBNs on the even-odd task for different problem sizes: $I = 3, 4, 5, 7$. With increasing I , the training process requires a smaller fraction of input space in order to reach a higher learning probability. $N = 20$, $G_{max} = 50$, init. population = 50, crossover rate = 0.7, mutation rate = 0.0. We calculate the data over 400 runs for all I s. RIGHT: The learning probability of RBNs on the mapping task for $I = 3, 4, 5$. We observe the same behavior, but the networks generalize even worse because the task is more difficult. $N = 40$, same GA parameters.

To investigate the effect of the average connectivity K on the learning probability, we repeat the even-odd task for networks with $K \in \{1.0, 1.5, 2.0, 2.5, 3.0\}$. The network size was held constant at $N = 20$. In order to describe the training performance, we defined the *perfect training likelihood* measure $\alpha(s)$ as the probability for the algorithm to be able to train the network with the given fraction (s) of the input space (see section 4 for definition).

Considering the perfect training likelihood, the results in Figure 4 (RIGHT) show that for networks with subcritical connectivity $K < 2$, the patterns are harder to learn than with supercritical connectivity $K > 2$. Close to the “edge of chaos”, i.e., for $K = 2$ and $K = 2.5$, we see an interesting behavior: for sample sizes above 40% of the patterns, the perfect training likelihood increases again. This transition may be related to the changes in information capacity of the network at $K = 2$ and needs further investigation with different tasks.

The significant difference between the learning probability and the perfect training likelihood for $s < 0.5$ in Figure 4 is due to the small sample size. It is thus very easy for the network to solve the task correctly, but over all r runs of the experiment, there is no network that can generalize successfully despite

achieving a perfect training score. Also, according to the definitions in Section 4, it is not surprising that for a fraction $s = 1$ of the input space, i.e., all patterns are presented, the learning probability and the perfect training likelihood are different. Out of r runs, the GA did not find perfect networks for the task for all example, but if the networks solve the training inputs perfectly, they will also generalize perfectly because in this case, the training sample input includes all possible patterns.

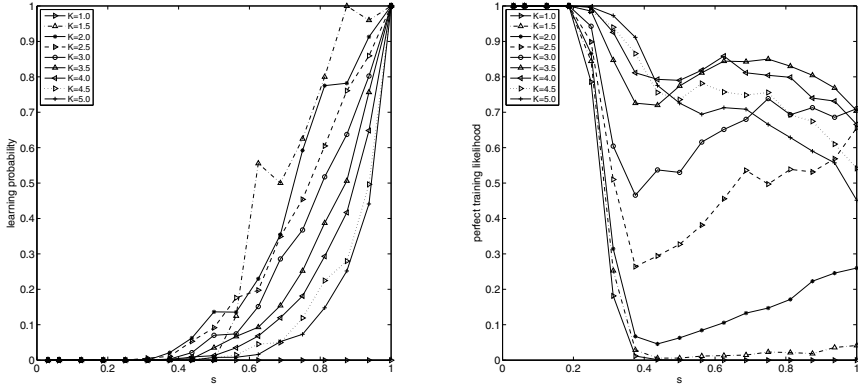


Fig. 4. LEFT: The learning probability of networks with size $N = 15$ and $K \in \{1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.5, 5.0\}$ for the even-odd task of size $I = 5$. Networks with connectivity $K = 1.5, 2, 2.5$ have a higher learning probability. RIGHT: The perfect training likelihood for the same networks and the same task. As the training sample size increases, subcritical connectivity networks are not able to correctly classify all training patterns, while for $K \geq 2$, correctly classifying them is easier.

7 Mean Generalization and Training Score

Figure 5 shows the learning probability (LEFT) and the perfect training likelihood (RIGHT) measured as Patarnello and Carnevali did, i.e., they only counted the number of networks with perfect generalization scores (see Section 4). Thus, if a network generalizes only 90% of the patterns, it is not counted in their score. That means that the probabilistic measures of performance that we used so far have the drawback of describing the fitness landscape of the space of possible networks rather than the performance of a particular network, which we are more interested in. To address this issue, we introduce a new way of measuring both the learning and the generalization capability. We define both of these measures as the average of the generalization and learning fitness over r runs (see section 4).

Figure 6 shows the generalization (LEFT) and the training score (RIGHT) with this new measure. As opposed to Carnevali and Patarnello’s work, where higher K led to a lower learning probability, our results with the new measures

for higher K lead to a higher performance with a better generalization and training score. Our measures therefore better represent the performance of the networks with regards to a given task because they also include networks that can partially solve the task.

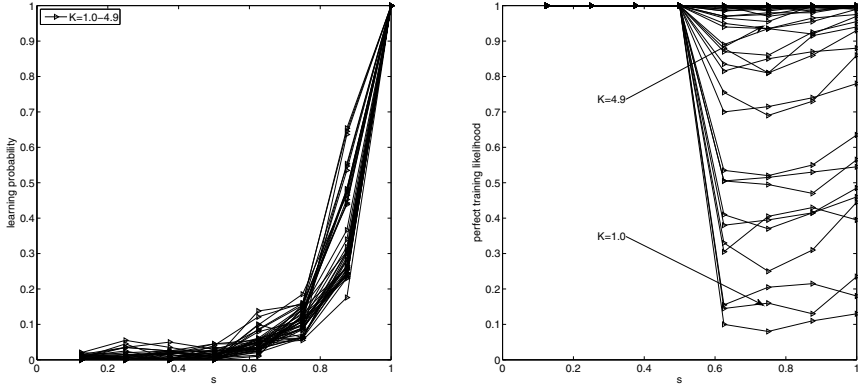


Fig. 5. Learning probability (LEFT) and perfect training likelihood (RIGHT). $I = 3$, $N = 15$, even-odd task. Compared to the learning probability for $I = 5$, there is not much difference between the learning probability of networks with various K for $I = 3$ because of the small input space. However, the perfect training likelihood still increases with K . K ranges from 1.0 to 4.9 with 0.1 increments.

8 Cumulative Measures

In all the previous generalization figures, the question arises which networks are “better” than others, in particular if they do not reach a maximal generalization score when less than 100% of the patterns are presented. This behavior can be observed in Figure 4 (LEFT) for the even-odd task.

Figure 7 shows the *cumulative learning probability* (LEFT) and the *cumulative training likelihood* (RIGHT) determined by integrating numerically (see Section 4 for definitions) the area under the curves of Figure 5. Figure 7 (LEFT) shows that K has no effect on the generalization and that the generalization capability is very low. Figure 7 (RIGHT) shows that higher K increases the chance of perfect training, i.e., the network can be trained to memorize all training patterns. Each cluster of connectivities in Figure 5 (RIGHT) corresponds to a “step” in the curves of Figure 7 (RIGHT).

Figure 8 shows the *cumulative generalization score* (LEFT) and the *cumulative training score* (RIGHT) based on the new measure as introduced in Section 7. We have used the even-odd task for two input sizes, $I = 3$ and $I = 5$. We observe that K has now a significant effect on the generalization score. The higher K , the better the generalization. Moreover, different intervals of K result

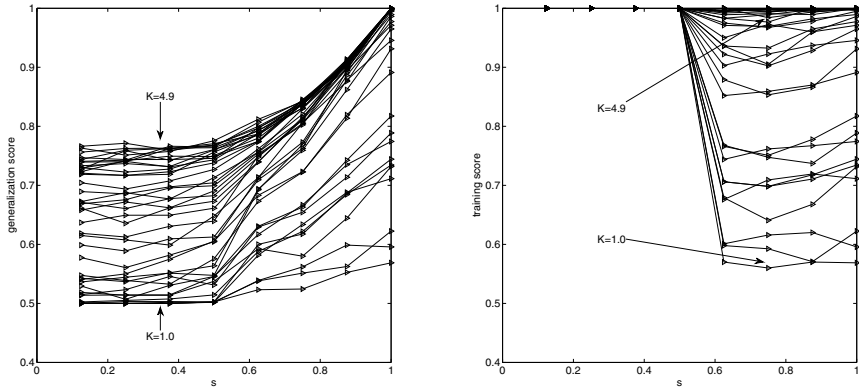


Fig. 6. The new generalization (LEFT) and training score (RIGHT), which better reflects the performance of the networks with regards to a given task. $I = 3, N = 15$, even-odd task. K ranges from 1.0 to 4.9 with 0.1 increments.

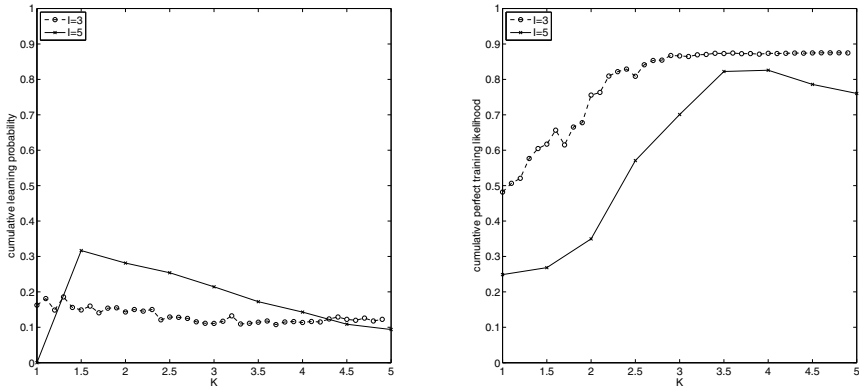


Fig. 7. LEFT: Cumulative learning probability. RIGHT: Cumulative training likelihood. Each data point in this figure corresponds to the area under the curves shown in Figure 5. $N = 15$ in both figures, even-odd task. As one can see, perfect memorization is more likely with higher K , but perfect generalization is more likely for near-critical connectivity $1.5 \leq K \leq 3$. The cumulative learning probability and the perfect training likelihood represent the area under the learning probability and perfect training likelihood curves respectively (see Figure 4 and 5, and Section 4).

in a step-wise generalization score increase. Figure 8 (RIGHT) shows that the cumulative training score for higher K increases the chance of perfect training, i.e., the network can be trained to memorize all training patterns. Also, the higher the input size I , the better the generalization, which was already observed by Patarnello and Carnevali (see also Section 5).

In summary, we have seen so far that according to our new measures, higher K networks both generalize and memorize better, but they achieve perfect generalization less often. The picture is a bit more complicated, however. Our data also shows that for networks around $K = 1.5$, there are more networks in the space of all possible networks that can generalize perfectly. For $K > 1.5$, the networks have a higher generalization score on average, but there is a lower number of networks with perfect generalization. That is because the fraction of networks with perfect generalization is too small with respect to the space of all the networks. For $K < 1.5$, the networks are hard to train, but if we manage to do so, they also generalize well.

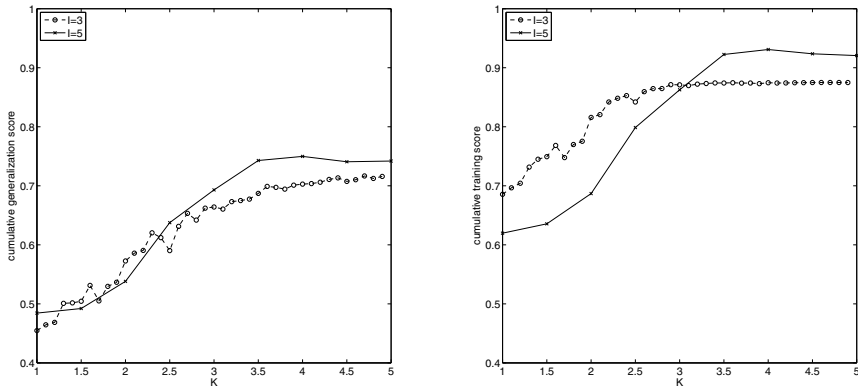


Fig. 8. LEFT: Cumulative generalization score. RIGHT: Cumulative training score. $N = 15$ in both figures, even-odd task, $I = 3$ and $I = 5$. As one can see, both the network's generalization and the memorization capacity increase with K . The cumulative generalization and training score represent the area under the mean generalization and training score curves respectively (see Figure 6 and Section 4).

Figure 9 shows the complete cumulative learning probability (LEFT) and cumulative training likelihood (RIGHT) landscapes as a function of K and N . We observe that according to these measures, neither the system size nor the connectivity affects the learning probability. Also, the networks have a very low learning probability, as seen in Figure 7. That means that the performance of the training method does not depend on the system size and the connectivity and confirms our hypothesis that Carnevali and Patarnello's measure is more about the method than the network's performance.

Finally, Figure 10 shows the same data as presented in Figure 9 but with our own score measures. For both the cumulative generalization score and the cumulative training score, the network size N has no effect on the generalization and the training, at least for this task. However, we see that for the cumulative generalization score, the higher K , the higher the generalization score. The same applies to the cumulative training score. This contrasts what we have seen in Figure 9.

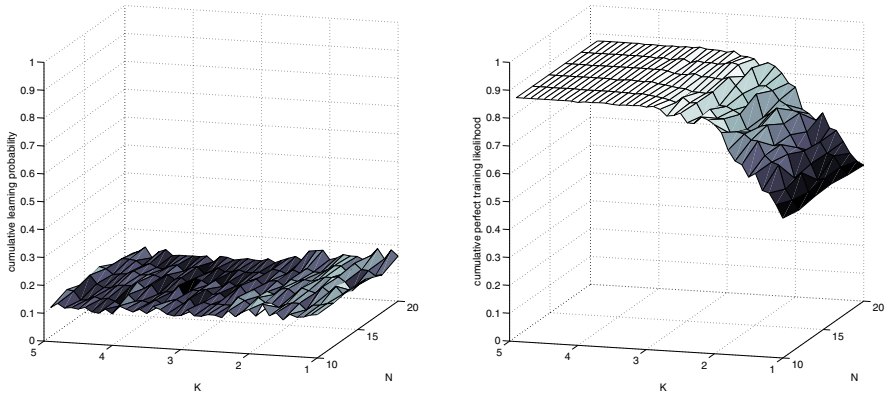


Fig. 9. LEFT: Cumulative learning probability. RIGHT: Cumulative training likelihood. Even-odd task.

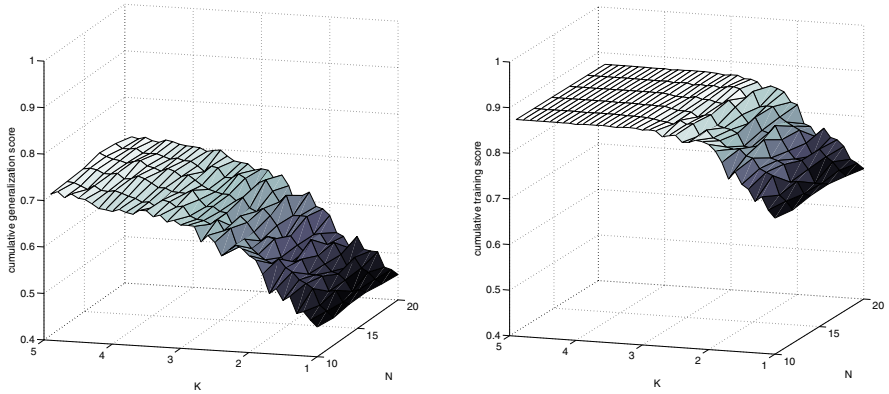


Fig. 10. LEFT: Cumulative generalization score. RIGHT: Cumulative training score. Even-odd task.

9 Discussion

We have seen that Patarnello and Carnevali’s measure quantifies the fitness landscape of the networks rather than the network’s performance. Our newly defined measures applied to RBNs have shown that higher K networks both generalize and memorize better. However, our results suggest that for large input spaces and for $K < 1.5$ and $K > 3$ networks, the space of the possible networks changes in a way that makes it difficult to find perfect networks (see Figures 4 and 5). On the other hand, for $1.5 \leq K < 3$, finding the perfect

networks is significantly easier. This is a direct result of the change in the number of possible networks and the number of networks that realize a particular task as a function of K .

In [13], Lizier *et al.* investigated information theoretical aspects of phase transitions in RBNs and concluded that subcritical networks ($K < 2$) are more suitable for computational tasks that require more of an information storage, while supercritical networks ($K > 2$) are more suitable for computations that require more of an information transfer. The networks at critical connectivity ($K = 2$) showed a balance between information transfer and information storage. This finding is purely information theoretic and does neither consider input and outputs nor actual computational tasks. In our case, solving the tasks depends on the stable network states and their interpretations. The results in Lizier *et al.* do not apply directly to the performance of our networks, but we believe there is a way to link the findings in future work. Compared to Lizier *et al.*, our experiments show that supercritical networks do a better job at both memorizing and generalizing. However, from the point of view of the learning probability, we also observe that for networks with $1.5 \leq K < 3$, we are more likely to find perfect networks for our specific computational tasks.

10 Conclusion

In this paper we empirically showed that random Boolean networks can be evolved to solve simple computational tasks. We have investigated the learning and generalization capabilities of such networks as a function of the system size N , the average connectivity K , problem size I , and the task. As feedforward Boolean networks, RBNs can learn simple tasks, however, their generalization capabilities are not as extraordinary as for feedforward Boolean networks. We have seen that the learning probability measure used by Patarnello and Carnevali [16] was of limited use and have thus introduced new measures, which better describe what the networks are doing during the training and generalization phase. The results presented in this paper are invariant of the training parameters and are intrinsic to both the learning capability of dynamical automata networks and the complexity of the computational task. Future work will focus on the understanding of the Boolean function space, in particular on the function bias.

References

1. Aleksander, I.: Random logic nets: Stability and adaptation. *International Journal of Man-Machine Studies* 5, 115–131 (1973)
2. Aleksander, I.: From Wisard to Magnus: A family of weightless virtual neural machines. In: Austin, J. (ed.) *RAM-Based Neural Networks*. Progress in Neural Processing, vol. 9. World Scientific (1998)
3. Aleksander, I., Thomas, W.V., Bowden, P.A.: WISARD: A radical step forward in image recognition. *Sensor Review* 4, 120–124 (1984)
4. Amari, S.I.: Characteristics of randomly connected threshold-element networks and network systems. *Proceedings of the IEEE* 59(1), 35–47 (1971)

5. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
6. Carnevali, P., Patarnello, S.: Exhaustive thermodynamical analysis of Boolean learning networks. *Europhysics Letters* 4(10), 1199–1204 (1987)
7. Van den Broeck, C., Kawai, R.: Learning in feedforward Boolean networks. *Physical Review A* 42(10), 6210–6218 (1990)
8. Gershenson, C.: Classification of random Boolean networks. In: Standish, R.K., Bedau, M.A., Abbass, H.A. (eds.) *Artificial Life VIII. Proceedings of the Eight International Conference on Artificial Life*, pp. 1–8. MIT Press, Cambridge, MA (2003)
9. Kauffman, S.A.: Metabolic stability and epigenesis in randomly connected genetic nets. *Journal of Theoretical Biology* 22, 437–467 (1968)
10. Kauffman, S.A.: Emergent properties in random complex automata. *Physica D* 10(1-2), 145–156 (1984)
11. Kauffman, S.A.: *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, New York (1993)
12. Lawson, J., Wolpert, D.H.: Adaptive programming of unconventional nano-architectures. *Journal of Computational and Theoretical Nanoscience* 3, 272–279 (2006)
13. Lizier, J., Prokopenko, M., Zomaya, A.: The information dynamics of phase transitions in random boolean networks. In: Bullock, S., Noble, J., Watson, R., Bedau, M.A. (eds.) *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pp. 374–381. MIT Press, Cambridge (2008)
14. Martland, D.: Auto-associative pattern storage using synchronous boolean networks. In: *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, CA, vol. III, pp. 355–366 (1987)
15. Martland, D.: Behaviour of autonomous (synchronous) boolean networks. In: *Proceedings of the First IEEE International Conference on Neural Networks*, San Diego, CA, vol. II, pp. 243–250 (1987)
16. Patarnello, A., Carnevali, P.: Learning networks of neurons with Boolean logic. *Europhysics Letters* 4(4), 503–508 (1987)
17. Patarnello, S., Carnevali, P.: Learning capabilities of boolean networks. In: Aleksander, I. (ed.) *Neural Computing Architectures: The Design of Brain-Like Machines*, ch. 7, pp. 117–129. North Oxford Academic, London (1989)
18. Rozonoér, L.I.: Random logical nets I. *Automation and Remote Control* 5, 773–781 (1969); translation of *Avtomatika i Telemekhanika*
19. Selfridge, O.G.: “Pandemonium”: A paradigm for learning. In: *Mechanisation of Thought Processes: Proceedings of a Symposium Held at the National Physical Laboratory*, pp. 513–526 (1958)
20. Selfridge, O.G., Neisser, U.: Pattern recognition by machine. *Scientific American* 203(2), 60–68 (1960)
21. Teuscher, C.: *Turing’s Connectionism. An Investigation of Neural Network Architectures*. Springer, London (2002)
22. Teuscher, C., Gulbahce, N., Rohlf, T.: Learning and generalization in random Boolean networks. In: *Dynamics Days 2007: International Conference on Chaos and Nonlinear Dynamics*, Boston, MA, January 3-6 (2007)
23. Teuscher, C., Gulbahce, N., Rohlf, T.: An assessment of random dynamical network automata for nanoelectronics. *International Journal of Nanotechnology and Molecular Computation* 1(4), 39–57 (2009)

24. Tour, J., Van Zandt, W.L., Husband, C.P., Husband, S.M., Wilson, L.S., Franzon, P.D., Nackashi, D.P.: Nanocell logic gates for molecular computing. *IEEE Transactions on Nanotechnology* 1(2), 100–109 (2002)
25. Turing, A.M.: Intelligent machinery. In: Meltzer, B., Michie, D. (eds.) *Machine Intelligence*, vol. 5, pp. 3–23. Edinburgh University Press, Edinburgh (1969)
26. Weisbuch, G.: *Dynamique des systèmes complexes: Une introduction aux réseaux d'automates*. InterEditions, France (1989)
27. Weisbuch, G.: *Complex Systems Dynamics: An Introduction to Automata Networks*. Lecture Notes, Santa Fe Institute, *Studies in the Sciences of Complexity*, vol. 2. Addison-Wesley, Redwood City (1991)
28. Wittaker, E.T., Robinson, G.: The trapezoidal and parabolic rules. In: *The Calculus of Observations: A Treatise on Numerical Mathematics*, pp. 156–158. Dover, New York (1969)

Collective Evolutionary Dynamics and Spatial Reciprocity under the N-Person Snowdrift Game

Marta D. Santos¹, Francisco C. Santos², and Jorge M. Pacheco^{1,3}

¹ ATP-group, CMAF, Complexo Interdisciplinar, 1649-003 Lisboa, Portugal

² DEI & INESC-ID, Instituto Superior Técnico, TU Lisbon, Lisboa, Portugal

³ Departamento de Matemática e Aplicações, Universidade do Minho,
4710-057 Braga, Portugal

{marta.santos,pacheco}@cii.fc.ul.pt, fcsantos@fct.unl.pt

Abstract. The evolution of cooperation has been gathering increasing attention during the last decades. Most of the times, cooperative behavior involves more than two individuals, and the N-person Prisoner's Dilemma, which is the most studied generalized social dilemma in this context, not always manages to capture those situations that often occur to humans. In such cases, the N-person Snowdrift Game (NSG) often provides an adequate alternative. Here we show, making use of the NSG, how spatial populations affect the average levels of cooperation, when compared with the results obtained under conventional evolutionary game theory, that is, for well-mixed populations.

Keywords: Cooperation, Evolution, Evolutionary Game Theory, Diversity.

1 Introduction

Cooperation is on the basis of some of the major transitions in evolution [1]. Genes cooperate to form cells, which in turn cooperate to form multi-cellular organisms; individuals cooperate to form groups and societies, and human culture is a cooperative process. To understand how cooperative behavior emerges and evolves is therefore a quest which has received growing attention during the last decades, and to which Evolutionary Game Theory (EGT) [2, 3] has been able to provide fundamental insights [2-15]. One-shot, symmetric 2-person games are the traditional approach adopted to investigate the emergence and evolution of cooperation; however, one cannot ignore that many real-life situations are actually associated with collective action based on joint decisions made by groups involving more than 2 individuals. There are many examples, in our everyday life and throughout our history, where instances of N-person games are, or have been, at stake. The effort to protect the Earth's environment, on which every single "player" has to make a choice whether to adopt a more conscious behavior or not; the sharing of common resources among different countries; the participation in open source projects; the payment of taxes and

social welfare; etc..., the examples abound. Furthermore, and very commonly, performing a given task which is beneficial to an entire group requires the cooperation of several individuals of that group, who often share the workload required to perform that task. In this case, the N-person generalization of the so-called Snowdrift Game provides suitable description [16].

In its 2-person version, two individuals are driving on a road which gets blocked by a snowdrift. To proceed with their journey home, the snow must be removed, and this removal may or may not be done: if no one shovels, no one gets home; if the two drivers cooperate and shovel, both get home, each one sharing the workload of shovelling the snow. If only one driver decides to shovel, both get home despite one driver incurring the entire cost of snow shovelling. If we define the benefit of getting home as b and the cost of removing the snow as c , then if both drivers cooperate and shovel, each gets $b - c/2$. If both defect, no one gets anything (or goes home) – 0. If one cooperates and the other defects, the Cooperator (C) gets $b - c$ whereas the Defector (D) gets b . Assuming, as usual, that the benefit is greater than the cost (i.e., $b > c$), we get a payoff ranking characteristic of a chicken, hawk-dove or snowdrift dilemma [2, 17, 18]. The N-person generalization of this dilemma is immediate. In keeping with the previous example, we can imagine that the snowdrift occurs at a cross-road where N drivers meet. Again, all want to go home (getting all the same benefit b), but perhaps not all are willing to shovel. If all shovel, then each gets $b - c/N$. But if only $k > 0$ individuals cooperate, each gets $b - c/k$ while those who defect get home without shoveling (and hence get b).

Our goal on this paper is to understand the impact of the structure of the population on the outcome of cooperation on a simple model of the N-person Snowdrift Game. In the framework of EGT, populations are conventionally modelled as *infinite* and *well-mixed* (each and every individual is equally likely to interact with everyone else). We will explore the consequences of each of these assumptions, and ultimately show how structured populations generally lead to higher levels of cooperation at low costs, inhibiting cooperation at high costs, raising new and exciting questions.

This work is organized as follows: on sections 2 and 3 we explain in further detail the model at study and some background on this topic regarding well-mixed populations, respectively; and on section 4 we present the results obtained for structured populations. Finally on section 5 we discuss the results obtained and highlight some future lines of study.

2 The Model

As referred to above, two types of strategies are considered: Cooperators (C, individuals that are willing to pay a certain cost so that the benefit is obtained) and Defectors (D, who do not contribute at all). The payoffs are as follows [19]:

$$P_C(k) = b - c/k \quad k \in [1, N]$$

$$P_D(k) = \begin{cases} 0 & k = 0 \\ b & k \in [1, N-1] \end{cases} \quad (1)$$

where k is the number of cooperators in the group of N individuals including the one concerned, c is the cost, and b is the benefit obtained by each individual of the group, regardless of her strategy, when the task is performed, with $b > c$ (throughout the manuscript, we take $b = 1$). From this definition one concludes that a single cooperator is able to afford the benefit. This is the simplest case, on which we will be focusing on this work; interesting results also arise when more than one cooperator is necessary to be able to attain the benefit - that is, a higher threshold is introduced [16].

3 Background on Well-Mixed Populations

Well-mixed populations (the so-called mean-field approximation in Physics) constitute the simplest approach possible to this problem, for which analytical results can be obtained. The individual fitness of an individual is the result of averaging over all possible groups of size N ; as a consequence, all cooperators have the same fitness, the same happening with all defectors. Evolution is implemented by means of the replicator equation [3],

$$\dot{x} = x(1-x)(f_C(x) - f_D(x)). \quad (2)$$

where x stands for the fraction of Cs on the population, and $f_C(x)$ and $f_D(x)$ correspond to the average fitness of Cs and Ds for that x , respectively. Strategies' evolution follows the gradient of natural selection determined by the relative fitness difference.

At this point, a distinction has to be made, between *infinite* and *finite* populations. In *infinite* populations the sampling is binomial, and consequently the average fitness of cooperators and defectors in the population is, respectively, given by

$$f_C(x) = \sum_{k=0}^{N-1} \binom{N-1}{k} x^k (1-x)^{N-1-k} P_C(k+1)$$

$$f_D(x) = \sum_{k=0}^{N-1} \binom{N-1}{k} x^k (1-x)^{N-1-k} P_D(k) \quad (3)$$

Solving (2) for the steady state $\dot{x} = 0$ on the N-person Snowdrift Game, one obtains [19]

$$\frac{c}{b}(1-x^*)^N + Nx^*(1-x^*)^{N-1} - \frac{c}{b} = 0, \quad (4)$$

which can be solved numerically for arbitrary N , leading to the results shown in Figure 1. It shows us that smaller groups are more advantageous for cooperation, since the equilibrium abundance of cooperators decreases with increasing group size N and decreasing benefit-cost ratio c/b .

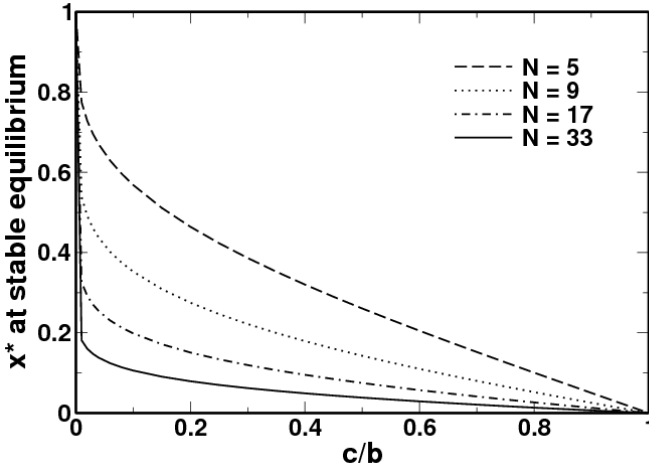


Fig. 1. Fraction of cooperators x^* at the stable equilibrium, as a function of c/b , for different group sizes, in infinite well-mixed populations. One observes that the bigger the group, the smaller the equilibrium fraction of cooperators, for the same ratio c/b .

However, the *infinite* assumption is one that certainly is not compatible with the real world. EGT on a finite population (of size Z) introduces some modifications: the fraction of cooperators is no longer a continuous variable, but varying in steps of $1/Z$; and sampling of individuals is no longer binomial, following now a hypergeometric distribution:

$$P_C(j+1) = \binom{Z-1}{N-1}^{-1} \sum_{j=0}^{N-1} \binom{k-1}{j} \binom{Z-k}{N-j-1} P_C(j-1) \quad (5)$$

$$f_D(k) = \binom{Z-1}{N-1}^{-1} \sum_{j=0}^{N-1} \binom{k}{j} \binom{Z-k-1}{N-j-1} P_D(j)$$

Following the conventional approach, we compute the fraction of cooperators in the population after a transient period of evolutionary dynamics of the population. The fitness of individual i is given by the accumulated payoff resulting from all the games in which she participates. Strategy evolution is implemented via the pairwise

comparison rule [20-22]: in each time-step, an individual (A) is selected at random from the population to revise her strategy, comparing her accumulated payoff with one of her neighbors (B) also randomly chosen. Individual A adopts the strategy of B with a probability given by the ubiquitous Fermi expression from statistical physics

$$p = [1 + e^{-\beta(f_B - f_A)}]^{-1} \quad (6)$$

where f_A and f_B are the fitness of individuals A and B respectively, and β , which in physics corresponds to an inverse temperature, denotes here the intensity of selection. For $\beta \ll 1$ selection is weak, and in the limit on infinite populations (that is, $Z \rightarrow \infty$), one recovers the replicator equation. Increasing β increases the intensity of selection, reaching pure imitation dynamics whenever $\beta \gg 1$.

When one performs evolutions on large yet finite well-mixed populations, the agreement is, as expected, almost perfect. Only small discrepancies can be observed for the higher and lower values of c/b , due to the finiteness of the population. This effect disappears once sufficiently big populations are considered (on this type of processes, for $Z = 10^4$ convergence to the infinite population case is excellent).

One can also define the probability to increase and to decrease the number k of cooperators in the population by one, at each time-step:

$$T^\pm(k) = \frac{k}{Z} \frac{Z-k}{Z} \frac{1}{1 + e^{\mp\beta(f_C(k) - f_D(k))}}. \quad (7)$$

The first term relates to the probability of selecting a cooperator, the second one to the probability of selecting a defector, and the last to the take-over probability, taking into account the average payoffs of cooperators and defectors for that specific k . For arbitrary β , the quantity corresponding to the right hand side of the replicator equation, which specifies the gradient of selection, is given by

$$g(k) \equiv T^+(k) - T^-(k) = \frac{k}{Z} \frac{Z-k}{Z} \tanh\left\{\frac{\beta}{2}[f_C(k) - f_D(k)]\right\}. \quad (8)$$

The right-hand side of $g(k)$ is similar to the replicator equation, only that the pairwise comparison leads to the appearance of the hyperbolic tangent of the fitness difference, instead of the fitness difference. This has implications in the characteristic evolutionary times, which now depend on β , but not in what concerns the roots of $g(k)$. Also, adoption of this specific social learning hypothesis, combined with the finite population size, means that internal equilibria are no longer possible: the evolutionary dynamics will only stop whenever the system reaches one of the two

absorbing states, full cooperation or full defection. Hence, the sign of $g(k)$, which indicates the direction of selection, is important in that it may strongly influence the evolutionary time required to reach any of the absorbing states.

For well-mixed populations, calculating $g(k)$ is generally straightforward, as every C will have the same fitness, the same happening with every D . In structured populations this property no longer holds, and the task of computing $g(k)$ becomes considerably harder. Here we shall compute $T^+(k)$ and $T(k)$ at a mean-field level, that is, we compute the average frequency of transitions increasing (and decreasing) the number of cooperators for each random configuration with k cooperators.

4 NSG on Structured Populations

Real world populations have yet additional degrees of complexity. For instance, individuals do not potentially interact with everyone else in the population, but only with a limited number of neighbors. Graph theory constitutes a natural and very convenient framework to describe population structure: individuals are placed on the vertices of a graph, whose edges define the existence of interaction between them.

In network structured populations, each individual has z social ties, which means her payoff is determined by the $z + 1$ games she participates in: the one centered on herself, and also those centered on her neighbors [14], as depicted on Figure 2.

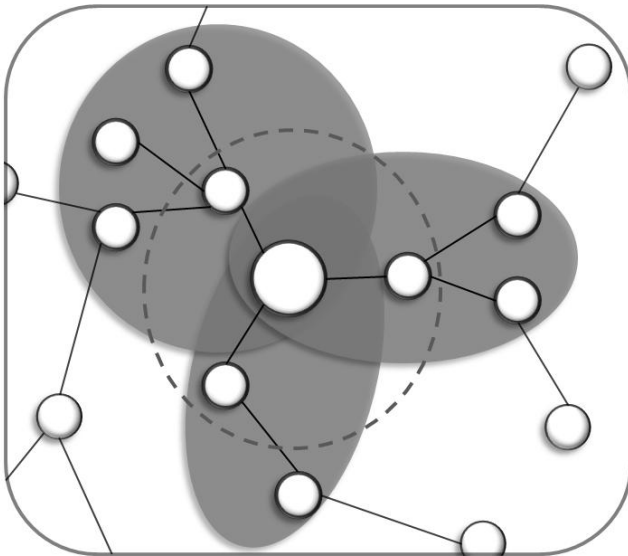


Fig. 2. In this example, the focal individual (largest sphere) has $z = 3$ social ties, and therefore her payoff is determined by the participation on the game centered on herself (grey dashed line) and the games centered on her neighbors (grey shaded shapes), in a total of 4 games

On this work we will focus on homogeneous networks, and particularly on regular ones. On the other hand, a regular network can be understood as some form of spatial organization, leading to a process known as “spatial reciprocity”.

Figure 3 shows the gradient of selection defined in equation (8) for some values of the ratio c/b , as well as the comparison between the coexistence points obtained in this way and those corresponding to infinite, well-mixed populations.

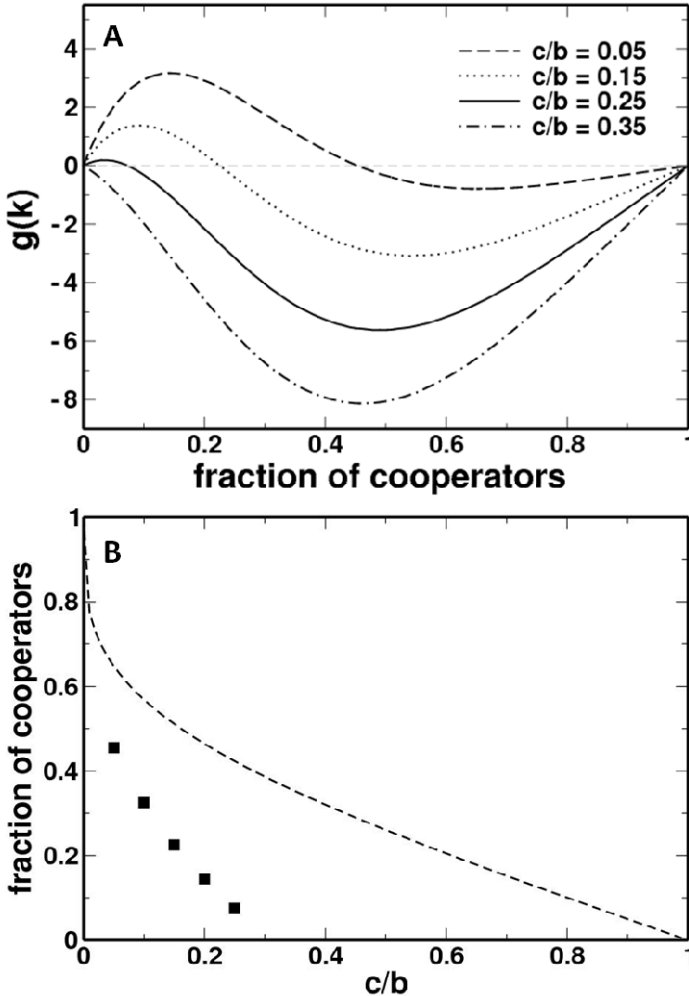


Fig. 3. A. Gradients of selection $g(k)$ for NSG on a regular network for some values of the ratio c/b . B. Comparison of the coexistence points (full circles) obtained in A (depicted by full squares) with those associated with well-mixed populations (dashed line). **Parameters:** $Z=1000$, $z=4$, $b=1.0$, $\beta=1.0$

Unlike what happens in the infinite population case, now the mean-field gradients show the occurrence of internal coexistence-like points up to $c/b = 0.3$. One observes that these points differ significantly from the analytical results previously obtained, suggesting that the evolutionary dynamics of the NSG under spatial reciprocity differs significantly from that in infinite, well-mixed populations. For $c/b > 0.3$, the gradient is always negative, showing that, at a mean-field level, cooperators do not stand a chance.

The fact that the calculations of the gradients are mean-field in nature points to the qualitative nature of these results. Therefore, one might expect that, qualitatively, cooperators will stand a chance for $c/b < 0.3$, whereas beyond this regime defectors will win the evolutionary race.

Figure 4 shows the evolution of cooperation under spatial reciprocity for the NSG.

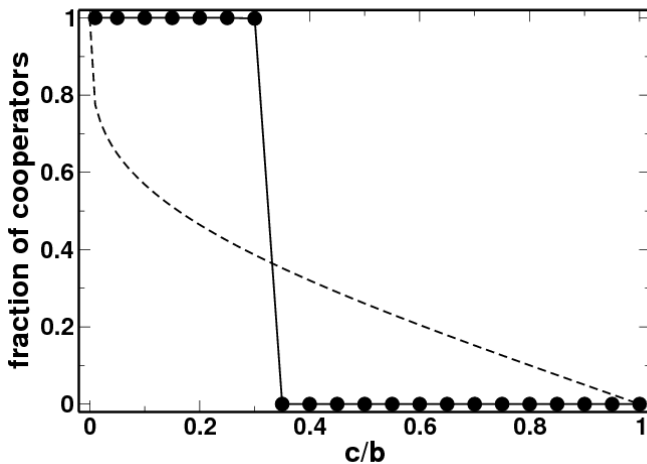


Fig. 4. Average level of cooperation, as a function of c/b , under spatial reciprocity for NSG, for group size $N = 5$ (depicted by solid circles), compared with the correspondent result for well-mixed populations. One observes that cooperation is favored for low values of the cost, being inhibited for large values. **Parameters:** $Z=1000$, $z=4$, $b=1.0$, $\beta = 1.0$

Similar to what has been obtained for the 2-person SG [18] cooperation is favoured for low values of the cost while it is inhibited for large values. The solid circles were obtained by averaging over 2000 generations, after a transient period of 10^5 generations, and each circle corresponds to an average over 10^3 runs. These results, in turn, are qualitatively similar to those one obtains by plotting the fraction of times the population converges to full cooperation for a given value of the cost-to-benefit ratio (not shown).

In order to understand better the origin of this result, we focus on a particular value of $c/b < 0.30$, and follow the time evolution of the fraction of cooperators, starting from two different initial conditions – above and below the coexistence point $x_{coexistence}$

obtained for the same value of c/b in the infinite, well-mixed case (dashed line in Figure 3b).

In Figure 5 we start from a population of 1% and 70% of cooperators respectively, randomly distributed in the network, and allow the system to evolve 10^5 time-steps. In all our simulations we adopted asynchronous update in populations of size 10^3 and connectivity $z = 4$; each dot corresponds to an average over 1000 runs (as stated, we set $b = 1$ in all simulations, such that the only game parameter is the ratio c/b).

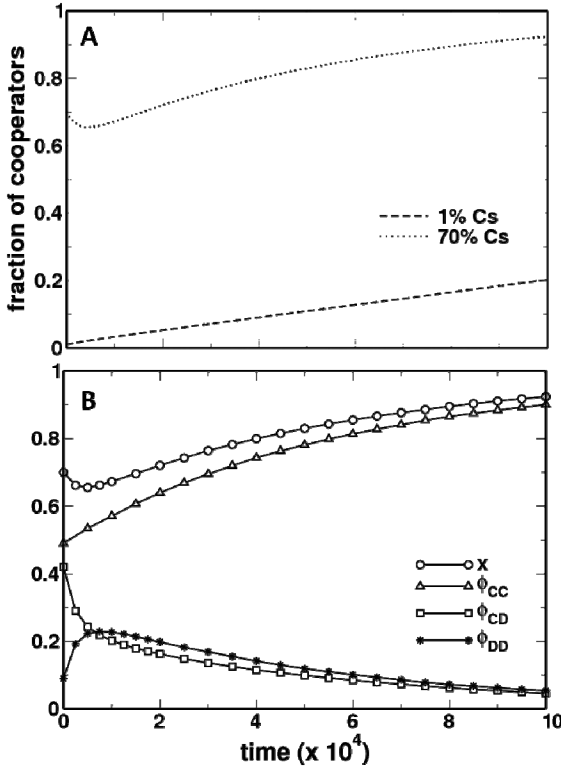


Fig. 5. A. Fraction of cooperators (x) as a function of time steps, starting above $x_{coexistence}$ (70% of cooperators) and below it (1% of cooperators). B. x, ϕ_{CC}, ϕ_{DD} and ϕ_{CD} as a function of time, starting from 70% of cooperators, for regular networks. **Parameters:** $Z=1000, z=4, b=1.0, c=0.15, \beta = 1.0$

Figure 5A shows that, although the final outcome is the same (the population is heading to the absorbing state of full cooperation), the dynamic behavior observed throughout evolution is quite different. While for $x < x_{coexistence}$ the fraction of cooperators increases right from the start, for $x > x_{coexistence}$ it starts by first decreasing, increasing only after a considerable amount of generations. What is the mechanism responsible for the decrease of the number of cooperators? To answer this question, let us define the fraction of links between individuals playing strategies i and j as:

$$\phi_{ij} = \frac{\sum_{i=1}^Z \#ij \text{ links}}{Zz/2} \quad (9)$$

where $Zz/2$ is the total number of links of the network and i and j is either C or D.

Figure 5B shows the average time evolution of these quantities when starting with a fraction of cooperators $x > x_{coexistence}$, along with the above defined quantities. Analysis of these results shows that isolated cooperators are the ones specifically being eliminated, i.e., cooperators and defectors on the population organize themselves increasingly more in an *assorted* manner – individuals that adopt a certain strategy are not isolated but organized in such a way that have at least one neighbor following the same strategy. This is corroborated by the several curves shown: ϕ_{CC} increases slightly, accompanied by a sudden decrease of ϕ_{CD} , representing the self-organization of cooperators and defectors; the maximum “saturation value” reached by ϕ_{DD} corresponds to the moment in which cooperators are less represented in the population. For these values of c/b , regular structures can, therefore, be favorable for cooperation under the NSG, potentiating the self-organization of the population towards the full cooperation absorbing state.

As $c/b > 0.3$, cooperators no longer resist the increased capacity of assortment exhibited by defectors, and indeed in the majority of cases the population evolves into full defection.

5 Discussion

The present study puts in evidence the impact of regular structures on the evolution of cooperation making use of a model of collective cooperation based on the N-person SG. We have shown that, below a critical cost-to-benefit ratio, regular networks facilitate cooperation, by enhancing the self-organization of the population regarding the distribution of strategies. This capacity of self-organization of cooperators breaks-down as one surpasses this critical cost-to-benefit ratio, which we have shown can be qualitatively associated with the mean-field value of the gradient of selection numerically computed for the structured population.

Acknowledgments. Financial support from FNRS Belgium (FCS) and FCT Portugal (MDS, JMP) is gratefully acknowledged. The authors would also like to thank Sven Van Segbroek and Flávio L. Pinheiro for the useful discussions.

References

1. Maynard-Smith, J., Szathmáry, E.: The Major Transitions in Evolution. Freeman, Oxford (1995)
2. Maynard-Smith, J.: Evolution and the Theory of Games. Cambridge University Press, Cambridge (1982)

3. Hofbauer, J., Sigmund, K.: *Evolutionary Games and Population Dynamics*. Cambridge Univ. Press, Cambridge (1998)
4. Axelrod, R., Hamilton, W.D.: The evolution of cooperation. *Science* 211(4489), 1390–1396 (1981)
5. Boyd, R., Richerson, P.J.: *Culture and the Evolutionary Process*. University of Chicago Press (1985)
6. Skyrms, B.: The Stag Hunt. *Proceedings and Addresses of the American Philosophical Association* 75(2), 31–41 (2001)
7. Macy, M.W., Flache, A.: Learning dynamics in social dilemmas. *Proc. Natl. Acad. Sci. U S A* 99(suppl. 3), 7229–7236 (2002)
8. Hammerstein, P.: *Genetic and Cultural Evolution of Cooperation*. MIT Press, Cambridge (2003)
9. Nowak, M.A., et al.: Emergence of cooperation and evolutionary stability in finite populations. *Nature* 428(6983), 646–650 (2004)
10. Skyrms, B.: *The Stag Hunt and the Evolution of Social Structure*. Cambridge University Press (2004)
11. Santos, F.C., Pacheco, J.M.: Scale-free networks provide a unifying framework for the emergence of cooperation. *Phys. Rev. Lett.* 95(9), 098104 (2005)
12. Ohtsuki, H., et al.: A simple rule for the evolution of cooperation on graphs and social networks. *Nature* 441(7092), 502–505 (2006)
13. Santos, F.C., Pacheco, J.M., Lenaerts, T.: Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proc. Natl. Acad. Sci. U S A* 103(9), 3490–3494 (2006)
14. Santos, F.C., Santos, M.D., Pacheco, J.M.: Social diversity promotes the emergence of cooperation in public goods games. *Nature* 454(7201), 213–216 (2008)
15. Pacheco, J.M., Pinheiro, F.L., Santos, F.C.: Population structure induces a symmetry breaking favoring the emergence of cooperation. *PLoS Comput. Biol.* 5(12), e1000596 (2009)
16. Souza, M.O., Pacheco, J.M., Santos, F.C.: Evolution of cooperation under N-person snowdrift games. *Journal of Theoretical Biology* 260(4), 581–588 (2009)
17. Sugden, R.: *The Economics of Rights, Co-operation and Welfare*, p. 191. Blackwell, Oxford (1986)
18. Hauert, C., Doebeli, M.: Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature* 428(6983), 643–646 (2004)
19. Zheng, D.F., et al.: Cooperation behavior in a model of evolutionary snowdrift games with N-person interactions. *Europhysics Letters* 80, 18002–18006 (2007)
20. Traulsen, A., Nowak, M.A., Pacheco, J.M.: Stochastic dynamics of invasion and fixation. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 74(1 Pt 1), 011909 (2006)
21. Traulsen, A., Nowak, M.A., Pacheco, J.M.: Stochastic payoff evaluation increases the temperature of selection. *J. Theor. Biol.* 244(2), 349–356 (2007)
22. Traulsen, A., Pacheco, J.M., Nowak, M.A.: Pairwise comparison and selection temperature in evolutionary game dynamics. *J. Theor. Biol.* 246(3), 522–529 (2007)
23. Ohtsuki, H., Nowak, M.A., Pacheco, J.M.: Breaking the symmetry between interaction and replacement in evolutionary dynamics on graphs. *Phys. Rev. Lett.* 98(10), 108106 (2007)

An Evolutionary Game Theoretic Framework for Adaptive, Cooperative and Stable Network Applications

Chonho Lee¹, Junichi Suzuki¹, and Athanasios V. Vasilakos²

¹ Department of Computer Science
University of Massachusetts, Boston, USA
{chonho,jxs}@cs.umb.edu

² Department of Computer and Telecommunication Engineering
University of Western Macedonia, Greece
vasilako@ath.forthnet.gr

Abstract. This paper investigates a bio-inspired framework, iNet-EGT/C, to build adaptive, cooperative and stable network applications. In this framework, each application is designed as a decentralized set of agents, each of which provides a functional service and possesses biological behaviors such as migration, replication and death. iNet-EGT/C implements an adaptive behavior selection mechanism for agents. Its selection process is modeled as a series of evolutionary games among behaviors. iNet-EGT/C allows agents to seek to operate at evolutionarily stable equilibria and adapt to dynamic networks by invoking evolutionarily stable behaviors. It is theoretically proved that each behavior selection process retains stability (i.e., reachability to at least one evolutionarily stable equilibrium). iNet-EGT/C leverages the notion of coalitions for agents to select behaviors as coalitional decisions in a cooperative manner rather than individual decisions in a selfish manner. This cooperative behavior selection accelerates agents' adaptation speed by up to 78%.

Keywords: Evolutionary game theory, Adaptive and cooperative network applications, Stability in adaptive networking.

1 Introduction

Large-scale network applications such as data center applications and cloud computing applications are required to autonomously adapt to dynamic changes in network conditions such as workload and resource availability [1]. To address this requirement, this paper investigates a bio-inspired framework for autonomous adaptive network applications. In the framework, each application is designed as a decentralized group of software agents. This is analogous to a bee colony (an application) consisting of multiple bees (agents). Each agent implements a functional service (e.g., web service) and follows biological behaviors such as migration, replication and death. This paper focuses on an adaptive behavior selection mechanism for agents, called iNet-EGT/C (iNet-EGT, coalitional edition). It is

designed after immunological *antigen-antibody reaction*, which produces antibodies specific to antigens (e.g., viruses) for eliminating them. iNet-EGT/C models a set of network conditions (e.g., workload and resource availability) as an antigen and models an agent behavior as an antibody. Each agent contains iNet-EGT/C as its behavior selection mechanism (or its immune system). iNet-EGT/C allows each agent to autonomously sense its surrounding network conditions (an antigen) and select a behavior (an antibody) suitable for the sensed conditions. For example, agents may invoke the replication behavior at the network hosts that accept a large number of user requests for their services. This leads to the adaptation of agent availability. As a result, agents can improve their throughput. Also, agents may invoke the migration behavior to move toward the network hosts that receive a large number of user requests for their services. This leads to the adaptation of agent locations; agents can improve their response time.

In iNet-EGT/C, antigen-antibody reaction (i.e., behavior selection) process is modeled with evolutionary game theory. Each agent contains a set (or population) of behaviors. In the population, randomly-selected two behaviors play a game. The game distinguishes a winning behavior and a losing behavior based on their *payoffs*, which indicate the likelihood for the behaviors to adapt an agent to the current network conditions. The winner replicates itself and increases its share in the population. The loser disappears in the population. This way, the population state (i.e., behavior distribution in the population) changes as a series of games are repeatedly performed. iNet-EGT/C theoretically proves that the population state converges to an *evolutionarily stable equilibrium*. It is the state that, regardless of the initial population state, the population always converges to. In that state, no other behaviors except a dominant behavior, called an *evolutionarily stable behavior*, can dominate the population. Thanks to this property, iNet-EGT/C allows each agent to seek to operate at equilibria by invoking evolutionarily stable behaviors as rational and adaptive decisions.

iNet-EGT/C leverages the notion of *coalitions* to compute payoffs. A coalition is a group of agents that reside on neighboring hosts. Each agent computes the payoffs for its behaviors based on (1) its surrounding network conditions and (2) the behaviors that agents in its coalition intend to invoke. This way, agents seek evolutionarily stable behaviors as coalitional decisions in a cooperative manner rather than individual decisions in a selfish manner. This coalitional payoff computation is designed to accelerate the adaptation speed of agents.

This paper describes the design of iNet-EGT/C and evaluates it through theoretical and simulation studies. Both studies verify that iNet-EGT/C allows agents to seek to operate at evolutionarily stable equilibria and adapt to dynamic networks. The notion of coalitions accelerates agents' adaptation speed by 78%.

2 Related Work

iNet-EGT/C extends its predecessors: iNet [2] and iNet-EGT [3]. It shares the same goal with them; bio-inspired adaptive behavior selection. However, they are different in their approaches to the goal. iNet takes a stochastic approach with

a genetic algorithm; it does not guarantee stability (i.e., reachability to at least one equilibrium) in behavior selection. iNet-EGT takes an evolutionary game theoretic approach; however, it does not consider the notion of agent coalitions.

Game theoretic approaches have been studied for several aspects in networks; for example, job allocation [4], security [5-7] and routing [8,9]. They seek rational networking strategies in static networks, but do not consider adaptation in dynamic networks. [9] does not guarantee stability in its strategic decision making. None of [4-9] considers the notion of coalitions in games.

[10,11] leverage evolutionary game theory for adaptive routing in dynamic networks. Unlike [10,11], iNet-EGT/C performs the mutation operation to better adapt to future changes in network conditions. Moreover, iNet-EGT/C considers the notion of agent coalitions, which is beyond the scope of [10,11].

[12-16] study behavior selection mechanisms for agent-based systems. [12,13] propose rule-based mechanisms, which are similar to iNet-EGT/C in that they implement deterministic behavior selection. However, unlike [12,13], iNet-EGT/C guarantees stability in behavior selection. Moreover, in general, it maintains lower time complexity in behavior selection ($O(k)$ where k denotes the number of games) than rule-based mechanisms ($O(M \log M)$ where M denotes the number of behavior types). [14-16] propose non-deterministic behavior selection with stochastic algorithms. Due to their stochastic processes, they fail to retain stability; they often search inconsistent solutions under the same problem setting (or the same set of network conditions) in different runs/trials. In contrast, iNet-EGT/C considers determinism in behavior selection to retain stability.

3 Preliminaries: Evolutionary Game Theory

In a conventional game, the objective of a player is to choose a strategy that maximizes its payoff. In contrast, evolutionary games are played repeatedly by players randomly drawn from a population [17]. This section overviews key elements in evolutionary games: evolutionarily stable strategies and replicator dynamics.

3.1 Evolutionarily Stable Strategies (ESS)

Suppose all players in the initial population are programmed to play a certain (incumbent) strategy k . Then, let a small population share of players, $x \in (0, 1)$, mutate and play a different (mutant) strategy ℓ . When a player is drawn for a game, the probabilities that its opponent plays k and ℓ are $1 - x$ and x , respectively. Thus, the expected payoffs for the player to play k and ℓ are $U(k, x\ell + (1 - x)k)$ and $U(\ell, x\ell + (1 - x)k)$, respectively.

Definition 1. A strategy k is called evolutionarily stable if, for every strategy $\ell \neq k$, a certain $\bar{x} \in (0, 1)$ exists, such that the inequality

$$U(k, x\ell + (1 - x)k) > U(\ell, x\ell + (1 - x)k) \quad (1)$$

holds for all $x \in (0, \bar{x})$.

If the payoff function is linear, Equation 1 derives

$$(1-x)U(k,k) + xU(k,\ell) > (1-x)U(\ell,k) + xU(\ell,\ell) \quad (2)$$

If x is close to zero, Equation 2 yields either

$$U(k,k) > U(\ell,k), \text{ or } U(k,k) = U(\ell,k) \text{ and } U(k,\ell) > U(\ell,\ell) \quad (3)$$

This indicates that a player associated with the strategy k gains a higher payoff than the ones associated with the other strategies. Thus, no players can benefit by changing their strategies from k to the others. This means an ESS is a solution on a Nash equilibrium. An ESS is a strategy that cannot be invaded by any alternative (mutant) strategies that have small population shares.

3.2 Replicator Dynamics

The replicator dynamics describes how population shares associated with different strategies evolve over time [18]. Let $\lambda_k(t) \geq 0$ be the number of players that plays the strategy $k \in K$, where K is the set of available strategies. The total population of players is given by $\lambda(t) = \sum_{k=1}^{|K|} \lambda_k(t)$. Let $x_k(t) = \lambda_k(t)/\lambda(t)$ be the population share of players that play k at time t . The population state is defined by $\mathbf{x}(t) = [x_1(t), \dots, x_k(t), \dots, x_K(t)]$. Given \mathbf{x} , the expected payoff of playing k is denoted by $U(k, \mathbf{x})$. The population's average payoff, which is same as the payoff of a player drawn randomly from the population, is denoted by $U(\mathbf{x}, \mathbf{x}) = \sum_{k=1}^{|K|} x_k \cdot U(k, \mathbf{x})$. In the replicator dynamics, the dynamics of the population share x_k is described as follows. \dot{x}_k is the time derivative of x_k .

$$\dot{x}_k = x_k \cdot [U(k, \mathbf{x}) - U(\mathbf{x}, \mathbf{x})] \quad (4)$$

This equation states players increase (or decrease) their population shares when their payoffs are higher (or lower) than the population's average payoff.

Theorem 1. *If a strategy k is strictly dominated, then $x_k(t)_{t \rightarrow \infty} \rightarrow 0$.*

A strategy is said to be *strictly dominant* if its payoff is strictly higher than any opponent strategies. As its population share grows, it dominates the population over time. Conversely, a strategy is said to be *strictly dominated* if its payoff is lower than that of a strictly dominant strategy. Thus, strictly dominated strategies disappear in the population over time.

There is a close connection between Nash equilibria and the steady states of the replicator dynamics, in which the population shares do not change over time. Since no players change their strategies on Nash equilibria, every Nash equilibrium is a steady state of the replicator dynamics. As described in Section 3.1, an ESS is a solution on a Nash equilibrium. Thus, an ESS is a solution at a steady state of the replicator dynamics. In other words, an ESS is the strictly dominant strategy in the population on a steady state.

In iNet-EGT/C, an agent maintains a population of its behaviors. In a single behavior selection process, behaviors are randomly drawn from the population

to play games repeatedly until the population reaches a steady state. Then, iNet-EGT/C allows each agent to identify a strictly dominant behavior in the population and invoke it as an evolutionarily stable behavior (or ESS).

4 iNet-EGT/C

The immune system is an adaptive defense mechanism that regulates the body against dynamic environmental changes such as antigen invasions. Through a number of interactions among antibodies, the immune system evokes *antigen-antibody reaction* to produce antibodies specific to detected antigens. In each interaction, an antibody stimulates or suppresses another one according to its *affinity* to an antigen. A stimulated antibody replicates itself and increases its population. Conversely, a suppressed one dies and decreases its population. This way, the population of specific antibodies rapidly increases following the recognition of an antigen and decreases again after eliminating the antigen. Adaptive immune response is an emergent product of interactions among antibodies.

iNet-EGT/C is designed after antigen-antibody reaction. Each agent contains iNet-EGT/C as its own behavior selection mechanism (or as its own immune system). An antigen is modeled as a set of network conditions: $C = \{c_1, c_2, \dots, c_L\}$ where L denotes the number of network conditions that each agent senses on a host. This paper considers three network conditions ($L = 3$):

- *Queue length*: The number of user requests in a request queue, which each host operates to store incoming user requests until they are processed by the agents running on the same host.
- *Workload change rate*: The rate of workload change per a unit time. Workload is computed as the number of incoming user requests per minute given to the agents running on the same host.
- *Resource utilization*: Memory consumption, in percentage, by the agents running on the same host.

An antibody is modeled as an agent behavior. $B = \{b_1, b_2, \dots, b_M\}$ denotes a set of available behavior types. This paper considers four types ($M = 4$):

- *Migration*: Agents may migrate from one platform to another.
- *Replication*: Agents may make a copy of themselves. A replicated (child) agent is placed on the host that its parent agent resides on.
- *Death*: Agents may die and disappear in the network. When an agent dies, its underlying host releases the resources (e.g. memory space) it consumes.
- *Do-nothing*: Agents may choose to do nothing.

Each agent maintains a population (P) of behaviors, each of which is of a certain behavior type. The population's size is given by $N = \sum_{b \in B} n_b$ where n_b denotes the number of behaviors of the behavior type $b \in B$.

4.1 Evolutionary Games in iNet-EGT/C

An interaction (stimulation or suppression) between antibodies is modeled as an evolutionary game between behaviors. Listing 1 shows how iNet-EGT/C allows each agent to select/invoke a specific behavior under a given set of network conditions through a series of evolutionary games.

The population P is initialized with `initializePopulation()` (Line 2). Initially, all behavior types have the equal population share. ($n_{b \in B}$ is equal for every b .) `randomlySelect()` draw two behaviors randomly from the population (Line 6), and `performGame()` distinguishes them to a winning one and a losing one according to their payoffs, which indicate the likelihood for the behaviors to adapt an agent to the current network conditions (Line 8). The notion of payoffs is modeled after the notion of affinity in the immune system. The loser behavior is suppressed and disappears in the population. The winner behavior is stimulated and replicated to increase its population share (Line 9). It is also mutated at the probability of p_m (Lines 10 and 11).

A behavior whose population share is the largest is called a *current major behavior*. An agent invokes the current major behavior when its population share (x_b) exceeds a threshold (t_s) (Lines 15 and 16). Similar to the immune system, the behavior selection in iNet-EGT/C is designed as an emergent product of games (interactions) among behaviors (antibodies).

```

1  function selectBehavior()
2       $\mathcal{P} \leftarrow \text{initializePopulation}(N)$ 
3      while true do
4           $\mathcal{W} \leftarrow \emptyset$ 
5          for  $i \leftarrow 1$  to  $|\mathcal{P}|/2$  do
6               $\{\text{behavior}_1, \text{behavior}_2\} \leftarrow \text{randomlySelect}(\mathcal{P})$ 
7               $\mathcal{P} \leftarrow \mathcal{P} \setminus \{\text{behavior}_1, \text{behavior}_2\}$ 
8               $\text{winner} \leftarrow \text{performGame}(\text{behavior}_1, \text{behavior}_2)$ 
9               $\text{replica} \leftarrow \text{replicate}(\text{winner})$ 
10             if  $\text{random}() \leq p_m$  then
11                  $\text{replica} \leftarrow \text{mutate}(\text{replica})$ 
12              $\mathcal{W} \leftarrow \mathcal{W} \cup \text{winner} \cup \text{replica}$ 
13         end for
14          $\mathcal{P} \leftarrow \mathcal{W}$ 
15         if  $\exists b$  where  $b \in \mathcal{P}$  and  $x_b > t_s$  then
16             return  $b$ 
17         end if
18     end while
19 end function
20
21 function performGame( $\text{behavior}_1, \text{behavior}_2$ )
22      $\mathcal{C} \leftarrow \text{getNetworkConditions}()$ 
23      $\mathcal{O} \leftarrow \text{getCurrentMajorBehaviors}()$ 
24      $p_1 \leftarrow |\{\text{behavior}_1, \mathcal{R}\}| - \text{rank}(\text{behavior}_1, \mathcal{O}, \mathcal{C})$ 
25      $p_2 \leftarrow |\{\text{behavior}_1, \mathcal{R}\}| - \text{rank}(\text{behavior}_2, \mathcal{O}, \mathcal{C})$ 
26     if  $p_1 > p_2$  then return  $\text{behavior}_1$ 
27     else if  $p_1 < p_2$  then return  $\text{behavior}_2$ 
28     else return  $\text{randomlySelect}(\{\text{behavior}_1, \text{behavior}_2\})$ 
29 end function

```

Listing 1. Pseudocode of Behavior Selection

4.2 Coalitional Payoff Functions

In order to compute the payoffs of an agent’s behaviors, iNet-EGT/C considers the agent’s coalition, which consists of the agents running on the local host and direct neighbor hosts. Equation 5 shows the coalitional payoff function $F_i(b)$ for the behavior type b of agent i under a set of network conditions C .

$$F_i(b) = |\{b, O\}| - \text{rank}(b, O, C) \tag{5}$$

$O = \{o_1, o_2, \dots, o_q\}$ denotes a set of current major behaviors of the other agents in the same coalition. (q indicates the number of those agents.) See also Line 22 to 25 in Listing 1.

The function $\text{rank}()$ in Equation 5 compares b and O (i.e., $q + 1$ behaviors in total) with respect to *domination factors* and yields b ’s *domination rank*. The domination ranking is a ranking scheme that considers the Pareto optimality among multiple factors (or objectives) [19]. A behavior $b \in B$ is said to dominate a behavior $b' \in B$ if b ’s factor values are better than, or equal to, b' ’s in all domination factors, and b ’s factor values are better than b' ’s in at least one domination factors. This paper considers three domination factors:

- *Queue length*
- *Resource utilization*
- *Load balancing*: The variance of queue lengths in the local and neighboring hosts

A behavior b ’s factor values are computed as the network conditions in the case where agent i invokes b and the other agents in its coalition invokes O .

Figure 1 shows an example that illustrates how to use the proposed coalitional payoff functions. In this example, four agents (a , b , c and d) runs on a host. (For simplicity, this example ignores neighbor nodes.) Agent b ’s current major behavior is migration (M). c ’s and d ’s are death (D) and replication (R), respectively. Thus, agent a ’s payoff functions ($F_a(R)$, $F_a(M)$, $F_a(N)$ and $F_a(D)$) are formulated for its four behaviors, as shown in Figure 1.

Figure 2 depicts an example domination ranking among agent a ’s four behaviors. In a three dimensional space whose axes represent domination factors, each behavior is plotted based on its factor values. (In this space, the smaller factor values, the better.) For example, R ’s factor values are computed as the network conditions in the case where agent a invokes the replication behavior and the other three agents invoke their current major behaviors.

R dominates all the other behaviors; it is given a rank value of 1. D dominates N , but does not dominate M . However, D is superior to M in two factors, and M is superior in one factor. Thus, D and M are given the rank values of 2 and 3, respectively. N is given a rank value of 4. Given rank values, $F_a(R) = 4 - 1 = 3$, $F_a(D) = 4 - 2 = 2$, $F_a(M) = 4 - 3 = 1$, and $F_a(D) = 4 - 4 = 0$.

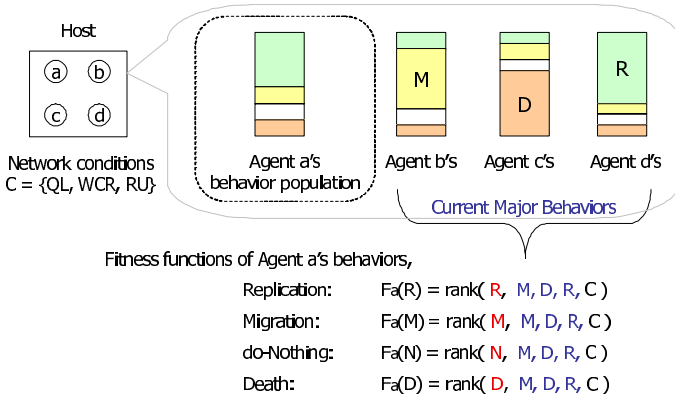


Fig. 1. Example Payoff Functions

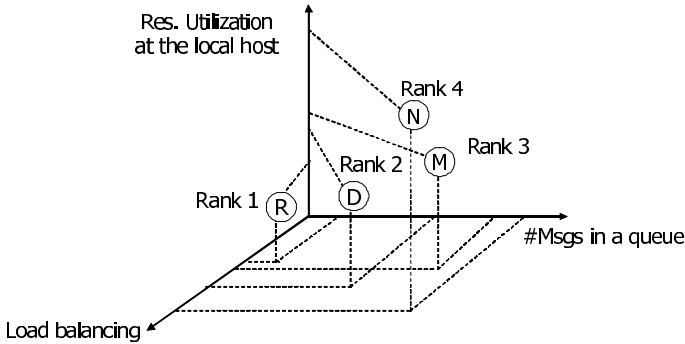


Fig. 2. An Example Domination Ranking

5 Stability Analysis

This section theoretically analyzes the stability of cooperative behavior selection in iNet-EGT/C. More specifically, this theoretical analysis is intended to prove that the population states of all agents in the same coalition converge to evolutionarily stable equilibria. The proof consists of three steps: (1) designing a set of differential equations that describe the dynamics of the population state, (2) proving a cooperative behavior selection process has equilibria, and (3) proving the equilibria are asymptotically (or evolutionarily) stable. The proof uses the following terminology and variables.

- $X^{(i)}(t) = \{x_1^{(i)}(t), x_2^{(i)}(t), \dots, x_M^{(i)}(t)\}$ denotes the population state of agent i at time t , where $x_b = \frac{n_b}{N}$ is the population share of $b \in B$ ($\sum_b x_b = 1$).
- $F_b^{(i)}$ denotes the coalitional payoff of $b \in B$ in agent i .

– $p_k^b = x_b \cdot \phi(F_b - F_k)$ denotes the probability that a behavior of $b \in B$ replicates itself by winning a game against a behavior of $k \in B$. $\phi(F_b - F_k)$ is the conditional probability that the fitness value of b is higher than that of k .

The dynamics of b 's population share is described as follows.

$$\begin{aligned} \dot{x}_b &= \sum_{k \in B, k \neq b} \{x_k \cdot p_k^b - x_b \cdot p_b^k\} = x_b \sum_{k \in B, k \neq b} x_k \{\phi(F_b - F_k) - \phi(F_k - F_b)\} \\ &= x_b \sum_{k \in B, k \neq b} x_k \cdot c_{bk} \quad \text{where } c_{bk} = \phi(F_b - F_k) - \phi(F_k - F_b) \end{aligned} \tag{6}$$

Note that, if k is strictly dominated, $x_k(t)_{t \rightarrow \infty} \rightarrow 0$. (See Theorem [II](#))

Theorem 2. *The population state of an agent converges to an equilibrium.*

Proof. It is true that, given the design of fitness functions (Figure [II](#)), different behavior types have different fitness values under the same set of network conditions. In other words, given a particular set of network conditions, a behavior type becomes strictly dominant. Assume that $F_1 > F_2 > \dots > F_M$, and by Theorem 1, the population state converges to an equilibrium: $X(t)_{t \rightarrow \infty} = \{x_1(t), x_2(t), \dots, x_M(t)\}_{t \rightarrow \infty} \rightarrow \{1, 0, \dots, 0\}$.

Theorem 3. *The equilibrium found in Theorem [2](#) is asymptotically stable.*

Proof. At an equilibrium where $X = \{1, 0, \dots, 0\}$, Equation [6](#) can be downsized by substituting $x_1 = 1 - x_2 - \dots - x_M$.

$$\dot{z}_b = z_b [c_{b1}(1 - z_b) + \sum_{i=2, i \neq b}^M z_i \cdot c_{bi}] \quad \text{where } b = 2, \dots, M \tag{7}$$

$Z(t) = \{z_2(t), z_3(t), \dots, z_M(t)\}$ denotes the downsized population state. Given Theorem [II](#), $Z(t)$ converges to an equilibrium: $Z(t)_{t \rightarrow \infty} = Z_{eq} = \{0, 0, \dots, 0\}$.

If all Eigenvalues of the Jaccobian matrix of $Z(t)$ has negative Real parts, Z_{eq} is asymptotically stable. The Jaccobian matrix J 's elements are:

$$J_{bk} = \left[\frac{\partial \dot{z}_b}{\partial z_k} \right]_{|Z=Z_{eq}} = \left[\frac{\partial z_b [c_{b1}(1 - z_b) + \sum_{i=2, i \neq b}^M z_i \cdot c_{bi}]}{\partial z_k} \right]_{|Z=Z_{eq}} \tag{8}$$

where $b, k = 2, \dots, M$

Therefore, J is given as follows, where $c_{21}, c_{31}, \dots, c_{M1}$ are J 's Eigenvalues.

$$J = \begin{bmatrix} c_{21} & 0 & \dots & 0 \\ 0 & c_{31} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_{M1} \end{bmatrix} \tag{9}$$

$c_{b1} = -\phi(F_1 - F_b) < 0$ for every b ; therefore, Z_{eq} is asymptotically stable.

6 Simulation Evaluation

This section evaluates iNet-EGT/C through simulations. Figure 3 shows a simulated server farm (or cloud computing environment) that consists of 100 (10×10) hosts in a grid topology. User requests travel from users to agents via user access point. This simulation study assumes that a single (emulated) user runs on the access point and transmits user requests to agents. At the beginning of a simulation, four agents are deployed on randomly-selected hosts. Each agent has its own iNet-EGT/C that contains the population of 100 behaviors. ($N = 100$ in Listing 1). 25 behaviors are of each of four behavior types: migration, replication, death and do-nothing. Mutation rate (p_m in Listing 1) and behavior selection threshold (t_s in Listing 1) are set to 0.05 and 0.95, respectively.

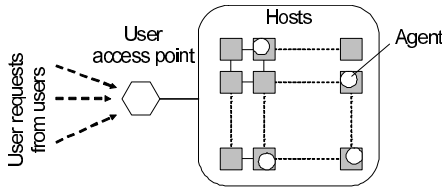


Fig. 3. Simulated Server Farm

Figure 4 shows a trace of workload (the number of user requests) given to agents. It follows an empirical workload measurement at www.ibm.com [20]. The largest workload spike occurs at 12:00 from 3,000 to 9,000 messages/min.

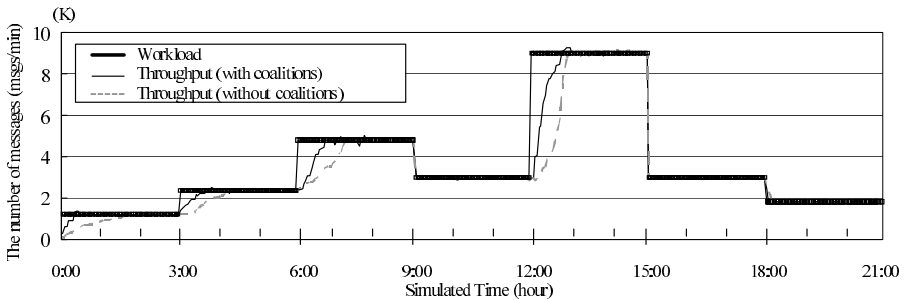


Fig. 4. Workload and Throughput

6.1 Adaptability and Stability

Figure 5 shows how the population state (i.e., behavior distribution) changes over time (from 0:00 to 6:00) in an agent deployed at the beginning of a simulation. The number of replication behaviors increases in the first 15 minutes, and the population converges to an evolutionarily stable state. This means that the agent in question replicates itself to process a given workload. (Initially-deployed four agents are not enough to efficiently process a given workload.) Then, the do-nothing behavior takes over the replication behavior to dominate the population; the population converges to another evolutionarily stable state. At this point, agents have replicated enough to process the current workload; the agent in question does not replicate itself anymore. As illustrated in Figure 5, iNet-EGT/C allows agents to successfully seek evolutionarily stable equilibria in their behavior selection according to dynamic network conditions.

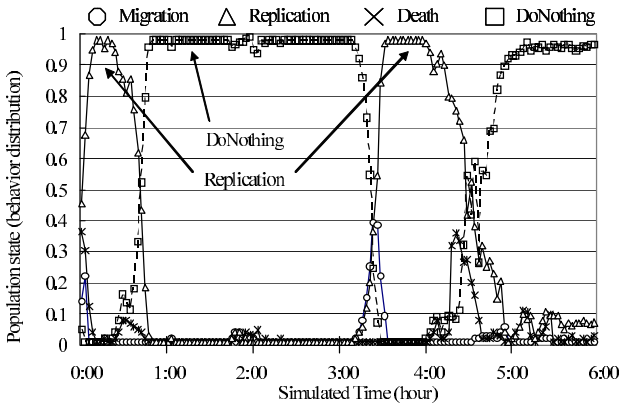


Fig. 5. Changes in Population State

Figure 6 shows how agent availability (i.e., the number of agents) changes dynamically. It increases when the replication behavior dominates the behavior population (e.g., around 0:00 and 3:00; See Figure 5) in response to workload spikes. Conversely, agent availability decreases when the death behavior dominates the behavior population (e.g., around 9:00 and 15:00) in response to workload drops. Figure 6 demonstrates that iNet-EGT/C allows agents to dynamically adapt their availability by invoking their behaviors according to the evolutionarily stable states they are on.

Figures 4 and 7 show the throughput (i.e., the number of processed requests) and response time that agents yield. At the beginning of a simulation, they yield low throughput and high response time because four agents are not enough to efficiently process all user requests. During a simulation, throughput and response time degrade when workload spikes. However, as agents perform their

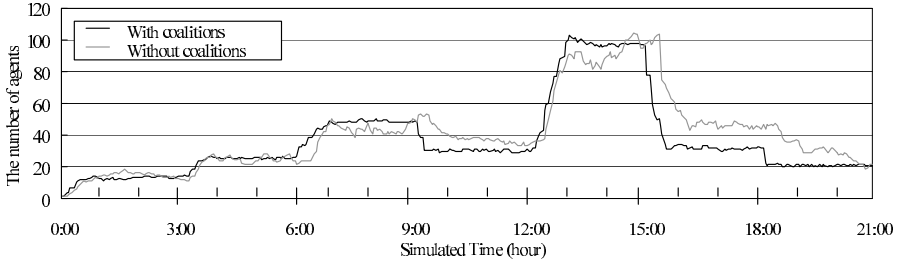


Fig. 6. Agent Availability

behaviors by seeking evolutionarily stable equilibria, they adapt their throughput and response time to dynamic workload changes.

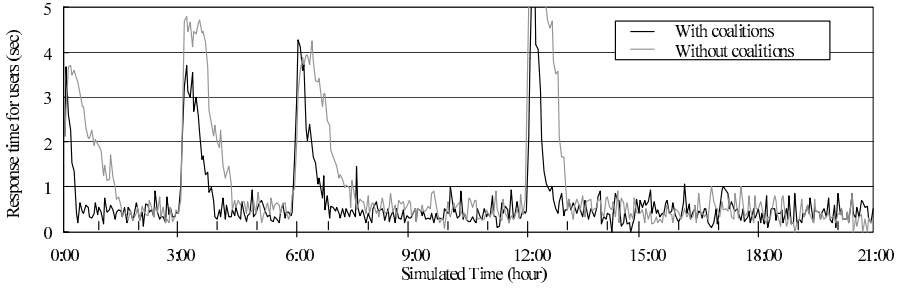


Fig. 7. Response Time

Figure 8 depicts the average behavior population state among agents: $S_{avg}(t) = \frac{1}{A(t)} \sum_i \max_{b \in B} \{x_b(t)\}$. $A(t)$ denotes the total number of agents running in the network. i and b index agents and behavior types, respectively. $\max_{b \in B} \{x_b(t)\}$ denotes the share of the current major behavior in an agent’s behavior population. $S_{avg}(t)$ increases as the current major behavior’s share increases in each agent. It approaches 1.0 (e.g., > 0.95) when the current major behaviors dominate behavior populations and remain effective for agents to adapt to the current network conditions. This means that agents’ behavior population states reaches evolutionarily stable equilibria. $S_{avg}(t)$ decreases when the current major behaviors are no longer effective and the other behaviors take over to dominate behavior populations. For example, $S_{avg}(t)$ remains high from 1:00 to 3:00 because agents have adapted to network conditions by 1:00 and the do-nothing behavior is effective until 3:00 (See also Figure 5). However, the do-nothing behavior becomes ineffective when workload spikes at 3:00; $S_{avg}(t)$ decreases until another behavior (the replication behavior in this case; See Figure 5) takes over and dominates the behavior population. $S_{avg}(t)$ stays over 0.95 during 82% of

the total simulation time. Figure 8 demonstrates that iNet-EGT/C allows agents to seek to operate at evolutionarily stable equilibria in dynamic networks.

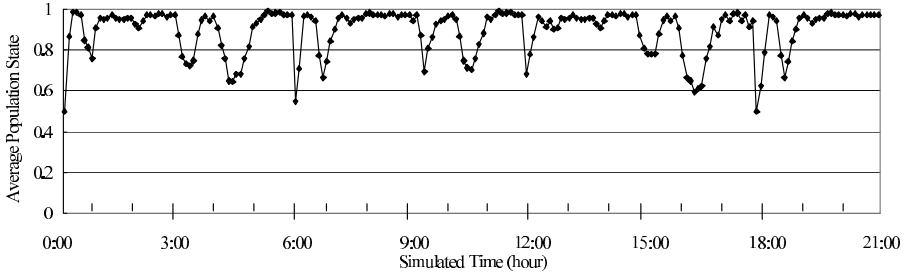


Fig. 8. Average Behavior Population State

6.2 Impacts of Agent Coalitions on Adaptability

Figures 4 and 7 illustrate agents’ throughput and response time with the notion of coalitions disabled. Both figures show that, upon workload spikes (at 0:00, 3:00, 6:00 and 12:00), agents adapt their performance faster by computing payoffs in a cooperative manner with the notion of coalitions. Figure 9 depicts throughput in percentile (i.e., ratio of throughput over workload). The throughput of 100% indicates that agents process all of user requests. Similar to Figure 4, Figure 9 demonstrates that agents improve their throughput faster with coalitions enabled. With coalitions enabled, throughput stays over 90% during 85% of the total simulation time, while 69% with coalitions disabled.

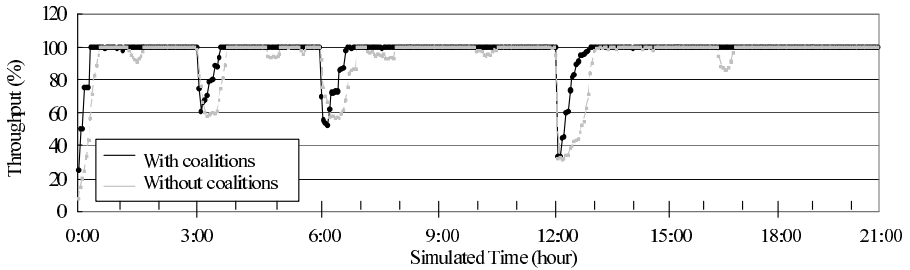


Fig. 9. Throughput (%)

Table 1 shows how soon agents yield the throughput of 100% and 75% upon workload spikes. Agents consistently accelerate their throughput adaptation by

Table 1. Adaptation Speed in Throughput (Minutes)

Workload Spike at	0:00		3:00		6:00		12:00	
Throughput Improvement Rate	100%	75%	100%	75%	100%	75%	100%	75%
Without coalitions	32	22	46	32	64	36	58	48
With coalitions	18	10	36	12	48	14	51	22
Speedup	178%	220%	127%	267%	133%	257%	114%	218%

leveraging the notion of coalitions. The average speedups of 100% and 75% throughput adaptation are 138% and 241%, respectively.

Figure 6 depicts agent availability with coalitions disabled. It verifies that, upon workload spikes and drops (at 0:00, 3:00, 6:00, 9:00, 12:00, 15:00 and 18:00), agents adapt their availability faster by leveraging the notion of coalitions.

Figure 10 shows agent availability in percentile. It is computed as the ratio of expected throughput over workload. The expected throughput is calculated as $A(t)/t_{service}$ where $A(t)$ denotes the total number of agents running in the network and $t_{service}$ denotes the time that an agent is expected to spend to process a single user request. If agent availability is over or under 100%, the number of agents is excess or insufficient, respectively, to process a given workload. The agent availability of 100% indicates that the number of agents perfectly fits with the current workload. Similar to Figure 6, Figure 10 demonstrates that agents adapt their availability faster with coalitions enabled. With coalitions enabled, agent availability stays at 100% during 88% of the total simulation time, while 73% with coalitions disabled.

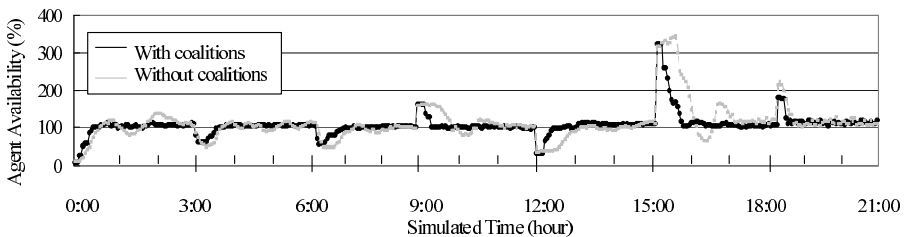
**Fig. 10.** Agent Availability (%)

Table 2 shows how soon agents yield 100% availability upon workload spikes and drops. Agents consistently accelerate their availability adaptation by leveraging the notion of coalitions. The average speedup is 144%.

Table 2. Adaptation Speed in Agent Availability (Minutes)

Workload Spike/Drop at	0:00	3:00	6:00	9:00	12:00	15:00	18:00
Without coalitions	32	42	62	50	56	68	34
With coalitions	20	30	46	28	50	54	22
Speedup	160%	140%	135%	179%	112%	126%	155%

7 Conclusion

This paper proposes and evaluates an evolutionary game theoretic framework, iNet-EGT/C, which aids building adaptive, cooperative and stable network applications. Both theoretical and simulation studies demonstrate that iNet-EGT/C allows network applications to seek to operate at evolutionarily stable equilibria and adapt to dynamic network conditions. The notion of agent coalitions in pay-off computation allows agents to yield the speedup of up to 178% in adaptation to dynamic network conditions.

References

1. Weiss, A.: Computing in the clouds. *ACM netWorker Magazine* 11(4) (2007)
2. Lee, C., Suzuki, J.: An immunologically-inspired autonomic framework for self-organizing and evolvable network applications. *ACM Trans. Autonomous and Adaptive Systems* 4(4) (2009)
3. Lee, C., Vasilakos, A.V., Suzuki, J.: iNet-EGT: An evolutionarily stable adaptation framework for network applications. In: *Proc. of Int'l Conference on Bio-inspired Models of Network, Information and Computing Systems* (December 2009)
4. Subrata, R., Zomaya, A.Y., Landfeldt, B.: Game theoretic approach for load balancing in computational grids. *IEEE Transactions on Parallel and Distributed Systems* 19(1) (2008)
5. Kodialam, M., Lakshman, T.V.: Detecting network intrusions via sampling: a game theoretic approach. In: *Proc. of IEEE Int'l. Conf. on Computer Comm.* (2003)
6. Agah, A., Basu, K., Das, S.K.: Preventing dos attack in sensor networks: a game theoretic approach. In: *Proc. of IEEE Int'l. Conf. on Comm.* (May 2005)
7. Otrok, H., Mehrotra, M., Assi, C.: Game theoretic models for detecting network intrusions. *Computer Communications* 31(10) (June 2008)
8. Kannan, R., Iyengar, S.: Game theoretic models for reliable path-length and energy constrained routing with data aggregation in wireless sensor networks. *IEEE Journal on Selected Areas in Communications* 22(6) (2004)
9. Yan, L., Hailes, S.: Cooperative packet relaying model for wireless ad hoc networks. In: *Proc. of the First ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing* (May 2008)
10. Vasilakos, A.V., Anastasopoulos, M.: Application of evolutionary game theory to wireless mesh networks. In: Jain, L.C., Palade, V., Srinivasan, D. (eds.) *Advances in Evolutionary Computing for System Design*. Springer (2007)
11. Anastasopoulos, M.P., Petraki, D.K., Kannan, R., Vasilakos, A.V.: Tcp throughput adaptation in wimax networks using replicator dynamics. *IEEE Transactions on Systems, Man, and Cybernetics* (January 2010)

12. Li, Z., Parashar, M.: Rudder: A rule-based multi-agent infrastructure for supporting autonomic grid applications. In: Proc. of IEEE International Conference on Autonomic Computing (2004)
13. Hagimont, D., Bouchenak, S., Palma, N., Taton, C.: Self-sizing of clustered databases. In: Proc. of International Symposium on World of Wireless, Mobile and Multimedia Networks (June 2006)
14. Wang, Y., Li, S., Chen, Q., Hu, W.-l.: Biology Inspired Robot Behavior Selection Mechanism: Using Genetic Algorithm. In: Li, K., Fei, M., Irwin, G.W., Ma, S. (eds.) LSMS 2007. LNCS, vol. 4688, pp. 777–786. Springer, Heidelberg (2007)
15. Damas, B.D., Custódio, L.: Emotion-based decision and learning using associative memory and statistical estimation. *Informatica* 27(2) (2003)
16. Kim, K.-J., Cho, S.-B.: Bn+bn: Behavior network with bayesian network for intelligent agent. In: Proc. of AFOSR Australian Conf. on Artificial Intelligence (December 2003)
17. Weibull, J.W.: *Evolutionary Game Theory*. MIT Press (1996)
18. Taylor, P., Jonker, L.: Evolutionary stable strategies and game dynamics. *Elsevier Mathematical Biosciences* 40(1) (1978)
19. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6(2) (2002)
20. Chase, J., Anderson, D., Thakar, P., Vahdat, A., Doyle, R.: Managing energy and server resources in hosting centers. In: 18th Symposium on Operating Systems Principles (October 2001)

Evolving the Asymmetry of the Prisoner's Dilemma Game in Adaptive Social Structures

João Moreira¹, Jorge M. Pacheco^{2,1}, and Francisco C. Santos^{3,1}

¹ ATP-group, CMAF, Complexo Interdisciplinar, Lisboa, Portugal

² Departamento de Matemática e Aplicações, Universidade do Minho, Braga, Portugal

³ DEI & INESC-ID, Instituto Superior Técnico, TU Lisbon, Lisboa, Portugal

Abstract. Empirical studies show that most real social networks exhibit both a significant average connectivity and marked heterogeneity. While the first precludes the emergence of cooperation in static networks, it has been recently shown that the latter induces a symmetry breaking of the game, as cooperative acts become dependent on the social context of the individual. Here we show how adaptive networks can give rise to such diversity in social contexts, creating sufficient conditions for cooperation to prevail as a result of a positive assortment of strategies and the emergence of a symmetry breaking of the game. We further show that realistic heterogeneous networks of high average connectivity where cooperation prevails can result from a simple topological dynamics.

Keywords: Evolution of Cooperation, Evolutionary Game Theory, Distributed Prisoner's Dilemma, Dynamic Networks, Evolutionary Dynamics.

1 Introduction

Conventional Evolutionary Game Theory (EGT) predicts that natural selection favors the selfish and strong [1, 2], in contrast with empirical evidence which shows that cooperation is widespread in nature. The issue of cooperation has been traditionally dealt with in EGT making use of the Prisoner's Dilemma (PD), and several mechanisms have been proposed which make cooperation evolutionary viable [3, 4]. Among those, the structure of the network along which individuals interact drastically affects the chances of cooperation. While homogeneous networks (degree-homogeneous) open a small window of opportunity for cooperation to thrive [5-9], heterogeneous networks (degree-heterogeneous) induce a remarkable boost of cooperative behavior [6, 10-13]. This enhancement, however, is limited to social networks exhibiting low average connectivity, whereas data on realistic networks [14-19] shows that values of the average connectivity (z) up to 170 are possible. This requires yet another mechanism to allow the survival of cooperation.

In this work we shall explore this new mechanism, making use of a PD game in which the benefits collected by the participants may be proportional to the costs expended. Besides the conventional scenario in which every cooperator contributes the same amount to each game they participate in, we shall also explore the limit in

which every cooperator contributes the same overall amount, irrespective of the total number of games they participate in. This is particularly relevant whenever heterogeneous networks are at stake. In such setting the evolution of the interaction network (see below) may break the original symmetric game into an asymmetric game, as the actual game played by each player becomes dependent on their social context [11, 20].

Here, we use a simple adaptive network model [21] that combines strategy evolution with topological evolution [3, 22, 23]. We consider individuals with limited cognitive capacities and investigate the necessary conditions for cooperation to thrive. We will show that network heterogeneity, which emerges as a result of a co-evolution of strategy and topology, is crucial for the appearance and stability of cooperative action. This break of symmetry is naturally induced by a simple dynamics in which individuals revise their contacts uniquely based on their myopic self-interest.

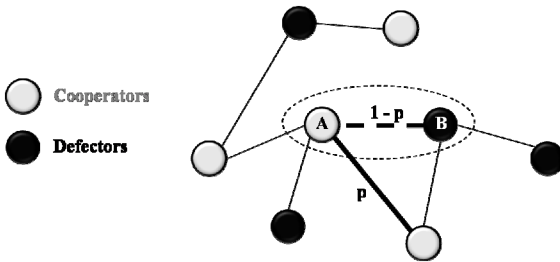


Fig. 1. Readjusting social ties. Cooperators and defectors interact along the links of a network. A (B) is dissatisfied (satisfied) since B (A) is defector (cooperator). Consequently A wants to change the link whereas B does not. The success of the rewiring will depend on the

fitness values $\Pi(A)$ and $\Pi(B)$ of A and B, respectively. With probability p (see section 3.3) A redirects the link to a random neighbor of B. Otherwise, with a probability $1 - p$, A will stay linked to B. Other possibilities may occur depending on the strategies of the chosen individuals (see section 3.3).

2 Co-evolution of Strategy and Topology

We consider a population of individuals that can be either cooperators (C) or defectors (D). They only keep information on their first neighbors, and engage in 2-person PD, where Cs contribute a cost c whereas Ds do not contribute any cost. The total amount is multiplied by an enhancement factor F and then shared equally between the two players. Hence, a player i ($i = 1, 2$) using strategy s_i ($s_i = 1$ if C, 0 if D) will get the payoff $P_i = Fc(s_1 + s_2)/2 - cs_i$ [20]. For $1 < F < 2$ we get the payoff ranking characteristic of a two-person PD.

In the equation above we have considered that Cs contribute a fixed cost c per game. We can consider a somewhat different scenario in which Cs now distribute the same cost c among all games they play. In this case, if player i is a C, she/he will pay a cost $c_i = c/k_i$, where k_i is the player's connectivity (number of neighbors). Consequently, the payoff that a player gets from this game is $P_i = F(c_1s_1 + c_2s_2)/2 - c_i s_i$.

It is reasonable to make this distinction in the costs paid by C s when we allow different individuals to have different number of neighbors (k), a situation that naturally occurs on a heterogeneous graph. For instance, the interaction between two C s directly connected but with different k s will result in a higher payoff for the player with the smaller number of links. This will translate in a symmetry breaking of the game.

Following the convention of Ref. [20] we will refer to the game where C s contribute a fixed cost per game as the *conventional prisoner's dilemma* (CPD); and to the game where C s contribute a fixed cost per individual we call the *distributed prisoner's dilemma* (DPD).

In addition, both C s and D s are able to decide, on an equal footing, which ties they want to maintain and which they want to change. Given an edge with individuals A and B at the extremes, we say that A (B) is satisfied with that edge if the B (A) is a C , being dissatisfied otherwise. If, for instance, A is satisfied, then they will keep the link. If not, then they will compete with B to rewire the link (see Fig. 1 and section 3.3), rewiring being attempted to a random neighbor of B .

This is justifiable on the fact that individuals, who have a limited knowledge of their social environment, will look for new social ties by proxy [24]. In this sense, A is more likely to encounter one of B 's friends and become neighbors with them. In addition, selecting a neighbor of an inconvenient partner may turn out to be a good choice, since this partner also tries to establish links with C s, making it more likely that the rewiring results in a tie to a C .

The fact that in our model C s and D s interact via social ties they both decide upon establishes a coupling between individual strategy and population structure: the game payoff induces an entangled co-evolution of strategy and structure. The adaptive nature of the social structure explained above introduces a new time scale, τ_a , not necessarily equal to the time scale associated with strategy evolution, τ_e . We define a ratio of time scale $W = \tau_e/\tau_a$, which determines the cooperative state of the population at the end of the evolution.

Indeed, whenever $\tau_e \ll \tau_a$, that is, $W=0$, we recover the results of [6, 25]. On the other hand, with increasing W , individuals become ever more proficient at adapting their ties. In general, however, one expects the two time scales to be of comparable magnitude in realistic situations (cf. Figs. 2 and 3).

More intuitively, W provides a measure of individuals' responsiveness to adverse ties: large values of W reflect populations in which individuals react promptly to adverse ties, whereas smaller values of W reveal the opposite behavior.

3 Materials and Methods

We place individuals on the nodes of a graph, to a total of N . A total of N_E links represent the social ties between individuals. Graphs will evolve in time as individuals change their ties. The average connectivity $z = 2N_E/N$ is conserved since we do not create nor destroy links. We require that graphs remain connected at all times. To enforce this condition we impose that nodes connected by a single link cannot lose

this link. Each simulation starts from a homogeneous random graph in which all nodes have the same number of links randomly connected to other nodes [26].

We also computed the cumulative degree distribution $D(k) = N^{-1} \sum_{i=k}^{N-1} N_i$.

N_i indicates the number of nodes with i links so $D(k)$ gives the probability of finding nodes in the graph with degree greater or equal to k . The maximum value of the connectivity of a graph is k_{max} which provides a simple measure of the heterogeneity of a graph since $D(k) = 0$ for $k > k_{max}$.

Whenever $W > 0$, evolution of strategy and structure proceed together under asynchronous updating. Choice of type of update event depends on W . If we assume, without loss of generality, $\tau_e = 1$, then a strategy update event is chosen with probability $1/(1+W)$, a structural update event being selected otherwise.

A strategy update event is defined in the following way, corresponding to the so-called pairwise comparison rule [27]: One node A is chosen at random and another node B is chosen randomly among A 's first neighbors. The individuals A and B interact with all their first neighbors, according to CPD or DPD. As a result, they accumulate the total payoffs $\Pi(A)$ and $\Pi(B)$, respectively. The individual A will imitate the strategy of B with a probability that increases with the payoff difference, which is given by the Fermi distribution function $p = 1/[1 + e^{-\beta[\Pi(B) - \Pi(A)]}]$.

The value of $\beta \geq 0$ (which plays the role of an inverse temperature in statistical physics), controls here the intensity of selection [27]: $\beta \rightarrow 0$ leads to neutral drift whereas $\beta \rightarrow \infty$ leads to the so-called imitation dynamics, often used to model cultural evolution.

C s and D s interact via the links of a network. Two individuals, A and B , connected by one link, may be satisfied or dissatisfied. In Fig. 1, B is satisfied, whereas A is not, since A (B) is a C (D). Therefore, A wants to change the link whereas B does not. The action taken depends on the fitness $\Pi(A)$ and $\Pi(B)$ of A and B , respectively. With a probability p , defined above in terms of the Fermi distribution, A redirects the link to a random neighbor of B . With probability $1 - p$, A stays linked to B . Whenever both A and B are satisfied, nothing happens. When both A and B are dissatisfied, rewiring takes place such that the new link keeps attached to A with probability p and attached to B with probability $1 - p$.

We start our simulations from a homogeneous random graph [26], in which all nodes have the same number of links (z), randomly linked to arbitrary nodes. The population size is $N = 10^3$ with average connectivities $z = 20, 30$, and 40 (the value $z = 30$ used in Fig. 2 and Fig. 3, right panel, reflects the mean value of the average connectivities reported in [13] for social networks). We always start with 50% of C s randomly distributed in the population. In all cases we used $c = 1$ for the cost of cooperation ($c_i = 1/k_i$ for an individual i playing the DPD).

Each value in the figures corresponds to an average over 10^4 different randomly generated configurations and graphs. In each of those we average and evolution over 9×10^5 generations after discarding a transient period of 10^5 generations.

At the end of each evolution we also computed the maximal connectivity k_{max} associated with the final graph and the cumulative degree distribution, which are the basis of the results plotted in Figs. 3.

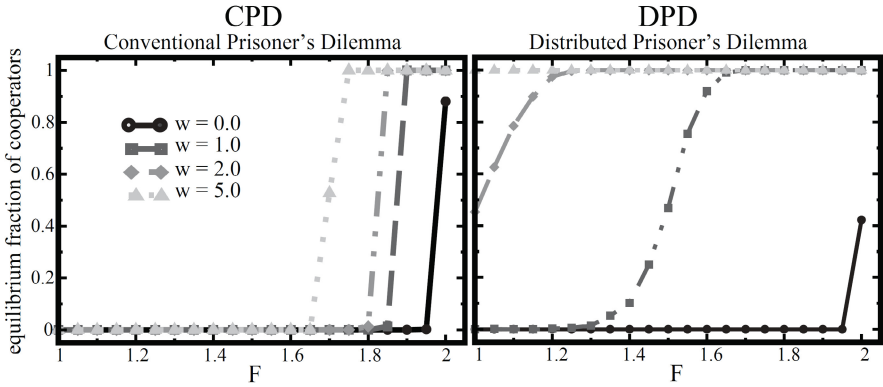


Fig. 2. Co-evolution for different time scales. Equilibrium fraction of cooperators as a function of the enhancement factor F using a homogeneous random graph of $z = 30$ and $\beta = 1.0$. **Left panel:** Under CPD it is difficult for cooperation to emerge unless we allow a fast adaptation of the network structure. As W increases, the rate of link rewiring also increases, and so does the viability of cooperation. **Right panel:** Under DPD, in addition to the adaptive assortment of Cs , cooperation benefits from the break of symmetry associated with the nature of the dilemma and emerging heterogeneity of the network.

4 Results and Discussion

The results of Fig. 2 show the fraction of Cs that survive at the end of evolution (see section 3.4) for different values of W . We plot the graphs for the interval $1 < F < 2$, for both the CPD and the DPD. For $W = 0$ cooperation can be hardly sustained, since the network remains static and equal to the initial homogeneous random network (see above). Moreover, in homogeneous networks the CPD and DPD games are equivalent, as the differences between both amounts to a rescaling of the intensity of selection. It is only when we give individuals the chance to change their social ties, than we begin to see differences. As Cs (Ds) seek for Cs to cooperate (exploit), Cs tend to acquire a higher number of links when compared with Ds . This self-organized heterogeneity benefits the emergence of cooperation [10], in particular when highly connected nodes are occupied by Cs [28].

Yet, in the DPD paradigm, as the network changes, the actual game played by each individual may also change with her degree. The DPD can represent a situation where individuals have limited resources and therefore, as the network becomes more heterogeneous with increasing W , so do the amounts contributed by different Cs . As shown in Ref. [20], in the DPD paradigm the condition for a highly connected C to become advantageous becomes less stringent the larger their connectivity. On the contrary, under the CPD paradigm, the cost of cooperation plays a major role in the overall fitness of the cooperative hub, which means that the larger their connectivity, the harder it will be for the cooperative hub to become advantageous with respect to

any D in their neighborhood. Consequently, when compared with the CPD paradigm, the DPD will promote cooperation as it benefits from the additional break of symmetry of the game induced by evolution of the social structure.

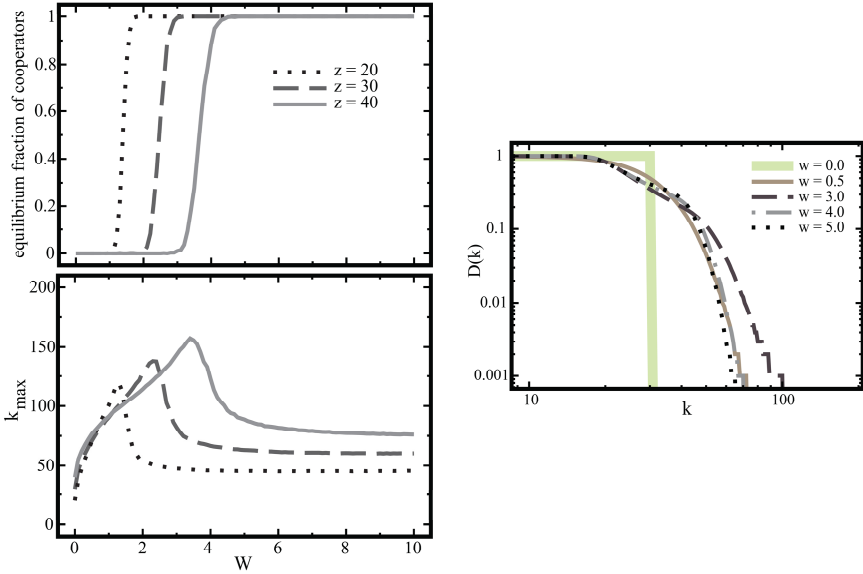


Fig. 3. Co-evolution for different networks. CPD game for $F = 1.8$ and $\beta = 1.0$ using homogeneous random graphs. **Upper left panel:** Equilibrium fraction of cooperators as a function of W for different values of z . For each value of z , there is a critical value of the time scale W_{crit} , above which cooperators wipe out defectors. **Lower left panel:** Maximum value of the connectivity in the population as a function of W . W_{crit} increases monotonically with z . **Right panel:** Cumulative distributions for different values of W . Starting from a homogeneous random network with $k_{max} = z = 30$, just as we increase W , the distribution widens, resulting in both single scale networks ($W = 0.5$, solid brown line) and broad-scale networks ($W > 3$, dotted black and grey lines). W_{crit} is also the value for which the heterogeneity of the associated network reaches a maximum. The results obtained for DPD are qualitatively the same.

Nevertheless, in both games, for a sufficient large W , we will get a full cooperative scenario: The quicker the individuals are able to alter their social ties, the easier it is for D s to become extinct. This behavior is better understood from the upper left panel of Fig. 3: For small W , C s never survive long, but, as W approaches a critical value W_{crit} , they become increasingly better at wiping out D s. The W_{crit} increases monotonically with z , which makes sense because there are more links to be rewired. In practice, W_{crit} is determined as the value of W at which cooperation reaches 50%. Thus, the survival of cooperation relies on the capacity of individuals to adjust to adverse ties, even when the average connectivity is high.

Fig. 3 also provides evidence of the detailed interplay between strategy and structure. On one hand, strategy updating promotes a local assortment of strategies, since C s breed C s and D s breed D s. On the other hand, structural updating promotes

local assortative interactions between C s and disassortative interactions between D s and C s. When both are active, strategy update will promote assortativity among C s, but will restrain disassortativity between D s and C s, which overall will benefit the emergence of cooperation. Additionally, since graphs will become heterogeneous for any $W > 0$, as a result of structural update (we are starting from homogeneous graphs), it will become easier for strategy update to promote cooperation, and even more when we play the DPD (see Fig. 2).

From the left panels of Fig. 3, the overall onset of increase of heterogeneity qualitatively follows the wave of cooperation for the corresponding z [21]. Indeed, the overall heterogeneity of the graph increases with W until it reaches a maximum at W_{crit} , above which heterogeneity again decreases down to a stationary value [21]. This is clearly shown in the right panel of Fig. 3 for a CPD with an enhancement factor $F = 1.8$. The results shown suggest that the adaptive dynamics of social ties accounts for the heterogeneities observed in realistic social networks [16]. The DPD produces results similar to the ones represented in Fig. 3. Also, similar analytic results were already obtained in a simpler model of link rewiring [29].

Our results show that to understand the emergence of cooperative behavior in a realistic scenario, one should consider simultaneously the evolution of the social network of interactions and the evolution of individual strategies. We show how an adaptive social network can easily transform a defection dominance scenario into a different one where cooperation may thrive. Moreover, the co-evolutionary process of strategy and structure can produce realistic heterogeneous networks. Hence, besides providing a bottom-up answer to the problem of cooperation, the proposed mechanism also shows how complex social topologies can result from simple social dynamical processes, exclusively based on local assumptions. In addition, the emergence of such heterogeneous structures with diverse social contexts becomes particularly relevant whenever individuals contributions are correlated with the social context they are embedded in. In this regime, network dynamics is able to remove the game symmetry of the PD in homogeneous networks, opening a route for cooperation to thrive.

Finally, the DPD used here relies on the fact that all cooperators are effectively assessed as cooperators, irrespectively of the amount contributed. In fact, such assessment should rely on a social norm [30], which may evaluate an action as “Good” or “Bad”. From this perspective, our setting considers a social norm where the act of giving is seen as more important than the amount given, under which, as we show, cooperation prevails.

References

1. Maynard Smith, J., Szathmáry, E.: *The Major Transitions in Evolution*. W. H. Freeman, Oxford (1995)
2. Sigmund, K.: *The Calculus of Selfishness*. Princeton University Press (2010)
3. Zimmermann, M.G., Eguíluz, V.M.: Cooperation, social networks, and the emergence of leadership in a prisoner's dilemma with adaptive local interactions. *Phys Review E* 72 (2005)
4. Nowak, M.A.: Five rules for the evolution of cooperation. *Science* 314, 1560–1563 (2006)
5. Nowak, M.A., May, R.M.: Evolutionary Games and Spatial Chaos. *Nature* 359, 826–829 (1992)

6. Santos, F.C., Pacheco, J.M., Lenaerts, T.: Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proc. Natl. Acad. Sci. U S A* 103, 3490–3494 (2006)
7. Ebel, H., Bornholdt, S.: Coevolutionary games on networks. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 66, 056118 (2002)
8. Szabó, G., Fáth, G.: Evolutionary games on graphs. *Physics Reports* 446, 97–216 (2007)
9. Szabó, G., Tóke, C.: Evolutionary prisoner's dilemma game on a square lattice. *Physical Review E* 58, 69–73 (1998)
10. Santos, F.C., Pacheco, J.M.: Scale-free networks provide a unifying framework for the emergence of cooperation. *Phys. Rev. Lett.* 95, 098104 (2005)
11. Santos, F.C., Santos, M.D., Pacheco, J.M.: Social diversity promotes the emergence of cooperation in public goods games. *Nature* 454, 213–216 (2008)
12. Gómez-Gardeñes, J., Campillo, M., Floría, L.M., Moreno, Y.: Dynamical Organization of Cooperation in Complex Topologies. *Phys. Rev. Lett.* 98, 108103 (2007)
13. Masuda, N.: Participation costs dismiss the advantage of heterogeneous networks in evolution of cooperation. *Proc. Biol. Sci.* 274, 1815–1821 (2007)
14. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* 286, 509–512 (1999)
15. Watts, D.J.: *Small worlds: The dynamics of networks between order and randomness.* Princeton University Press, Princeton (1999)
16. Amaral, L.A., Scala, A., Barthelemy, M., Stanley, H.E.: Classes of small-world networks. *Proc. Natl. Acad. Sci. U S A* 97, 11149–11152 (2000)
17. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74, 47–98 (2002)
18. Dorogotsev, S.N., Mendes, J.F.F.: *Evolution of Networks: From Biological Nets to the Internet and WWW.* Oxford University Press, Oxford (2003)
19. Watts, D.J.: The “new” science of networks. *Ann. Rev. Sociobiol.* 30, 243–270 (2004)
20. Pacheco, J.M., Pinheiro, F.L., Santos, F.C.: Population structure induces a symmetry breaking favoring the emergence of cooperation. *PLoS Comput. Biol.* 5, e1000596 (2009)
21. Santos, F.C., Pacheco, J.M., Lenaerts, T.: Cooperation prevails when individuals adjust their social ties. *PLoS Comput. Biol.* 2 e140 (2006)
22. Tanimoto, J.: Promotion of cooperation through co-evolution of networks and strategy in a 2 x 2 game. *Physica A* 388, 953–960 (2009)
23. Tanimoto, J.: Dilemma solving by the coevolution of networks and strategy in a 2x2 game. *Phys. Review E* 76, 021126 (2007)
24. Kossinets, G., Watts, D.J.: Empirical analysis of an evolving network. *Phys. Rev. Lett.* 311, 89–90 (2006)
25. Ohtsuki, H., Hauert, C., Lieberman, E., Nowak, M.A.: A simple rule for the evolution of cooperation on graphs and social networks. *Nature* 441, 502–505 (2006)
26. Santos, F.C., Rodrigues, J.F., Pacheco, J.M.: Epidemic spreading and cooperation dynamics on homogeneous small-world networks. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 72, 056128 (2005)
27. Traulsen, A., Nowak, M.A., Pacheco, J.M.: Stochastic dynamics of invasion and fixation. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 74, 011909 (2006)
28. Santos, F.C., Rodrigues, J.F., Pacheco, J.M.: Graph topology plays a determinant role in the evolution of cooperation. *Proc. Biol. Sci.* 273, 51–55 (2006)
29. Pacheco, J.M., Traulsen, A., Nowak, M.A.: Coevolution of strategy and structure in complex networks with dynamical linking. *Physical Review Letters* 97, 258103 (2006)
30. Nowak, M.A., Sigmund, K.: Evolution of indirect reciprocity. *Nature* 437, 1291–1298 (2005)

Feature and Kernel Evolution for Recognition of Hypersensitive Sites in DNA Sequences

Uday Kamath¹, Amarda Shehu^{1,2}, and Kenneth A. De Jong¹

¹ Department of Computer Science

² Department of Bioinformatics and Computational Biology

George Mason University, Fairfax, VA, 22030

kdejong@gmu.edu

Abstract. The annotation of DNA regions that regulate gene transcription is the first step towards understanding phenotypical differences among cells and many diseases. Hypersensitive (HS) sites are reliable markers of regulatory regions. Mapping HS sites is the focus of many statistical learning techniques that employ Support Vector Machines (SVM) to classify a DNA sequence as HS or non-HS. The contribution of this paper is a novel methodology inspired by biological evolution to automate the basic steps in SVM and improve classification accuracy. First, an evolutionary algorithm designs optimal sequence motifs used to associate feature vectors with the input sequences. Second, a genetic programming algorithm designs optimal kernel functions that map the feature vectors into a high-dimensional space where the vectors can be optimally separated into the HS and non-HS classes. Results show that the employment of evolutionary computation techniques improves classification accuracy and promises to automate the analysis of biological sequences.

Keywords: DNase I hypersensitive sites, evolutionary algorithms, support vector machines, genetic programming, kernel functions, motifs.

1 Introduction

Many diseases and phenotypical differences among cells are caused by variations in non-coding regions of DNA that regulate gene transcription [29,15]. Since the successful annotation of the human genome with functional coding regions [6], locating regulatory regions is now the remaining challenge to mapping out the entire human genome [29,15]. Based on the observation that regulatory regions bind with transcription-factor binding proteins to activate or repress following genes, many experimental techniques originally relied on detecting transcription-factor binding sites to locate regulatory regions. This approach has proved costly and time consuming [1]. Current techniques rely instead on identifying sites that precede regulatory regions and are particularly sensitive to DNA-modifying enzymes like non-specific endonuclease DNase I [41,13,26,4,35,29,15]. A wealth of short DNA sequences determined to be hypersensitive (HS) sites are now available from high-throughput experimental techniques [35,9].

Despite the discovery that HS sites are reliable markers of regulatory regions, rapid annotation of the entire human genome requires a combination of experimental and computational techniques [33]. The abundance of HS sequences already identified in wet laboratories allows applying statistical learning techniques to automate the annotation process. Recent work explores the employment of Support Vector Machines (SVM) to the binary classification problem of classifying short DNA sequences as HS or non-HS [33,21]. SVMs have a solid theoretical foundation in statistical learning theory and are the most widely used machine learning technique in binary classification problems [39]. In bioinformatics, SVMs have been applied to predict protein localization sites [14], DNA translation start sites [43], DNA splice sites [42,20,19], and more (cf. to [32]).

Despite their broad applicability, important decisions in an SVM classifier remain problem specific and require some understanding of the problem domain. Essentially, an SVM maps non-vector data, such as text, graphs, and strings, into a vector space where a hyperplane can be found to optimally separate the vectors into the two available classes. The process consists of two basic steps. In the context of classifying input DNA sequences as HS or non-HS, the first step involves associating feature vectors with the input sequences. The second step involves mapping the feature vectors into a high-dimensional space where labeled data can be linearly separable by a hyperplane. Once the hyperplane is computed, predicting the label of an unlabeled data point involves determining on which side of the hyperplane the point lies.

The success of an SVM classifier depends on both the choice of the feature space and the internal transformation, the *kernel function*, used. Often, the main novelty in applying an SVM to a new classification problem is the extraction of meaningful features that allow converting the input data into vectors. For instance, if training sequences belonging to one class are known to contain specific subsequences with higher frequency than the sequences belonging to the other class, these subsequences could be used as features, and their frequency of occurrence can be used to convert an input sequence into a feature vector. Such information is not available to non-experts, and significant time and resources are often devoted to finding features that give meaningful vectors.

The particular choice of a kernel function that transforms the feature vectors into a high-dimensional space where the data are linearly separable is also problem-specific. Finding an optimal kernel function is nontrivial, and many researchers rely on testing a small number of predefined kernels. Well-known kernel functions include the Linear, Polynomial, Radial Basis, Gaussian Radial, and Sigmoid kernel functions [2]. One needs to determine not only the kernel function that yields the highest classification accuracy, but also the optimal values for the various parameters contained in the kernel function. Finding the right kernel function and the right parameters for the selected kernel function is a tedious optimization process requiring many cycles of experimentation.

The contribution of this paper is a novel methodology that removes the need for expert input and automates the two basic components of an SVM classification, feature and kernel selection, all the while improving classification accuracy.

The methodology is inspired by biological evolution and employs evolutionary computation techniques to evolve optimal features and optimal kernel functions. We have recently proposed an evolutionary algorithm (EA) to design optimal features [21]. The novelty in the work presented here is the combination of these features with novel optimal kernel functions evolved through a novel genetic programming (GP) algorithm. Our results show that the employment of these evolutionary computation techniques to select optimal features and optimal kernel functions automates the process of SVM classification and significantly improves the classification accuracy.

The rest of this paper is organized as follows. A brief summary of related work is provided in section 1.1. Our method is described in section 2. Results follow in section 3. The article concludes with a discussion in section 4.

1.1 Related Work

The issue of extracting meaningful features from biological sequences is circumvented when employing implicit string kernels. These kernels directly associate distances in the feature space through suffix trees or other similarity measures [24]. In other applications, one first associates feature vectors with input sequences and then uses a kernel function to obtain distances in the feature space through dot product calculations [25]. Extracting explicit features has distinct advantages. The features can encapsulate important biological features, and their relative strength or contribution to learning can be directly measured.

Often, the main novelty in applying an SVM to a new classification problem is the extraction of meaningful features that allow converting the input data into vectors. When no prior knowledge is available to guide the design of meaningful features, spectrum features are often employed to explicitly map an input sequence to a vector space [25]. A k -spectrum is the set of $d = |\Sigma|^k$ features that correspond to all strings of length k (k -mers) generated from an alphabet Σ . A d -dimensional feature vector is then associated with an input sequence by recording the frequency of occurrence of each of the d k -mers in the sequence. Such an approach is employed in [33] to recognize HS sequences.

Using spectrum features, however, has the disadvantage of an exponential increase in the number of features as the spectrum length increases. For instance, the 6-mers employed in [33] result in 4096 features. A high number of features adversely impacts the performance of the SVM, both in terms of running time and classification accuracy. Analysis in [33] and our recent work [21] reveals that a very small percentage of the 6-spectrum features actually contribute to learning. These observations, together with our hypothesis that sequence motifs in HS sequences may make for better features, motivated our recent work [21].

Instead of enumerating fixed-length sequences, our work in [21] proposes an EA to explore the space of fixed-length sequences in search of optimal motifs that best discriminate between HS and non-HS sequences. Employing these motifs instead of spectrum features improves the classification accuracy by as much as 10% [21]. We have shown the benefits of employing these motifs over spectrum

features in other bioinformatics applications [22]. We employ the EA proposed in [21] and briefly summarized in section 2 to obtain features in this work.

The success and effectiveness of an SVM classifier depends not only on the choice of the feature space, but also the selected kernel function. Many researchers test a small number of predefined kernels functions, such as Linear, Polynomial, Radial Basis, Gaussian Radial, and Sigmoid [2], to select one that yields the highest classification accuracy. Many of the kernel functions contain parameters that need further tuning to improve accuracy. Finding the right kernel function and the right parameters can be a tedious optimization process.

A heuristics-based grid search technique is often employed to tune kernel parameters [28]. Particle Swarm Optimization and Genetic Algorithms (GAs) have been employed to find optimal parameters in a Gaussian kernel in [8,18]. Evolutionary-based methods are beginning to be applied not only to find optimal parameters in a selected kernel function, but also to design an optimal kernel function [34,11]. GP is employed in [12] to evolve kernel functions. The functions in [12] are not guaranteed to follow Mercer's theorem, so optimality is not guaranteed. Additionally, in all current applications of GP, a small predefined set of kernel functions is employed to evolve new kernel functions [12,38]. The set includes only the Linear, Gaussian, and Polynomial functions, excluding many other known kernel functions. Additionally, the cost parameter C , which controls the trade off between allowing misclassification errors during training of the SVM and forcing rigid margins, is kept at a fixed value in [12,38].

In this paper, the optimal features obtained with an EA are combined with an optimal kernel function obtained with a GP algorithm. The main novelty of the work presented here is a novel approach that allows simultaneously evolving kernel functions, their parameters, and the SVM cost parameter C . Unlike the existing work summarized above, an extensive list of available kernel functions is employed to evolve new functions. Additionally, the evolution of the kernel functions in our GP is subjected to the Mercer's theorem that kernel functions be positive semi-definite [36], thus guaranteeing optimality.

2 Methods

The EA introduced in our recent work [21] and employed here to obtain meaningful features is briefly summarized below. The rest of the section details the novel GP algorithm we propose to evolve kernel functions, their parameters, and the SVM cost parameter C .

2.1 Finding Over-Represented Motifs in DNA Sequences

The EA we introduce in [21] essentially searches for motifs that best discriminate between HS and non-HS sequences. The motifs are variable-length strings of length $l \in \{6, \dots, 12\}$ generated from the the IUPAC code [5] shown in Table 1. In addition to the 4-letter $\{A, T, C, G\}$ DNA alphabet, the IUPAC code contains ambiguous symbols that allow specifying groups of nucleotides with

shared chemical properties. The motifs vary in length from 6-mers to 12-mers because no a priori information is available on the length of optimal motifs. Additionally, work in [33], which employs 6-mers generated over the 4-letter alphabet of DNA, shows that no shorter than 6-mers are needed to achieve high classification accuracy with an SVM.

Table 1. IUPAC code is adapted from [5]

Symbol	Meaning	Description Origin
G	G	G uanine
A	A	A denine
T	T	T hymine
C	C	C ytosine
R	G or A	pu R ine
Y	T or C	p Y rimidine
M	A or C	a M ino
K	G or T	K etone
S	G or C	S trong interaction
W	A or T	W eak interaction
H	A or C or T	H follows G in alphabet
B	G or T or C	B follows A in alphabet
V	G or C or A	V follows U in alphabet
D	G or A or T	D follows C in alphabet
N	G or A or T or C	a N y

The EA searches the space of candidate motifs using a $(\mu + \lambda)$ -ES-style EA, where μ is the number of parents and λ is the number of offsprings generated as each population of motifs evolves. The first population contains μ randomly generated motifs. In each generation, parents are selected uniformly at random to produce λ offsprings through mutation and crossover. Truncation is employed to determine which of the μ fittest individuals (motifs) will survive as the next generation of parents. In our recent work [21] and experiments here, $\mu = 500$ and $\lambda = 200$. Our recent work [21] additionally shows that the island-model approach yields better motifs than crossbreeding motifs of different lengths, which is confirmed by other work [40,7]. In the island-model approach, each island contains motifs of the same length (i.e., one motif species) and evolves in isolation and in parallel with other islands without migration. It is also interesting to point out that both in nature and evolutionary algorithms, offsprings produced by structurally dissimilar parents are inviable.

Given a current population of motifs, there is an equal chance of applying either mutation or crossover to generate a new offspring. If the mutation operator is chosen, each motif in the current population has equal probability of being selected as a parent. The mutation operator is equivalent to the bit flip operator. Any of the symbols of the chosen parent has equal probability of being mutated into any of the symbols of the IUPAC code. When applying crossover, any pair of motifs has equal probability of being selected as parents. The genetic material of the parents is combined to produce an offspring. While the EA algorithm we

introduce in [21] allows for two-point or uniform crossover, the best results in [21] and the results in this work are obtained with the one-point crossover. While crossover is intended to mix the genetic code of each parent and confer it to fit children, the mutation operator is a fundamental evolutionary mechanism to provide diversity. Mutation is intended to prevent all offsprings/sought solutions of the fitness function to fall into a local optimum [7].

While the true fitness of an individual should be evaluated in the context of SVM-based classification, doing so on each offspring is computationally impractical. We employ a simpler fitness function that approximates how spectrum features in an SVM are employed in the kernel function [33]. Given a k -mer w , the fitness value $f(w) = 100 * |c(w)_{\text{HS}}/\text{total}_{\text{HS}} - c(w)_{\text{non-HS}}/\text{total}_{\text{non-HS}}|$, where $c(w)$ counts the number of sequences containing w , and total normalizes by the number of known sequences in each class (HS or non-HS). According to this fitness function, a motif that is found in all HS sequences but no non-HS sequences, or alternatively in all non-HS sequences but no HS sequences, will have the highest fitness score of 100. A motif found with the same frequency in non-HS and HS sequences will have the lowest score of 0. Analysis in our recent work [21,22] shows that the fitness value of the top motifs strongly correlates with the classification accuracy these motifs confer to an SVM.

In the results in section 3, an upper bound 5000 generations is used (convergence in the top fitness scores is generally obtained within the first 500 generations). The top 200 motifs of the final population are then employed to construct feature vectors from the input DNA sequences. These feature vectors need to be transformed by a kernel function. The GP algorithm proposed below searches the space of positive semi-definite kernel functions for the one that yields the highest SVM classification accuracy when applied to the feature vectors.

2.2 Genetic Programming Algorithm

The GP algorithm we propose searches over the space of positive semi-definite kernel functions in order to guarantee optimality. We first discuss the concept of *kernel closure* and then detail the elements of the proposed GP algorithm.

Kernel Closure. Kernel functions must be continuous, symmetric, and preferably have a positive (semi-)definite Gram matrix [36]. Kernels that satisfy Mercer’s theorem are positive semi-definite, meaning that their kernel matrices have non-negative eigenvalues. The use of a positive definite kernel insures that the optimization problem will be convex and the solution will be unique [36]. Work in [3] has demonstrated that kernels which are only conditionally positive definite can outperform most classical kernels in many applications. Moreover, new kernel functions can be constructed by combining well-known positive (semi-)definite kernel functions through specific mathematical manipulations that guarantee closure; that is, the new kernel functions inherit the properties of the kernel functions used to construct them [37].

Guaranteeing Closure. In this work, we employ an extensive list of kernel functions that have been proved to be positive (semi)-definite to construct new kernel functions that obey the closure property. The mathematical manipulations that allow doing so are listed in Table 2.

Table 2. Mathematical operations that guarantee closure. x, y refer to two feature vectors on which the kernel operates. k_1 and k_2 refer to two kernel functions used to construct a new kernel function k .

Add:	$k(x, y) = k_1(x, y) + k_2(x, y)$
Scalar:	$k(x, y) = a \cdot k_1(x, y)$
Multiply:	$k(x, y) = k_1(x, y) \cdot k_2(x, y)$
Exponential:	$k(x, y) = e^{k_1(x, y)}$

Kernel Functions. Below is the extensive list of kernel functions that have been proved to be positive (semi)-definite and have been successfully employed in SVM classification. Our GP algorithm employs all of these kernel functions to evolve new ones.

(a) *Linear Kernel.* The linear kernel is the simplest kernel function. It is given by the inner dot product between two vectors and employs an optional coefficient c : $k(x, y) = x^T y + c$.

(b) *Polynomial Kernel.* The polynomial kernel is a non-stationary kernel well suited for problems where all the training data is normalized. The kernel contains parameters for slope α , coefficient constant c , and degree d : $k(x, y) = (\alpha x^T y + c)^d$.

(c) *Gaussian Kernel.* The Gaussian kernel is an example of the radial basis function kernel that is successfully employed as the default kernel in mainstream SVM implementations and applications. The adjustable parameter σ plays a major role in the performance of the kernel: $k(x, y) = e^{-\sigma \|x-y\|^2}$.

(d) *Laplace Kernel.* The Laplace kernel is another radial basis functional kernel, which is less sensitive to changes in the σ parameter: $k(x, y) = e^{-\frac{\|x-y\|}{\sigma}}$.

(e) *Anova Kernel.* The Anova kernel is another radial basis function kernel that performs well in multidimensional regression problems [16]: $k(x, y) = (\sum_{k=1}^n e^{-\sigma(x_k - y_k)^2})^d$.

(f) *Sigmoid Kernel.* The sigmoid kernel is popular due to its origin from neural network theory. Despite being only conditionally positive definite, it has been successful in various applications: $k(x, y) = \tanh(\alpha x^T y + c)$.

(g) *Rational Quadratic Kernel.* This kernel is often employed for faster computations to obtain similar results to the Gaussian kernel: $k(x, y) = 1 - \frac{\|x-y\|^2}{\|x-y\|^2 + c}$.

(h) *Inverse Multiquadratic Kernel.* The Inverse Multiquadratic kernels has also been shown to be positive definite: $k(x, y) = \frac{1}{\sqrt{\|x-y\|^2+c^2}}$.

(i) *Circular Kernel.* The Circular kernel is inspired from statistics. It is an example of an isotropic stationary kernel and is positive definite in \mathcal{R}^2 : $k(x, y) = \frac{2}{\pi} \arccos(-\frac{\|x-y\|}{\sigma}) - \frac{2}{\pi} \frac{\|x-y\|}{\sigma} \sqrt{1 - \frac{\|x-y\|^2}{\sigma^2}}$.

(j) *Spherical Kernel.* The Spherical kernel is similar to the circular kernel, but it is positive definite in \mathcal{R}^3 : $k(x, y) = 1 - \frac{3}{2} \frac{\|x-y\|}{\sigma} + \frac{1}{2} (\frac{\|x-y\|}{\sigma})^3$ if $\|x - y\| < \sigma$. Otherwise, $k(x, y) = 0$.

(k) *Wave Kernel.* The Wave kernel is also symmetric positive definite: $k(x, y) = \frac{\theta}{\|x-y\|} \sin(\frac{\|x-y\|}{\theta})$.

(l) *Spline Kernel.* The Spline kernel is given as a piecewise cubic polynomial and is positive semi-definite: $k(x, y) = \prod_{i=1}^d 1 + x_i y_i + x_i y_i \min(x_i, y_i) - \frac{x_i + y_i}{2} \min(x_i, y_i)^2 + \frac{\min(x_i, y_i)^3}{3}$, where $x, y \in \mathcal{R}^d$.

(m) *Bessel Kernel.* The Bessel kernel is well known in the theory of function spaces of fractional smoothness. It is given by $k(x, y) = \frac{J_{\nu+1}(\sigma\|x-y\|)}{\|x-y\|^{-\nu(\nu+1)}}$, where J is a Bessel function of order 1.

(n) *Cauchy Kernel.* The Cauchy kernel is a long-tailed kernel that can be used to give long-range influence and sensitivity over a high-dimensional space of feature vectors: $k(x, y) = \frac{1}{1 + \frac{\|x-y\|^2}{\sigma}}$.

(o) *Chi-square Kernel.* The Chi-square kernel comes from the well known chi-square distribution: $k(x, y) = 1 - \sum_{i=1}^n \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}$.

(p) *Histogram Kernel.* The Histogram Intersection kernel has been successfully used for image classification but is generally applicable to a variety of other applications: $k(x, y) = \sum_{i=1}^n \min(x_i, y_i)$.

(q) *Generalized T-Student Kernel.* The Generalized T-Student kernel has been proven to have a positive semi-definite matrix: $k(x, y) = \frac{1}{1 + \|x-y\|^d}$.

Genetic Programming. Our GP algorithm simultaneously searches the space of kernel functions (and their parameters) and SVM cost parameters C . Each population consists of 2,000 individuals. The first generation contains the extensive list of kernel functions detailed above, together with default values for their parameters and the SVM cost parameter C . A total of 30 generations are evolved to obtain the kernel function and cost parameter C that yield the highest classification accuracy (convergence in the fitness scores is observed after 20

generations). The fitness of each individual in a population is the classification accuracy obtained by applying the SVM on the feature vectors obtained with the above EA using the kernel function and parameter C contained in the individual.

Representation. Each individual is represented as a forest of two trees. One tree maintains the kernel function, and the other the cost parameter C . The tree that maintains C consists of only one node, and is subjected only to the mutation operator. The representation of the kernel tree in each individual is analogous to the parse trees employed for Lisp expressions. Each non-terminal node in a kernel tree is either a mathematical operator (add, scalar, multiply, exponential) or a predefined kernel function (from the extensive list detailed above). Each terminal node in a kernel tree is either a kernel parameter (e.g., σ , d , θ) or one of the input feature vectors x , y . See Fig. 1 for some examples of kernel functions.

Closure Constraint on Kernel Tree. Under the principle of closure, each node in the kernel tree may take any subtree as a child. In basic form, closure allows any non-terminal node to be a parent of any other node. We employ strongly typed GP (STGP) to evolve kernel functions. STGP places additional type constraints on the nodes, specifying the nodes that may link with others. STGP is typically used to allow child nodes to pass data to a parent that is guaranteed to be able to read the data [31]. This constraint allows the mutation and crossover operators to generate semantically-correct trees.

Ephemeral Constants. The cost parameter C and the various parameters of a kernel function (e.g., σ , d , θ) are terminal nodes internally implemented as ephemeral constants (ERC) [7] in our GP algorithm. All ERCs undergo mutation only. Parameters that take values in \mathcal{R} are sampled uniformly at random from preset ranges each time they undergo mutation. For instance, the range of values for C is $[2^{-5}, 2^5]$. Parameters that take integer values (e.g., in the Anova and Bessel kernel functions) are internally implemented as ERC-int nodes. Table 3 gives a summary of all the non-terminal (mathematical operators and predefined kernel functions from the extensive list detailed above) and terminal nodes (ERC and ERC-int nodes representing kernel parameters and the SVM cost parameter C).

Mutation. The mutation operator is applied both to parameters and kernel functions. A tree is first picked at random (out of the tree maintaining the kernel function and the single-node tree maintaining the cost parameter C). A node is then picked at random from the selected tree. Every ERC located in the subtree rooted at the selected node is then mutated according to a Gaussian probability distribution over the preset range of the parameter maintained in the ERC node. When the selected tree is the kernel tree, the mutation operator changes the structure of the kernel function. Mutation is implemented through the growth method in [23]. A node picked at random from the kernel tree is replaced by a randomly-generated subtree, as illustrated in Fig. 1.

Table 3. All terminal and non-terminal nodes are shown

Name (Symbol)	Formula	Args, Constrs
Polynomial (P)	$k(x, y) = (\alpha x^T y + c)^d$	$3, c, d \in \mathcal{R}$
Linear (L)	$k(x, y) = x^T y + c$	$3, c \in \mathcal{R}$
Sigmoid (S)	$k(x, y) = e^{-\sigma \ x-y\ ^2}$	$3, \sigma \in \mathcal{R}$
Laplace (Lp)	$k(x, y) = e^{-\frac{\ x-y\ }{\sigma}}$	$3, \sigma \in \mathcal{R}$
Anova (A)	$k(x, y) = (\sum_{k=1}^n e^{-\sigma(x_k - y_k)^2})^d$	$3, \sigma \in \mathcal{R}$
Rational Quadratic (RQ)	$k(x, y) = 1 - \frac{\ x-y\ ^2}{\ x-y\ ^2 + c}$	$3, c \in \mathcal{R}$
Inv. MultiQuadratic (IQ)	$k(x, y) = \frac{1}{\sqrt{\ x-y\ ^2 + c^2}}$	$3, c \in \mathcal{R}$
Circular (CLR)	$k(x, y) = \frac{2}{\pi} \arccos(-\frac{\ x-y\ }{\sigma}) - \frac{2}{\pi} \frac{\ x-y\ }{\sigma} \sqrt{1 - \frac{\ x-y\ ^2}{\sigma^2}}$	$3, \sigma \in \mathcal{R}$
Spherical (SPL)	$k(x, y) = 1 - \frac{3}{2} \frac{\ x-y\ }{\sigma} + \frac{1}{2} (\frac{\ x-y\ }{\sigma})^3$ if $\ x-y\ < \sigma$	$3, \sigma \in \mathcal{R}$
Wave (W)	$k(x, y) = \frac{\theta}{\ x-y\ } \sin(\frac{\ x-y\ }{\theta})$	$3, 0 \leq \theta < 2\pi$
Spline (SLN)	$k(x, y) = \prod_{i=1}^d 1 + x_i y_i + x_i y_i \min(x_i, y_i) - \frac{x_i + y_i}{2} \min(x_i, y_i)^2 + \frac{\min(x_i, y_i)^3}{3}$	$3, d \in \mathcal{I}$
Bessel (B)	$k(x, y) = \frac{J_{v+1}(\sigma \ x-y\)}{\ x-y\ ^{-n(v+1)}}$	$4, n \in \mathcal{I}$
Cauchy (Cy)	$k(x, y) = \frac{1}{1 + \frac{\ x-y\ ^2}{\sigma}}$	$3, \sigma \in \mathcal{R}$
Chi-Square (CHI)	$k(x, y) = 1 - \sum_{i=1}^n \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}$	2
Histogram (HI)	$k(x, y) = \sum_{i=1}^n \min(x_i, y_i)$	2
T-Student (T-s)	$k(x, y) = \frac{1}{1 + \ x-y\ ^d}$	$3, \sigma \in \mathcal{R}$
Add (+)	$k(x, y) = k_1(x, y) + k_2(x, y)$	2Kernels
Multiply (*)	$k(x, y) = k_1(x, y) \cdot k_2(x, y)$	2Kernels
Scalar (Sc)	$k(x, y) = a \cdot k_1(x, y)$	1Kernel, $a \in \mathcal{R}$
Exponential (E)	$k(x, y) = e^{k_1(x, y)}$	1Kernel
x_i, \dots, x_n		0,input
y_i, \dots, y_n		0,input
ERC-int		Integer
ERC		Real

Crossover. the crossover operator applies only to the kernel tree. Parents are selected through the standard Tournament selection, and the standard Koza-style crossover mechanism is employed to generate an offspring kernel function. Compatible parents are sought; that is, individuals are randomly sampled from a population until two are found whose kernel trees have the same constraints. A random node is then chosen in each parent tree such that the two nodes have the same return type. If, by swapping subtrees at these nodes, the two trees do not violate maximum depth constraints, the swap is performed. Otherwise, the hunt for random nodes is repeated. Fig. 1 provides an illustration of crossover.

Bloat Control. One issue in GP is the unconstrained growth of individuals with no performance improvement. This growth, called bloat, may be limited by special genetic operators that restrict the maximum tree depth of an individual. An alternative technique, parsimony pressure, penalizes the size of an individual by making it less fit. We employ a simple technique, lexicographic tournament selection, to control bloat. Lexicographic tournament selection is similar to

tournament selection. The only difference is that, if multiple individuals have the same fitness, the individual with the shortest tree depth is chosen [30].

Putting it All Together. The optimal features are evolved separately through the EA we propose in [21] and employ here to convert input DNA sequences into feature vectors. The fitness of each individual in the GP algorithm is measured as the classification accuracy obtained when applying an SVM with the kernel function and cost parameter C in the individual on the feature vectors. The data set with the input DNA sequences and methodology employed to measure classification accuracy are detailed below.

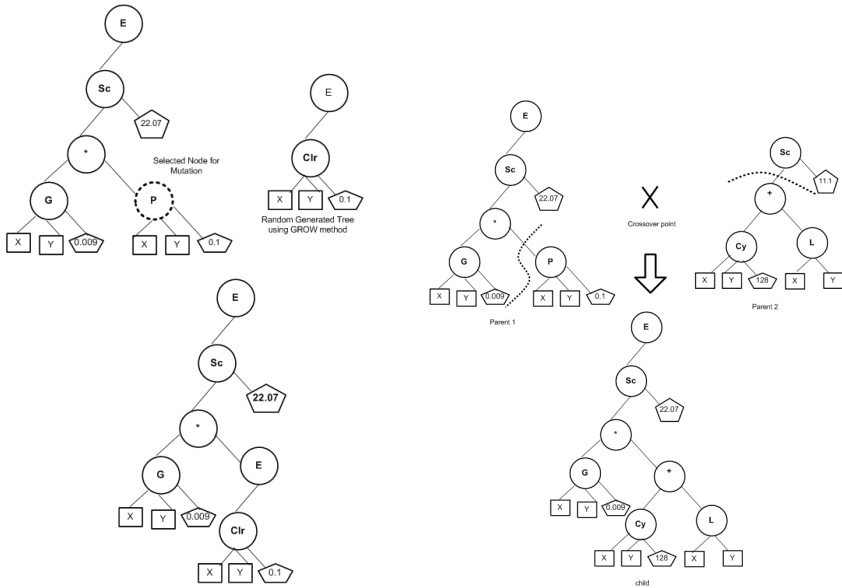


Fig. 1. Mutation (left) and crossover (right) are illustrated on sample kernel trees

Data Set. Input sequences are obtained from noble.gs.washington.edu/proj/hs. They consist of 280 HS sequences and 737 non-HS sequences experimentally obtained from throughout the human genome. The HS sequences were identified through a novel experimental methodology that employs cloning and in-vivo activity of K562 erythroid cells [35]. The non-HS sequences were not hypersensitive when tested in the same cell type. HS and non-HS sequences have similar average lengths of 242 nucleotides.

Test Methodology. The performance of the SVM is tested via 10-fold cross-validation on the total set of 1017 sequences (280 HS and 737 non-HS). The set is randomly divided into 10 subsets of equal size. The SVM is trained on 90% of the subsets and tested on the 10% held out. This is referred to as 10-fold validation. The area under the ROC curve is reported as an average over the 10-fold validations.

Implementation Details. The method is implemented in Java and run on an Intel Core2 Duo machine with 4GB RAM and 2.66GHz CPU. The EA implementation builds upon the GP implementation of ECJ software [27]. We use the open-source Biojava project [17] to integrate bioinformatics utilities for genomic sequences and the libSVM package [10] for the SVM implementation.

3 Results

The EA is run 30 different times to obtain 30 different sets of top motifs. In each case, the resulting feature vectors are tested with the SVM cross-validation test, employing a default kernel and default parameters. The kernel used is the radial basis function shown to outperform other predefined kernels in our recent work [21]. The average accuracy obtained with this kernel over 30 different runs, each of which results in a different set of 200 top-scoring motifs, is 82.9, with a standard deviation 1.1. The maximum and minimum accuracies obtained over these runs are 77.1 and 85.15, respectively.

Table 4 shows how the classification accuracy changes when the kernel parameters are tuned with the grid search technique in [10]. Only three sets of feature vectors are tested, the worst, average, and best features, respectively associated with the minimum, average, and maximum accuracies obtained with the default RBF kernel. Reported values are averaged over the 10-fold cross-validation.

Table 4. Accuracies obtained with the tuned RBF kernel when employing worst, average, and best features

	Worst Features	Average Features	Best Features
Tuned Kernel	79.30	83.78	85.39

Table 5 shows the classification accuracies obtained when the tuned RBF kernel is replaced with the top kernel function and parameter values obtained by our GP algorithm. The values reported in Table 5 are averaged over the top kernel and parameters obtained from 30 different runs of our GP algorithm. Standard deviations vary from 0.2 to 0.4.

Table 5. Accuracies obtained when replacing the tuned RBF kernel with the evolved kernel and parameters reported by the GP algorithm

	Worst Features	Average Features	Best Features
Evolved Kernel	82.17	85.97	87.21

Finally, Table 6 shows some of the top individuals (kernel and cost parameters C) obtained. The fitness of each individual (SVM classification accuracy on best features vectors) is shown in the last column.

Table 6. Fitness values are shown for some of the top individuals (kernel function and cost parameter) obtained with our GP algorithm

Evolved Kernel	Cost Parameter	Fitness
$(* (CY(x)(y)(6.1394E - 4))(S(x)(y)(7.4104004E - 4)))$	17.85	87.35
$(CLR(x)(y)(10.746466))$	1.18	86.85
$(CY(x)(y)(2.3240509E - 4))$	1.68	86.81
$(Sc(SPL(x)(y)(56.92857))(48.207794))$	1.00	86.79
$(* (CY(x)(y)(30.50425))(SPL(x)(y)(92.285255)))$	1.24	86.77

4 Conclusions

Our results show that employing an EA algorithm to obtain meaningful features and a GP algorithm to obtain new better kernel functions and values for the SVM cost parameter C significantly improve the accuracy of an SVM classification for the HS recognition problem. Moreover, the employment of evolutionary computing automates important decisions in SVM, which often require previous expert knowledge or significant experimentation.

Some of the top kernel functions obtained by our GP algorithm are a combination of operators, such as exponential, multiplication, or scalar over predefined kernels, such as Gaussian, Sigmoid, and Linear. The presence of kernel functions, such as Cauchy, Circular, Spiral, and Student-T among top-scoring kernels further justify our employment of an extensive list of kernel functions in our GP algorithm. We emphasize that our GP algorithm maintains closure and guarantees the optimality of obtained kernels.

While the fitness of an evolved kernel function is estimated in the context of the SVM classification, the fitness of the motifs sought to associate meaningful feature vectors with input sequences employs a simpler filter function in the interest of keeping computational cost low. Since the fitness of a kernel function is intimately dependent on the feature vectors employed in the SVM classification, our future work will consider integrated evolutionary schemes that evolve kernels and motifs in tandem. Co-evolution of features and kernels may further boost classification accuracies but at an expected computational cost. Distributed implementations will be sought to manage the computational cost.

Acknowledgments. We are indebted to Sean Luke and Keith Sullivan for insights on GP and Rezarta Dogan and Sarang Kayande for discussions on this work.

References

1. Blanchette, M., Bataille, A.R., Chen, X., Poitras, C., Laganier, J., Lefebvre, C., Deblois, G., Giguere, V., Ferretti, V., Bergeron, D., Coulombe, B., Robert, F.: Genome-wide computational prediction of transcriptional regulatory modules reveals new insights into human gene expression. *Genome Res.* 16(5), 656–668 (2006)

2. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Haussler, D. (ed.) 5th Annual ACM Workshop on COLT, pp. 144–152. ACM Press (1992)
3. Boughorbel, S., Tarel, J.-P., Boujemaa, N.: Conditionally positive definite kernels for svm based image recognition. In: Proceedings of IEEE International Conference on Multimedia and Expo (ICME 2005), Amsterdam, The Netherlands (2005), <http://perso.lcpc.fr/tarel.jean-philippe/publis/icme05.html>
4. Burgess-Beusse, B., Farrell, C., Gaszner, M., Litt, M., Mutskov, V., Recillas-Targa, F., Simpson, M., West, A., Felsenfeld, G.: The insulation of genes from external enhancers and silencing chromatin. *Proc. Natl. Acad. Sci. USA* 99(S4), 16433–16437 (2002)
5. I. Committee. Nomenclature committee of the international union of biochemistry (nc-iub). nomenclature for incompletely specified bases in nucleic acid sequences. recommendations 1984. *Biochemistry* 229(2), 75–88 (1985)
6. Consortium IHGS. Finishing the euchromatic sequence of the human genome. *Nature* 431(7011), 931–945 (2004)
7. De Jong, K.A.: *Evolutionary computation: a unified approach*. MIT Press, Cambridge (2001)
8. de Souza, B.F., de Carvalho, A.C., Calvo, R., Ishii, R.P.: Multiclass svm model selection using particle swarm optimization. In: Sixth International Conference on Hybrid Intelligent Systems (2006)
9. Dorschner, M.O., Hawrylycz, M., Humbert, R., Wallace, J.C., Shafer, A., Kawamoto, J., Mack, J., Hall, R., Goldy, J., Sabo, P.J., Kohli, A., Li, Q., McArthur, M., Stamatoyannopoulos, J.A.: High-throughput localization of functional elements by quantitative chromatin profiling. *Nat. Methods* 1(3), 219–225 (2004)
10. Fan, R.-E., Chen, P.-H., Lin, C.-J.: Working set selection using the second order information for training SVM. *J. Mach. Learn. Res.* 6(1532-4435), 1889–1918 (2005)
11. Friedrichs, F., Igel, C.: Evolutionary tuning of multiple svm parameters. In: 12th European Symposium on Artificial Neural Networks (ESANN 2004), pp. 519–524 (2004)
12. Gagné, C., Schoenauer, M., Sebag, M., Tomassini, M.: Genetic Programming for Kernel-Based Learning with Co-evolving Subsets Selection. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 1008–1017. Springer, Heidelberg (2006)
13. Gross, D.S., Garrard, W.T.: Nuclear hypersensitive sites in chromatin. *Annu. Rev. Biochem.* 57, 159–197 (1988)
14. Habib, T., Zhang, C., Yang, J.Y., Yang, M.Q., Deng, Y.: Supervised learning method for the prediction of subcellular localization of proteins using amino acid and amino acid pair composition. *BMC Genom.* 9(suppl. 1), S1–S16 (2008)
15. Higgs, D.R., Vernimmen, D., Hughes, J., Gibbons, R.: Using genomics to study how chromatin influences gene expression. *Annu. Rev. Genom. Human Genet.* 8, 299–325 (2007)
16. Hofmann, T., Schölkopf, B., Smola, A.: Kernel methods in machine learning. *The Annals of Statistics* 36(3), 1171–1220 (2008)
17. Holland, R.C., Down, T.A., Pocock, M., Prlic, A., Huen, D., James, K., Foisy, S., Draeger, A., Yates, A., Heuer, M., Schreiber, M.J.: BioJava: an open-source framework for bioinformatics. *Bioinformatics* 24(18), 2096–2097 (2008)
18. Huang, C.-L., Wang, C.-J.: A ga-based feature selection and parameter optimization for support vector machines. *Expert Systems with Applications*, 231–240 (2006)

19. Islamaj-Dogan, R., Getoor, L., Wilbur, W.J.: A feature generation algorithm with applications to biological sequence classification. In: Liu, H., Motoda, H. (eds.) *Computational Methods of Feature Selection*. Springer, Berlin (2007)
20. Islamaj-Dogan, R., Getoor, L., Wilbur, W.J., Mount, S.M.: Features generated for computational splice-site prediction correspond to functional elements. *BMC Bioinformatics* 8, 410–416 (2007)
21. Kamath, U., De Jong, K.A., Shehu, A.: Selecting predictive features for recognition of hypersensitive sites of regulatory genomic sequences with an evolutionary algorithm. In: *GECCO: Gen. Evol. Comp. Conf.*, pp. 179–186. ACM, New York (2010)
22. Kamath, U., Shehu, A., De Jong, K.A.: Using evolutionary computation to improve svm classification. In: *WCCI: IEEE World Conf. Comp. Intel.* IEEE Press (2010) (in press)
23. Koza, J.: *On the Programming of Computers by Means of Natural Selection*. MIT Press, Boston (1992)
24. Leslie, C., Kuang, R., Bennett, K.: Fast string kernels using inexact matching for protein sequences. *Journal of Machine Learning Research* 5, 1435–1455 (2004)
25. Leslie CS, N.W., Eskin E.: The spectrum kernel: a string kernel for svm protein classification. In: *Pacific Symposium on Biocomputing*, Baoding, China, vol. 7, pp. 564–575 (2002)
26. Lowrey, C.H., Bodine, D.M., Nienhuis, A.W.: Mechanism of DNase I hypersensitive site formation within the human globin locus control region. *Proc. Natl. Acad. Sci. USA* 89(3), 1143–1147 (1992)
27. Luke, S., Panait, L., Balan, G., Paus, S., Skolicki, Z., Popovici, E., Sullivan, K., Harrison, J., Bassett, J., Hubley, R., Chircop, A., Compton, J., Haddon, W., Donnelly, S., Jamil, B., OBeirne, J.: ECJ:Java-based evolutionary computation research (2010)
28. Staelin, C.: *Parameter Selection for Support Vector Machines*, Internal publication of HP Laboratories, Israel (approved for external publication) Technion City, Haifa, 32000. Israel Copyright Hewlett-Packard Company (2002), <http://www.hpl.hp.com/techreports/2002/HPL-2002-354R1.pdf>
29. Maston, G.A., Evans, S.K., Green, M.R.: Transcriptional regulatory elements in the human genome. *Annu. Rev. Genom. Human Genet.* 7, 29–59 (2006)
30. Mierswa, I.: Evolutionary learning with kernels: A generic solution for large margin problems. In: *GECCO: Gen. Evol. Comp. Conf.*, pp. 1553–1560 (2006)
31. Montana, D.J.: Strongly typed genetic programming. *Evolutionary Computation* 3(2), 199–230 (1993)
32. Noble, W.S.: Support vector machine applications in computational biology. In: Schölkopf, B., Tsuda, K., Vert, J.-P. (eds.) *Kernel Methods in Computational Biology*. MIT Press, Cambridge (2004)
33. Noble, W.S., Kuehn, S., Thurman, R., Yu, M., Stamatoyannopoulos, J.A.: Predicting the in vivo signature of human gene regulatory sequences. *Bioinformatics* 21(suppl. 1), i338–i343 (2005)
34. Phientrakul, T., Kijisirikul, B.: Evolutionary strategies for multi-scale radial basis function kernels in support vector machines. In: *Genetic and Evolutionary Computation Conference*, Washington D.C., USA, pp. 905–911 (2005)
35. Sabo, P.J., Humbert, R., Hawrylycz, M., Wallace, J.C., Dorschner, M.O., McArthur, M., Stamatoyannopoulos, J.A.: Genome-wide identification of DNase I hypersensitive sites using active chromatin sequence libraries. *Proc. Natl. Acad. Sci. USA* 101(13), 4537–4542 (2004)

36. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Boston (2002)
37. Shawe-Taylor, J., Cristianini, N.: Kernel methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
38. Sullivan, K., Luke, S.: Evolving kernels for support vector machine classification. In: Genetic and Evolutionary Computation Conference (2007)
39. Vapnik, V.N.: Statistical learning theory. Wiley & Sons, New York (1998)
40. Vertanen, K.: Genetic adventures in parallel: Towards a good island model under PVM (1998)
41. Wu, C.: The 5' ends of drosophila heat shock genes in chromatin are hypersensitive to DNase I. *Nature* 286(5776), 854–860 (1980)
42. Zhang, X.H., Heller, K.A., Hefter, I., Leslie, C.S., Chasin, L.A.: Sequence information for the splicing of human pre-mrna identified by support vector machine classification. *Genome Res.* 13(12), 2637–2650 (2003)
43. Zien, A., Raetsch, G., Mika, S., Schölkopf, B., Lengauer, T., Mueller, K.R.: Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics* 16(9), 799–807 (2000)

A Phenomic Algorithm for Inference of Gene Networks Using S-Systems and Memetic Search

Rio G.L. D'Souza¹, K. Chandra Sekaran², and A. Kandasamy²

¹ St Joseph Engineering College, Mangalore, India

² National Institute of Technology Karnataka, Surathkal, Mangalore, India
{Rio,kchnitk}@ieee.org, kandy@nitk.ac.in

Abstract. In recent years, evolutionary methods have seen unprecedented success in elucidation of gene networks, especially from microarray data. We have implemented the Phenomic Algorithm which is an evolutionary method for inference of gene networks based on population dynamics. We have used S-systems to model gene interactions and applied memetic search to fine tune the parameters of the inferred networks. We have tested the novel algorithm on artificial gene expression datasets obtained from simulated gene networks. We have also compared the results to those obtained from two other similar algorithms. Results showed that the new method, which we call as Phenomic Algorithm with Memetic Search (PAMS), is an effective method for inference of gene networks.

Keywords: Microarray data analysis, Gene networks, Evolutionary algorithms, S-systems, Memetic search, Phenomic algorithms.

1 Introduction

Ever since the advent microarray technology scientists have been able to study thousands of genes at a time, and this has helped them to analyze the relationships between them. Most microarray experiments result in large datasets which need to be analyzed in order to understand the underlying relationships. There is vast potential for methods that can yield useful patterns from such large datasets without compromising the dimensionality [1]. Gene networks represent relationships between genes, based on observations of how the expression level of each gene affects the expression levels of the others [2]. Several researchers have used evolutionary methods [3] to analyze the relationships between thousands of genes. The Phenomic Algorithm [4], [5] is an approach based on population dynamics. We have implemented the proposed algorithm and validated it on artificial gene network datasets.

The rest of this paper is organized as follows: In Section 2, we provide a review of similar work done by others. We introduce the models and also the basis of the methods that we employ in Section 3; and in Section 4 we discuss the results of our experiments. This is followed by Section 5 which concludes the paper.

2 Related Work

The Inference of gene networks from the ever-growing mass of microarray data has become an important research activity in Systems Biology. Among the initial attempts, Somogyi et al. [6] developed a simple method which inferred Boolean networks. Several gene network reconstruction algorithms have been studied by Akutsu et al. [7] and D’haeseleer et al. [8]. While some of these methods infer only qualitative relationships between genes, others which infer quantitative relationships are limited by the scale of networks that they can deduce.

Reliable inference of gene networks is dependent on how closely the chosen model represents the real gene networks. One such model, which is nonlinear and dynamic is the S-System proposed by Savageau [9]. Several researchers [10] have used this model to reverse engineer gene networks. Recently, Ahmed, Song and Xing [11] have used a variant of S-System to construct graphical models for inferring time-varying gene regulatory networks. Most problems in this field can be viewed as some type of optimization and multiobjective evolutionary algorithms (MOEAs) have found remarkable success in reconstructing gene networks from expression data [12].

3 Models and Methods

3.1 S-System Model of Gene Networks

To establish that a change in the expression of gene B was caused by a change in the expression of gene A, it is necessary to show that a dependency exists between the two genes, whereby gene B is dependent on gene A. The Power-law formalism called S-System, which was proposed by Savageau [9] is a nonlinear and dynamic model which we used to capture the relationships between genes.

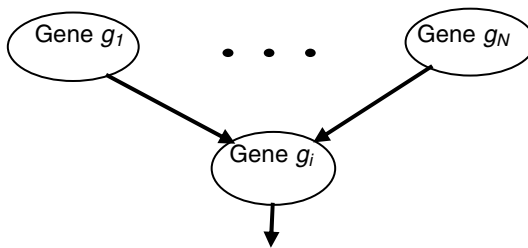


Fig. 1. Generalized gene network model

The behaviour of a cell can be abstracted by a gene regulatory network of N genes and other intermediate gene-products. Each gene g_i produces a certain amount of RNA x_i whenever it expresses. This causes a change in the concentration of this RNA over a time-period. This situation, shown in Fig. 1, can be represented in equation (1):

$$x(t + 1) = h(x(t)), \quad \text{where } x(t) = (x_1, x_2, \dots, x_N) . \quad (1)$$

The S-system model for this gene network can be described by a set of nonlinear differential equations in equation (2):

$$\frac{dx_i(t)}{dt} = \alpha_i \prod_{j=1}^N x_j(t)^{G_{i,j}} - \beta_i \prod_{j=1}^N x_j(t)^{H_{i,j}} . \tag{2}$$

Here $G_{i,j}$ and $H_{i,j}$ are kinetic exponents and α_i and β_i are positive rate constants. Ignoring genes that do not influence gene g_i , applying the rule of finite differences and rearranging equation (2), the calculated value of gene expression level for gene g_i , $x_{i,cal,t+1}$ can be written as in equation (3):

$$x_{i,cal,t+1} = x_i(t+1) = x_i(t) + \left\{ \alpha_i \prod_{j \in S_i} x_j(t)^{G_{i,j}} - \beta_i \prod_{j \in S_i} x_j(t)^{H_{i,j}} \right\} \Delta t . \tag{3}$$

After each evaluation of $x_{i,cal,t+1}$, the error at that time-step between the calculated and observed gene expression level is given in equation (4) as:

$$e_{i,t} = x_{i,cal,t} - x_{i,exp,t} . \tag{4}$$

The link that is being verified is retained if the error is less than E (which is the maximum error allowed at that point in the inference algorithm), otherwise the link is not retained. The kinetic exponents and rate constants are optimized separately through a memetic search mechanism.

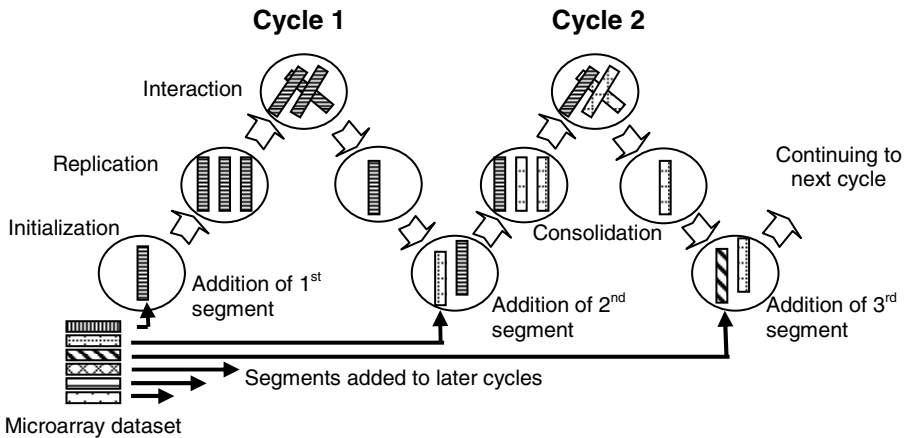


Fig. 2. First two cycles of processing in the phenomic algorithm

3.2 The Phenomic Approach

In order to elucidate gene-to-gene relationships from expression measurements taken from microarray data, the expression patterns of thousands of pairs of genes need to be compared. Carrying out this huge number of comparisons in a coordinated manner,

such that all relationships of interest are discovered, is a challenging task. Modeled carefully gene expression data could be used to characterize the relationship between the phenotype and the genotype.

In this approach, as seen in Fig. 2, the individuals interact and it is during these interactions that the relationships between genes are captured. A link is set up between two genes that are causally related. After a predetermined number of randomized interaction cycles, the population is consolidated to remove replicated individuals. The links captured by dying individuals are carried over to the survivors.

3.3 Memetic Search

In equation (3), the set of kinetic rate constants and exponents $\{\alpha_i, \beta_i, G_{i,j}, H_{i,j}\}$ determine the change in the expression level of gene g_i at any time t , due to the change in expression levels of all genes at that time. The parameters associated with a gene network are optimized by the memetic search procedure, as given below:

The memetic search procedure.

```

Procedure MemeticSearch(geneNetwork, kineticString)
begin
  t := 0;
  P(t) := initPop(kineticString);
  similarity := 0.0;
  while (similarity < 0.6)
    P'(t) := generateOffspringPop(P(t));
    P(t+1) := retainBetterString(P(t), P'(t));
    similarity := evaluateSimilarity(P(t+1));
    t := t+1;
  endwhile
  kineticString := findBest(P(t));
end

```

3.4 The Phenomic Algorithm with Memetic Search (PAMS)

Like most evolutionary algorithms, the phenomic algorithm with memetic search starts with a population of n individuals each of which embeds the expression profile of one gene taken from the microarray data. The algorithm thereafter goes into the evolutionary cycle which starts with the first generation of phenotypic processing. During phenotypic processing the triplets of individuals (or genes) are allowed to meet in the interaction phase. The possibility of causal relationship between them is verified by using equation (4). The memetic search procedure is used at this stage to fine-tune the network parameters.

The process is repeated starting with the interaction phase, until all microarray data segments are processed. The gene networks at this stage are output as the optimal

networks that represent the given microarray data. The pseudo-code of PAMS is given below:

The phenomic algorithm with memetic search (PAMS) and its main functions.

Algorithm PAMS

begin

$t := 0;$

segments[] := divideMicroarrayData();

$P(t) := \text{replicateSeg}(\text{segments}[t]);$

while ($t < n$)

$P'(t) := \text{interactPop}(P(t));$

$P''(t) := \text{consolidatePop}(P'(t));$

$P(t+1) := P''(t) + \text{replicateSeg}(\text{segments}[t+1]);$

$t := t+1;$

endwhile

geneLinks := readLinks($P(t)$);

displayNetworks(geneLinks);

end

In the next section, PAMS is validated and compared with other extant multiobjective algorithms for inferring gene networks.

4 Results and Discussion

The validation of PAMS was done by using datasets derived from artificial gene networks developed by researchers at the Virginia Bioinformatics Institute and downloadable from their website [13]. Two datasets were selected: one is called Century and the other is called Jumbo. A typical gene network inferred when running PAMS on the Century dataset is shown in Fig. 3.

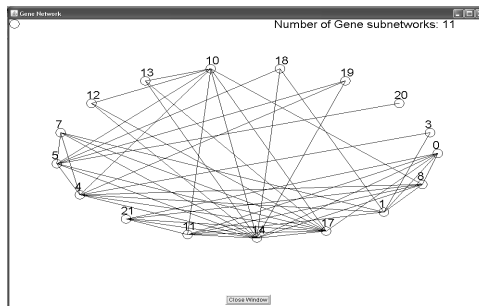


Fig. 3. A gene network inferred by PAMS when $E = 0.05$

We now define and use three metrics as the basis for comparison of the three methods. The first metric, NOL, is defined as in equation (5). The second metric, SWSF, measures the closeness of the inferred networks to small-world networks and is defined in equation (6).

$$NOL = \sum_{i=1}^N \sum_{j=1}^N l_{ij} \quad (5)$$

$$SWSF = \frac{1}{C} \sum_{k=1}^C k n_k \quad (6)$$

Here, $l_{ij} = 1$ if gene g_i is linked to gene g_j , else $l_{ij} = 0$ (taken from the adjacency matrix of the network), N is the total number of genes in the target network, n_k is the number of nodes with out-degree of k , and C is the maximum degree of the network.

The third metric, GED, is the minimal number of edit operations (edge insertions/edge removals) that transform one graph into another one [14]. In our case, we use the formula in equation (7) to calculate this distance.

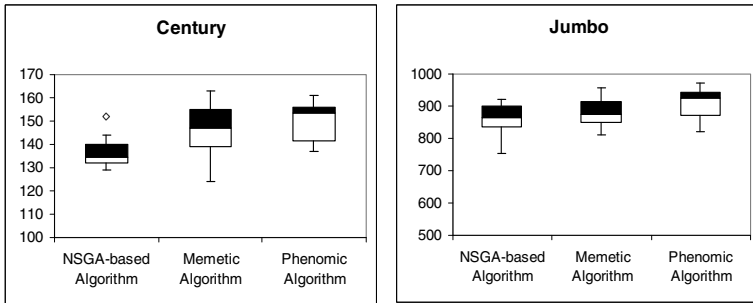


Fig. 4. Boxplots of NOL obtained with Century and Jumbo gene network datasets

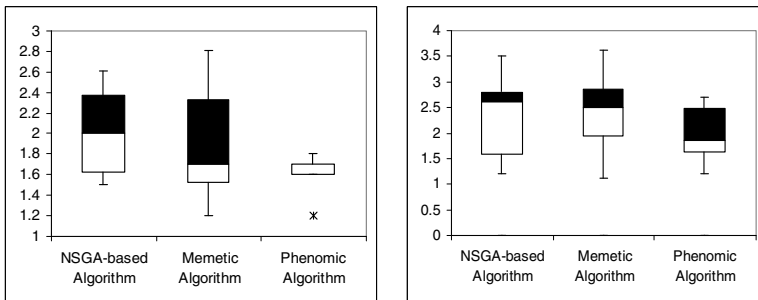


Fig. 5. Boxplots of SWSF obtained with Century and Jumbo gene network datasets

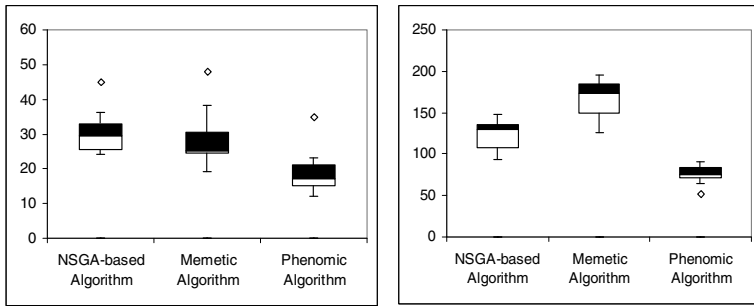


Fig. 6. Boxplots of GED obtained with Century and Jumbo gene network datasets

$$GED = \sum_{i=1}^N \sum_{j=1}^N D_{ij} \begin{cases} D_{ij} = 1, \text{ if } l_{ij} \neq m_{ij} \\ D_{ij} = 0, \text{ if } l_{ij} = m_{ij} \end{cases} \quad (7)$$

Here D_{ij} is the distance between corresponding nodes of the two networks that are being compared; l_{ij} and m_{ij} are elements of the adjacency matrices of the inferred gene network and source artificial gene network, respectively.

The phenomic algorithm (PAMS) is compared with two other evolutionary algorithms that are used for inferring gene networks. The first of these algorithms is based on Non-dominated Sorting Genetic Algorithm (NSGA-based) which was proposed by Deb et al. [15], combined with a gene network inference algorithm as implemented by Spieth et al. [16]. The second algorithm is a Memetic Algorithm developed by Spieth et al. [17]. For further details readers are referred to the original papers. The descriptive statistics of the three metrics over ten runs of each of the above methods are shown as boxplots in Fig. 4, Fig. 5 and Fig. 6.

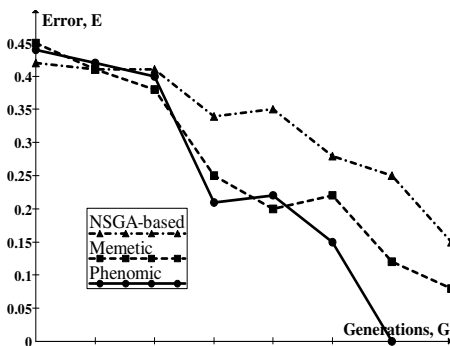


Fig. 7. Variation of error observed against generations

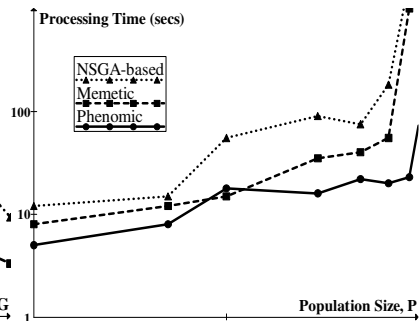


Fig. 8. Variation of processing time against population size

The boxplots in Fig. 4, Fig. 5 and Fig. 6 show that PAMS infers qualitatively better networks than both the NSGA-based and the Memetic algorithm. This is achieved without compromising on the runtime performance as shown in Fig. 7 and Fig 8. In terms of error E, as shown in Fig. 7, the phenomic algorithm achieves zero error after about 6000 generations. All values plotted are the average of 10 runs. The processing time taken by PAMS for processing even large populations (approaching 1000 individuals), is less than a minute.

5 Conclusion

The problem of inferring gene networks is getting increased attention from researchers worldwide and many new methods are introduced each year. Multiobjective evolutionary algorithms have been yielding consistently good results [12] and the phenomic algorithm adds to this successful trend. The benefits of local search through memetic mechanisms are also evident in the results described earlier.

In future work, it is planned to apply PAMS to natural gene expression datasets. The robustness of the novel method to noise, which is a frequent spoiler in natural datasets, will be under scrutiny in such experiments.

References

1. Schulze, A., Downward, J.: Navigating Gene Expression Using Microarrays – A Technology Review. *Nature Cell Biology* 3, E190–E195 (2001)
2. Soinov, L.A., Krestyaninova, M.A., Brazma, A.: Towards Reconstruction of Gene Networks from Expression Data by Supervised Learning. *Genome Biology* 4(1), R6 (2003)
3. D'haeseleer, P., Liang, S., Somogyi, R.: Gene Expression Analysis and Genetic Network Modelling: Tutorial. In: *Pacific Symposium on Biocomputing* (1999)
4. D'Souza, R.G.L., Chandra Sekaran, K., Kandasamy, A.: A Phenomic Algorithm for Reconstruction of Gene Networks. In: *IV International Conference on Computational Intelligence and Cognitive Informatics, CICI 2007*, pp. 53–58. WASET, Venice (2007)
5. D'Souza, R.G.L., Chandra Sekaran, K., Kandasamy, A.: Reconstruction of Gene Networks using Phenomic Algorithms. *Intl. Jour. of Artificial Intell. and Appl.* 1(2) (2010)
6. Somogyi, R., Fuhrman, S., Askenazi, M., Wuensche, A.: The Gene Expression Matrix: Towards the Extraction of Genetic Network Architectures. In: *Proc. of Second World Cong. of Nonlinear Analysts (WCNA 1996)*, vol. 30(3), pp. 1815–1824 (1997)
7. Akutsu, T., Miyano, S., Kuhara, S.: Identification of Genetic Networks from a Small Number of Gene Expression Patterns under the Boolean Network Model. In: *Pacific Symp. on Biocomputing*, vol. 4, pp. 17–28 (1999)
8. D'haeseleer, P., Liang, S., Somogyi, R.: Genetic Network Inference: From Co-Expression Clustering to Reverse Engineering. *Bioinformatics* 16(8), 707–726 (2000)
9. Savageau, M.A.: Power-law Formalism: A Canonical Nonlinear Approach to Modelling and Analysis. In: *Proc. of the World Congress of Nonlinear Analysts 1992*, pp. 3323–3334 (1995)

10. Spieth, C., Streichert, F., Speer, N., Zell, A.: Optimizing Topology and Parameters of Gene Regulatory Network Models from Time-Series Experiments. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 461–470. Springer, Heidelberg (2004)
11. Ahmed, A., Song, L., Xing, E.: Time-Varying Networks: Reconstructing Temporally Rewiring Genetic Interactions during the Life Cycle of *Drosophila melanogaster*. CMU-MLD Technical Report CMU-ML-08-118 (2008)
12. Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. *Evolutionary Computation* 8(2), 125–147 (2000)
13. A Collection of Artificial Gene Networks (2010), <http://www.comp-sys-bio.org/AGN/data.html> (accessed January 20, 2010)
14. Supper, J., Fröhlich, H., Spieth, C., Dräger, A., Zell, A.: Inferring Gene Regulatory Networks by Machine Learning Methods. In: APBC 2007, pp. 247–256 (2007)
15. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. *IEEE Trans. of Evol. Comp.* 6(2), 182–197 (2002)
16. Spieth, C., Streichert, F., Speer, N., Zell, A.: Multi-objective Model Optimization for Inferring Gene Regulatory Networks. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 607–620. Springer, Heidelberg (2005)
17. Spieth, C., Streichert, F., Speer, N., Zell, A.: A Memetic Inference Method for Gene Regulatory Networks Based on S-Systems. In: Congress on Evol. Comp (CEC 2004), Proc. Part I, pp. 152–157. IEEE Press (2004)

Fluctuation-Driven Adaptation and Symbiosis in Cellular Dynamics

Chikara Furusawa, Kota Ijichi, and Hiroshi Shimizu

Graduate School of Information Science and Technology, Osaka University, 1-5
Yamadaoka, Suita, Osaka 565-0871, Japan

furusawa@ist.osaka-u.ac.jp

<http://www-shimizu.ist.osaka-u.ac.jp/furusawa>

Abstract. Biological systems can generally adapt environmental changes and to create symbiotic relationship with other species, by changing their intra-cellular states flexibly. However, the mechanisms for such flexible adaptation and creation of symbiotic relationship remain unclear. In this study, by using simple computer models of cells, we show that for cells whose gene expression fluctuate stochastically, the adaptive cellular state is inevitably selected by noise, even without sophisticated mechanisms. Furthermore, by the fluctuation-induced adaptation mechanism, we show that symbiotic relationships naturally appear in systems of interacting cells. This mechanism can provide clues to understand flexible adaptation and creation of symbiotic relationship. Applications of this mechanism for designing artificial systems are also discussed.

Keywords: fluctuation, attractor, adaptation, symbiosis.

1 Introduction

Cells adapt to a variety of environmental conditions by changing the pattern of gene expression and metabolic flux distribution. Furthermore, the flexible changes in intra-cellular states of cells make it possible to create symbiotic relationship between different species through cell-cell interactions. Although these adaptation and creation of symbiotic relationship are ubiquitous in nature, the mechanisms for the flexible changes in intra-cellular state are not yet fully understood. In contemporary molecular biology, these adaptive responses are generally explained by signal transduction mechanisms, where external events are interpreted by gene regulatory networks. For example, the Lac operon of *Escherichia coli* encodes proteins involved in lactose metabolism, and expression of the operon is controlled by a regulatory protein so that, when lactose is available, these proteins are expressed in an efficient and coordinated manner [1]. However, such program-like descriptions of adaptive response are not always able to explain the flexible adaptive behavior and creation of symbiotic relationship, since such program-like responses require evolution of regulatory network, and the program for adaptation to novel environmental changes and creation of symbiotic relationship that the species has not experienced cannot be programmed in advance through evolutionary process.

Furthermore, a recent study indicated the possibility that cells can respond to environmental changes adaptively without pre-programmed signal transduction mechanisms. Kashiwagi et al. demonstrated that *E. coli* cells select an appropriate intra-cellular state according to environmental conditions without the help of signal transduction mechanisms [2]. There, an artificial gene network composed of two mutually inhibitory operons was introduced into *E. coli* cells, so that states of gene expression are bistable. These authors found that the cells shift to the adaptive cellular state by expressing the gene required to survive in the environment. They also demonstrated that the selection of the adaptive attractor between bistable states by noise is possible by introducing phenomenological activity that governs the synthesis and degradation of protein.

In our previous study [3], using an abstract cell model we demonstrated that cells can select states most favorable for their survival among a large number of other possible states as an inevitable outcome. By studying a model that consists of a protein regulatory network and a metabolic reaction network, we showed that cellular states with high growth rates are selected among a huge number of possible cellular states, and this selection is only mediated by fluctuations of gene expressions [4,5,6]. This selection of a higher growth state is theoretically explained by noting that a state with lower growth speed is more influenced by stochasticity in gene expression, so that it is easily kicked away triggering a switch to a state with a higher growth rate. We showed that there is generally a negative correlation between the rate of noise-driven escape from a given state and the cellular growth rate. Due to this negative correlation, an optimal growth state is selected spontaneously. The results indicated the possibility that cells can respond to environmental changes adaptively without finely-tuned preprogrammed signal transduction mechanisms.

In this study, we analyze the possibility that this fluctuation-driven flexible adaptation mechanism can explain flexible formation of symbiotic relationship among cells. We construct a model of interacting cells in which each cell has gene expression dynamics with stochastic fluctuation within. Using this model of interacting cells, we show that the cell-cell interactions can cause mutually adaptive response of cells by fluctuation of gene expression, that eventually results the formation of symbiotic relationship. In the first part of this paper, we briefly present the mechanism of fluctuation-driven adaptation using a single cell model. In the latter part, we show the result of the interacting cell model to show the mechanisms of emergence of symbiotic relationship. The application of the noise-based mechanisms for adaptation and formation of symbiotic relationship are also discussed.

2 Fluctuation-Driven Adaptation

2.1 Cell Model

A schematic representation of the single cell model is shown in Fig.1. The cell includes two networks, i.e., a gene regulatory network which controls expression levels of proteins through each other, and a metabolic reaction network whose

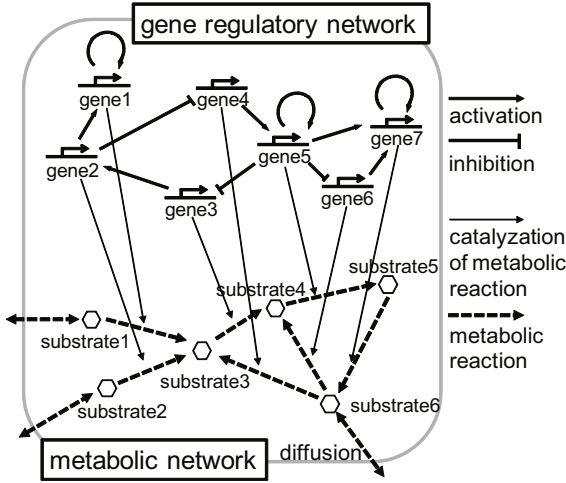


Fig. 1. A cell model with gene-metabolic networks

fluxes are regulated by the expression levels of the proteins. The internal state of a cell is represented by a set of expression levels of n proteins (x_1, x_2, \dots, x_n) and concentrations of m metabolic substrates (y_1, y_2, \dots, y_m). The time development of protein expression is determined by (i) the synthesis of proteins, (ii) the dilution of proteins by the growth in cell volume, and (iii) fluctuations in protein expressions arising from stochasticity in chemical reactions. The dilution of proteins is proportional to the growth rate of cell volume v_g , which is determined by the metabolic fluxes. Also, it is natural to assume that the rates of protein synthesis are proportional to the growth rate v_g , since the decrease in protein concentration by dilution due to the cell growth has to be compensated by synthesis to maintain a steady state. In fact, some experimental studies showed that the total protein concentration is relatively unchanged with the growth rate [7], which suggests that the change of protein dilution rate was compensated by changing protein synthesis rate. The adaptation mechanism presented below works, even if the rigorous proportionality of protein synthesis and dilution rate to the growth rate is replaced by just a positive correlation between the synthesis rate and the cell volume growth rate. Following this argument, the dynamics of concentration of the i -th protein is chosen as follows:

$$\frac{dx_i(t)}{dt} = f\left(\sum_{j=1}^n W_{ij}x_j(t) - \theta\right)v_g(t) - x_i(t)v_g(t) + \eta(t) \quad (1)$$

The first and second terms in r.h.s. represent synthesis, dilution of the protein i , respectively. In the first term, the regulation of protein expression levels by other proteins are indicated by regulatory matrix W_{ij} , which takes 1, 0, or -1 representing activation, no regulatory interaction, and inhibition of the i -th protein expression by the j -th protein, respectively. The synthesis of proteins is given by the sigmoidal regulation function $f(z) = 1/(1 + \exp(-\mu z))$, where

$z = (\sum W_{ij}x_j(t) - \theta)$ is the total regulatory input with the threshold θ for activation of synthesis, and μ indicates gain parameter of the sigmoid function. The regulatory interactions are determined randomly with the rate ρ_a, ρ_i , indicating the connection rate of excitatory paths and inhibitory paths, respectively. The third term represents the noise in protein concentration with a certain amplitude σ satisfying $\langle \eta_i(t)\eta_j(t') \rangle = \delta(t - t')\delta_{ij}$, where i and j represent different proteins. For simplification, we assume that the amplitude of the noise is independent of the growth rate v_g , whereas the inclusion of v_g dependence does not alter our results qualitatively

Temporal changes in concentrations of metabolic substrates are given by metabolic reactions and transportation of substrates from the outside of the cell. Each metabolic reaction is catalyzed by a corresponding protein. Some nutrient substrates are supplied from the environment by diffusion through the cell membrane, to ensure the growth of a cell. Here, the dynamics of i -th substrate concentration y_i is represented as:

$$\frac{dy_i}{dt} = \epsilon \sum_{j=1}^n \sum_{k=1}^m \text{Con}(k, j, i)x_j y_k - \epsilon \sum_{j'=1}^n \sum_{k'=1}^m \text{Con}(i, j', k')x_{j'} y_i + D(Y_i - y_i) \quad (2)$$

where ϵ indicates the coefficient for the metabolic reactions, and $\text{Con}(i, j, k)$ represents the reaction matrix of the metabolic network, which takes 1 if there is a metabolic reaction from i -th substrate to k -th substrate catalyzed by j -th protein, and 0 otherwise. The first and second terms of r.h.s. correspond to synthesis and consumption of i -th substrate by metabolic reactions, respectively. The third term of r.h.s. represents the transportation of the substrate through the cell membrane, which is approximated by the linear term in the diffusion process with a diffusion coefficient D . Y_i is a constant representing the concentration of i -th substrate in the environment. The concentration Y_i is nonzero only for nutrient substrates.

The cellular growth rate v_g is determined by the dynamics in the metabolic reactions. We assume that some of metabolic substrates are necessary for cellular growth, and the growth rate v_g is determined as a function of the concentrations of them. Several choices of the function are possible, and the results to be discussed are generally observed as long as the growth rate varies drastically depending on the concentrations. Here we assume that the growth rate is proportional to the minimal concentration among these necessary substrates. In other words, among m metabolic substrates there are r substrates (y_1, y_2, \dots, y_r) required for cellular growth, and the growth rate is represented as $v_g \propto \min(y_1, y_2, \dots, y_r)$.

We carried out numerical experiments with the model using several sets of parameter values obeying the above constraints that allows for multiple attractors, and evaluated thousand of different randomly generated reaction networks. We found that the adaptation processes triggered by noise shown below are generally observed, as long as the intra-cellular dynamics has multiple attractors. In the next section, we present the typical behaviors obtained by using networks consisting of $n \sim 96$ proteins and $m \sim 32$ metabolic substrates.

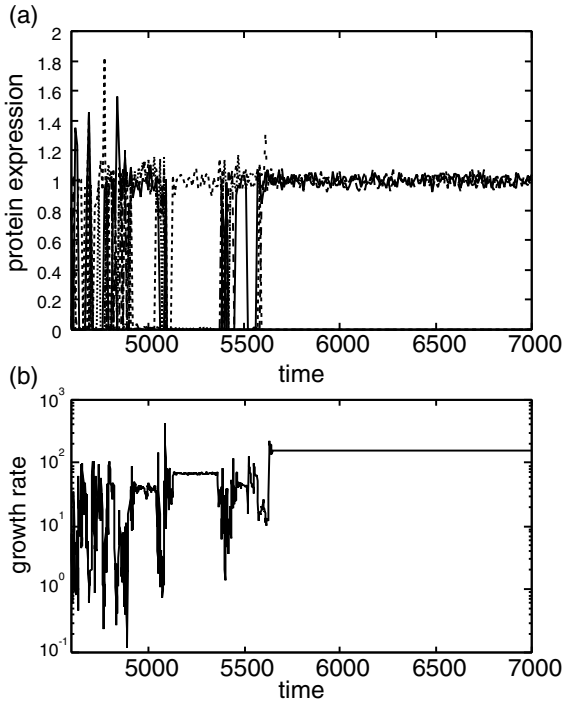


Fig. 2. Time series of protein expressions and growth rate. Randomly generated gene regulatory network and metabolic network with $n=96$ and $m=32$ were used for the simulation. (a) Time series of protein expressions. The vertical axis show the expression levels of proteins, and the horizontal axis represents time. Six out of 96 protein species are displayed. (b) Time series of growth rate observed during the time interval shown in (a). Initially, the growth rate of the cell is relatively low and it fluctuates due to the highly stochastic time course of protein expressions. After a few short-lived nearly optimal states (c.f. 4800 and 5600 time steps), the cell finds a state of protein expression that realize a relatively high growth rate. The parameters are $\theta=0.5$, $\mu=10$, $\epsilon=0.1$, $D=0.1$. In addition, we enhanced the rate of positive autoregulatory paths, so that the regulatory network has multiple attractors. In the simulations, 30% of activating paths are chosen as autoregulatory paths.

2.2 Simulation Results

In Fig.2, an example of the selection process of rapidly growing states, starting from randomly chosen initial expression state, is shown by taking an adequate noise amplitude in expression dynamics. Time series of concentrations of arbitrarily chosen proteins and growth rate of the cell v_g are plotted in 2(a) and 2(b), respectively. In the example, cells are set initially at a state with a low growth rate. In such a state, stochasticity dominates the evolution of protein concentrations with time. After itinerating among various expression patterns, the cellular dynamics arrive at a state with a higher growth rate. Such a transition repeats

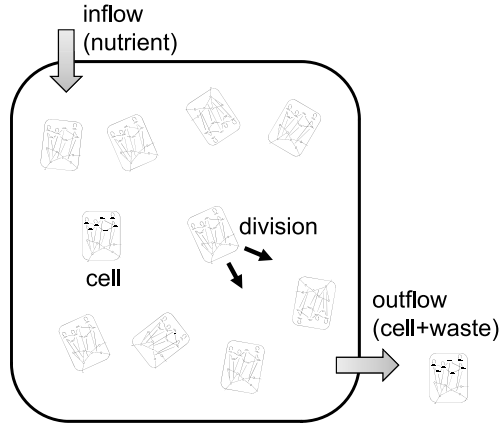


Fig. 3. A model of interacting cells. Each cell has gene-metabolic networks as shown in Fig.1. The cells interact with each other through the transport of metabolites into and out of the surrounding medium. A certain amount of fresh medium containing nutrient metabolites is continuously supplied from outside the environment, and the same amount of medium with cells is discarded to keep the total amount of medium constant. A cell divides in a time interval which is inversely proportional to the growth rate.

until the growth rate becomes sufficiently high. Once a gene expression pattern supporting the optimal growth is reached, the system maintains it over time.

This selection of higher growth states was observed for all of the one thousand networks we simulated. It also worked independently of initial conditions. Note that once one of the expression patterns is selected as an attractor, the flux pattern on the metabolic network is uniquely determined. As a result, the cellular growth rate v_g is also fixed, which in turn affects the protein expression dynamics. Here the influence of noise depends on the growth rate v_g for each attractor. When v_g is small, the deterministic part of protein expression dynamics (i.e., the first and second terms of r.h.s. in eq.(1)) is small, so that the stochastic part in the dynamics is relatively dominant in the protein expression dynamics. Then, the probability to escape the attractor due to fluctuation is large. In contrast, when the growth rate v_g is large in the attractor, the magnitude of the deterministic part of expression dynamics is larger than that of the stochastic part. As a result, the probability to escape the state becomes small.

3 Emergence of Symbiotic Relationship by Cell-Cell Interactions

In the previous section, we show that cells can change their internal state to achieve higher growth rate by stochastic fluctuation, even without sophisticated signal transduction machinery. The next question addressed here is what happens when cells with such fluctuation-driven adaptation mechanism interact each

other? We can expect that, cells adapt to environment that is maintained by other cells, which can result mutual adaptation and formation of symbiotic relationship without finely-tuned preprogram. To study this fluctuation-driven symbiotic formation, we constructed an interacting cell model in which each cell has internal expression dynamics with fluctuation as in the previous section and they interact each other through environment.

3.1 Interacting Cell Model

A schematic representation of the interacting cell model is shown in Fig.3. Each cell has gene regulatory network and metabolic network, which are identical to those in the previous section. We assume that some metabolites can penetrate the cell membrane, and each cell communicates with its environment through the transport of metabolic substrates. Thus, interactions between cells occur throughout the environment. Also, we assume that the medium is well stirred by neglecting the spatial variation of the concentration, so that all cells interact with each other through an identical environment. In this model, we consider only diffusion processes through the cell membrane. Thus, the rates of chemicals transported into a cell are proportional to differences of chemical concentrations between the inside and the outside of the cell. The cells are cultured in a tank with a fixed volume, where fresh medium containing nutrient metabolite are continuously supplied and the same amount of medium including the cells are discarded. Thus, when the number of cells in the medium is q , the concentration of i -th substrate in the medium Y_i obeys the following differential equation:

$$\frac{dY_i}{dt} = - \sum_{j=1}^q D(Y_i - y_i^j) + \hat{D}(\hat{Y}_i - Y_i) \quad (3)$$

where y_i^j represents the concentration of i -th substrate in j -th cell. The first term in r.h.s. represents consumption and production of i -th substrate by the cells. The second term represents the flow of the substrate from/to the environment, in which \hat{D} and \hat{Y}_i are constants. As an initial condition, we take a single cell, with randomly chosen expression pattern. The number of cells increases due to cell divisions, where the doubling time of a cell is inversely proportional to its growth rate v_g . As a result of increase of cell number, the concentrations of nutrients in the tank decrease, which result a decrease of the growth rate v_g .

3.2 Simulation Results

In this interacting cell system, cells can adapt the dynamically changing environmental condition by the fluctuation-driven adaptation mechanism discussed in the previous section. That is, the transition between attractor is driven by the fluctuation in the expression dynamics, only when the growth rate v_g is small. In Fig.4, we show a typical example of such adaptation process of interacting cells, in which time evolutions of the number of cells in different attractors are

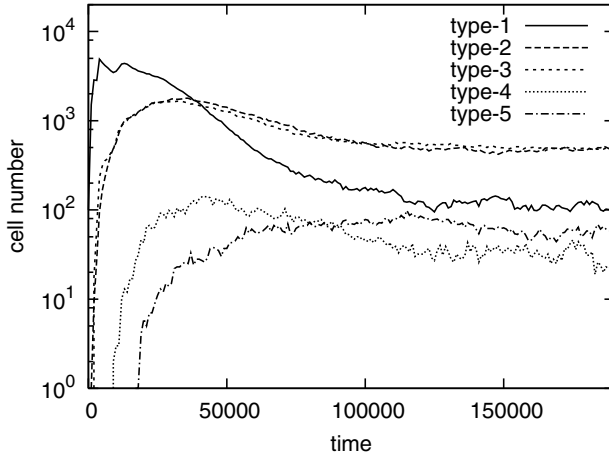


Fig. 4. The emergence of symbiotic system driven by noise in cellular dynamics. Each line represents the number of cells in each cell type. Here, the cell type is defined as a cellular state which falls into a certain attractor. The noise amplitude $\sigma=0.1$.

plotted. In this simulation, the amplitude of noise $\sigma = 0.1$, while other parameters of intra-cellular dynamics are identical to those used in Fig.2. In this example, the initial cell falls into a fixed cellular state (denoted by “type-1” in Fig.4), and the number of cells having type-1 state increases by cell divisions. As the result of increase of type-1 cells, the nutrients in the environment which are required for the growth of type-1 cell are consumed and the concentrations of these nutrients decrease. Then, the growth rate v_g of type-1 cells decrease, and the stochastic fluctuation starts to dominate the cellular dynamics of type-1 cells. By this fluctuation, the intra-cellular states of some cells are kicked out of the type-1 state, and fall into different cellular states. The cells appeared after this transition (e.g., type-2 and type-3) consume the waste products of type-1 cells in the environment as nutrients. Furthermore, after the increase of type-2 and type-3 cells, other cell types emerge by the fluctuation. The network of production and consumption of substrates in the medium by the cells with different states can form symbiotic relationships, for example, type-2 and type-5 cells supply the nutrients for their growth to each other. In this simulation, even though the gene regulatory and metabolic networks are identical for all cells, complex parasitic and symbiotic dynamics emerges by fluctuation-driven adaptive mechanism. Important point here is that, by this emergence of complex eco-system, the cells can utilize the nutrient supplied from outside the environment more efficiently than without such symbiotic relationship. In Fig.5, we plot how the total cell number at the steady state depends on the amplitude of the noise. As shown in the figure, when the amplitude of noise is small, the cellular state of the cells is homogeneous and the total number of cells is relatively small. In contrast, when the noise amplitude exceeds a threshold, the total cell number increases. In this phase, cells start to change their intra-cellular state by noise,

and parasitic and symbiotic relationships appear. Here, the cell-cell interactions through production and consumption of metabolic substrates make it possible use the nutrient from outside the environment more efficiently in total, even though each cell does not have sophisticated sensory machineries and pre-programmed regulatory machineries for such formation of ecosystem. This spontaneous emergence of complex ecosystem driven by the noise provides clues to understand the emergence and maintenance of real complex ecosystems including networks of parasitic and symbiotic relationships.

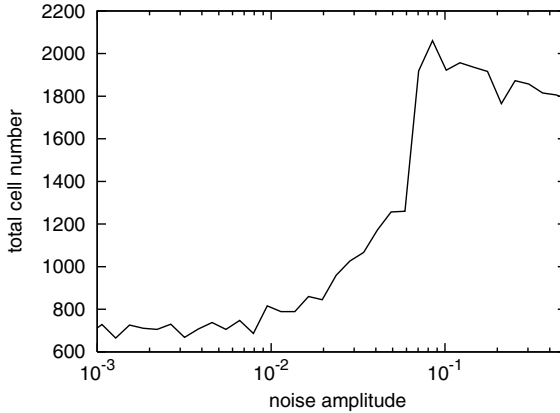


Fig. 5. The relationship between the noise amplitude and cell number in the steady state. When the noise amplitude is small, the state of cells are homogeneous, and the total cell number is relatively small. With the increase of the noise amplitude, a symbiotic relationship as shown in Fig.4 emerges, and the exchange of metabolites among cells with different states can realize efficient utilization of nutrient, which result total cell number in a steady state.

4 Discussion

We have carried out numerical experiments with our models using several sets of parameter values that allow for multiple attractors in expression dynamics, and have evaluated thousands of different randomly generated reaction networks. The emergence of adaptation and symbiotic processes triggered by noise is observed generally, independently of the details of the model. In fact, it emerges as long as the following four requirements are satisfied: i) the coexistence of multiple attractors; ii) the dependence of growth rate on attractors; iii) an increase of cellular reaction processes with the speed of growth; and iv) the presence of stochasticity in reaction dynamics. We have confirmed the robustness of our results against changes in model parameters and rules. For example, the results did not change when the model parameters such as coefficients of reactions were changed, provided the above requirements were satisfied. Also, the specific form

on how the growth rate depends on the expression dynamics is not important for the result, instead the same results are obtained as long as the growth rate is somehow determined by the expression dynamics.

This study provides a possible explanation for the flexible adaptation and formation of symbiotic relationship. Although symbiotic relationship is quite ubiquitous in nature [8], the mechanism of formation of symbiotic relationship, that is, how species maximizing their growth can create a cooperative relationship, still remains unclear. This study shows that, even when each cell in a cell society maximizes the growth rate, as the result of transitions of cellular states by the noise and selection of adaptive state, the cells with the cooperative relationship naturally emerges.

The noise-driven symbiotic formation process presented in this study might provide a novel control mechanism of multi-unit artificial systems in the field of engineering. Modern artificial systems are generally controlled by complicated computer programs and interactions among units are precisely designed. However, in general, such control mechanism by complicated programs and precisely designed interactions cannot respond adequately to circumstances for which no response is pre-programmed and the system never faced. For example, if an unpredictable interaction with other units occurs, to respond adequately is difficult for such elaborate artificial systems. Thus, a control mechanism that can respond to unexpected condition is desirable for the robust control mechanism for artificial systems. The noise-driven adaptive and symbiotic process presented in this study can be applied to such robust control mechanism. For example, recently based on this noise-driven adaptive mechanism, a method for virtual topology controls of internet traffic was proposed [9]. In this study, it was shown that the noise-driven topology control method can successfully adapt changes of traffic around twice higher variance comparing with conventional control method. For more complex traffic networks, such as interacting multiple overlay networks, the noise-driven mechanisms for the formation of symbiotic relationship we proposed in this study might be applicable to robust traffic control method, since this mechanism enables multi-unit systems to respond to unexpected condition with aid of stochastic fluctuation in internal dynamics, as discussed throughout this paper. We expect that this noise-driven robust control mechanism will be applied for controlling multi-unit artificial systems in future.

Acknowledgments. This work was supported by a grant from the Global COE (Centers of Excellence) Program in Osaka University from the Ministry of Education, Culture, Sports, Science and Technology, Japan. We would like to thank T. Yomo for stimulating discussions.

References

1. Jacob, F., Monod, J.: Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.* 3, 318–356 (1961)
2. Kashiwagi, A., Urabe, I., Kaneko, K., Yomo, T.: Adaptive response of a gene network to environmental changes by fitness-induced attractor selection. *PLoS One* 1 e49 (2006)

3. Furusawa, C., Kaneko, K.: A Generic Mechanism for Adaptive Growth Rate Regulation. *PLoS Comp. Biol.* 4(1), e3 (2008)
4. Elowitz, M.B., Levine, A.J., Siggia, E.D., Swain, P.S.: A Generic Mechanism for Adaptive Growth Rate Regulation. *Science* 297, 1183–1186 (2002)
5. Kaern, M., Elston, T.C., Blake, W.J., Collins, J.J.: Stochasticity in gene expression: from theories to phenotypes. *Nature Rev. Genet.* 8, 451–464
6. Pedraza, J.M., Van Oudenaarden, A.: Noise propagation in gene networks. *Science* 307, 1965–1969
7. Marr, A.G.: Growth rate of *Escherichia coli*. *Microbio. Rev.* 55(2), 316–333
8. Currie, C.R.: A community of ants, fungi, and bacteria: a multilateral approach to studying symbiosis. *Annu. Rev. Microbiol.* 55, 357–380 (2001)
9. Minami, Y., Koizumi, Y., Arakawa, S., Murata, M.: Adaptability of Virtual Network Topology Control based on Attractor Selection. In: *Proceedings of 2009 International Symposium on Nonlinear Theory and its Applications, NOLTA 2009* (2009)

Enhancing Sampling of the Conformational Space Near the Protein Native State

Brian Olson¹, Kevin Molloy¹, and Amarda Shehu^{1,2}

¹ Department of Computer Science

² Department of Bioinformatics and Computational Biology
George Mason University, Fairfax, VA, 22030
amarda@gmu.edu

Abstract. A protein molecule assumes specific conformations under native conditions to fit and interact with other molecules. Due to the role that three-dimensional structure plays in protein function, significant efforts are devoted to elucidating native conformations. Many search algorithms are proposed to navigate the high-dimensional protein conformational space and its underlying energy surface in search of low-energy conformations that comprise the native state. In this work, we identify two strategies to enhance the sampling of native conformations. We show that employing an enhanced fragment library with greater structural diversity to assemble low-energy conformations allows sampling more native conformations. To efficiently handle the ensuing vast conformational space, only a representative subset of the sampled conformations are maintained and employed to further guide the search for native conformations. Our results show that these two strategies greatly enhance the sampling of the conformational space near the native state.

Keywords: protein native state, conformational ensemble, probabilistic search, tree-based projection-guided exploration, fragment library.

1 Introduction

The genomic revolution has resulted in millions of protein sequences for which little functional information is available [24]. Due to the central role that protein molecules play in biochemical processes in the cell, knowledge of the biological function of a protein molecule promises to advance our understanding of the living cell and various diseases. The spatial arrangement of a protein's atoms, interchangeably referred to as a structure or conformation, determines to a great extent biological function. A protein molecule assumes specific conformations under physiologic (native) conditions to fit and interact with other molecules.

Due to the role that structure plays in the biological function of a protein, significant efforts are devoted to elucidating native structures. The Protein Structure Initiative has pushed experimental efforts and yielded native structures of many proteins [27]. The great number of novel protein sequences with no known structures and the time and cost associated with resolving structures in the wet lab call for computational methods to complement wet-lab efforts.

The Anfinsen experiments have shown that the amino-acid sequence governs the folding of a protein chain into a “biologically-active conformation” under a “normal physiological milieu” [2]. Anfinsen posited that, if one were to understand how the amino-acid sequence determines the biologically-active or native conformation, one could find such a conformation *in silico*. Research shows that proteins are not rigid and that the biologically-active state is an ensemble of (native) conformations [15, 12, 18]. Probing this ensemble when employing only knowledge of the amino-acid sequence of a protein at hand continues to challenge structural biology and has been proved NP-hard [13].

A protein chain consists of smaller building blocks, amino acids, each of which contains many atoms. Amino acids connect their backbone atoms to form a backbone chain, as shown in Fig. 1(a), with side-chain atoms dangling off the backbone of each amino acid. Tracking the various conformations of a protein chain involves exploring a vast conformational space of many dimensions. Many degrees of freedom (dofs) are needed to represent a protein chain. One can reduce the representational detail through coarse-grained representations, such as backbone-only representations, which track only conformations of the backbone. Once a native backbone conformation is found, computational techniques can be used to find physically-relevant placements of the side chains [8, 17].

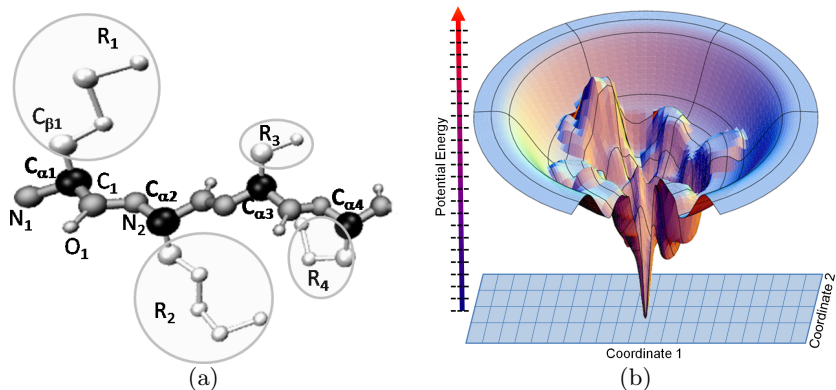


Fig. 1. (a) A chain of four amino acids is shown. Backbone atoms are labeled N (gray), C_α (black), C (gray), and O (silver). A peptide bond between N_i and C_{i+1} links two amino acids together (i proceeds from N- to C-terminus, which refer to backbone N and C atoms not involved in peptide bonds). Atoms in white are labeled R for residue. There are 20 distinct residues or side chains in natural proteins. Side chain atoms dangle off the backbone. (b) We cross-section energy landscape (grid on z axis) and projection of conformational space (grid on xy axis, 2 coordinates shown for visualization).

Many coarse-grained representations have been proposed [10]. Even when focusing on the backbone, many dofs remain. A protein chain of n amino acids poses $2n$ backbone dihedral angles that can be modified to obtain backbone conformations. The conformational space of interest is narrowed when focusing on native conformations. These conformations are associated with the lowest energies in a funnel-like energy surface that underlies the protein conformational

space [12]. The totality of atomic interactions in a protein conformation results in a potential energy that is directly related to the probability of that conformation to be populated under native conditions [12].

The search for the low-energy native conformations is challenging, because the energy surface is rich in local minima. Some of the minima may be introduced by coarse-grained energy functions designed to operate on coarse-grained conformations. By reducing the number of atoms modeled, coarse-grained representations and the energy functions that operate on them are more computationally appealing than all-atom functions. All available energy functions are empirical. However, it is generally accepted that modern functions do not significantly hamper a powerful conformational search [10].

A powerful search algorithm needs to populate a sufficient number of energy minima in order to probe the native state without spending impractical resources on irrelevant regions of the search space. Without any a priori information, it is not possible to know what a sufficient number of minima is or where the relevant regions are. The only knowledge is that native conformations are associated with low energies. This is the main reason why it remains challenging for search algorithms to obtain native conformations. Computing these conformations, however, is crucial in associating structural and functional information with novel protein sequences, engineering novel proteins, predicting protein stability, and modeling protein-ligand or protein-protein interactions [6, 39, 21].

We have recently proposed a probabilistic search algorithm that essentially addresses the question of where to devote exploration time [31]. The algorithm gathers information about regions of the conformational space and energy surface it explores. Discretizations of the explored conformational space and energy surface are employed to further guide the search in the conformational space.

The algorithm essentially grows a search tree in conformational space, reconciling two goals: (i) expanding towards low-energy conformations while (ii) not oversampling geometrically-similar conformations. The first goal guides the tree deep in the energy surface. The second goal grows the tree wide in conformational space. Energies of computed conformations are partitioned into levels through a discretized one-dimensional (1d) grid. The grid helps select conformations associated with lower energy levels more often for expansion. The search keeps track of computed conformations in a low-dimensional projection space, which is discretized to select for expansion low-energy conformations that fall in under-explored regions (see Fig. II(b)). The employment of discretization layers is inspired by sampling-based motion-planning work that uses decompositions, subdivisions, and projections of the search space to balance the exploration between coverage and progress toward the goal [35, 30, 28, 38, 23, 22, 29, 9, 36].

In this work, we focus on further enhancing the sampling of the conformational space near the native state while employing reasonable resources. This goal is crucial, given the potential inaccuracies inherent in a coarse-grained energy function and the fact that the native state is an ensemble of conformations. We identify two strategies to enhance sampling. We propose to increase the complexity of the conformational space while reducing the granularity of the

conformational ensemble maintained in the search tree. An enhanced library of structurally diverse fragment configurations is used to assemble low-energy conformations and increase the complexity of the search space. Increasing the complexity appears counterintuitive to efforts to expedite search. The discretizations employed in our search algorithm, however, allow exploiting the complexity without wasting resources. Moreover, a second strategy is proposed to efficiently handle the vast conformational space that ensues from employing the enhanced fragment library. Only a representative subset of the sampled conformations are maintained and employed to further guide the search for native conformations. Results show the proposed strategies enhance the sampling of the conformational space near the native state. Our work may be promising for large-scale proteomics applications, where the focus is on quickly probing the native state and then refining selected conformations in detailed biophysical studies.

The rest of this paper is organized as follows. A brief summary of related work is provided in section [1.1](#). Our method is described in section [2](#). Results follow in section [3](#). The article concludes with a discussion in section [4](#).

1.1 Related Work

Where should a search algorithm devote its time? Regions that lead to the solution space are not known a priori, since stochastic search of a high-dimensional space affords only a local view. An effective search algorithm needs to strike the right balance between populating a large number of distinct low-energy regions and focusing further resources to regions likely to lead to the energy basin corresponding to the native state. Ingredients for success were identified most notably in [\[25, 26\]](#). Work in [\[25\]](#) introduced the idea of a two-stage hierarchical exploration that searches the whole conformational space first and then narrows the search in a later stage to smaller regions with low energy and distinct geometry.

Since the success of locating the energy basin in the second stage depends on the regions populated by the first stage, the emerging state-of-the-art template is to sample a large number of low-energy conformations in the first stage, essentially to build a broad map of the energy landscape [\[33, 6, 7, 5, 11, 32\]](#). Clustering is then conducted over the conformations to reveal distinct minima that constitute good starting points from which expensive (in finer detail) local searches in the second stage can reach the basin. In contrast, coarse graining is employed to reduce the computational cost of the first stage. It still takes weeks on multiple CPUs to obtain a large number of low-energy conformations potentially relevant for the native state [\[33, 6, 7, 11, 32\]](#). Since the local searches employed in the second stage are computationally expensive, it is important that the first stage reveal few distinct local minima worth exploring in greater detail.

The first stage of the search and the analysis over the conformations are often independent of each-other. As a result, computed conformations cannot be ensured to be geometrically-distinct. Incorporating geometric diversity during the exploration is non-trivial, in part because it remains difficult to find meaningful conformational (reaction) coordinates on which to measure geometric diversity. Popular measures like least Root-Mean-Squared-Deviation (lRMSD) and radius

of gyration (Rg) are confined to the analysis because they can mask away important differences. Specifically, work in [33] has shown that important minima can be missed even when employing Rg to select distinct conformations obtained at a current temperature to initiate MC trajectories at the next temperature in a Simulated Annealing MC search. Significant work in biophysics is devoted to finding effective reaction coordinates for proteins (cf. to [10]).

The search algorithm we have recently proposed [31, 34] incorporates analysis over explored regions of the conformational space and where they map in the protein energy surface in order to adaptively determine where next to devote resources. The analysis is carried out over discretizations of the explored space in order to properly guide the search over the continuous conformational space (a brief summary of the essential ingredients of the algorithm is provided in section 2). As the description of our method and results shows, the sampling of the conformational space near the native state can be further enhanced if one increases the complexity of the space while reducing the size of the conformational ensemble maintained in the search tree.

2 Methods

We first summarize the main steps of the algorithm proposed in [31, 34] (shown below). Given a protein sequence α , the goal is to obtain an ensemble Ω_α , where the lowest-energy backbone-only conformations are sufficiently close to the native state that they can be further refined to recover this state in all-atom detail.

Input: α , amino-acid sequence
Output: ensemble Ω_α of conformations
1: $C_{\text{init}} \leftarrow$ extended coarse-grained conf from α
2: $\text{ADDCONF}(C_{\text{init}}, \text{Layer}_E, \text{Layer}_{\text{Proj}})$
3: while TIME AND $ \Omega_\alpha $ do not exceed limits do
4: $\ell \leftarrow \text{SELECTENERGYLEVEL}(\text{Layer}_E)$
5: cell $\leftarrow \text{SELECTGEOMCELL}(\ell, \text{Layer}_{\text{Proj}}.\text{cells})$
6: $C \leftarrow \text{SELECTCONF}(\text{cell}.\text{confs})$
7: $C_{\text{new}} \leftarrow \text{EXPANDCONF}(C)$
8: $\text{ADDCONF}(C_{\text{new}}, \text{Layer}_E, \text{Layer}_{\text{Proj}})$
9: $\Omega_\alpha \leftarrow \Omega_\alpha \cup \{C_{\text{new}}\}$

An explicit 1d grid is defined over interval $[E_{\text{min}}, E_{\text{max}}]$, where E_{min} is the minimum energy over computed conformations, and E_{max} is the energy of the extended conformation. Energy levels ℓ are generated every δE units, which is set to a small 2 kcal/mol, so that the average energy $E_{\text{avg}}(\ell)$ over conformations in a level $\ell \in \text{Layer}_E$ captures well the distribution of energies in ℓ . This discretization is used to bias the selection towards conformations in lower energy levels through the quadratic weight function $w(\ell) = E_{\text{avg}}(\ell) \cdot E_{\text{avg}}(\ell) + \epsilon$, where $\epsilon = 2^{-22}$ ensures a non-zero probability of selection for conformations with higher energies. A level ℓ is selected with probability $w(\ell) / \sum_{\ell' \in \text{Layer}_E} w(\ell')$.

An implicit 3d grid is associated with ℓ based on a uniform discretization of geometric coordinates. Three coordinates that capture extrema in a 3d structure

are adapted from the ultrafast shape recognition (USR) features proposed in [3]. A second weight function selects cells with fewer conformations as in $1.0/[(1.0 + \text{nse1}) \cdot \text{nconfs}]$, where `nse1` records how often a cell is selected, and `nconfs` is the number of conformations that project to the cell. Once a cell is chosen, the actual conformation selected for expansion is obtained at random over those in the cell, since conformations in the same cell have similar energies (within δE).

A new conformation C_{new} that expands the tree (and grows the conformational ensemble Ω_α) from a selected C conformation is sampled through a Metropolis Monte Carlo technique that employs fragment-based assembly. The backbone dihedral angles of a selected fragment of three amino acids (trimer) in C are exchanged with angles from a library of trimer configurations built from a non-redundant subset of known protein native structures. A total of $n - 2$ (n amino acids in the chain) exchanges are evaluated and accepted with probability according to the Metropolis criterion to obtain C_{new} .

Applications on different protein sequences reveal that the ensemble Ω_α of low-energy backbone conformations sampled for a sequence in a few CPU hours contains many conformations similar to the known native structure [31]. Comparisons with a Monte Carlo trajectory show the algorithm has a higher sampling capability [31, 34]. However, detailed inspection of how the algorithm navigates the conformational space near the native state reveals that the ability to add low-energy conformations diminishes significantly with time. It becomes more difficult to find new low-energy conformations in underexplored regions of the conformational space. Moreover, the multitude of conformations retained in Ω_α imposes restrictions on execution time, further restricting the search.

We propose two strategies to help the exploration find more low-energy conformations near the native state. An enhanced fragment library with greater structural diversity is proposed to assemble low-energy conformations and sample more conformations near the native state. To efficiently handle the ensuing vast conformational space, only a representative subset of the sampled conformations are maintained and employed to further guide the tree in conformational space. We detail each of these strategies next.

2.1 Enhancing the Trimer Configuration Library

In recent years fragment-based assembly has been incorporated into most state-of-the-art protein conformational search algorithms [16, 6, 7, 5, 11, 20]. The diversity of the fragment library influences the quality of the assembled conformations [20]. Indeed, the domain of the conformational search space is primarily determined by the fragment library. To provide the exploration a greater domain in which to search for native conformations, we propose an enhanced fragment library that essentially adds complexity to the conformational space.

The original fragment library (OFL) used in our recent work [31, 34] contains trimer configurations, organized by trimer amino-acid sequence. A subset of nonredundant protein structures is extracted through the PISCES server [37] from the Protein Data Bank (PDB) [4]. The subset contains only proteins that have $\leq 40\%$ sequence similarity, $\leq 2.5\text{\AA}$ resolution and R-factor ≤ 0.2 . The

40% cutoff reduces the topologies that are over-populated by similar protein sequences in the PDB. The remaining 6,000 protein chains are split into all overlapping trimers. The configurations, backbone dihedral angles, of these trimers are recorded in a fragment library indexed by trimer amino-acid sequences.

When a conformation is selected for expansion, each of the $n - 2$ Monte Carlo moves propose to replace a trimer configuration with a configuration extracted from the fragment library. In OFL, the candidate configurations are only those with the same amino-acid sequence as the sequence of the trimer configuration chosen for replacement. Focusing only on trimer configurations with the same amino-acid sequence does not allow considering configurations that, while slightly different in sequence, may allow assembling novel conformations that meet the Metropolis criterion. Analysis of protein structures reveals that proteins have similar native structures with as little as 15% sequence identity [14]. Excluding trimer configurations simply because their amino-acid sequence is not identical to that of the trimer configuration selected for replacement restricts the conformational search space. This may prevent sampling novel conformations potentially relevant for the native state of the given protein sequence.

We propose to expand the conformational space available to our algorithm with an enhanced fragment library (EFL). Local features predicted from the given sequence α are employed to design a structurally-diverse high-quality library of configurations. The candidate trimer configurations in EFL are dependent on α , and we refer to a specific library instance designed from a given α as EFL_α . Our construction of EFL_α biases towards trimer configurations that share features with those predicted from α . Essentially, EFL_α , whose construction is detailed below, allows selecting configurations that have *similar* (not necessarily identical) sequences to a trimer configuration selected for replacement. While containing a more diverse set of configurations at the disposal of the expansion routine in the algorithm, EFL_α does not contain more configurations than OFL. The configurations are limited to those that share secondary structure annotations with the annotation predicted on α .

EFL_α is constructed as follows. A multiple sequence alignment (MSA) lists proteins that have similar sequences to the given α . PSI-BLAST [1] is then employed to analyze the MSA and yield for each position i in α a list of amino acids that can replace the amino-acid at position i . The resulting position-specific profile for α reveals what alternative trimer sequences can be considered as similar to a trimer from position i to $i + 2$. The configurations of these trimers, extracted from a nonredundant database of protein structures as detailed above, can be added as candidate configurations to those extracted for the trimer sequence from i to $i + 2$. A filtering step improves the quality of the resulting configurations. Only configurations with the same secondary structure (as present in the known protein structures from which the trimer configurations are extracted) as that predicted for α with PSI-PRED [19] are added as candidate configurations for a trimer. Considering configurations of similar sequences but identical secondary structures has become very popular in ab-initio structure prediction methods that employ fragment-based assembly [6].

The resulting EFL_α represents (in the number of ways conformations can be assembled with the configurations in the library) a conformational space that is not only larger, but also more likely to share local structural motifs with the native structure of the given sequence α . Results in section 3 show that our algorithm is able to take advantage of this more complex conformational space to discover more conformations relevant for the native state than when employing the original fragment library.

2.2 Reducing the Granularity of the Conformational Ensemble Ω_α

One of the benefits of employing trimer configurations to assemble conformations is that hundreds of thousands of conformations can be sampled this way in less than a day on one CPU. Maintaining all these conformations in the ensemble Ω_α introduces both a practical memory limitation and unnecessary difficulty in selecting a conformation for expansion. Our recent work limits the exploration to three hours on one CPU in order to limit the size of the conformational ensemble [31, 34]. Limiting the size of the conformational ensemble, however, limits the explorative power of the algorithm. Moreover, the enhanced fragment library increases the size of the conformational space to be sampled. In order to explore this broader space while not limiting the sampling capability of the algorithm, we change the purpose of the conformational ensemble Ω_α . Instead of maintaining every sampled conformation in Ω_α , the ensemble now maintains only a carefully selected subset of the sampled conformations through which to represent the explored conformational space.

By essentially reducing the granularity of Ω_α , the linear relationship between running time and memory requirements is removed. Each C_{new} generated is first evaluated for geometric novelty before being added to Ω_α . Clustering by lRMSD is computationally prohibitive to be performed after every sampled conformation C_{new} . Instead, we propose a less costly but effective strategy, which reduces the size of Ω_α by a factor of 10 to 100 (see Fig 2 in section 3). The strategy adds minimal computation overhead and does not impact the ability of the algorithm to sample low-energy conformations near the native state.

The granularity reduction exploits a feature of the energetic and geometric projection layers that is actually exploited in the selection process: two conformations that lie in the same energy level ℓ and projection cell r will be geometrically similar (for some similarity threshold τ). Analysis shows that for the chosen granularity of 30 geometric cells per dimension (in the geometric projection grid) the value of τ is less than 1\AA (using lRMSD). For this value of τ , an arbitrary cutoff of one conformation per ℓ and r would suffice. However, the strategy we employ is not dependent on the chosen granularity of the geometric projection grid. Instead, if two conformations share the same ℓ and r , their similarity is determined using lRMSD. If the lRMSD is below a chosen τ (set at 1\AA in our experiments), then only one of the conformations, selected at random, is retained; either the existing conformation is replaced or the new conformation is discarded with equal probability.

2.3 Implementation Details

The algorithm is implemented in C++ and runs single-threaded on an AMD 2.66 GHz Dual-Core Opteron with 4 GB of RAM. All reported times are based on CPU user time. The similarity threshold τ is set to 1Å. All other parameters are as in previous work [31, 34]. Results that compare the enhanced fragment library to the original library are obtained after 48 hours. This gives the algorithm ample time to sample different combinations of fragment configurations in the libraries and reduces the role of stochastic variations in our comparisons.

3 Results

We apply the proposed strategies to enhance the sampling of the native state of the six protein sequences listed in section 3.1. Section 3.2 compares the quality of the enhanced fragment library with the original one. Section 3.3 then shows the degree to which granularity reduction compresses the conformational ensemble Ω_α . Finally, section 3.4 shows how the proposed strategies enhance the sampling of conformations near the native state for each of the chosen protein sequences.

3.1 Target Proteins

Table 1 lists the six targeted protein sequences, Pin1 Trp-Trp ww domain (wwD), human β -defensin 2 (hbd2), bacterial ribosomal protein (L20), immunoglobulin binding domain of streptococcal protein G (GB1), calbindin D_{9k}, and the African Swine Fever Virus pB119L protein. The proteins are selected to span different sizes (number of amino-acids) and known native topologies.

Table 1. PDB Id, fold, and number of amino acids are shown for each of the six proteins. PDB Id refers to a unique identifier associated with an experimentally-resolved native structure deposited for a protein in the PDB.

Protein	wwD	hbd2	L20	GB1	Calbindin D _{9k}	pB119L
PDB Id	1I6C	1FD4	1GYZ	1GB1	4ICB	3GWL
Fold	β	α/β	α	α/β	α	α
Nr. AAs	26	41	60	60	76	106

3.2 Quality of the Enhanced Fragment Library

The quality of the fragment libraries is evaluated using the local-fit score introduced in [20]. The local-fit score measures the degree to which a fragment library fits a given native protein structure not used to construct the library. The chain of a given protein is broken into all its overlapping trimers. The configurations available for each resulting trimer in the library are then scanned to find the configuration closest (in terms of IRMSD) to the configuration of the trimer in the given native structure. The local-fit score associated with a given protein is

the average over the lowest IRMSDs obtained for all trimers defined over the chain of the protein. The local-fit score, referred to as IRMSD_f , is calculated for each of the six proteins listed above and is reported in Table 2. The scores obtained when employing the original fragment library are compared with those obtained when employing the enhanced fragment library.

Table 2. Local-fit IRMSD_f scores and IRMSDs between assembled conformations and known native structures are shown when employing the original fragment library (normal font) and enhanced fragment library (bold) for each of the six target proteins

Protein	wwD	hbd2	L20	GB1	Calbindin D _{9k}	pB119L
IRMSD_f (Å)	0.09 0.08	0.02 0.04	0.06 0.06	0.06 0.06	0.03 0.02	0.05 0.03
IRMSD (Å)	5.36 5.02	8.46 6.21	9.74 7.94	6.39 9.01	6.04 5.78	25.04 10.69

Table 2 shows (row 2) that overall lower local-fit scores are obtained when employing the enhanced fragment library. Similarly, if one assembles the conformation with the lowest IRMSD configurations from each library for each selected trimer in a given protein chain, the enhanced fragment library yields conformations that are closer to the known native structures. Lower IRMSDs from the native structure are reported for most of the proteins in Table 2 (row 3). This is not surprising, since the enhanced fragment library does not limit the search for fragment configurations to those with the same amino-acid sequence as the selected trimer. The high IRMSDs between the assembled conformations and the known native structures, especially for GB1 and pB119L, make the case that suboptimal fragment configurations are needed to assemble an optimal conformation. This further attests to the difficulty of assembling native conformations and the need for non-trivial search methods with powerful sampling capability.

3.3 Reduction of Ensemble Ω_α

Reducing the granularity of the Ω_α ensemble significantly reduces the number of conformations retained in memory. The rate of memory consumption is now directly related to the algorithm’s ability to discover geometrically-novel conformations with similar energies. In practice, this enhancement allows exploring the conformational space for an indefinite period of time. Fig 2 illustrates the relationship between runtime and memory requirement for the algorithm on Calbindin D_{9k} (similar results are observed for all other tested systems).

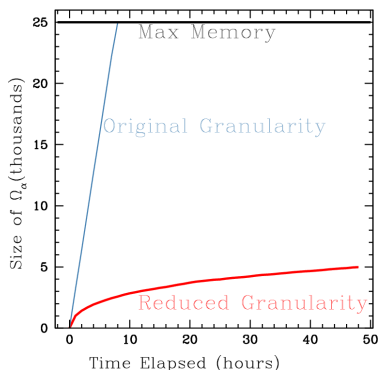


Fig. 2. Granularity reduction lowers the rate of growth of Ω_α (red line vs. blue line). The Black line shows maximum ensemble size stored in a 32-bit machine.

3.4 Effectiveness of Enhanced Fragment Library

The ensemble Ω_α contains low-energy coarse-grained conformations that are good candidates for all-atom energetic refinement. In Table 3 we report the lowest IRMSD between the conformations in Ω_α and the known native structure for each protein. The lowest IRMSDs are compared when employing the original versus the enhanced fragment library. Table 3 shows that lower IRMSDs are obtained when employing the enhanced fragment library. This library allows the search algorithm to assemble conformations that are closer in IRMSD to the native state compared to the original fragment library. Fig. 3 shows the lowest-IRMSD conformation obtained with the enhanced fragment library superimposed over the known native structure for each of the six targeted proteins.

Table 3 also shows the lowest IRMSD obtained on each protein when employing the state-of-the-art Rosetta structure prediction method [6]. To keep the comparisons similar, only the coarse-grained structure prediction component of

Table 3. The minimum IRMSD to the native structure is shown for each of the six target proteins. Data obtained when employing the original library is in normal font. Data obtained with the enhanced fragment library is highlighted in bold. The final row shows the data obtained when employing Rosetta [6].

Protein	wwD	hbd2	L20	GB1	Calbindin D_{9k}	pB119L
min-IRMSD (Å)	4.52 3.47	5.34 5.84	5.11 3.66	6.89 6.31	5.76 4.70	10.32 8.30
Ros-IRMSD (Å)	2.90	6.17	3.68	2.67	2.73	9.13

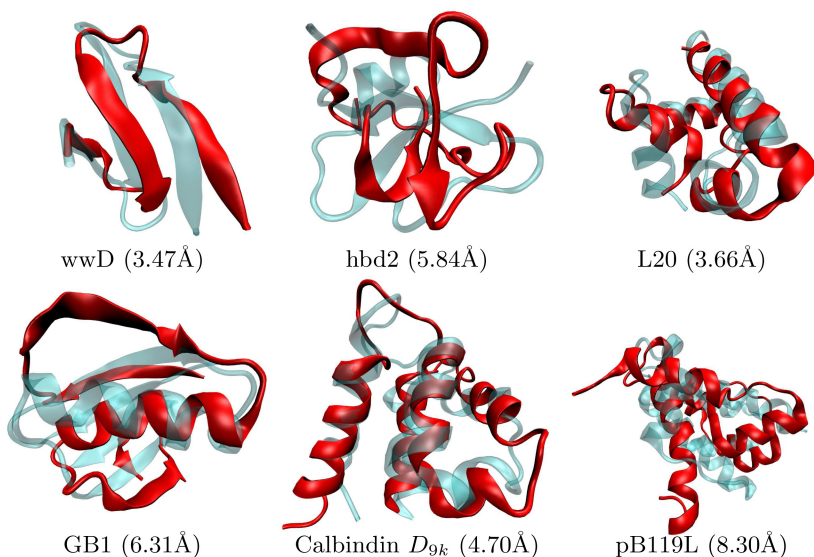


Fig. 3. The lowest IRMSD conformation obtained with the enhanced fragment library is drawn in red and superimposed over the known native structure in transparent blue

Rosetta is employed. This component is initiated from each of the six target sequences and allowed to run for the same amount of time, 48 CPU hours. Comparison of the lowest IRMSDs obtained with Rosetta to those obtained with our method when employing the enhanced fragment library shows that Rosetta significantly outperforms our method by more than 2 Å on only one protein, GB1. Our method obtains better results on three of the target proteins. The ability of Rosetta to perform better on GB1 may originate from the coarse-grained energy function and the modulation of temperature during its coarse-grained search. Our discussion in section 4 lists a more accurate energy function and incorporation of temperature modulation as interesting directions for future research.

The enhanced fragment library, coupled with the reduction of the conformational ensemble Ω_α , allows the search algorithm to enhance its sampling of the native state. Figure 4 shows histograms of IRMSDs of conformations in Ω_α from the known native structure, superimposing the histograms obtained when employing both the enhanced and original fragment ensemble. These histograms show that the enhanced fragment ensemble allows the search algorithm to increase the number of computed conformations with lower IRMSD to the known native structure. This increase is significant for wwD, L20, calbindin, and pB119L. pB119L is longer than the other proteins and is used here to test the upper limits of the search algorithm, with neither library allowing us to obtain conformations below 8Å IRMSD from the native structure.

The histogram representation in Figure 4 is useful, because local maxima in the histograms correspond to potential clusters of conformations that can be detected with simple clustering techniques. The ensembles obtained with the enhanced fragment library for each protein contain more of these maxima at low IRMSDs. A technique interested in selecting a few conformations would obtain more native-like conformations if the enhanced fragment library is employed.

4 Discussion

This paper investigates the effect of increasing the complexity of the conformational search space while decreasing the sample size required to represent it on a probabilistic search algorithm. We propose a more structurally diverse fragment library to provide our search algorithm with a larger conformational space. To efficiently handle the vast search space, we reduce the granularity of the conformational ensemble that the algorithm maintains to represent the space it has explored. Our results show that these two strategies allow the search algorithm to enhance the sampling of conformations relevant for the native state.

Our search algorithm, recently introduced in [31, 34], makes use of discretizations over projection layers of the energy surface and conformational space to guide its search towards diverse low-energy conformations. The algorithm is a first step towards rapidly computing coarse-grained native conformations from amino-acid sequence alone. The strategies proposed here address the need to enhance the sampling capability of the algorithm. Our results show that the proposed strategies confer the algorithm with the capability to conduct a longer, more detailed exploration and improve its sampling of native conformations.

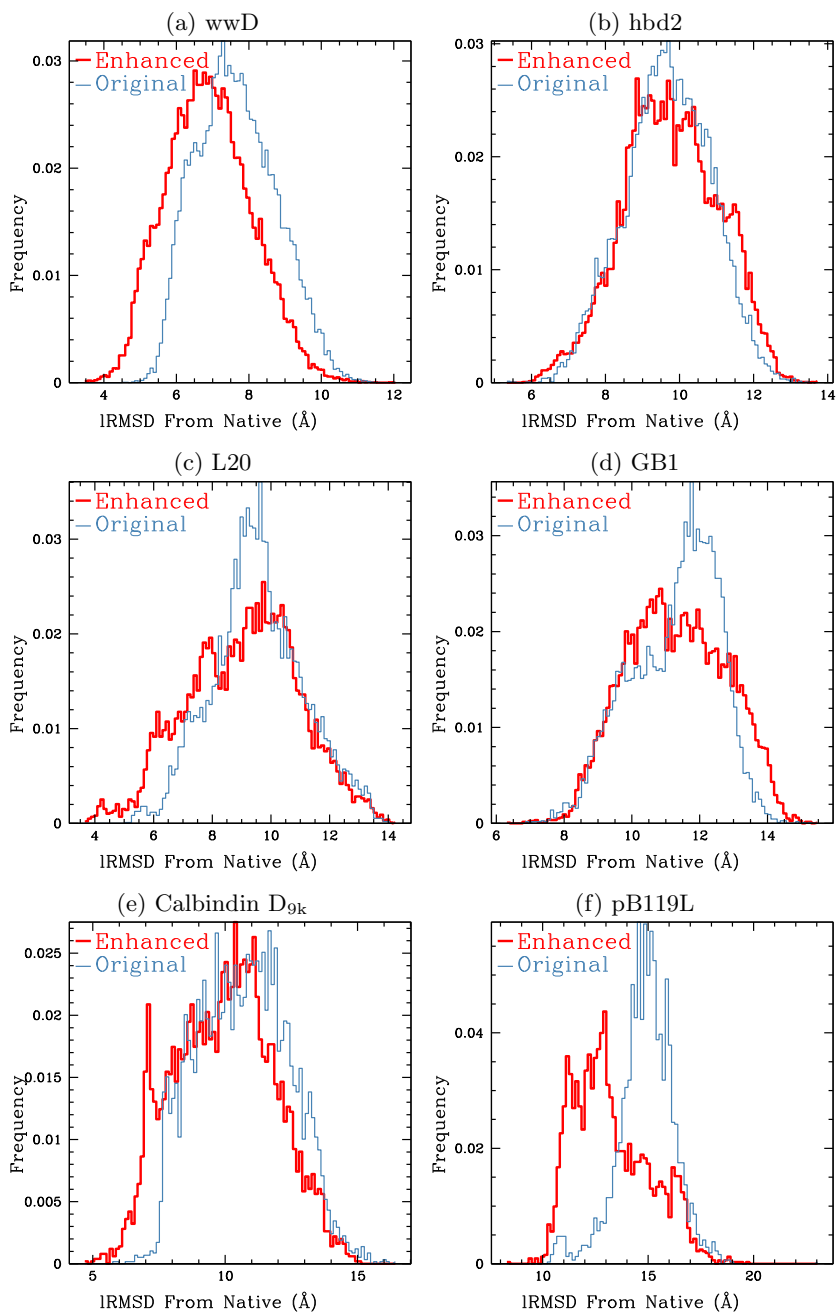


Fig. 4. (a-f) show the percentage of conformations in the ensemble Ω_α for a given IRMSD from the native structure. Data obtained with the enhanced fragment library are shown with a thick red line and those obtained with the original fragment library are shown with a thin blue line.

The work conducted in this paper lays the foundations for further future work. The enhanced sampling capability shown in this work will allow investigating different selection-related weight functions, novel projection coordinates, and coarser representations to further enhance the sampling capability of the algorithm on more complex high-dimensional conformational spaces of larger protein systems with challenging native topologies. Furthermore, state-of-the-art coarse-grained energy functions and a temperature modulation scheme will be pursued to further enhance the sampling capability of the method.

References

1. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl. Acids Res.* 25(17), 3389–33402 (1997)
2. Anfinsen, C.B.: Principles that govern the folding of protein chains. *Science* 181(4096), 223–230 (1973)
3. Ballester, P.J., Richards, G.: Ultrafast shape recognition to search compound databases for similar molecular shapes. *J. Comput. Chem.* 28(10), 1711–1723 (2007)
4. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The Protein Data Bank. *Nucl. Acids Res.* 28(1), 235–242 (2000)
5. Bonneau, R., Baker, D.: De novo prediction of three-dimensional structures for major protein families. *J. Mol. Biol.* 322(1), 65–78 (2002)
6. Bradley, P., Misura, K.M.S., Baker, D.: Toward high-resolution de novo structure prediction for small proteins. *Science* 309(5742), 1868–1871 (2005)
7. Brunette, T.J., Brock, O.: Guiding conformation space search with an all-atom energy potential. *Proteins: Struct. Funct. Bioinf.* 73(4), 958–972 (2009)
8. Canutescu, A.A., Shelenkov, A.A., Dunbrack Jr., R.L.: A graph-theory algorithm for rapid protein side chain prediction. *Protein Sci.* 12(9), 2001–2014 (2003)
9. Choset, H., et al.: *Principles of Robot Motion: Theory, Algorithms, and Implementations*, 1st edn. MIT Press, Cambridge (2005)
10. Clementi, C.: Coarse-grained models of protein folding: Toy-models or predictive tools? *Curr. Opin. Struct. Biol.* 18, 10–15 (2008)
11. DeBartolo, J., Colubri, A., Jha, A.K., Fitzgerald, J.E., Freed, K.F., Sosnick, T.R.: Mimicking the folding pathway to improve homology-free protein structure prediction. *Proc. Natl. Acad. Sci. USA* 106(10), 3734–3739 (2009)
12. Dill, K.A., Chan, H.S.: From Levinthal to pathways to funnels. *Nat. Struct. Biol.* 4(1), 10–19 (1997)
13. Dill, K.A., Ozkan, B., Shell, M.S., Weikl, T.R.: The protein folding problem 37, 289–316 (2008)
14. Fersht, A.R.: *Structure and Mechanism in Protein Science. A Guide to Enzyme Catalysis and Protein Folding*, 3rd edn. W.H. Freeman and Co., New York (1999)
15. Frauenfelder, H., Sligar, S.G., Wolynes, P.G.: The energy landscapes and motion on proteins. *Science* 254(5038), 1598–1603 (1991)
16. Haspel, N., Tsai, C., Wolfson, H., Nussinov, R.: Reducing the computational complexity of protein folding via fragment folding and assembly. *Protein Sci.* 12(6), 1177–1187 (2003)
17. Heath, A.P., Kaviraki, L.E., Clementi, C.: From coarse-grain to all-atom: Towards multiscale analysis of protein landscapes. *Proteins: Struct. Funct. Bioinf.* 68(3), 646–661 (2007)

18. Huang, Y.J., Montellione, G.T.: Structural biology: Proteins flex to function. *Nature* 438(7064), 36–37 (2005)
19. Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* 292(2), 195–202 (1999)
20. Kolodny, R., Koehl, P., Guibas, L., Levitt, M.: Small libraries of protein fragments model native protein structures accurately. *J. Mol. Biol.* 323(2), 297–307 (2002)
21. Kortemme, T., Baker, D.: Computational design of protein-protein interactions. *Curr. Opin. Struct. Biol.* 8(1), 91–97 (2004)
22. Kurniawati, H., Hsu, D.: Workspace-based connectivity oracle: An adaptive sampling strategy for PRM planning. In: WAFR. Springer Tracts in Advanced Robotics. vol. 47, pp. 35–51 (2006)
23. Ladd, A.M., Kavraki, L.E.: Motion planning in the presence of drift, underactuation and discrete system changes. In: *Robotics: Sci. and Syst.*, pp. 233–241 (2005)
24. Lee, D., Redfern, O., Orengo, C.: Predicting protein function from sequence and structure. *Nat. Rev. Mol. Cell Biol.* 8(12), 995–1005 (2007)
25. Lee, J., Scheraga, H.A., Rackovsky, S.: New optimization method for conformational energy calculations on polypeptides: Conformational space annealing. *J. Comput. Chem.* 18(9), 1222–1232 (1997)
26. Lee, J., Scheraga, H.A., Rackovsky, S.: Conformational analysis of the 20-residue membrane-bound portion of melittin by conformational space annealing. *Biopolymers* 46(2), 103–115 (1998)
27. Matthews, B.W.: Protein structure initiative: getting into gear. *Nat. Struct. Biol. & Mol. Biol.* 14(6), 459–460 (2007)
28. Plaku, E., Kavraki, L., Vardi, M.: Discrete search leading continuous exploration for kinodynamic motion planning. In: *Robotics: Sci. and Syst.*, Atlanta, GA, USA (2007)
29. Rodriguez, S., Thomas, S., Pearce, R., Amato, N.: RESAMPL: A Region-Sensitive Adaptive Motion Planner. In: WAFR. Springer Tracts in Advanced Robotics, vol. 47, pp. 285–300 (2006)
30. Sánchez, G., Latombe, J.-C.: On delaying collision checking in PRM planning: Application to multi-robot coordination. *Int. J. Robot. Res.* 21(1), 5–26 (2002)
31. Shehu, A.: An ab-initio tree-based exploration to enhance sampling of low-energy protein conformations, Seattle, WA, USA (2009)
32. Shehu, A., Kavraki, L.E., Clementi, C.: Unfolding the fold of cyclic cysteine-rich peptides. *Protein Sci.* 17(3), 482–493 (2008)
33. Shehu, A., Kavraki, L.E., Clementi, C.: Multiscale characterization of protein conformational ensembles. *Proteins: Struct. Funct. Bioinf.* 76(4), 837–851 (2009)
34. Shehu, A., Olson, B.: Guiding the search for native-like protein conformations with an ab-initio tree-based exploration. *Int. J. Robot. Res.* 29(8), 1106–1127 (2010)
35. Stilman, M., Kuffner, J.J.: Planning among movable obstacles with artificial constraints. *Int. J. Robot. Res.* 12(12), 1295–1307 (2008)
36. van den Berg, J.P., Overmars, M.H.: Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners. *Int. J. Robot. Res.* 24(12), 1055–1071 (2005)
37. Wang, G., Dunbrack, R.L.: Pisces: a protein sequence culling server. *Bioinformatics* 19(12), 1589–1591 (2003)
38. Yang, Y., Brock, O.: Efficient motion planning based on disassembly. In: *Robotics: Sci. and Syst.*, Cambridge, MA, pp. 97–104 (2005)
39. Yin, S., Ding, F., Dokholyan, N.V.: Eris: an automated estimator of protein stability. *Nat. Methods* 4(6), 466–467 (2007)

Self-organized Data Aggregation among Selfish Nodes in an Isolated Cluster

K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine

Department of Information and Communications Technology,
Graduate School of Engineering, Osaka University,
2-1 Yamadaoka, Suita-shi, Osaka 565-0871, Japan

habib@post.comm.eng.osaka-u.ac.jp, {sasabe,takine}@comm.eng.osaka-u.ac.jp

Abstract. This paper considers a delay tolerant network, where a message ferry travels multiple isolated clusters, collects data from nodes in the clusters, and finally delivers the data to a sink node. In our previous work, we proposed a self-organized data aggregation technique for collecting data from nodes efficiently, which can automatically accumulate data from cluster members to a limited number of cluster members called aggregators. The proposed scheme was developed based on the evolutionary game theoretic approach, in order to take account of the inherent selfishness of the nodes for saving their own battery life. The number of aggregators can be controlled to a desired value by adjusting the energy that the message ferry supplies to the aggregators. In this paper, we further examine the proposed system in terms of success of data transmission and system survivability. We first introduce a new type of game model with retransmissions. Through both theoretic and simulation approaches, we then reveal feasible parameter settings which can achieve a system with desirable characteristics: Stability, survival, and successful data transfer.

Keywords: delay tolerant networks (DTNs), evolutionary game theory, self-organized, aggregators, message ferry.

1 Introduction

In ambient information society, it is expected that each user can automatically obtain its desired information from environments equipped with a numerous number of devices. The underlying network supporting the ambient information society can be regarded as a kind of delay tolerant networks (DTNs) [1,2] due to lack of reliable continuous end-to-end connectivity. In DTNs, a *store-carry-forward* [1] message delivery scheme and *custody transfer* [3] mechanism are used to confirm reliable transfer of data (bundle) with custody among nodes (devices) by delegating the responsibility of custody-bundle transfer through intermediate nodes in a hop-by-hop manner. The intermediate nodes keeping custody bundles are called *custodians*. Each custodian must reserve a sufficient amount of storage and energy to receive and hold the custody bundles until successful delivery or

delivery expiration of the custody bundles. Due to the lack of the storages, custodians sometimes face storage congestion where they must refuse to receive any custody bundle from other nodes. In addition, each battery powered node must be awake while holding the bundles. Since each custodian also generates its own custody bundles, it is naturally selfish in behavior and rejects requests of custody transfer from other nodes to save its storage as well as its energy. Intuitively, this problem increases in long-term isolated networks.

In such a situation, some movable vehicles referred to as *message ferries* [9,10] can solve the storage congestion problem by actively visiting the network and gather bundles from the custodians. Note that the message ferry has a sufficient amount of storages and energy to carry the bundles to the corresponding destination, i.e., a base station referred to as *sink node*, and it can also supply energy to the nodes if required. When there are several isolated networks referred to as *clusters*, the message ferry must visit each of the cluster and collects bundles from the custodians.

In such kind of scenarios, however, sometimes it is difficult for the message ferry to visit all of the nodes in a certain period of time. Taking account of the challenges, we developed a self-organized data aggregation technique in [4]. With the help of the evolutionary game theoretic approach [6,7,8], our system can automatically select some special custodians referred to as *aggregators*, which are cooperative in nature and willingly hold custody bundles of other nodes referred to as *senders*. Therefore, the message ferry needs to collect the bundles only from the aggregators. Note here that in this scheme, each aggregator must keep awake to receive and hold the bundles until transferring them to the message ferry, while each sender awakes only when generating and sending the bundles. In addition, each aggregator can obtain energy supply from the message ferry only when it finds a sender as its neighbor. In our scheme, each node appropriately selects strategy, i.e., sending or aggregating, depending on neighbors' strategies. This interaction among nodes is modeled as a game in game theory. The detail will be given in succeeding sections.

In this paper, we further examine the characteristics of the proposed scheme by focusing on unevaluated viewpoints in our previous work. We first introduce a new type of game model taking account of bundle retransmission when a sender cannot find an aggregator as its neighbor. Then, we evaluate the system stability through theoretic analysis based on replicator dynamics. Since the replicator dynamics only focuses on the strategy distribution, we further consider condition for system survivability. To grasp the node-level behavior, we also apply agent-based dynamics which is a simulation-based approach. Through simulation experiments, we evaluate the validity of the theoretic analysis and reveal feasible parameter settings to achieve successful bundle transfer.

The rest of the paper is organized as follows. We introduce our self-organized data aggregation scheme in section 2. Section 3 gives theoretic analysis of the system dynamics and the stable condition with the help of replicator equation on graphs. We also discuss the system survivability in section 3. After a brief introduction of agent-based dynamics, we evaluate the validity of the theoretic

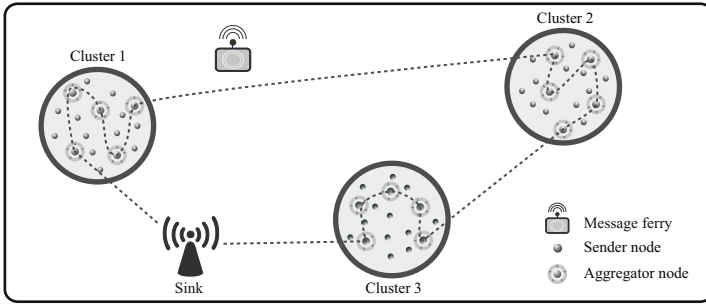


Fig. 1. Model scenario: Message ferry visits a limited number of aggregators in each cluster and delivers collected bundles to the sink node

analysis and reveal feasible parameter settings achieving high successful probability of bundle transfer through simulation experiments in section 4. Finally, section 5 concludes this paper.

2 Self-organized Data Aggregation

We assume that a fixed sink node collects bundles from nodes in isolated clusters with the help of the message ferry as shown in Fig. 1. Each node in a cluster can directly/indirectly communicate with other cluster members (neighbors) within the transmission range but cannot communicate with the sink node and/or nodes in other clusters due to the long distances among the clusters. The message ferry serves the inter-cluster communication and visits only a limited number of aggregators in each cluster.

In this paper, we focus on bundle aggregation in a cluster. The bundle aggregation is conducted through three phases: a) Aggregator selecting phase, b) bundle aggregating phase, and c) bundle collecting phase. These three phases are repeated at each node and the unit of the repetition is referred to as *round*. Initially, each node randomly chooses to be an aggregator or a sender because it cannot know the neighbors' behavior. In the succeeding rounds, each node selects their role depending on the results of the previous round with the help of evolutionary game theory. During bundle aggregating phase, each sender transmits its bundles to one of the aggregators within the transmission range. Then, in bundle collecting phase, each aggregator allows to transmit its service request to the message ferry, transfers all bundles, and obtains energy supply from the ferry.

Due to the lack of reliable connectivity among arbitrary nodes, it is difficult to achieve a centralized control in DTNs. Therefore, the selection of aggregators should be performed in a decentralized way. Note that each node communicates only with its neighbors within the transmission range and is synchronized with each other. It determines to be an aggregator or a sender by mutual interaction based on its own benefit depending on the surrounding conditions. Since aggregators are always awake during a round while senders only wake up when

Table 1. Abstracted payoff matrix.

node 2 \ node 1	send	aggregate
send	R, R	S, T
aggregate	T, S	P, P

Table 2. Payoff matrix for no-retransmission case.

node 2 \ node 1	send	aggregate
send	$-s, -s$	$-s, b-c$
aggregate	$b-c, -s$	$-c, -c$

Table 3. Payoff matrix for retransmission case.

node 2 \ node 1	send	aggregate
send	$-c, -c$	$-s, b-c$
aggregate	$b-c, -s$	$-c, -c$

generating and transmitting their bundles, it is assumed that each node loses energy proportional to the length of time it keeps awake. Let c and s denote the amount of energy consumption for aggregators and senders, respectively, per round. s increases with retransmissions times but never exceeds c , i.e., $c \geq s > 0$. The energy supplied by the message ferry to each aggregator is represented by b . Intuitively, the larger b is, the more the aggregators increases. $b > c$ should be satisfied to keep battery of nodes alive.

The interaction among nodes is modeled as a game between two neighboring nodes in evolutionary game theory and is summarized as a payoff matrix. There are two roles (strategies) for each node: Aggregator (aggregate) and sender (send). There are four possible combinations of the strategies of the two nodes where each node obtains different payoff from each combination of strategy. Table 1 illustrates the abstracted payoff matrix while Tables 2 and 3 illustrate the payoff matrices for no-retransmission case and retransmission case, respectively. Note that no-retransmission case is same as that proposed in our previous work [4]. The resulting payoffs for each combination can be modeled by taking the energy supply and energy consumption into account. If both nodes select to be aggregators, they lose the largest energy $P=c$ without any energy supply from the message ferry, because they are not be able to collect a sufficient number of bundles to request the message ferry to visit. An aggregator paired with a sender obtains the largest energy $T=b-c$; it loses c but obtains b from the message ferry, while the corresponding sender loses the smallest energy $S=s$. When both nodes select to be senders, two possible cases can take place depending on the presence of retransmission. For retransmission case, both of the senders consume $R=c$. This is equivalent to the worst case where each sender spends all the period of a round on achieving successful bundle transfer using retransmission mechanism. Note that we assume failure of bundle transfer is mainly caused by mismatch of waking time of sending and receiving nodes.

We obtain $T > S > R = P$ and $T > S = R > P$ for retransmission case and no-retransmission case, respectively. Every node not only has a temptation to be an aggregator ($T > R$) but also a fear to be an aggregator ($S > P$). The larger b is, the more the temptation is. This indicates that the sink node can control the number of aggregators (senders) by changing b . On the other hand, condition $T > R$ and $S > P$ indicate that taking a strategy different from the

opponent is better than taking the same strategy as the opponent. As a result, both aggregating and sending strategies stably coexist [6]. Thus, with the help of the payoff-matrix and evolutionary game theory, when each node undertakes suitable strategies to optimize its own payoff, then the system converges to a fully stable situation where both senders and aggregators stably coexist.

3 Theoretical Analysis

In this section, we first analyse the relationship between the ratio of the number of aggregators and the parameters of the payoff matrix through replicator dynamics of evolutionary game theory on graph [6,7,8]. The basic concept of replicator dynamics is that the growth rate of nodes taking a specific strategy is proportional to the payoff acquired by the strategy, and the strategy that yields more payoff than the average payoff of the whole system increases. Replicator dynamics on graphs additionally takes account of the effect of the topological structure of the network which is suitable for DTNs. Moreover, we discuss a condition for the system survivability, under which each node can permanently be alive without battery shortage. Finally, some numerical results will be given.

3.1 Replicator Equation on Graphs

First, we introduce the replicator equation on graphs [7] for no-retransmission case, which was originally obtained in [4]. Let x_1 denote the ratio of the number of aggregators to the total number of cluster members for no-retransmission case. Note that $1 - x_1$ represents the ratio of the number of senders. Let k denote the number of neighbors of each node, called degree. For non-regular graphs, k represents the average degree. With the help of the payoff matrix in Table 2, the replicator equation on graphs for no-retransmission case becomes

$$\dot{x}_1 = x_1(1 - x_1) \left[\frac{b(k^2 + k - 3) - (c - s)(k^2 + 2k)}{(k + 3)(k - 2)} - bx_1 \right], \quad k > 2.$$

Substituting $\dot{x}_1 = 0$ yields three equilibria: $x_1^* = 0, 1$, and

$$x_1^* = \frac{b(k^2 + k - 3) - (c - s)(k^2 + 2k)}{b(k + 3)(k - 2)}, \quad k > 2. \tag{1}$$

Note that the equilibrium in Eq. (1) is feasible if $0 < x_1^* < 1$, i.e.,

$$\frac{k^2 + 2k}{k^2 + k - 3} < \frac{b}{(c - s)} < \frac{k^2 + 2k}{3}, \tag{2}$$

satisfies. We have for all $k > 2$, $0 < (k^2 + 2k)/(k^2 + k - 3) < (k^2 + 2k)/3$. Also $c - s > 0$ always holds. As a result, for any c, s , and k , there exists $b > 0$ which satisfies Eq. (2). Thus the equilibrium in Eq. (1) is controllable. Further, x_1^* in Eq. (1) is stable because $\dot{x}_1 > 0$ if $0 < x_1 < x_1^*$, and otherwise, $\dot{x}_1 < 0$.

Similarly, for retransmission case, with the help of the payoff matrix in Table 3, the stable and controllable equilibrium becomes

$$x_2^* = \frac{b(k^2 + k - 3) - 3(c - s)}{(b + c - s)(k + 3)(k - 2)}, \quad k > 2,$$

which is valid for

$$\frac{3}{k^2 + k - 3} < \frac{b}{(c - s)} < \frac{k^2 + k - 3}{3}. \quad (3)$$

In what follows, we call Eqs. (2) and (3) as *stable conditions*.

Note that at the equilibrium the ratio of the number of aggregators is fixed but the role (strategy) of each node may change [4]. This feature is suitable for our system such that each node can acquire opportunities to send bundles and obtain energy supply by changing its role (strategy) round by round.

3.2 Valid Parameter Settings for Permanently Alive System

Although we mentioned that each node has a chance to obtain energy supply by changing its role round by round, careful parameter tuning is required to achieve high system survivability. At the equilibrium, it is expected that each sender (aggregator) can find at least one aggregator (resp. sender) as its neighbor. Thus, expected payoff for each node becomes $E[p] = (b - c)x_i^* - s(1 - x_i^*)$ ($i = 1, 2$). If $E[p]$ is positive, the system could survive without loss of any node. Therefore, the valid combinations of b , c and s should satisfy the following condition for positive payoff (referred to as *running condition*):

$$E[p] = (b - c)x_i^* - s(1 - x_i^*) > 0 \quad (i = 1, 2). \quad (4)$$

In practice, the sink node tries to find appropriate x_i^* ($i = 1, 2$) which satisfies both the stability condition and the running condition. The amount of energy supply from the message ferry, b , can be fully controlled by the sink node while c and s seem to be partly controllable: They are proportional to the length of waking period. The average node degree, k , is given from the environment. As a result, the sink node achieve desirable x_i^* ($i = 1, 2$) by mainly controlling b . In the next subsection, we show some numerical results to illustrate the feasible parameter settings.

3.3 Numerical Results

In this section, we show some numerical examples of the adequate parameter settings satisfying the stable and/or running conditions according to the theoretic analysis in sections 3.1 and 3.2. First, we clarify the impact of stable and/or running conditions and the effect of retransmission mechanism. Fig. 2 depicts the valid range of controllable benefit b as a function of k when $c = 10$ and $s = 0.1$. Fig. 3 illustrates the corresponding range of x_i^* ($i = 1, 2$). Note that we

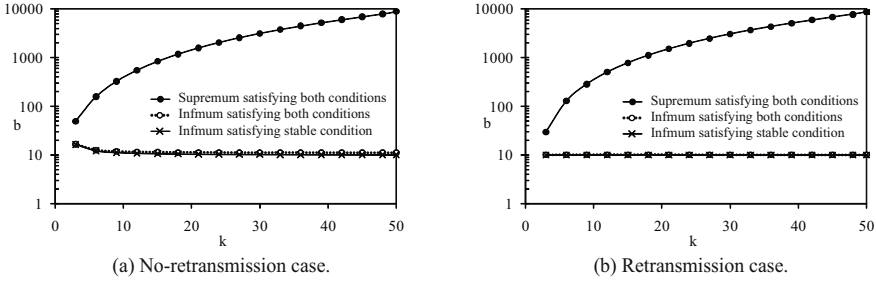


Fig. 2. The supremum and infimum of b satisfying stable and/or running conditions ($c=10, s=0.1$)

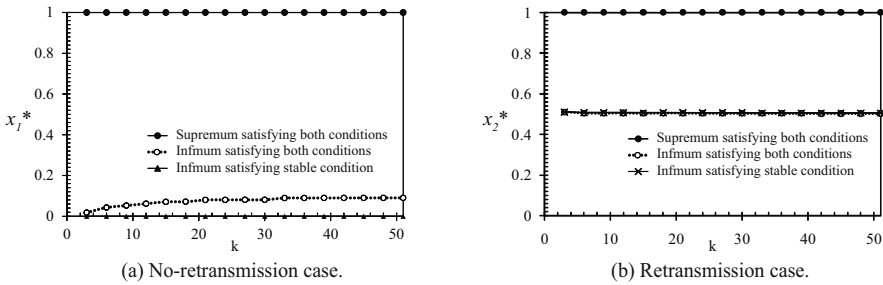


Fig. 3. The supremum and infimum of x_i^* ($i=1,2$) satisfying stable and/or running conditions ($c=10, s=0.1$)

show the results for larger k to reveal the basic characteristics even though they rarely occur in actual situations. We observe that the supremum of b increases with k and its has the same characteristic for both conditions. On the contrary, the infimum is almost constant while satisfying the two conditions. Although the presence of retransmission does not almost affect the valid range of b , Fig. 3 indicates that the retransmission mechanism narrows the valid range of equilibrium compared with no-retransmission case. Specifically, x_2^* must be greater than 0.505 to satisfy both conditions in retransmission case.

Next, we reveal how c and s affect the valid range of b and x_i^* ($i = 1, 2$). Fig. 4 illustrates the supremum and infimum of b satisfying stable and/or running conditions when c and s vary. We observe that for a specific k , the range of b shifts up with c . This simply means that $b - c$ should be positive. On the contrary, increase of s decreases the supremum of b . This is because when senders lose more energy, temptation b to become an aggregator can be smaller.

Fig. 5 presents the supremum and infimum of x_i^* ($i = 1, 2$) corresponding to Fig. 4. We observe that s has more impact on infimum than c . This is mainly caused by running condition. From Eq. (4), keeping low energy consumption of a sender is important for prolonging the battery life. We also find that no-retransmission case has wider feasible area than that with retransmission.

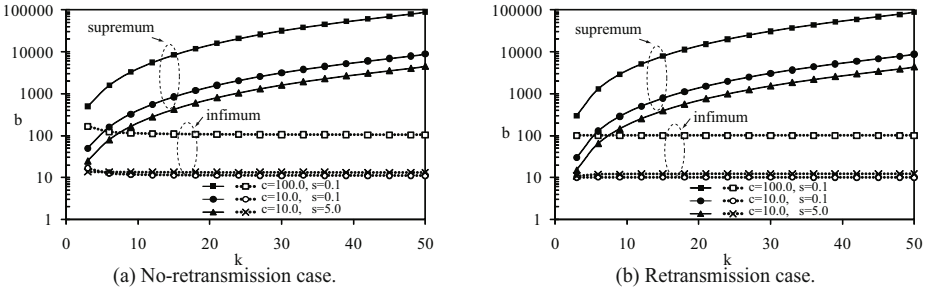


Fig. 4. The supremum and infimum of b satisfying stable and/or running conditions

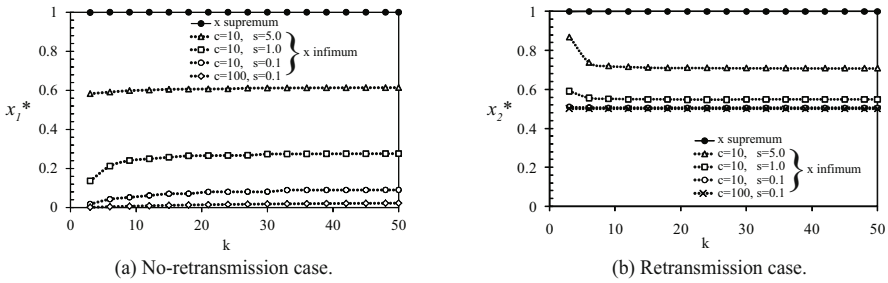


Fig. 5. The supremum and infimum of x_i^* ($i = 1, 2$) satisfying stable and/or running conditions

4 Simulation Experiments by Agent-Based Dynamics

4.1 Agent-Based Dynamics

In evolutionary game theory, the replicator dynamics predicts the macro-level system behavior which explains the effects of the corresponding parameters behaviors. On the other hand, the complementary method: The agent-based dynamics is used to understand the micro-level system behavior of the evolutionary game theory. It explains that with mutual interactions among neighboring nodes a superior strategy spreads over the network in a hop-by-hop manner.

In agent-based dynamics, in every round, each agent, i.e., node, first interacts with neighboring nodes and determines its strategy for the next round by comparing its own payoff with that of a randomly chosen neighboring node. The strategy update of node is conducted in a probabilistic manner where the more a strategy acquires the payoff, the more it spreads over the network through the imitation process in a hop-by-hop manner [4].

In what follows, we conduct simulation experiments for two purposes: 1) validation of analytical results, and 2) evaluations which are derived from micro-level system behavior.

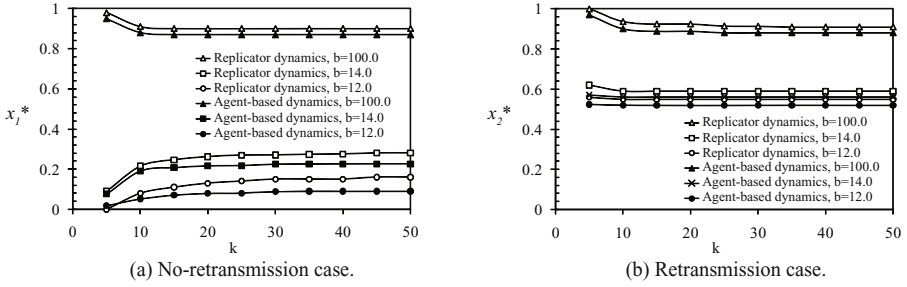


Fig. 6. Validity of the theoretic analysis ($c = 10$, $s = 0.1$, $k = 5$)

4.2 Simulation Models

The simulation experiments were conducted with a multi-agent programmable modeling simulator NetLogo [5] over unit disk graphs. The unit disk graphs are suitable for abstracting wireless networks because they are generated by randomly located nodes in 2-dimensional space where two nodes are adjacent if the distance between them is equal or less than a certain threshold, i.e., transmission range of each node. We set the number N of nodes to 100. The area size was set to 1×1 [km²], and the transmission range of each node was set to 100 [m] in default. We controlled the average degree k by adjusting the transmission range of each node adequately. We assumed that the duration of a round was fixed and each node periodically generated a fixed number of bundles per round. Therefore, for simplicity, we set $c = 10$ and $s = 0.1$ in our simulation experiments. In the following figures, the average of 100 independent simulation experiments are plotted.

4.3 Validity of the Theoretic Analysis

Fig. 6 illustrates both analytical and simulation results for three examples selected from the valid parameter settings satisfying both stable and running conditions in the analysis. We observe that the analytical results slightly differ from the simulation results in both cases. These differences come from the relatively small system scale ($N = 100$) and the variance of node degree in the unit disk graphs. Since these characteristics naturally exist in the real networks, we can conclude that the analytical results can predict the system behavior with a certain degree of accuracy.

Next, we evaluate the validity of running condition, Eq. (4). As discussed in Section 3.2, all nodes can survive forever under appropriate values of the parameters b , c , and s . Fig. 7 shows the transition of the number of nodes with positive cumulative payoffs for different x_i^* ($i = 1, 2$). Note that every node initially has no payoff. Each node obtains energy supply from the message ferry only when it acts as aggregator and has at least one sender as its neighbor. Our aim is to achieve all nodes having positive cumulative payoffs such that

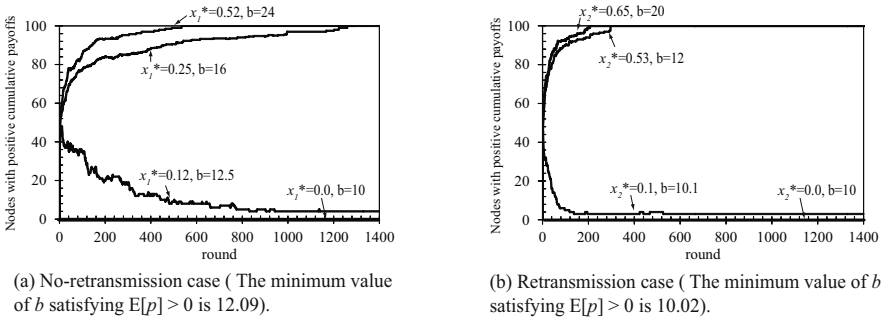


Fig. 7. Transition of the number of nodes with positive cumulative payoffs ($c = 10$, $s = 0.1$, $k=5$)

they can work permanently if they have a sufficient amount of initial battery which depends on the parameter settings. Given $c = 10, s = 0.1, k = 5$, through theoretic analysis, minimum b , b_{\min} , satisfying Eq. (4), becomes 12.09 for no-retransmission case and 10.02 for retransmission case, respectively. As shown in Fig. 7, we observe that these theoretical predictions approximately accord with the simulation results except for b close to b_{\min} .

4.4 Successful Bundle Transfer

For senders (aggregators), it is desirable that at least one aggregator (resp. sender) exists as a neighbor for successful bundles transferring. We define the probability of senders (aggregators) that have at least one aggregator (resp. sender) in their neighbors as *sender (resp. aggregator) success probability*. These probabilities are affected not only by b, c , and s but also by k . Fig. 8 depicts the probabilities as a function of k . At first, Figs. 8 (a) and 8 (b) show that as k increases, the probabilities almost become one. This is because each node has more neighbors in average if k increases. Note that we omit the sender successful probability in Fig. 8(a) because senders always success in bundle transfer independent of neighbors' strategies with the help of the retransmission mechanism.

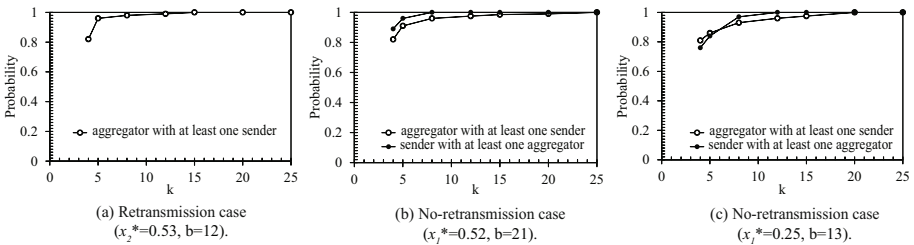


Fig. 8. Sender and aggregator successful probabilities satisfying stable and running conditions ($c = 10, s = 0.1$)

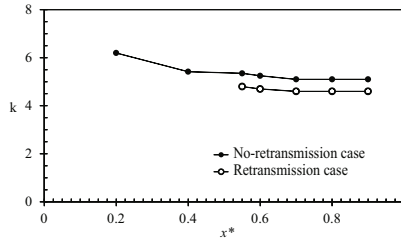


Fig. 9. Minimum k which satisfies sender and aggregator successful probabilities equal or greater than 0.9, and stable and running conditions ($c = 10, s = 0.1$)

Next, comparing Figs. 8 (b) and 8 (c), we observe that small x_1^* decreases the probabilities when k is small. Note that we want to keep x_1^* relatively low as mentioned above but these results indicates that small x_1^* and k do not necessarily satisfy the probabilities close to one. To clarify this, Fig. 9 illustrates the minimum k which satisfies both probabilities ≥ 0.9 as a function of x_i^* ($i = 1, 2$) where b is set adequately. We observe that the minimum k increases when x_i^* ($i = 1, 2$) decrease but is kept relatively low. This can be confirmed from the fact that k should be greater than $1/x_i^*$ ($i = 1, 2$) for senders to have at least one aggregator in their neighbors.

5 Conclusion

In this paper, we further examined the characteristics of the self-organized data aggregation scheme proposed in 4. We first introduced a new game model taking account of bundle retransmission when a sender cannot find an aggregator as its neighbor. Then, we derived the stable conditions through theoretic analysis based on replicator dynamics on graphs. In addition, we discussed running condition where all nodes can survive without battery outage. To evaluate the validity of theoretic analysis and reveal feasible parameter settings achieving successful bundle transfer, we conducted simulation experiments using agent-based dynamics. Both theoretic and simulation results presented appropriate parameter settings to achieve a system with desirable characteristics: Stability, survivability, and success probability in bundle transfer.

Acknowledgment. This research was supported by “Global COE (Centers of Excellence) Program” of the Ministry of Education, Culture, Sports, Science and Technology, Japan, as a part of “Ambient Networking” research group, Grant-in-Aid for Young Scientists (B) of JSPS under Grant No. 22700070, and Grant-in-Aid for Scientific Research (C) of JSPS under Grant No. 22500056.

References

1. Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., Weiss, H.: Delay tolerant network architecture. Work in progress as an IETF RFC 4838 draft, <http://www.ietf.org/rfc/rfc4838.txt>
2. Fall, K.: A delay-tolerant network architecture for challenged internets. In: ACM SIGCOMM, pp. 27–34 (2003)
3. Fall, K., Hong, W.: Custody transfer for reliable delivery in delay tolerant networks. Tech. Rep. IRB-TR-03-030, Intel Research Berkeley (2003)
4. Kabir, K.H., Sasabe, M., Takine, T.: Evolutionary game theoretic approach to self-organized data aggregation in delay tolerant networks. IEICE Transactions on Communications E93-B(3), 490–500 (2010)
5. NetLogo, version 4.1, <http://ccl.northwestern.edu/netlogo>
6. Nowak, M.A.: Evolutionary dynamics: exploring the equations of life. Harvard University Press (2006)
7. Ohtsuki, H., Nowak, M.A.: The replicator equation on graphs. Journal of Theoretical Biology 243(1), 86–97 (2006)
8. Ohtsuki, H., Hauert, C., Lieberman, E., Nowak, M.A.: A simple rule for the evolution of cooperation on graphs. Nature 441(7092), 502–505 (2006)
9. Zhao, W., Ammar, M., Zegura, E.: A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: ACM MobiHoc, pp. 187–198 (2004)
10. Zhao, W., Ammar, M., Zegura, E.: Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In: IEEE INFOCOM, pp. 1407–1418 (2005)

An Inter-networking Mechanism Using Stepwise Synchronization for Wireless Sensor Networks

Hiroshi Yamamoto, Naoki Wakamiya, and Masayuki Murata

Graduate School of Information Science and Technology, Osaka University,
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

{hiroshi-y,wakamiya,murata}@ist.osaka-u.ac.jp

Abstract. To realize the ambient information society, multiple wireless networks deployed in the region or carried by users are required to cooperate with each other. Since duty cycles and operational frequencies are different among networks, we need a mechanism to allow networks to efficiently exchange messages. In this paper, we propose a novel inter-networking mechanism where two networks are synchronized with each other in a moderate manner, which we call stepwise synchronization. With our proposal, nodes near the border of networks adjust their operational frequencies in a stepwise fashion to bridge the gap between intrinsic operational frequencies. For this purpose, we adopt the pulse-coupled oscillator model as a fundamental theory of synchronization. Through simulation experiments, we show that the communication delay and the energy consumption of border nodes are reduced, which enables wireless sensor networks to communicate longer with each other.

Keywords: Wireless Sensor Network, Synchronization, Pulse-Coupled Oscillator Model.

1 Introduction

The ambient information society is the concept and framework where intelligent environment detects, reasons, and satisfies overt and potential demands of people without their interaction [1,2,3]. In the ambient information society, people do not need to be aware of existence of networked information devices embedded in the environment. They do not need to intentionally access a network to control the environment to make it comfortable and satisfy their demands. Instead, the embedded network controls the environment and provides personalized information services to a user taking into account time, place, occasion, and person.

To realize the ambient information society, networks deployed and operating in the same environment must cooperate with each other in exchanging information, sharing information, and controlling each other. For example, a person has a wireless body area network which consists of vital sensors, accelerometers, PDA, and other devices. On the other hand, a room has embedded wired and wireless networks which consist of sensors and actuators for environmental control for example. Intelligent home appliances also constitute embedded networks. When the person enters the room, those networks

should cooperate with each other for smart environmental control. However, in general, those devices organize different and independent networks operating on different control policies, e.g. operational frequency. Therefore, so that the room provides the person with a comfortable environment, we need a mechanism for different networks to smoothly and dynamically connect and share their information. However, it is not a trivial task.

There are several proposals on dynamic composition of multiple networks [4,5]. In [4], they consider a mechanism for overlay networks to dynamically compose a hierarchical structure by two types of composition schemes, i.e. absorption and gatewaying. In [5], cooperation between wireless networks is accomplished by organizing an overlay network by connecting gateway nodes belonging to different wireless networks. Although they can be applied to ambient information networking to some extent, they have a major problem that they do not take into account the difference in operational policies, more specifically, operational frequencies of different wireless sensor networks.

In general, wireless sensor networks adopt a sleep scheduling or duty cycling mechanism to save energy. Operational frequencies, that is, frequencies that they wake up and resume operation, are different among networks depending on application's requirement and characteristics of devices. For example, an air conditioner would obtain and use the temperature information every minute to adjust its thermostat. On the other hand, devices to detect locations of people have to report their detection result very frequently at an order of seconds. When they want to exchange information among them for intelligent control of room temperature to intensively regulate the temperature around a person in the room, a node belonging to the location detection system has to stay active in order to wait for a node belonging to the thermal management system to wake up in transmitting a message. Even when an energy-efficient MAC protocol such as S-MAC [6] and X-MAC [7] is used, such communication consumes the substantial energy at the former node and it would bring danger of energy depletion.

Our research group considers stepwise synchronization between wireless sensor networks for smooth and moderate inter-networking. In [8], the concept of stepwise synchronization is introduced, where sensor nodes located near the border of two networks adjust their operational frequencies to bridge the gap in their intrinsic operational frequencies. Since only nodes near the border change their operational frequency, the remaining nodes can keep their frequency and thus energy consumption in inter-networking can be reduced. The stepwise synchronization is self-organized based on a nonlinear mathematical model of synchronization of oscillators, called the pulse-coupled oscillator (PCO) model [9]. The PCO model describes emergence of synchronization in a group oscillators with different frequencies through mutual interactions through stimuli. By adopting the PCO model to scheduling, operational frequencies of nodes can be appropriately adjusted without any centralized control in wireless sensor networks. However, in [8], only an idea of stepwise synchronization is shown and no detailed description on mechanisms is given.

Therefore, in this paper, we propose a realistic mechanism of stepwise synchronization for inter-networking among wireless sensor networks with different operational frequencies. In our mechanism, we strengthen the degree of entrainment at border nodes to intensively shift the operational frequency toward that of the other network while

the degree of entrainment is weakened as the distance to the border increase. As a result, the operational frequencies of nodes near the border are adjusted to somewhere between the original operational frequencies of wireless sensor networks. Through simulation experiments, we verify the practicality of our proposal in comparison with the case where each of networks keeps its intrinsic operational frequency.

The rest of this paper is organized as follow. First in section 2, we explain the pulse-coupled oscillator model. Next in section 3, we describe the details of our proposal. In section 4, we show and discuss results of our simulation experiments. Finally, we conclude the paper in section 5.

2 Pulse-Coupled Oscillator Model and Synchronization

A pulse-coupled oscillator model is a mathematical model which explains synchronized flashing of a group of fireflies [9]. It is considered that a firefly maintains a biological timer, based on which it intermittently flashes. The flashing frequency depends on its intrinsic timer frequency, which could be different among individuals. However, when fireflies form a group, they begin to flash in synchrony. A mechanism of biological synchronization is explained as follow. When a firefly observes a flash of another firefly, it is stimulated and its timer advances by a small amount. Because of nonlinearity in timer or stimulus, by repeatedly stimulating each other, their timers begin to expire synchronously, then flash at the same time. Among PCO models [9][10][11], in this paper we use the model proposed in [9].

In the PCO model [9], oscillator i maintains phase ϕ_i ($0 \leq \phi_i \leq 1$) of a timer and state x_i ($0 \leq x_i \leq 1$) given by a function of phase. The dynamics of phase ϕ_i is determined by the following differential equation.

$$\frac{d\phi_i}{dt} = F_i \tag{1}$$

where F_i stands for the intrinsic timer frequency of oscillator i . State x_i is determined from phase ϕ_i by the following monotonically increasing nonlinear function,

$$x_i = \frac{1}{b} \ln[1 + (e^b - 1)\phi_i] \tag{2}$$

where b ($0 < b$) is a dissipation parameter that dominates the rate of synchronization.

When phase ϕ_i and state x_i reach 1, oscillator i fires and both phase ϕ_i and state x_i go back to 0. When an oscillator fires, the oscillator stimulates oscillators that are coupled with the firing oscillator. If oscillator j is stimulated by oscillator i at time t , oscillator j increases its state x_j by a small amount ϵ and phase ϕ_j changes accordingly as

$$x_j(t^+) = B(x_j(t) + \epsilon), \tag{3}$$

where

$$B(x) = \begin{cases} x & (0 \leq x \leq 1) \\ 0 & (x < 0) \\ 1 & (x > 1) \end{cases}$$

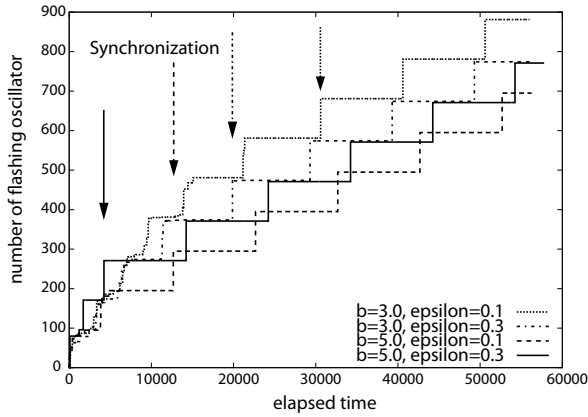


Fig. 1. Cumulative number of flashing oscillators

and

$$\phi_j(t^+) = \frac{e^{bx_j(t^+)} - 1}{e^b - 1} \tag{4}$$

When state $x_j(t^+)$ and phase $\phi_j(t^+)$ reach 1 by being stimulated, oscillator j also fires. At this time, oscillators i and j are considered synchronized. To avoid overshoot and instability, an oscillator is not stimulated by two or more oscillators at the same time, and an oscillator is not stimulated at the time when it fires.

In Fig. 1 we show how the cumulative number of flashing oscillators changes with different parameters b and ϵ against time. 100 nodes are arranged in a 10×10 grid. Each node is coupled with neighboring nodes in up, right, down, and left directions. The operational frequency of nodes is identical among nodes and their initial phase is set at random. The height of stepwise increase in the number corresponds to the number of oscillators simultaneously flashing. As indicated by arrows, the time when a group of oscillators reach synchronization and begins to flash in synchrony is about 30,000 with “ $b=0.1, \epsilon=3.0$ ”, about 20,000 with “ $b=3.0, \epsilon=3.0$ ”, about 13,000 with “ $b=1.0, \epsilon=5.0$ ”, and about 4,000 with “ $b=0.3, \epsilon=5.0$ ”, respectively. When we adopt larger b and ϵ , the speed of synchronization apparently increases. Although delay is not taken into account in drawing the figure, it is known that the synchronization is accomplished in the environment with loss and delay of stimuli [12][13].

3 Inter-networking Mechanism Using Stepwise Synchronization

In applying the PCO model to synchronization, a wireless sensor node corresponds to an oscillator. It stimulates neighbor nodes in the range of radio signals by broadcasting a message. A message is used for both of synchronization and data communication with such a mechanism where control messages for synchronization are embedded in messages for sensor data [14].

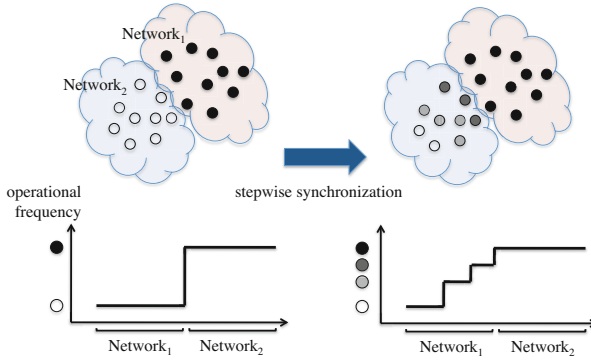


Fig. 2. Stepwise synchronization

Node i maintains state x_i and phase ϕ_i as variables of a timer of frequency F_i and calculates its new state and phase at regular intervals, e.g. at the granularity of timer. When state x_i and phase ϕ_i reach 1, node i sets both state x_i and phase ϕ_i at 0 and tries to broadcast a message which informs neighbor nodes that the node fires. Since a wireless channel is the shared medium, there is possibility that broadcasting is delayed to wait for the channel to become available. However, from our previous experiments, the influence of delay on synchronization is negligible [12]. When a node receives a broadcast message, it is stimulated. The stimulated node, say node j , increments its state x_j by a small amount ϵ by Eq. (3) and calculates new phase ϕ_j^+ based on the new state x_j^+ by using Eq. (4). If the new state x_j^+ and new phase ϕ_j^+ reach 1, node j also fires and broadcasts a message. Since duty cycling is adopted on a node, only neighboring nodes that are awake can receive stimuli. Details of integration of duty cycling and the PCO model will be given later.

Now we propose a stepwise synchronization-based inter-networking mechanism. In our mechanism, we assume that two or more wireless sensor networks operating in different operational frequencies, at which nodes wake up and resume operation by duty cycling, co-exist and nodes originally belonging to the same network are synchronized to the same frequency by a PCO-based synchronization mechanism. A node can communicate with any nodes in its communication range independently of whether they belong to the same network or not, as far as they are awake and ready to communicate. A node can know the distance, i.e. the number of hops, from the border of networks by using a mechanism which will be given later.

As an example, in Fig. 2, two wireless sensor networks with different operational frequencies are adjacent, and they attempt to cooperate. When we couple border nodes to let them stimulate each other, two wireless sensor networks will be unified to a single network with the operational frequency identical to the faster one. If the difference in the operational frequency is too large, they will remain independent. Therefore, we need a new mechanism to accomplish stepwise synchronization where only a part of network is involved in the synchronization and those networks with largely different operational frequencies can be synchronized. For this purpose, we adjust the degree of entrainment in accordance with the distance to the border. We focus on the fact that the dissipation b

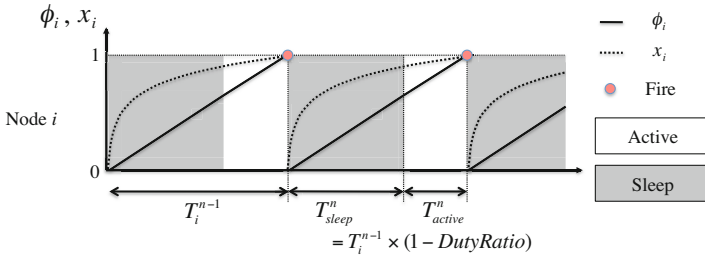


Fig. 3. Duty cycling

and the stimulus ϵ influence the degree of entrainment and the speed of synchronization (see Fig. 1).

In our proposal, we set larger values of b and ϵ , e.g. $b = 4.0$ and $\epsilon = 0.3$, at nodes located at the border of wireless sensor networks to strengthen entrainment and shift the operational frequency much. By receiving stimuli from the other network, nodes located at the border of wireless sensor networks actively changes their operational frequencies for larger parameters. Then smaller values are applied to nodes as the distance to the border becomes larger, e.g. $b = 3.5$ and $\epsilon = 0.1$. Nodes distant from the border of wireless sensor networks are also entrained by receiving stimuli from nodes located at the border, but the impact is smaller for smaller parameters and thus their operational frequencies stay rather closer to the original frequency. Consequently, we observe a stepwise change in operational frequencies around the border of two networks as illustrated in Fig. 2. Such stepwise synchronization can bridge the large gap in operational frequencies which cannot be overcome by the PCO model alone.

Now, we describe details of our proposal. Figure 3 shows the duty cycling in our proposal. Node i maintains state x_i and phase ϕ_i of a timer of frequency F_i . The phase automatically advances and the state accordingly changes independently of whether it is awake or sleeping. When state x_i and phase ϕ_i reach 1, node i sets both state x_i and phase ϕ_i at 0 and tries to broadcast a stimulus message. After the duration required to broadcast a stimulus message, node i goes to sleep for $T_{sleep}^n = T_i^{n-1} \times (1 - DutyRatio)$ independently whether it could successfully broadcast a stimulus message or not. T_i^{n-1} , called operational interval, is defined as the duration from the $n - 1$ -th firing timing to the n -th firing timing and $DutyRatio$ is the duty ratio parameter which is determined in advance. Although a sleeping node does not receive any stimulus message, there is a case that the state and phase occasionally reach 1. In such a case, a node wakes up to broadcast a stimulus message and after broadcasting it immediately resumes the sleep state for the remainder of T_{sleep}^n . When T_{sleep}^n has passed, the node wakes up and becomes capable of sending and receiving messages for the duration of $T_{active}^n = T_i^{n-1} \times DutyRatio$. When node i receives a stimulus message in its active period, it calculates its new state and phase by Eqs. (3) and (4). If the new state and new phase reaches 1, node i fires, broadcasts a stimulus message, and goes to sleep.

So that other nodes in the network can recognize their relative distance to the border, all nodes at the border, which have received messages from the other network, sets the distance field in the header of stimulus message it broadcasts as 1, meaning that the

message is from a node at the border. By receiving messages, a node can appropriately set its distance by adding 1 to the minimum value in the distance field of received messages. The distance information is also embedded in messages that it broadcasts, so that the distance information propagates through a network. Once a node recognizes its distance to the border, it adjusts b and ϵ in accordance with the distance. In this paper, the relationship between a pair of parameters b and ϵ and the distance is determined from preliminary experiments. A node at the border begins to use value 0 for the distance field, if it has not received any messages from the other network for a certain period of time to notify other nodes of the end of cooperation. Receiving this message, distance information is initialized to 0 on other nodes.

In addition to duty cycling based on the PCO-model, we further adopt duty cycling on the MAC layer. Low power listening (LPL) is an approach widely used in energy-efficient MAC protocols such as X-MAC [7]. With X-MAC, a node periodically wakes up by turning on a transceiver to see whether there is any message destined to itself. The duration that a node is ready for reception is denoted as R_l and the interval between successive active periods is denoted as R_s when the transceiver is off. A sender node that intends to send a message first transmits small messages, called Short Preamble, to notify a receiver of the existence of message. It keeps sending preambles until it receives an ACK for the preambles from an intended receiver. When a receiver, that is, a node that the sender wants to send the message to, wakes up, it would receive one of preambles during its active period. Then, the receiver sends back an ACK message to the sender and extends its active period accordingly. On receiving the ACK, the sender begins to send the message. After receiving the whole message, the receiver sends an ACK again to the sender. In a case of broadcasting, where a message is not intended for any specific node but all nodes in the vicinity, a sender begins to send a message itself repeatedly for the duration of slightly longer than R_s without communication initialization by Short Preamble. There is no acknowledgement either for broadcasting.

4 Performance Evaluation

4.1 Simulation Settings

We arranged 50 nodes in a 16×16 area as shown in Fig. 4. In Fig. 4, nodes in the upper-left area belong to Network 1 and the others do Network 2. Therefore, each of networks has four border nodes located in the overlapping area. A circle shown in Fig. 4 illustrates the communication range of the node centered at the circle and each node can communicate with nodes located within the communication range. Parameters are set as summarized in Table 1. In Table 1, S_{pre} , S_{ack} , S_{stim} and S_{data} stand for durations that a node sends a Short Preamble, an ACK, a stimulus message and a data packet, respectively. The operational interval between successive broadcasting is about 23 to 25 seconds in Network 1 and 130 to 143 seconds in Network 2. Initial states are set at random. Parameters b and ϵ used in cooperation are shown in Table 2 for each number of hop counts from the border. The duty ratio is set at 0.3 at all nodes.

In our simulation, the sink node of Network 1 sends a data message to the sink node of Network 2 by using multihop unicast communication once per 10 operational cycles. Data messages take the shortest path to the receiver node following the diagonal of the

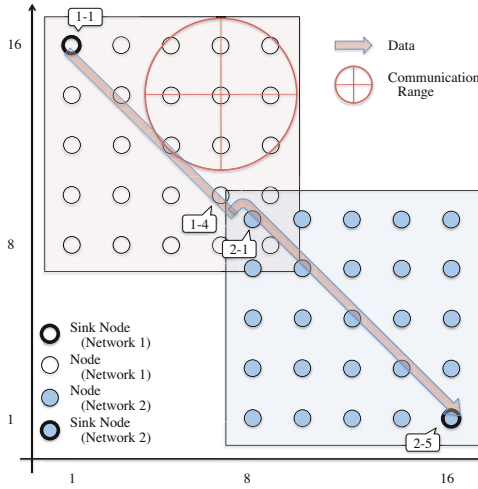


Fig. 4. Node layout in simulation

Table 1. Parameter settings

PCO		X-MAC			
param.	value	param.	value [ms]	param.	value [ms]
b	3.0	S_{pre}	2.0	R_s	200
ϵ	0.1	S_{ack}	2.0	R_l	20
f_1	0.040 - 0.044 [Hz]	S_{stim}	4.0		
f_2	0.007 - 0.0077 [Hz]	S_{data}	4.0		

networks. When a node between the sink nodes is active and receives a data message, it immediately tries sending the message to a next-hop node. It transmits preambles until it receives an ACK from the next-hop node, even after the end of the PCO-based active period, i.e. expiration of timer. When the transmission of the data message is completed, the node begins to sleep if the phase of timer is in the range of the PCO-based sleep period. Otherwise, it keeps awake in the PCO-based active period.

For the purpose of evaluation of energy consumption, we assume that each node is equipped with an Atmel ATmega 128L processor, a Texas Instruments CC2420 radio chip and two AA batteries. The details of energy consumption model is summarized in Table 3.

4.2 Results and Discussion

We compare two scenarios, where both networks keep their intrinsic frequencies denoted as “independent”, and our proposal is adopted denoted as “proposal”. As performance measures, we use communication delay which is defined as the duration between

Table 2. PCO parameters used in cooperation

param.	distance to border				
	border	1 hop	2 hop	3 hop	4 hop
b	4.0	3.5	3.0	2.0	1.5
ϵ	0.3	0.1	0.05	0.05	0.05

Table 3. Energy consumption model

param.	value
Initial energy	2000 [mAh]
Processor active current	8 [mA]
Sleeping current	15 [μ A]
Sending current	9.9 [mA]
Waiting and receiveing current	19.7 [mA]

the time when a node begins to send preambles for transmission of a data message and the time when a node receives an ACK for message reception.

First, we confirm that the stepwise synchronization is accomplished. Figure 5 shows how nodes in Network 2 (slower network) adapt their operational frequencies. Each square corresponds to a node and the color shows resultant operational intervals. In this figure, we see that the operational interval of nodes at the border, i.e. four nodes located upper-left of the network, becomes about 50 [s], closer to the operational interval of Network 1. On the other hand, the operational interval of other nodes become longer than that of those border nodes as the distances to the border become larger. As a result, the stepwise change in operational frequency emerges in accordance with the distance to the border.

Next, we evaluate per-hop communication delay. Figure 6 shows the median of the communication delay of all data messages transmitted in the simulation runs. In Fig. 6, “1-1, 1-2” corresponds to the communication delay from Node(1-1) to Node(1-2), for example. Those nodes from Node(1-1) to Node(1-4) belong to Network 1 (faster network), and those from Node(2-1) to Node(2-5) belong to Network 2 (slower network). Node(1-4), Node(2-1) and Node(2-2) are nodes located at the border of networks. In Fig. 6 in the case of “independent”, although all communication delays between nodes belonging to the same network are quite small, communication delay between nodes located at the border is 23 [s]. It is because Node(1-4) has to wait for Node(2-1) located at the border of Network 2 to wake up. On the other hand, in the case that our proposal is adopted, communication delays between nodes belonging to Network 2 become large. It is because that they do not wake up at the same time any more for different operational frequencies shown in Fig. 5. However, the communication delay between nodes located at the border of networks is reduced. The reason that the communication delay between Node(2-1) and Node(2-2) is small in both cases is that both Node(2-1) and Node(2-2) are located at the border and they are synchronized. As stated above, communication delay results from waiting intransmission, during which a node consumes energy. We next evaluate energy consumption, which is a major concern of a wireless sensor network.

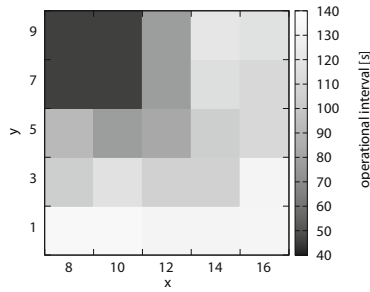


Fig. 5. Results: operational intervals in stepwise synchronization (Network2)

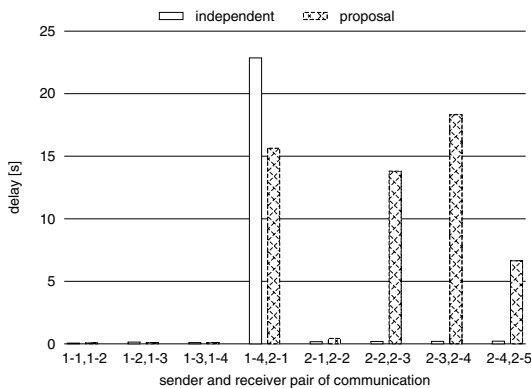


Fig. 6. Results: communication delay of each hop

Finally, simulation results of energy consumption are summarized in Fig. 7. In Fig. 7 in the case of “independent”, although the energy consumption of nodes located inside of networks is almost constant, Node(1-4) located at the border of Network 1 consumes the largest energy among all nodes in waiting for Node(2-1) to wake up. On the other hand, in the case that our proposal is adopted, the amount of energy consumed at the border node, i.e. Node(1-4), is reduced from 10.3 [mAh] to 8.9 [mAh] at the sacrifice of increased energy consumption at nodes in Network 2.

Although the total amount of consumed energy is larger with our proposal than the case of independent networks, we consider that our proposal benefits wireless sensor networks in (i) it balances energy consumption among nodes, which leads to prolongation of the lifetime of border nodes, (ii) it can enable wireless sensor networks with largely different frequencies to synchronize with each other (not shown in this paper), and (iii) since the stepwise synchronization emerges as a consequence of mutual interaction between nodes and there is no deterministic rule to determine stepwise operational frequencies, it can adapt to various situations, e.g. increase in the number of networks to cooperate, cooperation among moving networks, and different degree of cooperation.

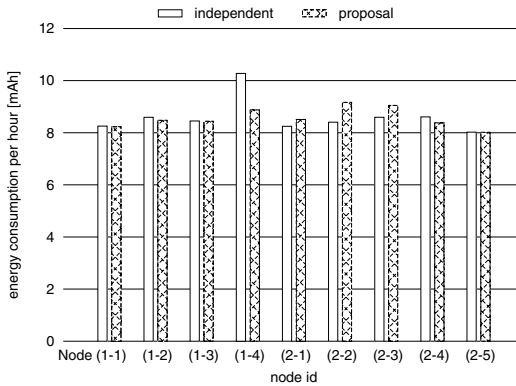


Fig. 7. Results: energy consumption per hour

5 Conclusion

In this paper, to achieve smooth and moderate inter-networking between wireless sensor networks with different operational frequencies, we propose a stepwise synchronization-based inter-networking mechanism. In this mechanism, we adopt the pulse-coupled oscillator model to autonomously accomplish stepwise synchronization. Through simulation experiments, it was shown that the delay in communication between border nodes and the energy consumption at the border nodes were reduced, but at the sacrifice of energy at nodes near the border in the slower network.

Since only preliminary and proof-of-concept evaluation is conducted in the paper, we further plan to comprehensively evaluate the proposal in such scenarios where the difference in operational frequencies is much larger between two networks, there are more than three networks to cooperate, and the degree of overlapping, i.e. the number of border nodes, dynamically changes.

Acknowledgements. This research was partly supported by “Early-concept Grants for Exploratory Research on New-generation Network” of National Institute of Information and Communications Technology, Japan and the “Global COE (Centers of Excellence) Program” and Grant-in-Aid for Scientific Research (B) 22300023 of Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. MEXT Global COE Program, Center of excellence for founding ambient information society infrastructure, <http://www.ist.osaka-u.ac.jp/GlobalCOE>
2. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.C.: Scenarios for Ambient Intelligence in 2010. IST Advisory Group Final Report (February 2001)
3. Preuvencers, D., Van den Bergh, J., Wagelaar, D., Georges, A., Rigole, P., Clerckx, T., Berbers, Y., Coninx, K., Jonckers, V., De Bosschere, K.: Towards an Extensible Context Ontology for Ambient Intelligence. LNCS, pp. 148–159 (October 2004)

4. Kersch, P., Szabo, R., Kis, Z.L.: Self Organizing Ambient Control Space - An Ambient Network Architecture for Dynamic Network Interconnection. In: Proceedings of the 1st ACM Workshop on Dynamic Interconnection of Networks (DIN), Cologne, Germany, pp. 17–21 (September 2005)
5. Poorter, E., Latre, B., Moerman, I., Demeester, P.: Symbiotic Networks: Towards a New Level of Cooperation Between Wireless Networks. *International Journal of Wireless Personal Communications*, 479–495 (June 2008)
6. Ye, W., Heidemann, J., Estrin, D.: An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In: Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), New York, USA, pp. 1567–1576 (June 2002)
7. Buettner, M., Yee, G., Anderson, E., Han, R.: X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. In: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys), Boulder, USA, pp. 307–320 (October 2006)
8. Wakamiya, N., Murata, M.: Dynamic Network Formation in Ambient Information Networking. In: Proceedings of the 1st International Workshop on Sensor Networks and Ambient Intelligence (SeNAml), Dunedin, New Zealand, pp. 443–448 (December 2008)
9. Mirollo, R.E., Strogatz, S.H.: Synchronization of Pulse-Coupled Biological Oscillators. *SIAM Journal on Applied Mathematics* 50(6), 1645–1662 (1990)
10. Goel, P., Ermentrout, B.: Synchrony, Stability, and Firing Patterns in Pulse-Coupled Oscillators. *Physica D: Nonlinear Phenomena* 163(3-4), 191–216 (2002)
11. Kuramoto, Y.: Collective Synchronization of Pulse-Coupled Oscillators and Excitable Units. *Physica D: Nonlinear Phenomena* 50(1), 15–30 (1991)
12. Kashihara, S., Wakamiya, N., Murata, M.: Implementation and evaluation of a synchronization-based data gathering scheme for sensor networks. In: Proceedings of IEEE International Conference on Communications, Wireless Networking (ICC), Seoul, Korea, pp. 3037–3043 (May 2005)
13. Mutazono, A., Sugano, M., Murata, M.: Evaluation of robustness in time synchronization for sensor networks. In: Proceedings of the 2nd International Conference on Bio-Inspired Models of Network (BIONETICS), Budapest, Hungary, pp. 89–92 (December 2007)
14. Wakamiya, N., Murata, M.: Synchronization-based Data Gathering Scheme for Sensor Networks. *IEICE Transactions on Communications* 88, 873–881 (2005)

An Empirical Study of Predictive Modeling Techniques of Software Quality

Taghi M. Khoshgoftaar¹, Kehan Gao², and Amri Napolitano¹

¹ Florida Atlantic University, Boca Raton, FL 33431, USA
taghi@cse.fau.edu, amrifau@gmail.com

² Eastern Connecticut State University, Willimantic, CT 06226, USA
gaok@easternct.edu

Abstract. The primary goal of software quality engineering is to apply various techniques and processes to produce a high quality software product. One strategy is applying data mining techniques to software metrics and defect data collected during the software development process to identify the potential low-quality program modules. In this paper, we investigate the use of feature selection in the context of software quality estimation (also referred to as software defect prediction), where a classification model is used to predict program modules (instances) as fault-prone or not-fault-prone. Seven filter-based feature ranking techniques are examined. Among them, six are commonly used, and the other one, named *signal to noise ratio* (SNR), is rarely employed. The objective of the paper is to compare these seven techniques for various software data sets and assess their effectiveness for software quality modeling. A case study is performed on 16 software data sets and classification models are built with five different learners. Our experimental results are summarized based on statistical tests for significance. The main conclusion is that the SNR technique performs better than or similar to the best performer of the six commonly used techniques.

Keywords: filter-based feature ranking techniques, software defect prediction, software metrics, software quality.

1 Introduction

The quality of a software product is a key factor to that product's success or failure, and must be monitored and managed throughout the software development process. Extensive studies have been dedicated towards improving software quality. One frequently used method is software defect prediction, in which practitioners utilize software metrics (attributes or features) gathered during the software development life cycle, along with various data mining techniques, to build classification models for predicting whether a given program module (instance or example) is in the fault-prone (*fp*) or not-fault-prone (*nfp*) class [18]. The primary benefit of such a strategy is that it facilitates intelligent project resource allocation for the potentially problematic modules. For example, modules predicted to be fault-prone receive more inspection and testing, resulting in better quality.

However, in the practice of software defect prediction, we find that superfluous software metrics often exist in data repositories. Moreover, the collected software metrics

may not be complete, consistent, or useful; some metrics may be redundant or irrelevant to classification results. Therefore, using all available software metrics to train a defect prediction model without considering the quality of the underlying software measurement data is often not the best strategy. Selecting a subset of features that are most important to the class attribute is needed prior to the model training process.

In this paper, we investigate seven filter-based feature ranking techniques to choose subsets of features (metrics). Among the seven techniques, six of them are commonly used methods, chi-square (CS), information gain (IG), gain ratio (GR), symmetrical uncertainty (SU), and ReliefF (two types, RF and RFW) [25]; the remaining technique is based on *signal to noise ratio* (SNR), which is a widely used concept in electrical and communication engineering but which has only started being used in data mining research very recently [17]. The idea of the filter-based feature ranking techniques is as follows: (1) each independent attribute is individually paired with the class attribute; (2) an intrinsic characteristic (score) of each attribute is evaluated; and (3) attributes are ordered according to the calculated scores. Note that no classifiers are built during the filter-based feature ranking process.

The main objective of this paper is to evaluate the effectiveness of seven filter-based feature selection methods in the context of software quality estimation. The case study data consists of software measurement and defect data from three real-world software projects, including four data sets from a very large telecommunications software system (LLTS) [12], nine data sets from the Eclipse project [28], and three data sets from NASA software project KC1 [16]. In the case study, feature selection is performed first, then defect prediction models are constructed using five different classifiers (naïve Bayes, multilayer perceptron, support vector machine, logistic regression, and K-nearest-neighbors) with the training data consisting of the software metrics selected by the seven different approaches. The empirical results demonstrate that the SNR technique has better performance than the other six commonly used feature selection approaches on average. Moreover, SNR exhibits more stable performance than some standard techniques such as RF and RFW with respect to different learners. From a software practice point of view, researchers and practitioners would like to work with a smaller set of metrics for defect prediction rather than analyze a large number of metrics.

The remainder of the paper is organized as follows: Section 2 describes some related work. Section 3 presents seven filter-based feature ranking techniques, five classifiers, and the performance metric used in this study. Section 4 describes the empirical case study, including software measurement data, results, analysis, and threats to empirical validity. Section 5 summarizes the conclusions of the paper and some directions for future work.

2 Related Work

Feature selection is the process of choosing some subset of the features and building a classifier based solely on those. It can remove as many unnecessary features as possible, leaving only those features which are useful in building a classifier. Feature selection

has been extensively studied for many years in the data mining and machine learning community. A good overview on various aspects of the attribute selection problem was done by Guyon and Elisseeff [7]. They outlined the key techniques and approaches used in attribute selection including feature construction, feature ranking, multivariate feature selection, efficient search methods, and feature validity assessment methods. Liu and Yu [19] provided a comprehensive survey of feature selection algorithms and presented an integrated approach to intelligent feature selection.

Typically, feature selection techniques are divided into two types: *wrapper*-based and *filter*-based. The wrapper-based approach involves training a learner during the feature selection process, while a filter-based approach uses the intrinsic characteristics of the data for feature selection and does not rely on training a learner. The primary advantage of the filter-based approach over the wrapper-based approach lies in its faster computation.

In this paper, we examine seven filter-based feature ranking techniques, six of which are commonly used filter-based methods, while the other one, named signal to noise ratio (SNR), has only been applied in data mining research very recently to select relevant features in classification problems. Studies [5] showed that SNR's ability to properly rank the features is affected by the number of features present. If the number of features is too large (very high dimensional data) then ranking performance may be affected by the presence of noise in the data. But in most cases, SNR combined with other approaches shows improved performance. Next, we briefly discuss some of the recent techniques that involve SNR in feature selection.

A PCC (Pearson correlation coefficient) SNR hybrid method is shown to be very efficient in selecting genes in microarray expression data [5]. ANN (artificial neural network) based feature selection using SNR for classification of Doppler ultrasound signals has been studied by Güler and İbeyli [6]. They devised an ANN-SNR based classification method to classify Ophthalmic arterial Doppler signals and internal carotid arterial Doppler signals. Lakshmi et al. [17] suggest a MSNR (maximized signal to noise ratio) technique in the field of text classification. A novel GA-Taguchi based feature selection method applies this SNR based technique on seventeen different real world data sets and shows its superiority in selecting useful features [27]. In this paper, we study the standard SNR technique and apply it to software quality modeling. To our knowledge, no research has done this before.

We also notice that although feature selection has been widely applied in various data mining problems [3,7,8,10,22], its application in software quality and reliability engineering has been rather limited. Rodríguez et al. [21] applied feature selection to five software engineering data sets using three filter-based models and three wrapper-based models. The authors state that the reduced data sets maintained the prediction capability of the original data sets while using fewer attributes. Chen et al. [2] have studied feature selection using wrappers in the context of software cost/effort estimation. In recent studies [24,13], we investigated various feature selection techniques, including filter-based and wrapper-based methods. It was concluded that the performances of the classification models either improved or were not affected when over 85% of the features were eliminated from the original data sets.

3 Methodology

3.1 Standard Filter-Based Feature Ranking Techniques

Feature ranking assigns a score to each feature according to a particular method (metric), allowing the selection of the best set of features. The six commonly used filter-based feature ranking techniques considered in this work include [25]: chi-square (CS), information gain (IG), gain ratio (GR), two types of ReliefF (RF and RFW), and symmetrical uncertainty (SU).

The chi-squared - χ^2 (CS) test is used to examine the distribution of the class as it relates to the values of the target feature. The null hypothesis is that there is no correlation, i.e., each value is as likely to have instances in any one class as any other class. Given the null hypothesis, the χ^2 statistic measures how far away the actual value is from the expected value:

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^{n_c} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

where r is the number of different values of the feature, n_c is the number of classes (in this work, $n_c = 2$), $O_{i,j}$ is the observed number of instances with value i which are in class j , and $E_{i,j}$ is the expected number of instances with value i and class j . The larger the χ^2 statistic, the more likely it is that the distribution of values and classes are dependent; that is, the feature is relevant to the class.

Information gain, gain ratio, and symmetrical uncertainty are measures based on the concept of entropy from information theory [25]. Information gain (IG) is the information provided about the target class attribute Y, given the value of another attribute X. IG measures the decrease of the weighted average impurity of the partitions compared to the impurity of the complete set of data. IG tends to prefer attributes with a larger number of possible values. If one attribute has a larger number of values, it will appear to gain more information than those with fewer values, even if it is actually no more informative. One strategy to solve this problem is to use the gain ratio (GR), which penalizes multiple-valued attributes. Symmetrical uncertainty (SU) is another way to overcome the problem of IG's bias toward attributes with more values, doing so by dividing by the sum of the entropies of X and Y.

Relief is an instance-based feature ranking technique introduced by Kira and Rendell [15]. ReliefF is an extension of the Relief algorithm that can handle noise and multiclass data sets, and is implemented in the WEKA¹ tool [25]. When the `WeightByDistance` (weight nearest neighbors by their distance) parameter is set as default (false), the algorithm is referred to as RF; when the parameter is set to 'true', the algorithm is referred to as RFW.

¹ WEKA (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato. WEKA is free software available under the GNU General Public License. In this study, all experiments and algorithms were implemented in the WEKA tool.

3.2 Signal to Noise Ratio

Signal to noise ratio (SNR) [5], in the context of classification problems in data mining, defines how well a feature discriminates two classes in a two-class problem. The equation to calculate SNR is

$$\text{SNR} = \frac{\mu_P - \mu_N}{\sigma_P + \sigma_N}$$

where μ_P and μ_N are mean values of a particular attribute for the samples from class P and class N , and σ_P and σ_N are standard deviations for this attribute from the sample set for class P and class N . Because of its discriminating power among the classes, SNR is highly efficient to properly rank the features in terms of its relation to the output class.

3.3 Classification Algorithms

The software defect prediction models are built using five different classification algorithms, including Naïve Bayes (NB) [25], Multilayer Perceptron (MLP) [9], Support Vector Machine (SVM) [23], Logistic Regression (LR) [25], and K-nearest-neighbors (KNN) [25]. These learners were selected for two key reasons: (1) they do not have a built-in feature selection capability, and (2) they are commonly used in both software engineering and data mining domains [11][8][20].

The WEKA data mining tool [25] is used to instantiate the different classifiers. Generally, the default parameter settings for the different learners are used (for NB and LR), except for the below-mentioned changes. A preliminary investigation in the context of this study indicated that the modified parameter settings are appropriate.

- In the case of MLP, the `hiddenLayers` parameter was changed to ‘3’ to define a network with one hidden layer containing three nodes, and the `validationSetSize` parameter was changed to ‘10’ to cause the classifier to leave 10% of the training data aside for use as a validation set to determine when to stop the iterative training process.
- For the KNN learner, the `distanceWeighting` parameter was set to ‘Weight by 1/distance’, and the `kNN` parameter was set to ‘5’.
- In the case of SVM, two changes were made: the `complexity` constant `c` was set to ‘5.0’, and `build Logistic Models` was set to ‘true’. A linear kernel was used by default.

3.4 Performance Metric

The Area Under the ROC (Receiver Operating Characteristic) curve, abbreviated as AUC, is used for evaluating the defect prediction models in this study. If the *fp* modules are considered positive and the *nfp* modules are considered negative, then the ROC curve plots the true positive rates versus the false positive rates. An ROC curve illustrates the classifier’s performance across all decision thresholds, i.e., a number between 0 and 1 that theoretically separates the *fp* and *nfp* modules. This is in contrast to most defect prediction studies that use performance metrics based on the default decision threshold value of 0.5. The AUC values range from 0 to 1, where a perfect classifier provides an AUC value of 1 [11][25].

Table 1. Software Data Set Characteristics

	Data	#Attri.	#Inst.	#fp	%fp	#nfp	%nfp
LLTS	SP1	42	3649	229	6%	3420	94%
	SP2	42	3981	189	5%	3792	95%
	SP3	42	3541	47	1%	3494	99%
	SP4	42	3978	92	2%	3886	98%
Eclipse	E2.0-10	208	377	23	6%	354	94%
	E2.0-5	208	377	52	14%	325	86%
	E2.0-3	208	377	101	27%	276	73%
	E2.1-5	208	434	34	8%	400	92%
	E2.1-4	208	434	50	12%	384	88%
	E2.1-2	208	434	125	29%	309	71%
	E3.0-10	208	661	41	6%	620	94%
	E3.0-5	208	661	98	15%	563	85%
	E3.0-3	208	661	157	24%	504	76%
NASA	KC1-5	62	145	36	25%	109	75%
	KC1-10	62	145	21	14%	124	86%
	KC1-20	62	145	10	7%	135	93%

4 Case Study

4.1 Software Measurement Data

Experiments conducted in this study used software metrics and defect data collected from real-world software projects, including a very large telecommunications software system (denoted as LLTS) [12], the Eclipse project [28], and NASA software project KC1 [16].

The software measurement data sets of LLTS consist of 42 software metrics, including 24 product metrics, 14 process metrics, and four execution metrics. More details about these software metrics can be seen in [12]. The dependent variable is the class of the program module. A module with one or more faults is considered *fp*, and *nfp* otherwise. The LLTS software system consists of four successive releases labeled SP1, SP2, SP3, and SP4, where each release is characterized by the same number and type of software metrics, but has a different number of instances (program modules). The SP1, SP2, SP3, and SP4 data sets consist of 3649, 3981, 3541, and 3978 program modules, respectively.

From the PROMISE data repository [28], we obtained the Eclipse defect counts and complexity metrics data set. The original data for the Eclipse packages consists of three releases denoted 2.0, 2.1, and 3.0, respectively. We chose three post-release defects thresholds *thd* to determine the defective instances for each release. A program module with *thd* or more post-release defects is labeled as *fp*, while those with fewer than *thd* defects are labeled as *nfp*. In our study, we use $thd \in \{10, 5, 3\}$ for release 2.0 and 3.0 while we use $thd \in \{5, 4, 2\}$ for release 2.1. All nine derived data sets contain 208 independent attributes. Releases 2.0, 2.1, and 3.0 contain 377, 434, and 661 instances, respectively.

We obtain the class-level data set for the KC1 data set also from the PROMISE repository. This is a publicly available data set that represents one of five NASA projects [16]. The original data contains a set of measures per module, including the number of defects reported for the module. It included 145 instances, each containing 94 independent attributes plus a defect attribute. After removing 32 Halstead derived measures,

we have 62 independent attributes left. We used three different thresholds to define defective instances, thereby obtaining three structures of the preprocessed KC1 data set. The thresholds are 20, 10, and 5, indicating instances with number of defects greater or equal to 20, 10, or 5 belong to the *fp* class, or the *nfp* class otherwise. The three data sets are named KC1-20, KC1-10, and KC1-5, respectively.

The 16 data sets used in this work reflect software projects of different sizes with different proportions of *fp* and *nfp* modules. Table 1 lists the characteristics of the 16 data sets utilized in this work.

4.2 Results of the Feature Selection Techniques

When using a filter-based feature ranking technique, the number of features (software metrics) that will be selected for modeling must be given in advance for the technique. In this study, we choose $\lceil \log_2 n \rceil$ features that have the highest scores, where n is the total number of the independent features. The reasons we adopt such a strategy include: (1) to our knowledge, related literature does not provide guidance on the appropriate number of features to select in such ranking techniques; (2) a recent study [14] recommended using $\lceil \log_2 n \rceil$ features to build Random Forests learners for binary classification for imbalanced data sets; and (3) a preliminary investigation showed that $\lceil \log_2 n \rceil$ is also appropriate for various learners. Consequently, a feature subset with size of $\lceil \log_2 n \rceil$ is used. That is, for the four LLTS data sets, we select $\lceil \log_2 42 \rceil = 6$ features; for the nine Eclipse data sets, we select $\lceil \log_2 208 \rceil = 8$ features; and for the three NASA KC1 data sets, we select $\lceil \log_2 62 \rceil = 6$ features.

Following the feature selection algorithms, the five different types of classification models are constructed with data sets containing only the selected attributes. The defect prediction models are evaluated in terms of the AUC performance metric, as stated earlier. Due to paper size limitations, we could not present each individual classifier's performance. We only present the results of the NB and LR learners individually and the average performance over five learners, as shown in Tables 2, 3, and 4 respectively. However, the discussions and summaries are made based on the facts observed over all five different classifiers.

In the experiments, ten runs of five-fold cross-validation were performed for model training. The values presented in the tables represent the average AUC for every classification model constructed over the ten runs of five-fold cross-validation. All the results of seven feature selection techniques and over 16 different software data sets are reported. The best feature selection technique in terms of their AUCs for each data set (row) is highlighted in **bold**. We also summarize the average performance (last row of the table) for each feature selection technique across the 16 data sets.

The results demonstrate that SNR outperformed the other feature selection techniques on average when the five classifiers are applied to the selected subset of features, since the average AUC of SNR over the five learners is 0.8380 (last row of Table 4), which is the highest score among the seven techniques. For the six commonly used techniques, IG performed best on average; GR, RF, and RFW performed more poorly than the other three techniques (IG, CS and SU). Of course, once the subset of features is set, the classification performance is determined by the learners we select. For instance, the feature subset selected by SNR demonstrated better performance than the

Table 2. Classification Performance in terms of AUC for NB Classifier

Data Set	CS	GR	IG	RF	RFW	SU	SNR
SP1	0.7846	0.7346	0.7831	0.7879	0.7882	0.7865	0.7995
SP2	0.8108	0.7613	0.8081	0.8053	0.8081	0.7729	0.8142
SP3	0.8184	0.7808	0.8118	0.8305	0.8190	0.7882	0.8067
SP4	0.7696	0.7519	0.7795	0.7731	0.7735	0.7592	0.8130
E2.0-10	0.7904	0.8074	0.8070	0.8455	0.8107	0.8158	0.8623
E2.0-5	0.8421	0.8078	0.8562	0.8617	0.8607	0.8464	0.8767
E2.0-3	0.7963	0.7458	0.8002	0.7857	0.8107	0.7940	0.8003
E2.1-5	0.8419	0.7919	0.8547	0.8022	0.8188	0.8269	0.8681
E2.1-4	0.8226	0.7917	0.8281	0.7688	0.7302	0.8170	0.8560
E2.1-2	0.7536	0.7614	0.7542	0.7993	0.7966	0.7551	0.8063
E3.0-10	0.8742	0.8463	0.8963	0.8044	0.8101	0.8540	0.9175
E3.0-5	0.8866	0.8785	0.8851	0.8481	0.8732	0.8817	0.9099
E3.0-3	0.8130	0.7974	0.8122	0.7789	0.8024	0.8072	0.8759
KC1-5	0.7484	0.7489	0.7438	0.7990	0.7832	0.7468	0.7847
KC1-10	0.7513	0.7729	0.7546	0.7585	0.7639	0.7719	0.7508
KC1-20	0.8525	0.8669	0.8569	0.8296	0.8987	0.8532	0.8646
Average	0.8098	0.7903	0.8145	0.8049	0.8093	0.8048	0.8379

Table 3. Classification Performance in terms of AUC for LR Classifier

Data Set	CS	GR	IG	RF	RFW	SU	SNR
SP1	0.8021	0.7688	0.8014	0.8103	0.8091	0.7993	0.8176
SP2	0.8230	0.7935	0.8176	0.8221	0.8233	0.7909	0.8279
SP3	0.8354	0.7805	0.8361	0.8354	0.8387	0.8040	0.8336
SP4	0.8153	0.7816	0.8216	0.8118	0.8142	0.7802	0.8232
E2.0-10	0.8067	0.8173	0.8367	0.8493	0.8184	0.8214	0.8595
E2.0-5	0.8898	0.8695	0.8999	0.8907	0.8923	0.8895	0.9029
E2.0-3	0.8665	0.8140	0.8658	0.8345	0.8468	0.8600	0.8611
E2.1-5	0.8729	0.8394	0.8765	0.8728	0.8824	0.8655	0.8887
E2.1-4	0.8673	0.8587	0.8652	0.8252	0.7582	0.8706	0.8768
E2.1-2	0.8852	0.8774	0.8839	0.8607	0.8726	0.8829	0.8868
E3.0-10	0.9090	0.8808	0.9015	0.8672	0.8604	0.8850	0.9152
E3.0-5	0.9424	0.9334	0.9420	0.8974	0.9083	0.9360	0.9418
E3.0-3	0.9096	0.9019	0.9098	0.8267	0.8415	0.9066	0.9204
KC1-5	0.7774	0.7669	0.7711	0.8134	0.7946	0.7648	0.7990
KC1-10	0.7613	0.7649	0.7791	0.7393	0.7072	0.7734	0.7701
KC1-20	0.8202	0.8293	0.8160	0.7433	0.8420	0.8336	0.8252
Average	0.8490	0.8299	0.8515	0.8312	0.8319	0.8415	0.8594

feature subsets selected by the other approaches for 11 out of 16 cases when NB was applied (see Table 2); 10 out of 16 cases when LR was used (see Table 3); seven out of 16 cases for MLP; six out of 16 for KNN; and four out of 16 for SVM. But on average, SNR performed better than other techniques for eight out of 16 cases (see Table 4).

We also performed a one-way ANalysis Of VAriance (ANOVA) F-test [1] on the classification performance for each technique across all the data sets to examine the significance level of the performance differences. The ANOVA tests were performed on the five classifiers individually. Once again, due to space limitation, we only present the test results for NB, LR, and the average of all five classifiers. The underlying assumptions of ANOVA were tested and validated prior to statistical analysis. The main factor of our ANOVA experiment is the seven feature ranking techniques. The null hypothesis for the ANOVA test is that all the group population means are the same, while the alternate hypothesis is that at least one pair of means is different.

Table 4. Classification Performance in terms of AUC for **Five Classifiers**

Data Set	CS	GR	IG	RF	RFW	SU	SNR
SP1	0.7556	0.7236	0.7579	0.7627	0.7557	0.7593	0.7678
SP2	0.7863	0.7437	0.7741	0.7594	0.7619	0.7570	0.7906
SP3	0.7800	0.7379	0.7792	0.7818	0.7839	0.7593	0.7752
SP4	0.7641	0.7180	0.7722	0.7469	0.7512	0.7346	0.7781
E2.0-10	0.8172	0.8092	0.8404	0.8499	0.8178	0.8188	0.8693
E2.0-5	0.8758	0.8440	0.8850	0.8699	0.8674	0.8754	0.8905
E2.0-3	0.8333	0.7850	0.8358	0.8023	0.8158	0.8322	0.8300
E2.1-5	0.8767	0.8252	0.8882	0.8162	0.8299	0.8695	0.8800
E2.1-4	0.8664	0.8310	0.8680	0.7875	0.7207	0.8636	0.8654
E2.1-2	0.8512	0.8460	0.8500	0.8243	0.8328	0.8482	0.8598
E3.0-10	0.9018	0.8644	0.9128	0.8435	0.8304	0.8744	0.9065
E3.0-5	0.9228	0.9144	0.9226	0.8633	0.8760	0.9175	0.9247
E3.0-3	0.8822	0.8693	0.8812	0.7962	0.8134	0.8772	0.8964
KC1-5	0.7774	0.7589	0.7676	0.8049	0.7895	0.7634	0.7837
KC1-10	0.7714	0.7731	0.7760	0.7358	0.7084	0.7803	0.7641
KC1-20	0.8288	0.8418	0.8353	0.7989	0.8664	0.8355	0.8261
Average	0.8307	0.8053	0.8341	0.8027	0.8013	0.8229	0.8380

Table 5. One-way ANOVA

(a) NB						(b) LR					
Source	Sum Sq.	d.f.	Mean Sq.	F	p-value	Source	Sum Sq.	d.f.	Mean Sq.	F	p-value
Techniques	0.1982	6	0.0330	16.61	0	Techniques	0.1289	6	0.0215	7.59	0
Error	2.2129	1113	0.0020			Error	3.1523	1113	0.0028		
Total	2.4111	1119				Total	3.2812	1119			

(c) Five Classifiers					
Source	Sum Sq.	d.f.	Mean Sq.	F	p-value
Techniques	1.2057	6	0.2010	43.11	0
Error	26.0682	5593	0.0047		
Total	27.2739	5599			

Table 5 shows the ANOVA results. It includes three subtables, each representing the result for each individual case (NB, LR, and five classifiers). All the p -values are less than the typical cutoff 0.05, indicating that for the main factor (Techniques), the alternate hypothesis is accepted, namely, at least two group means are significantly different from each other. We continued our statistical validation by performing a multiple comparison test on the main factor with Tukey’s Honestly Significant Difference (HSD) criterion [11]. Note that for both ANOVA and multiple comparison tests, the significance level α was set to 0.05.

The multiple comparison results are shown in Figure 11 displaying graphs with each group mean represented by a symbol (\circ) and the 95% confidence interval as a line around the symbol. Two means are significantly different if their intervals are disjoint, and are not significantly different if their intervals overlap. Matlab was used to perform the ANOVA and multiple comparisons presented in this work. Based on the multiple comparison results, we can conclude the following points:

- Among the six commonly used filter-based feature selection techniques, IG performed best. This is true irrespective of what classifier is used to build classification models. CS and SU performed averagely and GR, RF, and RFW performed poorly on average.

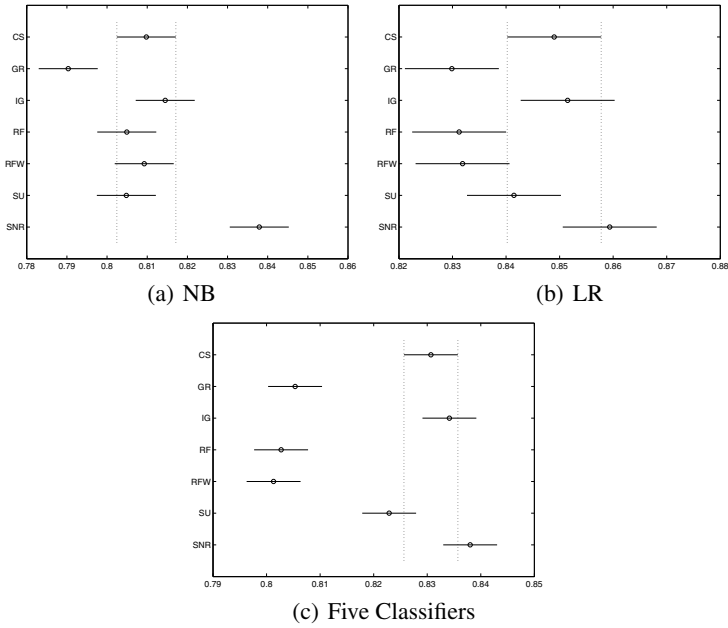


Fig. 1. Multiple Comparison

- SNR performed better than or similar to the best performer of the six commonly used techniques (i.e., IG) across 16 data sets on average.
- Some techniques demonstrate relatively stable performance with respect to different learners such as SNR and IG, while other techniques, such as RF and RFW, show more fluctuational performance with respect to different learners.
- Overall, IG and SNR demonstrated better and more stable performance than other approaches, and therefore are recommended by this study.

4.3 Discussion on Selected Software Metrics

From a software engineering point of view, a discussion on which software metrics were selected is warranted. Due to paper size limitations, we only present the results on LLTS SP1 as shown in Table 6. The table presents the selected subsets of software metrics for the seven techniques. For example, for SP1, the CS technique produces a subset of software metrics {5, 11, 15, 24, 33, 41}, where the values represent the software metric ID numbers. The 42 software metrics are labeled with an ID number ranging from 1 to 42. From the table, we can see that 27 metrics were selected by at least one feature selection technique and 15 metrics have never been selected by any techniques such as metric 3, 4, 10, and so on. The second column of the table indicates the number of times the metric is selected. All the detailed information related to the 42 software metrics are listed in Table 7.

Our recent work [4] has shown that classification models built on smaller subsets of features via the six commonly used filter-based feature selection techniques had similar

Table 6. Software Metrics Selected for SP1

Metric ID	Total # selected	Filter-Based Techniques						
		CS	GR	IG	RF	RFW	SU	SNR
1	2		x				x	
2	1		x					
5	2	x					x	
6	1				x			
7	1					x		
8	1		x					
9	1				x			
11	2	x			x			
12	1					x		
15	3	x				x	x	
17	1			x				
19	2		x					x
20	1			x				
21	1							x
24	2	x				x		
27	2				x			x
28	1		x					
29	2			x			x	
30	2				x		x	
31	1							x
32	1							x
33	2	x		x				
34	2			x				
35	2					x	x	
36	3		x		x			x
41	1	x						
42	1			x				

or better performances than those built with a complete set of attributes. Thus, we did not present the results for full data sets in this paper.

4.4 Threats to Validity

A typical software development project is very human intensive, which can affect many aspects of the development process including software quality and defect occurrence. Consequently, software engineering research that utilizes controlled experiments for evaluating the usefulness of empirical models is not practical. The case study presented in this paper is an empirical software engineering effort, for which the software engineering community demands that its subject have the following characteristics [26]: (1) developed by a group, and not by an individual; (2) be a large, industry-sized project, and not a toy problem; (3) developed by professionals, and not by students; and (4) developed in an industry/government organization setting, and not in a laboratory.

The software systems that are used in our case study were developed by professionals in large software development organizations using established software development processes and management practices. The software was developed to address real-world problems. We note that our case studies fulfill all of the above criteria specified by the software engineering community. The rest of this section discusses threats to external validity and threats to internal validity.

Threats to external validity are conditions that limit generalization of case study results. The analysis and conclusion presented in this article are based upon the metrics and defect data obtained from 16 data sets of three software projects. The same analysis

Table 7. Software Metrics for LLTS Data Sets

Metric ID	Symbol	Description
1	DES_PR	Number of problems found by designers during development of the current release.
2	BETA_PR	Number of problems found during beta testing of the current release.
3	DES_FIX	Number of problems fixed that were found by designers in the prior release.
4	BETA_FIX	Number of problems fixed that were found by beta testing in the prior release.
5	CUST_FIX	Number of problems fixed that were found by customers in the prior release.
6	REQ_UPD	Number of changes to the code due to new requirements.
7	TOT_UPD	Total number of changes to the code for any reason.
8	REQ	Number of distinct requirements that caused changes to the module.
9	SRC_GRO	Net increase in lines of code.
10	SRC_MOD	Net new and changed lines of code.
11	UNQ_DES	Number of different designers making changes.
12	VLO_UPD	Number of updates to this module by designers who had 10 or less total updates in entire company career.
13	LO_UPD	Number of updates to this module by designers who had between 11 and 20 total updates in entire company career.
14	UPD_CAR	Number of updates that designers had in their company careers.
15	USAGE	Deployment percentage of the module.
16	CALUNQ	Number of distinct procedure calls to others.
17	CAL2	Number of second and following calls to others. $CAL2 = CAL - CALUNQ$ where CAL is the total number of calls.
18	CNDSPNSM	Total span of branches of conditional arcs. The unit of measure is arcs.
19	CNDSPNMX	Maximum span of branches of conditional arcs.
20	CTRNSTMX	Maximum control structure nesting.
21	FILINCQU	Number of distinct include files.
22	KNT	Number of knots. A "knot" in a control flow graph is where arcs cross due to a violation of structured programming principles.
23	LOC	Number of lines of code.
24	CNDNOT	Number of arcs that are not conditional arcs.
25	IFTH	Number of non-loop conditional arcs (i.e., if-then constructs).
26	LOP	Number of loop constructs.
27	NDSENT	Number of entry nodes.
28	NDSEXT	Number of exit nodes.
29	NDSPND	Number of pending nodes (i.e., dead code segments).
30	NDSINT	Number of internal nodes (i.e., not an entry, exit, or pending node).
31	LGPATH	Base 2 logarithm of the number of independent paths.
32	STMCTL	Number of control statements.
33	STMDEC	Number of declarative statements.
34	STMEXE	Number of executable statements.
35	VARGLBUS	Number of global variables used.
36	VARSPNSM	Total span of variables.
37	VARSPNMX	Maximum span of variables.
38	VARUSDUQ	Number of distinct variables used.
39	VARUSD2	Number of second and following uses of variables. $VARUSD2 = VARUSD - VARUSDUQ$ where $VARUSD$ is the total number of variable uses.
40	RESCPU	Execution time (microseconds) of an average transaction on a system serving consumers.
41	BUSCPU	Execution time (microseconds) of an average transaction on a system serving businesses.
42	TANCPU	Execution time (microseconds) of an average transaction on a tandem system.

for another software system may provide different results which is a likely threat in all empirical software engineering research. However, we place our emphasis on the process of comparing the different feature selection techniques considered in this study. Our comparative analysis can easily be applied to another software system. Moreover, since all our final conclusions are based on ten runs of five-fold cross-validation and statistical tests for significance, our findings are grounded in using sound methods.

Threats to internal validity are unaccounted for influences on the experiments that may affect case study results. Poor fault proneness estimates can be caused by a wide variety of factors, including measurement errors while collecting and recording software metrics, modeling errors due to the unskilled use of software applications, errors

in model-selection during the modeling process, and the presence of outliers and noise in the training data set. Measurement errors are inherent to the data collection effort. In our comparative study, a common model-building and model-evaluation approach is used for all feature selection techniques and classifiers considered. Moreover, the experiments and statistical analysis were performed by only one skilled person in order to keep modeling errors to a minimum.

5 Conclusion

In the software quality modeling process, many practitioners often ignore a fact that excessive metrics exist in data repositories. They directly use the available set of software metrics to build classification models without regard to the quality of the underlying software measurement data, leading to inferior prediction accuracy and extension of training time. Selecting software metrics that are important for defect prediction is needed and critical before the model training process.

In this study, we investigated seven filter-based feature ranking techniques to select the most important software metrics. Among the seven techniques, six of them are standard filter-based techniques that are commonly used, while the remaining one is the signal to noise ratio (SNR) technique rarely used in feature selection. The main purpose of this paper is to examine and compare all seven filter-based feature selection techniques and evaluate their effectiveness in the context of software defect prediction. The experiments were conducted on 16 different data sets obtained from three different software projects. In order to alleviate the problem of potentially biased results generated by a specific classifier, we used five different learners to build classification models with data sets containing only the selected attributes. Moreover, ten runs of five-fold cross-validation were adopted to make the conclusions more persuasive.

The key conclusions are summarized as follows:

- For the six standard filter-based feature selection techniques, IG performed better than the other five techniques on average across all data sets. This is true no matter what classifier is used to build classification models. CS and SU performed slightly worse than IG, but better than GR, RF, and RFW.
- The SNR technique showed better or similar performance to the best performer of the six commonly used techniques, i.e., IG.
- Some techniques (such as SNR and IG) demonstrated more stable performance than other techniques (RF and RFW) with respect to different learners.
- IG and SNR exhibit overall better and more stable performance than other approaches, and therefore are recommended by this study.
- Selecting fewer software metrics for defect prediction is very important to the software quality assurance team, for instance, in our case study, working with six metrics is much easier and more practical than dealing with 42 metrics.

Future work will include more case studies with software measurement data sets of other software systems. In addition, as very few research works utilize SNR to select features, this concept presents a significant research area for computer scientists. More research works that use SNR to rank features are expected.

References

1. Berenson, M.L., Goldstein, M., Levine, D.: *Intermediate Statistical Methods and Applications: A Computer Package Approach*, 2nd edn. Prentice-Hall (1983)
2. Chen, Z., Menzies, T., Port, D., Boehm, B.: Finding the right data for software cost modeling. *IEEE Software* 22(6), 38–46 (2005)
3. Forman, G.: An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research* 3, 1289–1305 (2003)
4. Gao, K., Khoshgoftaar, T.M., Wang, H., Seliya, N.: Choosing software metrics for defect prediction: An investigation on feature selection techniques. *Software: Practice and Experience*. Special Issue: Practical Aspects of Search-Based Software Engineering 41(5), 579–606 (2011), doi:10.1002/spe.1043
5. Goh, L., Song, Q., Kasabov, N.: A novel feature selection method to improve classification of gene expression data. In: Chen, Y.P. (ed.) *Proceedings of the Second Conference on Asia-Pacific Bioinformatics*, pp. 161–166. Australian Computer Society, Darlinghurst (2004)
6. Güler, İ., İbeyli, E.D.: Feature saliency using signal-to-noise ratios in automated diagnostic systems developed for doppler ultrasound signals. *Engineering Applications of Artificial Intelligence* 19(1), 53–63 (2006)
7. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
8. Hall, M.A., Holmes, G.: Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transaction on Knowledge and Data Engineering* 15(6), 1437–1447 (2003)
9. Haykin, S.: *Neural Networks: A Comprehensive Foundation*, 2nd edn. Prentice-Hall (1998)
10. Ilczuk, G., Mlynarski, R.: W Kargul, and A Wakulicz-Deja. New feature selection methods for qualification of the patients for cardiac pacemaker implantation. *Computers in Cardiology* 34(2-3), 423–426 (2007)
11. Jiang, Y., Lin, J., Cukic, B., Menzies, T.: Variance analysis in software fault prediction models. In: *Proceedings of the 20th IEEE International Symposium on Software Reliability Engineering*, Bangalore, Mysore, India, November 16-19, pp. 99–108 (2009)
12. Khoshgoftaar, T.M., Bullard, L.A., Gao, K.: Attribute selection using rough sets in software quality classification. *International Journal of Reliability, Quality and Safty Engineering* 16(1), 73–89 (2009)
13. Khoshgoftaar, T.M., Gao, K.: Feature selection with imbalanced data for software defect prediction. In: *Proceedings of the 8th International Conference on Machine Learning and Applications*, Miami, Florida, USA, December 13-15, pp. 235–240 (2009)
14. Khoshgoftaar, T.M., Golawala, M., Van Hulse, J.: An empirical study of learning from imbalanced data using random forest. In: *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence*, Washington, DC, USA, vol. 2, pp. 310–317 (2007)
15. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: *Proceedings of 9th International Workshop on Machine Learning*, pp. 249–256 (1992)
16. Koru, A.G., Zhang, D., El Emam, K., Liu, H.: An investigation into the functional form of the size-defect relationship for software modules. *IEEE Transactions on Software Engineering* 35(2), 293–304 (2009)
17. Lakshmi, K., Mukherjee, D.S.: An improved feature selection using maximized signal to noise ratio technique for tc. In: *Proceedings of the Third international Conference on information Technology: New Generations*, pp. 541–546. IEEE Computer Society Press, Washington, DC (2006)
18. Lessmann, S., Baesens, B., Mues, C., Pietsch, S.: Benchmarking classification models for software defect prediction: A proposed framework and novel findings. *IEEE Transactions on Software Engineering* 34(4), 485–496 (2008)

19. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering* 17(4), 491–502 (2005)
20. Menzies, T., Greenwald, J., Frank, A.: Data mining static code attributes to learn defect predictors. *IEEE Transactions on Software Engineering* 33(1), 2–13 (2007)
21. Rodriguez, D., Ruiz, R., Cuadrado-Gallego, J., Aguilar-Ruiz, J.: Detecting fault modules applying feature selection to classifiers. In: *Proceedings of 8th IEEE International Conference on Information Reuse and Integration, Las Vegas, Nevada, August 13-15*, pp. 667–672 (2007)
22. Saeys, Y., Abeel, T., Van de Peer, Y.: Robust Feature Selection Using Ensemble Feature Selection Techniques. In: Daelemans, W., Goethals, B., Morik, K. (eds.) *ECML PKDD 2008, Part II. LNCS (LNAI)*, vol. 5212, pp. 313–325. Springer, Heidelberg (2008)
23. Shawe-Taylor, J., Cristianini, N.: *Support Vector Machines*, 2nd edn. Cambridge University Press (2000)
24. Wang, H., Khoshgoftaar, T.M., Gao, K., Seliya, N.: Mining data from multiple software development projects. In: *Proceedings of 2009 IEEE International Conference on Data Mining Workshops, Miami, Florida, USA, December 6*, pp. 551–557 (2009)
25. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann (2005)
26. Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: *Experimentation in Software Engineering: An Introduction. Kluwer International Series in Software Engineering. Software Engineering. Kluwer Academic Publishers, Boston* (2000)
27. Yang, C.-H., Huang, C.-C., Wu, K.-C., Chang, H.-Y.: A Novel GA-Taguchi-Based Feature Selection Method. In: Fyfe, C., Kim, D., Lee, S.-Y., Yin, H. (eds.) *IDEAL 2008. LNCS*, vol. 5326, pp. 112–119. Springer, Heidelberg (2008)
28. Zimmermann, T., Premraj, R., Zeller, A.: Predicting defects for eclipse. In: *Proceedings of the 29th International Conference on Software Engineering Workshops*, p. 76. IEEE Computer Society Press, Washington, DC (2007)

Sensor Based Time Series Classification of Body Movement

Swapna Philip¹, Yu Cao^{2,*}, and Ming Li¹

¹ Department of Computer Science
California State University, Fresno
Fresno, CA, U.S.A.

swapna.philip@gmail.com, mingli@csufresno.edu

² College of Engineering and Computer Science
University of Tennessee at Chattanooga
Chattanooga, TN, U.S.A.
yu-cao@utc.edu

Abstract. Advances in sensing and monitoring technology are being incorporated into today's healthcare practice. As a result, the concept of Body Sensor Networks (BSN) has been proposed to describe the wearable/wireless devices for healthcare applications. One of the major application scenarios for BSN is to detect and classify the body movements for long-term lifestyle and healthcare monitoring. This paper introduces a new approach for analyzing the time series obtained from BSN. In our research, the BSN record the acceleration data of the volunteer's movement while performing a set of activities such as jogging, walking, resting, and transitional activities. The main contribution of this paper is the proposed time series approximation and feature extraction algorithm that can convert the sensor-based time series data into a density map. We have performed extensive experiments to compare the accuracy in classifying the time series into different activities. It is concluded that the proposed approach would aid greatly the development of efficient health monitoring systems in the future. To the best of our knowledge, no similar research has been reported in the BSN field and we expect our research could provide useful insights for further investigation.

Keywords: Body Sensor Networks, Time Series Analysis, Human body movement, machine learning, classification.

1 Introduction

Body sensor networks (BSN) [1, 2] enable an inexpensive approach for data collection. The BSN include sensor nodes that communicate wirelessly and they function with less consumption of power and less memory capacity. These sensor networks are very useful especially in the field of health monitoring as they are capable of monitoring temperature, humidity, force, acceleration and heartbeat [3, 4]. The sensor networks can

* Corresponding author.

thus be specifically used in interpreting human motion and thus in health care for the elderly. The BSN have their own advantage over other types of motion or health monitoring systems as they are wearable, less expensive and more accurate with negligible environmental disturbances affecting data collection.

While monitoring the health of patients, it is necessary to deal with streaming data. However, the streaming data is usually too large to process as raw data in BSN environment because the nodes in BSN have a small memory and processing capacity. Hence for the classification of patients' activities, we need the suitable time series approximation and feature extraction techniques which can be easily deployed and the patterns can be analyzed in real-time. New approximation and feature analysis algorithms will make it easier to distinguish between activities such as jogging, resting, walking and transitional activities. The results from approximation and feature extraction will be feed into the machine learning based-classification algorithms.

Some approaches have been proposed to deal with the classification of sensor data with real time approximation of the series [5], but it has not been tested in the area of BSN. There has been work done in the automatic segmentation of the time series for classification [6]. The symbolic time series representation SAX [7] has been a popular approach for a wide variety of time series data. The SAX representation allows the application of string manipulation algorithms for feature extraction. One such feature extraction method is the time series bitmap [8], which has previously been applied to sensor data analysis.

In this paper, we proposed new approximation and feature extraction algorithms for BSN. We also test the accuracy of classification of the BSN-based time series data. The activities of four classes (jogging, resting, walking and transitional activities) are recorded in segments and applied with the SAX representation. The time series density map is then obtained from the symbolic representation. This map is then input to the classification algorithms. The classification algorithms are consisted of two steps: primary classification and secondary classification. The effectiveness of our proposed approach is validated using the percentage accuracy obtained from cross-validation.

2 Proposed Approach

The proposed approach in our research is divided into three steps: data collection, feature extraction, and classification. The BSN we use include two types of nodes: the sender and the receiver which communicate wirelessly with each other [9]. The sender has the accelerometer sensor attached to it, and it is controlled with the use of a sampling program available with the BSN. The sender node is fixed to the volunteer's hand and the acceleration data of the hand movement is recorded for a period of 8 seconds while he/she performs the following activities: jogging, walking, resting, picking up and putting down things, transitional activities (lying to sitting and, sitting to standing). This data is stored offline for further use. Fig. 1 shows the steps involved in this approach.

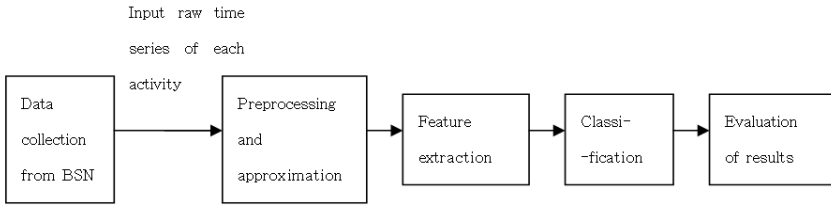


Fig. 1. The overview of our proposed approach

2.1 Data Collection from BSN

This is the first step in our proposed approach. The process of data collection involves setting up the BSN and recording of the acceleration of the volunteer's hand movements while the four classes of activity (jogging, walking, resting, and transitional) are being performed. The X, Y and Z acceleration data are recorded in a CSV file for as long as eight seconds. The yellow curve (Series 3) of Figure 2 represents the X acceleration. The pink curve (Series 2) of Figure 2 indicates the Y acceleration. The blue curve (Series 1) of Figure 2 is the Z acceleration. Thus for each class we record a collection of 100 instances. A sample time chart of the jogging activity is shown in Fig. 2.

2.2 Preprocessing and Approximation

Our second step is to adopt SAX like algorithm [7] to the time series of each activity. The Sax algorithm converts the time series into a symbolic representation. The advantages of using SAX include dimensionality reduction, discretization, lower distance bounding and numerosity reduction which makes it apt for the body sensor network data. The procedure is as follows: first the time series is normalized to remove noise, the time series of length n is divided into w equal parts and the average of each part is calculated. This is piecewise aggregate approximation (PAA) of the sequence. The cutpoints are determined from the Gaussian look up table. Usually the number of cutpoints is set as 3 and hence 4 symbols would be used for time series. Then the PAA which lies in the specified cut point slot is assigned a unique symbol or character. Thus we obtain the symbolic representation of the time series. For more details of the SAX algorithm, please refer to the original paper [7].

2.3 Feature Extraction

In the third step of our proposed approach, the SAX approximation is converted to a time series density map or bitmap. This approach is particularly useful while streaming real time data for classification. It is small and efficient enough to be handled by the BSN process or itself. The algorithm is described as follows. The Sax representation from the previous step (second step) has its alphabet size as four. We

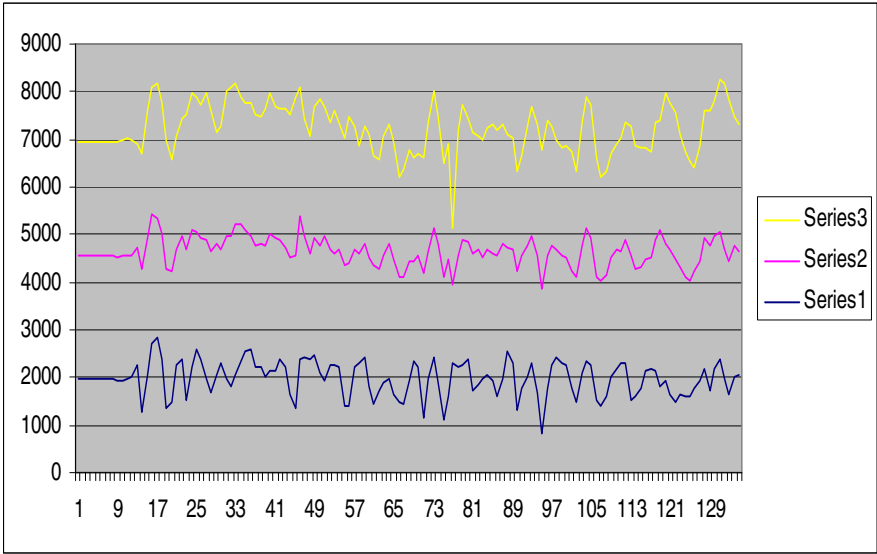


Fig. 2. The time chart for jogging activity. The yellow curve (Series 3) represents the X acceleration. The pink curve (Series 2) indicates the Y acceleration. The blue curve (Series 1) is the Z acceleration.

assign each alphabet a key. For example if A, G, C and T were the symbols used, they are assigned key values 1, 2, 3 and 4. Now the SAX representation is converted into a time series density map by moving a sliding window across the pattern and updating a frequency matrix of length $2^l \times 2^l$. The value of l is set as three since it has been found to work best in many experiments. It is also the length of the combination of symbols used in the Sax representation. An example of the matrix for length $l=1, 2$ and 3 is shown in Fig. 3.

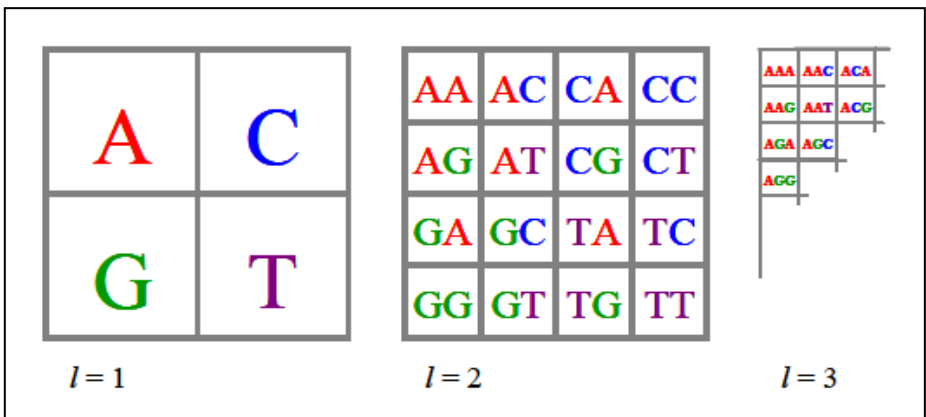


Fig. 3. The mapping of words of $l = 1, 2$ and 3

Suppose $l=4$, the first word in the sequence of the example string GATCACAGGTCTATCACCC is GATC. Let $k_0 = G$, $k_1 = A$, $k_2 = T$ and $k_3 = C$ be the key values. To map this word to a bitmap we use the formula,

$$col = \sum_{n=0}^{l-1} (k_n * 2^{l-n-1}) \bmod 2^{l-n} \quad row = \sum_{n=0}^{l-1} (k_n / 2) * 2^{l-n-1} .$$

Thus we obtain

the density map of patterns in the sequence or the bitmap of the sequence, by sliding a window of length l through it and updating the matrix based on the frequency of the possible combinations in the sequence. The output is an 8×8 matrix of cells which is time series density map of the specific activity instance. A training file is thus created with the 64 values as attributes for all the 400 instances of activity, including their class specified. This is fed as input to machine learning-based classifier.

2.4 Classification

This is the last step of our proposed approach. In order to evaluate the classification performance of the proposed approximation and feature extraction algorithms (step 2 and step 3), six machine learning-based classification algorithms were chosen. They are Bayes net, Naïve bayes multinomial, Simple logistic, Logit boost, Rotation forest, and Random forest. The classification was divided into primary and secondary steps. Primary classification performed the upper level classification for the four activities. The secondary classification performed lower division classification on the output of the primary i.e. the transitional activities were further classified into working (data pick up and put down) and transition activities (lying to sitting and sitting to stand). Training and cross validation testing are then performed on the training file. The evaluation of results is discussed next in the paper.

3 Experimental Result

3.1 Data Input to Classifier

The training data file contained 400 instances in total with four class divisions and 64 attributes. This was for the primary classification. The file for the secondary classification contained 200 instances with two class divisions and 64 attributes.

3.2 Classification Results

The classification results for the primary and secondary are shown in Table 1 and Table 2. Six classifiers have been used. The X, Y, and Z classification have been performed separately and their average result is also shown for number of folds as 8 for primary and 12 for secondary. The primary classification receives around 90 percent accuracy for all classifiers, while the secondary classification achieves only 70 percent. This shows that if the actions are well defined the time series density map would make a good choice for feature extraction in body sensor networks.

Table 1. Percentage classified correctly for primary classification

Eight Folder Cross Validation	Bayes Net	Naïve bayes Multinomial	Simple Logistic	Logit Boost	Rotation Forest	Random Forest
accelerometer X	90	90	90	88	90	90
accelerometer Y	90	90	90	90	88	89
accelerometer Z	90	88	91	89	89	89
average	90	89	90	89	89	89

Table 2. Percentage classified correctly for secondary classification

Eight Folder Cross Validation	Bayes Net	Naïve bayes Multinomial	Simple Logistic	Logit Boost	Rotation Forest	Random Forest
accelerometer X	67	71	74	71	76	75
accelerometer Y	48	63	85	78	82	77
accelerometer Z	48	57	62	58	67	68
average	54	64	74	69	75	73

4 Conclusion

We have shown in this paper that the time series density maps feature extraction method works well for the body sensor network data. It has also been estimated that this method has a high percentage of accuracy when it comes to classifying body movements and activities of BSN data. The feature extraction technique used here can be easily deployed on the BSN, enabling quick, time and space efficient and accurate activity classification with streaming sensor data. This can be implemented as future work, where we can use more sensors and nodes and improve the accuracy further by using combined data. We will also investigate the new feature extraction and classification algorithms under challenging environments, such missing sensor data, low quality of service, and unreliable sensor data. Our ultimate goal is to develop effective and efficient classifier for real-world clinical applications and thus serve a good purpose in health monitoring systems.

References

- [1] Yicka, J., et al.: Wireless sensor network survey. *Computer Networks* 5, 2292–2330 (2008)
- [2] Su, W., et al.: Wireless sensor network survey. *Wireless Sensor Networks: a Survey* 38, 393–422 (2002)
- [3] Yang, G.-Z.: *Body Sensor Networks*. Springer Science+Business Media LLC, New York (2006)
- [4] Schmidt, R., et al.: Body Area Network BAN—a key infrastructure element for patient-centered medical applications. *Biomedizinische Technik. Biomedical Engineering* 47, 365–368 (2002)

- [5] Kasetty, S., et al.: Real-Time Classification of Streaming Sensor Data. In: Proc. of 20th IEEE Int'l Conference on Tools with Artificial Intelligence (2008)
- [6] Guenterberg, E., et al.: An Automatic Segmentation Technique in Body Sensor Networks Based on Signal Energy. In: Proc. of The Fourth International Conference on Body Area Networks (BodyNets) (2009)
- [7] Lin, J., et al.: A symbolic representation of time series, with implications for streaming algorithms. In: Proc. of the 8th SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, California, USA (2003)
- [8] Kumar, N., et al.: Time-series Bitmaps: a Practical Visualization Tool for Working with Large Time Series Databases. In: Proc. of SIAM 2005 Data Mining Conference, Newport Beach, California, USA (2005)
- [9] Yang, G.-Z., et al.: Tutorial on Body Sensor Networks (2010),
<http://vip.doc.ic.ac.uk/bsn/tutorial>

Software Service Selection by Multi-level Matching and Reinforcement Learning

Rajeev R. Raje, Snehasis Mukhopadhyay, Sucheta Phatak,
Rashmi Shastri, and Lahiru S. Galge

Indiana University Purdue University Indianapolis, Indianapolis IN 46202, USA
{rraje,smukhopa}@cs.iupui.edu
<http://www.cs.iupui.edu/~rraje>

Abstract. The software realization of distributed systems is typically achieved as loose coalitions of independently created services. The selection of such services, to act as building blocks of a distributed system, is a critical task that requires discovery and matching activities. This selection task is generally based on simple matching techniques and without any notion of customization. This paper presents a method to achieve the service discovery process using the principles of multilevel matching based on multi-level specifications and customization based on reinforcement learning techniques. In this method, services are selected dynamically using an on-line performance-based reinforcement feedback. In contrast to methods which require the services to actually carry out a task before being selected, in the method proposed in this paper, service selection is carried out using only specification matching, thereby eliminating a large amount of redundant computation. Experimental results are presented in the context of a information classification system. These experiments demonstrate that a high degree of performance can be achieved at a much reduced computational cost using only multi-level specification-matching based reinforcement feedback signals.

Keywords: software services, multi-level specifications, discovery, classification, reinforcement learning, acquaintances.

1 Introduction

The selection of relevant software services is a necessary and critical step in the creation of distributed systems that are composed of independently developed and deployed services. The task of discovering and selecting such services for a specific query is carried out typically by a discovery system (DS). There have been many attempts of creating discovery systems such as, Jini [1], UPnP [2], SLP [3,4], UDDI [5], CORBA Trader [6], MDS [7], Ninja [8], and WSPDS [9], which have been classified as first-generation DS by [10]. A majority of these services carry the task of discovery by using a centralized publication mechanism and matching by using attribute-value pairs. In this paper, we present an approach to the discovery and selection of relevant services, for a particular

query, using the principles of customization and multi-level matching. These two features make the discovery process more comprehensive and efficient than the prevalent options.

The current discovery systems do not contain the notion of customization. Thus, they typically carry out an exhaustive search to identify appropriate services for a given query. As these current alternatives are limited in size and scope of the service search space, such an exhaustive search is still feasible. However, if the notion of a large-scale service bazaar is to be realized over a wide area network, the exhaustive approach will not be feasible due to the performance issues. Hence, there is a need to carry out selective search using the principles of customization. Customization, when added to the discovery mechanism will identify relevant services for a given query without the expense of an exhaustive search over a network. This could be achieved by storing the history of the previous service discoveries. Customization can be achieved by incorporating profiling techniques that use the concepts of machine learning (e.g., reinforcement learning [11]). The challenges associated with customization are related to the nature, size, and levels of these profiles, the entities to be profiled, and the exact learning techniques used in updating these profiles. The reward and penalty techniques used in such reinforcement learning can be realized by maintaining a history about the queries propagated.

A monolithic technique based on attribute and value pairs for matching, as used by a majority of the current approaches, is clearly not sufficient to identify the most appropriate services for a given query, as it does not differentiate between multiple similar services for a given query. As compared to this approach, matching based on multiple levels such as the type, the semantic contract, synchronization constraints, QoS values, and temporal attributes will allow the selection of the most appropriate (or relevant) services and/or also rank them using the outcome of the match for a given query. Thus, multi-level matching is a way to compare two software services and can help to determine whether one service can be substituted for another service or if one service can interact with the other service. An implicit requirement to support such a multi-level match is the presence of a multi-level specification that formally describes many facets (e.g., programmatic, semantic, QoS, etc.) of a service. The exact mechanisms for describing such a multi-level specification of a service and associated matching operators depend on the nature of a particular facet. For example, the type hierarchy can be used during the syntactical matching, while numerical operators can be used for the matching of QoS values. These definitions of matching operators help to capture the notions of generalization, specialization, substitutability, sub-typing, and interoperability of software services.

One machine learning approach for agent (or software service) selection (also called acquaintance learning), in the context of document classification, is described in [12]. It uses a vector space model, term frequency-inverse document

¹ In this paper, we have used the term *agent* interchangeably with the term *service*, as agents also offer specific services. In particular, we restrict ourselves to agents offering document classification services in this paper.

frequency method, with various and disparate document collections to produce classification agents with varying vocabularies that classify new documents by similarity to generated centroids. If an agent generates a null vector the document is unclassified, but might be classified by an agent with a different vocabulary. An 81 term Computer Science vocabulary was broken into nine disjoint sub-vocabularies creating agents that attempt to classify their own document sets, and time permitting, try to assist other remote agents. In the multi-opinion model all remote agents try to classify unclassified documents but as the number of agents available increases a saturation point is reached where more agents result in a small incremental increase in successful classification while response time increases linearly with the number of agents. Thus, proper selection of a small number of remote agents could achieve high performance at low response time and could be achieved by creating a small acquaintance list for each agent using a reinforcement learning algorithm called Pursuit Learning algorithm. On the basis of quickest return, or highest similarity value return, a best acquaintance is chosen and given a positive ranking weight which will modify the probability that its future choice will result in a reward. Algorithm performance compared to four best off line chosen agents resulted was 39% better than a random selection and 363% better than a worst four performance [12].

The limitation of the approach described above is that all agents need to carry out their task of document classification to generate the performance-based reinforcement signal, although only a small number are selected in the overall task execution. This leads to a large amount of duplicate, redundant computation. In contrast, in this paper, we propose that the reinforcement signals should be generated by multi-level specification matching, rather than actual execution results of all the contacted agents. Since such matching is substantially faster and less computationally expensive than the tasks for which the agents are designed, the overall discovery system will be computationally much more efficient. However, a critical question remains regarding the relative accuracy of such specification matching-based reinforcement learning of agents, since specifications are merely proxies of the capabilities of each agent. We argue that, due to the inherent robustness and noise tolerance of probabilistic reinforcement learning methods and due to the fact the discovery relies only on correct rank-ordering of the agents (rather than requiring very accurate values of the performance measures), such approximate matching-based reinforcement learning may result in near optimal discovery system, albeit with a much lower cost than performance-based discovery systems. Indeed, in this paper, we present experimental studies related to the same information classification domain as that reported in [12], to validate this claim.

2 Related and Past Work

There are many efforts at designing discovery systems in the domain of service-oriented architecture. For example, Jini [1], UPnP [2], SLP [3,4], UDDI [5], CORBA Trader [6], MDS [7], Agora [13], Ninja [8], and WSPDS [9] use the

attribute-value pairs, which are used in the matching process. DReggie [14], Structural ontology matching techniques [15,16], various UDDI enhancements [17,18,19,20] and SemB-UDDI [21] perform matching by the use of semantic ontology and markup languages such as DAML [22]. Such a combination does improve the matching performance over the basic attribute-value matching. GloServ [23], CDBMS [24] and OCTOPOS [25] use an hierarchy-based approach to matching. An enhancement of GloServ [26] is achieved by adding to it an ontology-based matching.

There have also been some efforts of designing DS in Cloud Computing. For example, the Cloud Service Discovery System [27] finds relevant Cloud services by using a simple Cloud Ontology, a frequency analysis, and similarity measures. [28] provides architecture for the cloud services to perform service selection with adaptive performances and minimum cost. The service selection algorithm still has the limitation of performing the basic keyword-based matching. [29] indicates the discovery process based on simple attribute matching provided by the experimental platform of Amazon EC2. All these efforts employ fairly minimal approaches to describe the services that are deployed in clouds and hence, their matching semantic tends to be simplistic. None of these approaches use customization and multi-level matching that is beyond the basic attribute-based and the ontology-based semantic matching schemes.

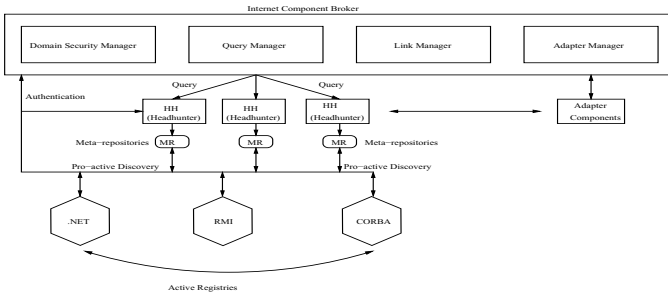


Fig. 1. The UniFrame Resource Discovery Service (URDS)

2.1 UniFrame Resource Discovery Service (URDS)

UniFrame [30] provides an environment for a seamless inter-operation of heterogeneous distributed services. A service in UniFrame is described by a multi-level specification, called UMM specification, which is an enhancement of the multi-level contract principle introduced in [31]. The UMM specification for a service is primarily made up of syntactic, semantic, synchronization, QoS, and deployment levels. Hence, in UniFrame the services are deployed on the network with their multi-level specifications. The URDS provides the infrastructure that supports the publication, deployment, and detection of the services. The URDS is a pro-active and hierarchical discovery system. The URDS architecture [32,33,34], shown in Figure 1, consists of: a) Headhunters (HH), b) Active Registries (ARs),

and c) Internet Component Broker (ICB). Each headhunter is associated with a meta-repository, which serves as a local store. Services, when implemented, are registered with the active registries using their UMM specifications. Once services are registered, the task of a headhunter is to discover them pro-actively, store their UMM specifications in its meta-repository, and perform a matching between the UMM specifications and any incoming queries issued by the user. The headhunters use multi-level matching (i.e., matching all the levels of the UMM specifications while responding to a specific query to identify relevant services. Headhunters may collaborate with each other if a requested service is not available in their local stores. The active registries are enhanced versions of the native registries of various implementation models (e.g., Java-RMI) with which services are registered. These enhancements allow Active Registries to listen and respond to multi-cast messages from the headhunters and have introspection capabilities to discover the services along with their specifications. The Internet Component Broker comprises of the Query Manager (QM), the Domain Security Manager (DSM), Adapter Manager (AM), and the Link Manager (LM). The Domain Security Manager serves as an authorized entity, which handles the secret key generation and distribution and enforces group memberships on other entities in URDS. The purpose of the Query Manager is to dispatch a user's query to the available headhunters. These headhunters then return lists of services matching the search criteria expressed in the query. The Link Manager serves to establish links with other Internet Component Brokers for the purpose of creating a federation and to propagate queries received from the Query Manager to other linked Internet Component Brokers. The Adapter Manager allows interoperability between heterogeneous services that may be needed to create a distributed system. In this paper, we have used the URDS (mainly the headhunter and its multi-level matching techniques) for the Multi-agent Classification System along with the principles of customization.

2.2 Reinforcement Learning

Originally motivated by mathematical psychology models of animal and child learning, reinforcement learning refers to the ability of an agent to learn long-term optimal behavior through the use of a reinforcement, i.e., an on-line performance feedback from a teacher or environment. The reinforcement, in turn, may be qualitative, infrequent, delayed, or stochastic. There is a rich body of reinforcement learning literature encompassing a wide array of learning models and algorithms. One of the earliest models of reinforcement learning is called a learning automaton [35] where the agent attempts to learn the optimal action from a finite set using reward/penalty reinforcement from a stationary teacher/environment with unknown reward probabilities. The learning problem is formulated as updating the agent's action probabilities on the basis of trials consisting of an action performed and the reinforcement received. While a wide variety of model-based and model-free learning algorithms has been proposed for a learning automaton with different asymptotic convergence properties, a

popular model-free algorithm is the so-called L_{RI} (Linear Reward-Inaction) algorithm described by

$$p_i(k+1) = p_i(k) + \alpha r(k)(1 - p_i(k))$$

$$p_j(k+1) = p_j(k) - \alpha r(k)p_j(k)$$

where $p_i(k)$ is the agent's probability of choosing action a_i at trial k , a_i is the action chosen at trial k , $r(k)$ is the reinforcement received (with $r(k) = 0$ signifying penalty, and $r(k) = 1$ signifying reward), and $\alpha > 0$ is the learning step-size. The idea is to increase the probability of the chosen action linearly if a reward is received (while reducing the other action probabilities) and not to change the action probabilities if a penalty is received. The L_{RI} algorithm has been shown to be ϵ -optimal, i.e., the asymptotic probability of converging to the optimal action can be made as close to 1 as desired by choosing a sufficiently small step-size α . While the L_{RI} algorithm belongs to the class of model-free learning algorithm (since it does not maintain or use any estimate of the environmental reward probabilities), there are also algorithms that are model-based. The Pursuit Learning Algorithm [35], for example, maintains and updates estimates of the reward probabilities, and adjusts the action probabilities by a step size in the direction of the unit vector that represents the current estimate of the optimal action. The Pursuit Learning Algorithm has also been proved to be ϵ -optimal.

It is quite clear that some form of machine learning technique needs to be used for the selection of remote services (i.e., acquaintance learning) to deal with uncertainty and/or dynamic changes in the environment. The advantage of reinforcement learning over other machine learning approaches (such as supervised learning and unsupervised learning) is that the former is inherently an on-line learning method, where the reinforcement data required for learning is generated during operation, thereby avoiding a large effort in off-line data generation and collection. Further, the reinforcement feedback signal needs to be only a qualitative and noisy indicator of how well the agent is performing, rather than a labeling of the data (as in supervised learning) to indicate the "correct" action. These advantages are particularly relevant in open, dynamic services environments where existing services may be removed and new services may be added at unknown instants of time. Reinforcement learning offers the possibility of adapting to these dynamic changes, by making use of the reinforcement feedback, without a large effort (and/or delay) in retraining of the agent. Hence, we have used the reinforcement learning technique in our proposed methodology for the design of a multi-agent classification system.

3 Methodology

Based on the work reported in [12], we propose a Multi-agent Classification System with specification matching. In this system, we propose the use of active registry and headhunters of the URDS for choosing the correct agent for classification of an incoming document. Multiple classification agents will register their

UMM specifications with the active registries and are eventually discovered by the headhunters. The headhunter in our experiments maintains an acquaintance list made up of top three classification agents. This list is dynamically updated using the reinforcement learning with the help of previous collaboration history. Unlike [12], the headhunter maintains its acquaintance based on matching of agent specifications with the clients query. For example (more details of our approach are provided in next section) in one scenario, the headhunter chooses the members of the acquaintance list by selecting the top three agents decided by the matching of their specifications against the incoming query and then the top acquaintance with highest matching score is chosen for the task of classification. Reward in the reinforcement learning algorithm is based on the matching score. Top acquaintance gets reward of 1 and all other acquaintance get reward of 0. Their Action Probability, and Estimated reward probability are updated by the formulas given above (and given in [12]). Headhunter now updates its acquaintance list by choosing the agent based on its action probability. To explore the learning we use following algorithm for choosing the first acquaintance:

SortedActionProbability holds the action probability values of all the agents in an ascending order.

```

Rand = RandomNum(0,1);
Sum = SortedActionProbability [0];
i=0;
While(Rand<Sum)
{
    Sum=Sum+SortedActionProbability[i++];
}
ChooseAgentAtIndex[i];

```

Other acquaintances are the agents with highest Estimated Reward Probability. Thus by using matching technique for choosing the acquaintance, we reduce the classification cost and final classification is only done by the top matching acquaintance. This technique is useful when the specification of an agent or service clearly describes the agent. If the specification is detailed enough then we can correctly identify the best service for that query. Next section analyzes the experimental results of this system. Similar to experiments described in [12], our experiments implement the pursuit learning algorithm for learning automata that works in feedback with environment. For each action, the environment provides reward or punishment reinforcement with some probability. We have designed a model-based pursuit algorithm that maintains the estimate of the action probabilities, which are unknown to classification agents. A query from the client is forwarded to a headhunter. Each headhunter maintains an acquaintance list and keeps track of each agent's action probability. It matches the user query with registered UMM specifications of agents. Whenever a successful match is found, the headhunter rewards and increases the action probability of the selected agent. Initially, all the agents have an equal probability and eventually each agent will be tried and explored.

We have implemented two different techniques for matching of agent UMM specification to queries. The first technique is called as the explicit matching. In

this technique, the queries contain a domain and sub domain of the document to be classified. Every agent when registered with the headhunter provides a detailed UMM specification which is extended to include its domain of classification. Hence in this approach, the headhunter matches the query domain along with other parts of the UMM specification and decides the matching score. This technique assumes that the clients have some idea about the classification domain. On the other hand, if the clients do not have any idea about the domain of the document they want to classify, then the explicit domain matching will not work. Hence, we use another method wherein the headhunter selects the acquaintances in a random manner initially. In both these cases, after the initial selection, the future selections of agents use the reinforcement learning technique.

4 Experimentation

We conducted a number of experiments to test the effectiveness of the proposed approach. The first set of experiments did not use the reinforcement learning algorithm used and the headhunter would just select any random agent for classification or would send the document to all the agents for classification. This formed as the base cases for comparing the classification performance. The next sets of experiments involved the use of acquaintance lists of headhunters selected by using the explicit (i.e., the direct description of classification domain) and implicit (i.e., using the thesaurus) matching techniques. Our experiments included ten agents with overlapping thesauri and ten sets of documents from a single sub domain. The results and analysis of experiments are provided below.

4.1 Experiments without Reinforcement Learning Algorithm

Experiment 1: In this experiment, all documents were sent to all classifier agents so that the resulting classification success rate was 100%. However, since all classifier agents attempted to classify all documents, this is by far the most expensive method computation and communication-wise.

Experiment 2: In this experiment, documents were sent to three classifier agents chosen at random. The classification success rate was observed to be only 40%, but the computation time required was less than the previous experiment.

Experiment 3: In this experiment, documents were sent to only one randomly chosen classifier agent. The success rate in classification was observed to be only 20% in this case but it was faster than the previous two cases.

4.2 Experiments with Reinforcement Learning Algorithm Using Explicit Matching

Experiment 1: Classification achieved by selecting the Top Acquaintance

In this experiment, the top acquaintance is selected by a headhunter for classification queries by means of the matching the document domain, along with other aspects of the UMM specification of agents. As the acquaintance list is

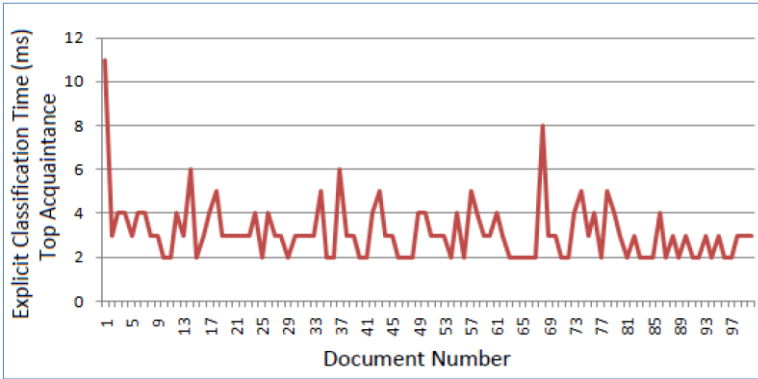


Fig. 2. Explicit Classification by Selecting the Top Acquaintance – Classification Time

constantly updated, using the reinforcement algorithm, during each classification only the top one is chosen. The performance (processing time), and the classification success rate are shown Figures 2 and 3 respectively. In Figure 2, the X-axis denotes the 100 documents (10 sets of 10 documents) and the Y-axis indicates the processing time (in milli-seconds) taken by the top acquaintance. After the initial period, the classification time saturates a value that is between 2 and 4 ms. The X-axis in Figure 3 indicates the set number (each set consisting of 10 documents), while the Y-axis indicates the number of successful document classifications for each set. From Figure 3, it can be seen that the classification performance quickly reaches the 100% level (i.e., 10 successful classifications in each set). This is hardly a surprise, as the top acquaintance which is most suitable for classification, is learnt by the headhunter quickly. When compared with the non-learning based exhaustive technique (Section 4.1), this scheme provides a comparable performance with a lot less computation (i.e., only one agent classifying the documents versus all 10 agents classifying the documents).

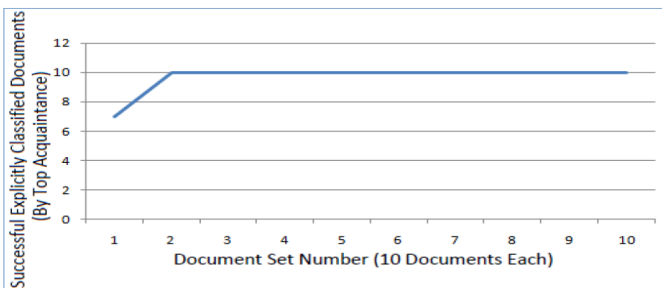


Fig. 3. Explicit Classification by Selecting the Top Acquaintance – Classification Success

Experiment 2: Classification done by Randomly chosen acquaintance.

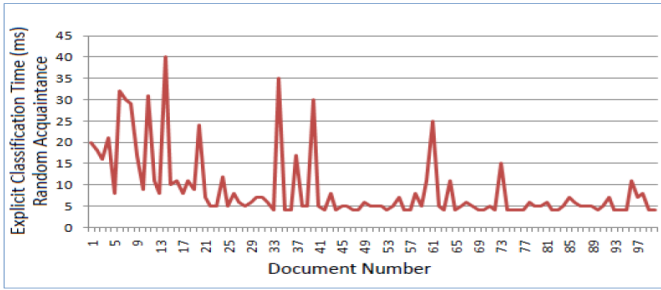


Fig. 4. Explicit Classification by Selecting a Random Acquaintance – Classification Time

In this case, an agent is selected by the headhunter from the acquaintance list in a random manner. Figures 4 and 5 show the performance of this scheme. The X and Y axes indicate the same entities as in Figures 2 and 3. As seen from Figure 4, the classification time varies randomly and in some cases is similar to the one in Figure 2, and the classification performance (Figure 5) is poorer than the case with the selection of the top-acquaintance (Figure 3). This is also expected, as the random selection does not guarantee the selection of the top acquaintance.

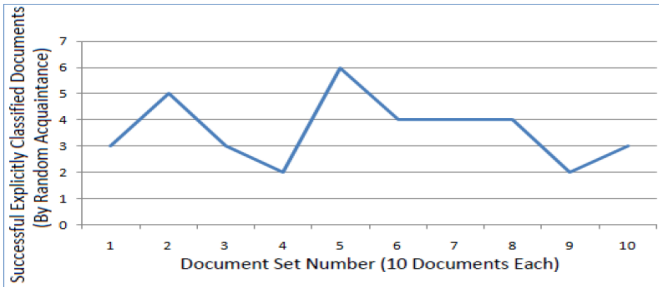


Fig. 5. Explicit Classification by Selecting a Random Acquaintance – Classification Success

4.3 Experiments with Reinforcement Learning Algorithm Using Implicit Matching

Experiment 1: Classification achieved by selecting the Top Acquaintance

In this experiment, the top acquaintance is selected by a headhunter for classification queries in a random manner initially. The selected agent gets a chance to

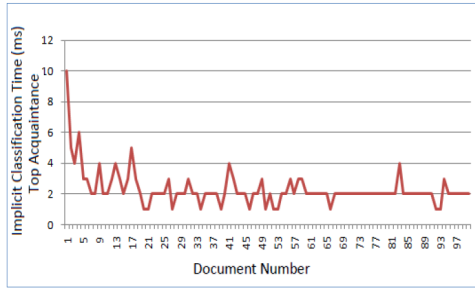


Fig. 6. Implicit Classification by Selecting the Top Acquaintance – Classification Time

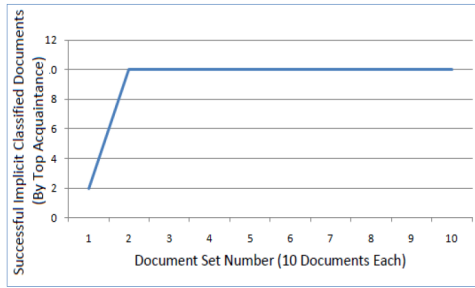


Fig. 7. Implicit Classification by Selecting the Top Acquaintance – Classification Success

classify the document and if it is successful, it gets a reward (as indicated above by the reinforcement algorithm), else it gets a penalty. The future iterations always use the top acquaintance during the selection process. The performance (processing time), and the classification success rate are shown Figures 6 and 7 respectively. In Figure 6, the X-axis denotes the 100 documents (10 sets of 10 documents) and the Y-axis indicates the processing time (in milli-seconds) taken by the top acquaintance. After the initial period, the classification time saturates a value that is close to 2 ms. The X-axis in Figure 7 indicates the set number (each set consisting of 10 documents), while the Y-axis indicates the number of successful document classifications for each set. From Figure 7, it can be seen that the classification performance quickly reaches the 100% level (i.e., 10 successful classifications in each set). This is hardly a surprise, as the top acquaintance is learnt by the headhunter quickly.

Experiment 2: Classification achieved by selecting Randomly chosen Acquaintance.

In this case, the initial acquaintance list is randomly selected and an agent is also selected by the headhunter from the acquaintance list in a random manner. Again, the selected agent, if successful, gets a reward, else gets a penalty and the list is updated accordingly. Figures 8 and 9 show the performance of this scheme. The X and Y axes indicate the same entities as in Figures 6 and 7. As

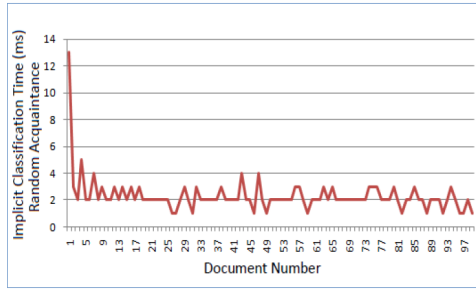


Fig. 8. Implicit Classification by Selecting a Random Acquaintance – Classification Time

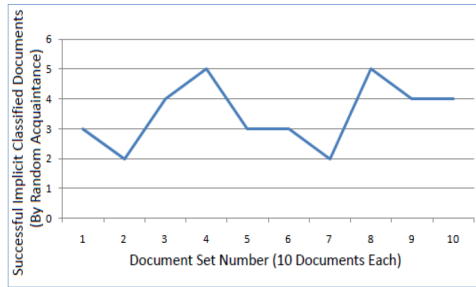


Fig. 9. Implicit Classification by Selecting a Random Acquaintance – Classification Success

seen from Figure 8, the classification time is similar to the case with the selection of the top acquaintance (Figure 6) but the classification performance (Figure 9) is poorer than the case with the selection of the top-acquaintance (Figure 7). This is also expected, as the random selection does not guarantee the selection of the top acquaintance.

4.4 Experiments Involving Fault Situations

Here to simulate the fault situations, during the classification process, the top acquaintance agent is deleted from the system after every 10 document classifications and after every next 10 passes a new agent is added to the system. The results are presented below Figures 10 and 11. As seen from Figure 10, the classification time does increase when the top acquaintance agent is deleted and it does reduce as the system learns the new top acquaintance. However, the classification performance (as shown in Figure 11) does not exhibit a predictive pattern. This can be attributed to the randomness of documents and the inclusion of a random agent into the system.

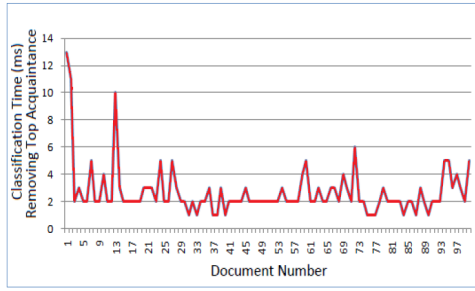


Fig. 10. Classification while Deleting the Top Acquaintance – Classification Time

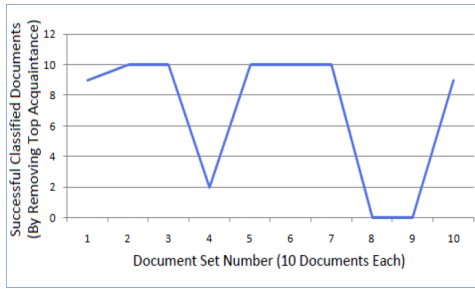


Fig. 11. Classification while Deleting the Top Acquaintance – Classification Time

4.5 Summary of Experimental Results

The above experiments indicate that the learning-based algorithm outperforms the discovery without learning (except the case where the documents are sent to all the agents) when the classification success is chosen as the metric of performance. When classification done by all classifiers, and with no learning, the classification success is 100% but it is at the cost of additional computation. In the case of explicit and implicit matching, with learning based algorithm, the scheme which uses the top acquaintance for classification outperforms the scheme where randomly selected acquaintances are used in the classification. The scheme involving top acquaintances for classification achieves a comparable classification success with the exhaustive case (i.e., on learning but the usage of all the agents in classification). Thus, incorporation of learning, in addition to multi-level matching, delivers a comparable performance while selecting appropriate agents for document classification.

5 Conclusion

In this paper, we have presented a method for learning of preferred acquaintance by using specification matching for classification agents. This is useful where agents are not only distributed in nature but they randomly join and leave the

system. Our experiments show that the learning based on specification matching not only reduces cost of sending document to all classifier, but also gives a good classification performance. We show that, with little overhead of maintaining model based pursuit learning algorithm and specification matching, the service discovery system converges fast and provides near optimal solution as far as the classification success is considered.

References

1. Sun Microsystems. Jini Specifications V2.0, <http://www.sun.com/software/jini/specs/>
2. UPnP Organization. UPnP Home Page (2005), <http://www.upnp.org>
3. Kemp, J., St. Pierre, P.: Service Location Protocol for Enterprise Networks. Wiley and Son Inc., ISBN 0-47-3158-7
4. OpenSLP Organization. OpenSLP Home Page (2005), <http://www.openslp.org>
5. UDDI Technical White Paper (2000), http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf
6. Object Management Group. Trading Object Service Specification (2000), <http://www.omg.org/docs/formal/00-06-27.pdf>
7. Globus Toolkit (2007), <http://www.globus.org/toolkit/>
8. von Behren, J., Brewer, E., Borisov, N., Chen, M., Welsh, M., MacDonald, J., Lau, J., Culler, D.N.: A Framework for Network Services. In: Proceedings of USENIX Annual Technical Conference (2002)
9. Banaei-Kashani, F., Chen, C., Shahabi, C.: WSPDS: Web Services Peer-to-peer Discovery Service (2004), http://infolab.usc.edu/DocsDemos/isws2004_WSPDS.pdf
10. Dabrowski, C., Mills, K., Quirolgico, S.: A Model-based Analysis of First-Generation Service Discovery Systems. Technical report, NIST Special Publication, 500-260 (October 2005), http://w3.antd.nist.gov/pubs/SP500_260final.pdf
11. Thathachar, M., Sastry, P.: A New Approach to the Design of Reinforcement Schemes for Learning Automata. IEEE Transactions on System Man Cybernetics 15, 168–175 (1985)
12. Mukhopadhyay, S., Peng, S., Raje, R., Palakal, M., Mostafa, J.: Multi-Agent Information Classification Using Dynamic Acquaintance Lists. Journal of the American Society for Information Science and Technology 54(10), 966–975 (2003)
13. Seacord, R., Hissam, S. and Wallnau, K. Agora: A Search Engine for Software Components. Technical report, Carnegie Mellon University, CMU/SEI-98-TR-011, ESC-TR-98-011 (1998)
14. Chakraborty, D., Perich, F., Avancha, S., Joshi, A.: DReggie: A Smart Service Discovery Technique for E-Commerce Applications. In: Proceedings, 20th Symposium on Reliable Distributed Systems (October 2001)
15. Di Martino, B.: Semantic web services discovery based on structural ontology matching. In: Proceedings of IJWGS (2009)
16. Lin, C., Wu, Z., Deng, S., Kuang, L.: Automatic Service Matching and Service Discovery Based on Ontology. In: Jin, H., Pan, Y., Xiao, N., Sun, J. (eds.) GCC 2004. LNCS, vol. 3252, pp. 99–106. Springer, Heidelberg (2004)

17. Paolucci, M., Kawamura, T., Payne, T., and Sycara, K. Importing the Semantic Web in UDDI. In: Workshop on EBusiness and Semantic Web (2001)
18. Kawamura, T., De Blasio, J.-A., Hasegawa, T., Paolucci, M., Sycara, K.: Preliminary Report of Public Experiment of Semantic Service Matchmaker with UDDI Business Registry. In: Orłowska, M.E., Weerawarana, S., Papazoglou, M.P., Yang, J. (eds.) ICSSOC 2003. LNCS, vol. 2910, pp. 208–224. Springer, Heidelberg (2003)
19. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.: Semantic Matching of Web Services Capabilities. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 333–347. Springer, Heidelberg (2002)
20. Colgrave, J., Akkiraju, R., Goodwin, R.: External Matching in UDDI. In: Proceedings of IEEE international Conference on Web Services (2004)
21. Aguilera, U., Abaitua, J., Diaz, J., Bujan, D., Ipiná, D.: Semantic Matching Algorithm for Discovery in UDDI. In: Proceedings of International Conference on Semantic Computing (2007)
22. DARPA. The DARPA Agent Markup Language (2006), <http://www.daml.org/>
23. Arabshian, K., Schulzrinne, H.: GloServ: global service discovery architecture. In: Mobile and Ubiquitous Systems: Networking and Services, pp. 319–325 (2004)
24. Skouteli, C., Samaras, G., Pitoura, E.: Concept-based discovery of mobile services. In: MDM 2005: Proceedings of the 6th International Conference on Mobile Data Management, pp. 257–261. ACM, New York (2005)
25. Gu, T., Qian, H., Yao, J., Pung, H.: An architecture for flexible service discovery in OCTOPUS. In: ICCCN, pp. 291–296 (2003)
26. Arabshian, K., Dickmann, C., Schulzrinne, H.: Ontology-Based Service Discovery Front-End Interface for GloServ. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 684–696. Springer, Heidelberg (2009)
27. Taekgyeong, H., Kwang, M.: An Ontology-enhanced Cloud Service Discovery System. In: Proceedings of International Multiconference of Engineers and Computer Scientists (2010)
28. Zeng, W., Zhao, Y., Zeng, J.: Cloud service and service selection algorithm research. In: Proceedings of ACM/SIGEVO Summit on Genetic and Evolutionary Computation (2009)
29. Rajiv, R., Liang, Z., Xiaomin, W., Anna, L.: Peer-to-Peer Cloud Provisioning: Service Discovery and Load-Balancing. In: Proceedings of CoR (2009)
30. Indiana University Purdue University Indianapolis. UniFrame Project (2010), <http://www.cs.iupui.edu/uniFrame>
31. Beugnard, A., Jezequel, J., Plouzeau, N., Watkins, D.: Making Components Contract Aware. IEEE Computer 32(7), 38–45 (1999)
32. Siram, N.: An Architecture for the UniFrame Resource Discovery Service. Master's thesis, Indiana University Purdue University Indianapolis, Department of Computer and Information Science (2002)
33. Siram, N., Raje, R., Bryant, B., Olson, A., Auguston, M., Burt, C.: An Architecture for the UniFrame Resource Discovery Service. In: van der Hoek, A., Coen-Porisini, A. (eds.) SEM 2002. LNCS, vol. 2596, pp. 20–35. Springer, Heidelberg (2003)
34. Raje, R., Gandhamaneni, J., Olson, A., Bryant, B.: MURDS: A Mobile-Agent-based Distributed Discovery System. In: Taniar, D. (ed.) Encyclopedia of Mobile Computing and Commerce, Hershey, USA, vol. 1, pp. 207–212 (2007)
35. Narendra, K.S., Thathachar, M.A.L.: Learning Automata: An Introduction. Prentice-Hall (1989)

Interoperating DNA Gene Sequences and Nutrition Provisions for Personalized Wellness

Jong P. Yoon¹ and Joyce Yoon²

¹ Mercy College, MATH/CIS Dept, Dobbs Ferry, New York

² Columbia University, Institute of Human Nutrition, College of Physicians and Surgeons
New York, New York

jyoon@mercy.edu, jny2106@columbia.edu

Abstract. The last decade has seen a large explosion of health-related human centered computing research and practice focused on wellness such as provisioning good nutrition. Human genome projects become successfully recognized and DNA sequencing technologies become affordable. While Health Informatics may appear to be the obvious home for these activities, interoperability between DNA gene sequence analysis and nutrition support is less satisfactorily achieved. To promote humans health and wellness, this paper describes two-tiered nutrition support system: rule-based and case-based reasoning about gene sequences and nutrients. In addition, this paper also protects human information from unauthorized accesses while preserving its privacy.

Keywords: Nutrition-net, Gene-net, Privacy Preservation, Rule-based Gene Sequence and Nutrient Reasoning, Case-based Reasoning.

1 Introduction

The last decade has seen a large explosion of health-related human centered computing research and practice focused on wellness such as good nutrition with the intention of helping people avoid needing medical care. Human genome projects become successfully recognized and DNA sequencing technologies become affordable [2,8,10]. While Health Informatics may appear to be the obvious home for these activities [3], it needs to provide the highest quality nutritional care for human wellness.

Malnutrition is not only nutritional imbalance or over nutrition, but also irrelevant or inaccurate nutrition. Nutrition usually refers to a deficiency of nutrients (under nutrition) relative to body requirements which contributes to an abnormality in body composition and/or its function [6,7]. This deficiency may arise from inadequate intake or absorption of good food, or exceeding intake of improper foods to a specific human body. It is likely that a food is good to a body while harm to another.

The resources listing accurate nutrition information is available nationwide and worldwide as well, with smart eating guidelines (e.g., <http://www.nutrition.gov>, <http://www.nal.usda.gov/fnic/pubs/bibs/gen/eatsmart.pdf>, or <http://fnic.nal.usda.gov/resourcelists>). It is important to eat healthy foods but more importantly to avoid the

foods which are adverse to the body state. Human body contains chemical compounds, such as water, carbohydrates (sugar, starch, and fiber), amino acids (in proteins), fatty acids (in lipids), and nucleic acids (DNA and RNA). These compounds are composed of elements such as carbon, hydrogen, oxygen, nitrogen, phosphorus, calcium, iron, zinc, magnesium, manganese, and so on. All of these chemical compounds occur in various forms and combinations (e.g., hormones, vitamins, phospholipids, hydroxyapatite) in the human body.

The contribution of this paper includes 1) personalized wellness promotion that can be achieved from interoperation between gene sequence analysis and nutrition development, and 2) a security technique that can protect human information from unauthorized accesses and providing services by preserving the privacy of human information. A broad architecture of the proposed technique is as shown in Figure 1. For wellness of human life, a list of foods is provisioned to a person based upon his or her DNA sequence. This provision is made to more general group of people (④ in Figure 1), and further specifically to a specific person (⑤ in figure). Since there exists the privacy issue in ⑤, a person can communicate securely.

The remainder of this paper is organized as followed: Section 2 describes about nutrition and nutrition-net. Section 3 describes about DNA and gene-net. Section 4 describes

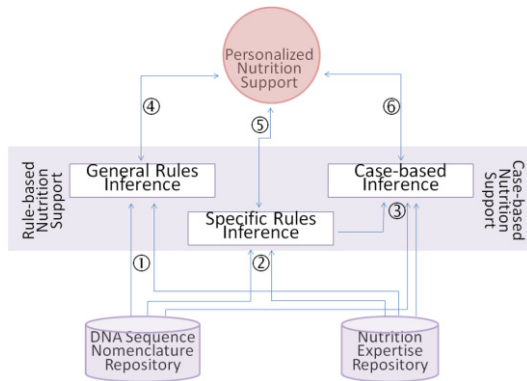


Fig. 1. General System Architecture

2 Nutrition

Nutrition is the provisioning process of the materials necessary foods to the organisms or cells of body [6,10]. There are six major classes of nutrients: water, carbohydrates, protein, vitamin, fats and minerals. Foods are those edible substances that can provide human body with certain kinds of nutrients, like proteins, fat, fiber, carbohydrates, and different kinds of vitamins and minerals.

The stomach has limited space and should only be filled foods that are rich in essential nutrients, and are therefore beneficial for the improvement and development

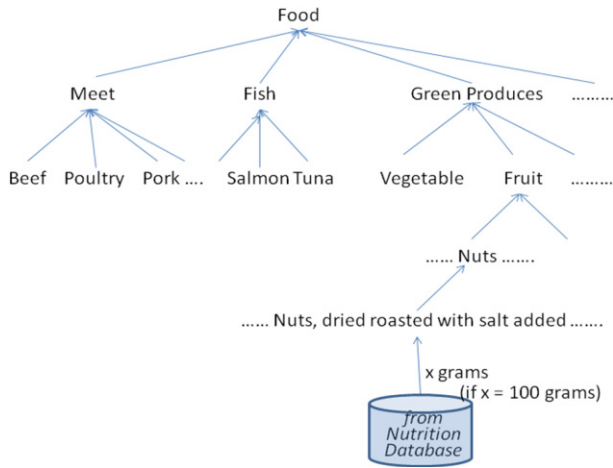


Fig. 2. Food Hierarchy of Nutrition-Net

of human brain and body. Incorporating additive nutrients into our food products is not new at all. Milk producers have been fortifying milk by adding extra vitamin D to it since the 1930s. Vitamin D helps prevent rickets in children, who are most vulnerable to this disease. For people who aren't able to consume lactose, adding vitamin D to their diets is a problem.

Also, the amount of nutrients that a human body can or need to consume is important. For example, more than 2000 milligrams of vitamin C per day in an adult human body can cause diarrhea and kidney stones. Stanols and plant sterols help reduce cholesterol inside one's body, and 2 grams of these substances per day is enough to keep cholesterol in control. Ingesting more than that amount can harm the body and does not provide any extra benefit. Iron is necessary for the growth of blood cells, but in some forms iron is not solvable is simple flushed out of the body.

Foods can be hierarchically organized [2]. One of the examples is shown in USDA National Nutrient Database (i.e., <http://www.nal.usda.gov/fnic/foodcomp/search/>). Foods are classified as being meat, fruits, vegetables, fishes, etc. Each can be also further specialized. For example, meats can be beef, pork, poultry, etc, each of which in turn can be specialized. Foods can be organized in a net, which we call "nutrition-net."

A nutrition-net is composed of a hierarchy of foods and the nutrition

Table 1. Nutrients Returned from the Nutrition Database

Nutrient	Value per 100 grams
Water	1.75g
Protein	17.30g
Lipid (fat)	51.45g
....
Carbohydrate	25.35g
Fiber, total dietary	9.0g
Sugars	4.65g
Calcium	70mg
....
Vitamin C	0.4mg
Vitamin B-6	0.296mg
Choline	54.3mg
....
Tryptophan	0.264g
Leucine	1.371g
....
Caffeine	0mg

database. As shown in Figure 2, the nutrition data may provide a list of the nutrients. In the figure, consider the food called “nuts, dried roasted with salt added” in the leaf node. It contains the nutrients as shown in Table 1 in the case of 100 grams of the food. Depending on the variable x , which indicates the weight of food, different values of nutrients will be provided. An example is listed as shown in Table 1.

3 DNA

Deoxyribonucleic Acid (DNA) is a nucleic acid that contains the genetic instructions used in the development and functioning of all known living organisms with the exception of some viruses. DNA occurs as linear chromosomes in eukaryotes, and circular chromosomes in prokaryotes. The set of chromosomes in a cell makes up its genome. The human genome has approximately 3 billion base pairs of DNA arranged into 46 chromosomes. The information carried by DNA is held in the sequence of pieces of DNA called genes.

An alternative form of a gene, i.e., one member of pairs, located at a specific position on a specific chromosome is called an allele. Each gene can have different alleles. Sometimes, different DNA sequences or alleles can result in different traits, for example skin color. It is likely that different DNA sequences or alleles may have the same result in the expression of a gene. At each locus among various individuals, multiple alleles exist. Allelic variation at a locus is measurable as the number of alleles present or the proportion of heterozygotes in the population.

Polymerase Changing Reaction (PCR) is a technique in molecular biology to amplify a single or few copies of the piece of DNA across several orders of magnitude, generating thousands to millions of copies of a particular DNA sequence. The method relies on thermal cycling, consisting of cycles of repeated heating and cooling of the reaction for DNA melting and enzymatic replication of the DNA. Primers (short DNA fragments) containing sequences complementary to the target

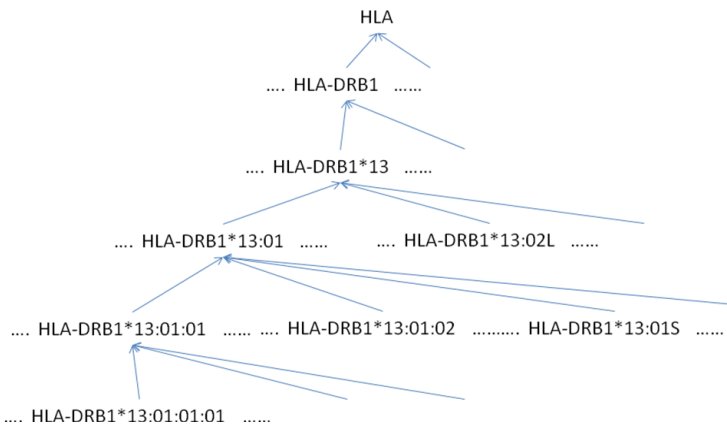


Fig. 3. An Example of Gene-Net

region along with DNA polymerase are key components to enable selective and repeated amplification. As PCR progresses, the DNA generated is itself used as a template for replication, setting in motion a chain reaction in which the DNA template is exponentially amplified.

Virtual Hybridization is a technique that rearranges chromosomes and enables to infer phylogenetic relationship between species using “gene” order. Human Leukocyte Antigen (HLA) is the name of the major histocompatibility complex in humans. The locus of the HLA contains a large number of genes related to immune system function in humans. This group of genes resides on chromosome 6, and encode cell-surface antigen-presenting proteins and many other genes.

Each HLA allele name has a unique number corresponding to up to four sets of digits separated by colons. The length of the allele designation is dependent on the sequence of the allele and that of its nearest relative. All alleles receive at least a four digit name, which corresponds to the first two sets of digits, longer names are only assigned when necessary. Modern HLA alleles are typically noted with a variety of levels of detail. Most designations begin with HLA- and the locus name, then * and some number of digits specifying the allele. The first two digits specify a group of alleles. Older typing methodologies often could not completely distinguish alleles and so stopped at this level. The third through fourth digits specify a synonymous allele. Digits five through six denote any synonymous mutations within the coding frame of the gene. The seventh and eighth digits distinguish mutations outside the coding region. Letters such L, N, Q, or S may follow an allele’s designation to specify an expression level or other non-genomic data known about it. Thus, a completely described allele may be up to 9 digits long, not including the HLA- prefix and locus notation.

For example, the nomenclature HLA-DRB1*13 specifies a group of alleles which encode the DR13 antigen or sequence homology to other DRB1*13 alleles, while HLA-DRB1*13:01 specifies a specific HLA allele. HLA-DRB1*13:01:02 specifies an allele that differs by a synonymous mutation from DRB1*13:01:01, while HLA-DRB1*13:01:01:02 specifies an allele which contains a mutation outside the coding region from DRB1*13:01:01:01. As exemplified, there are relationships amongst DNA gene or allele sequences. Such relationships can be depicted in a graph, called gene-net. An example of gene-net is shown in Figure 3.

4 Wellness Inference

As discussed in the previous sections, we know that foods can be organized in the nutrition-net, while DNA gene sequences are in a gene-net. In Figure 1, there are two types of logical expression: general expression and specific expression. A general expression is defined over general terms, while a specific expression over specific terms. For example, *Gene*(“freshmen in CS dept”, “UCC”), meaning that freshmen in the class CISC131 has a gene sequence codon UCC, is a general expression [9], and *Gene*(“Adam”, “UCC”) is a specific expression. In this section, we do not distinguish them. However, Section 5 considers the situation where specific rules need to inference, and therefore, the privacy of a specific information needs to be preserved.

This section describes how to suggest a food to a person according to his or her body state. There are basically two cases of nutrition suggestion: well-known rule-based suggestion and case-based reasoning suggestion [8].

4.1 Rule-Based Suggestion for Well-Known Cases

Consider the two rules; 1) a certain gene sequence causes a symptom, and 2) food is good for those who have specific DNA gene sequences. Such rules can be written in logic as follows:

$$\begin{aligned} \text{Gene } (s, g) &\rightarrow \text{Symptom } (s, x) \\ \text{Food } (s, f, a) &\rightarrow \text{Gene } (s, g). \end{aligned} \tag{1}$$

These two rules imply that if a person s has the gene sequence g , then that person has the symptom of (disease) x . Also, the gene sequence g is caused if a amount of food f is consumed.

From the rules in (1), the following rule is derived.

$$\begin{array}{l} \text{Gene } (s, g) \rightarrow \text{Symptom } (s, x), \text{ and Food } (s, f, a) \rightarrow \text{Gene } (s, g) \\ \hline \text{Gene } (s, g) \rightarrow \neg \text{Food } (s, f, a) \end{array} \tag{2}$$

meaning that if the two rules, $\text{Gene } (s, g) \rightarrow \text{Symptom } (x)$ and $\text{Food } (s, f, a) \rightarrow \text{Gene } (s, g)$, are given, the food f has mal-nutrient.

For example, it is known that the A1 allele associated with DRD2 gene binding in the striatum and compromised striatal dopamine signaling reduces obese individuals. In other words,

$$\begin{aligned} \text{Gene } (s, \text{"DRD2"}) &\rightarrow \text{Symptom } (s, \text{"obesity"}), \text{ and} \\ \text{Food } (s, \text{"Sugar"}, a) &\rightarrow \text{Gene } (s, \text{"DRD2"}). \end{aligned}$$

With these rules, it turns out that Sugar is not recommended to those who have gene "DRD2". That is, $\text{Gene } (s, \text{"DRD2"}) \rightarrow \text{Food } (s, \text{"Sugar"}, a)$.

We further consider both nutrition-net and gene-net. Since hierarchical, they can be also rewritten in logic as follows:

$$\begin{array}{l} \text{Gene } (s, g) \rightarrow \text{Food } (s, f_i, a), \text{ and } f_i \rightarrow f_j \\ \hline \text{Gene } (s, g) \rightarrow \text{Food } (s, f_j, a). \end{array} \tag{3}$$

and

$$\begin{array}{l} \text{Gene } (s, g_i) \rightarrow \text{Food } (s, f_i, a), \text{ and } g_i \rightarrow g_j \\ \hline \text{Gene } (s, g_j) \rightarrow \text{Food } (s, f_j, a). \end{array} \tag{4}$$

Equation (3) means that if rule $\text{Gene } (s, g) \rightarrow \text{Food } (s, f_i, a)$ exists and food f_j is super entity to another food f_i in a nutrition-net, then a new rule is generated such that $\text{Gene } (s, g) \rightarrow \text{Food } (s, f_j, a)$. Similarly, Equation (4) states that if rule $\text{Gene } (s, g_i) \rightarrow$

Food (s, f_i, a) exists and gene g_j is a super entity to another gene g_i in a gene-net, then a new rule $\text{Gene}(s, g_j) \rightarrow \text{Food}(s, f_i, a)$ is generated.

4.2 Cased-Based Reasoning

We know that there are many undiscovered rules, which however will be yet another useful food suggestion to human life. For this we employ the technique of case-based reasoning [8]. The case-based reasoning is the process of solving new problems on the solution of similar past problems. The idea comes from the fact that an auto mechanic who fixes an engine by recalling another car that exhibited similar symptoms. For wellness of human life (in Figure 1-Ⓞ), this case-based reasoning may be more frequently used in part because human DNA gene sequences are increasing and its new nomenclature is designed.

An example of case-based reasoning is depicted in Figure 4. We know how to suggest the healthy food N_i if gene sequence Ω_i exists. For the wellness for a person who has a gene sequence Ω_j , if there is no foods that are found healthy for the person, but we do know gene sequence Ω_i is similar to Ω_j as shown in the second diagram of Figure 4(2), then we may be able to suggest the food N_j . Of course, we need to have a transformation rule α_{ij} for gene sequence, and also β_{ij} for food transformation. Finding α_{ij} and β_{ij} is beyond the scope of this paper, but will be discussed in another paper.

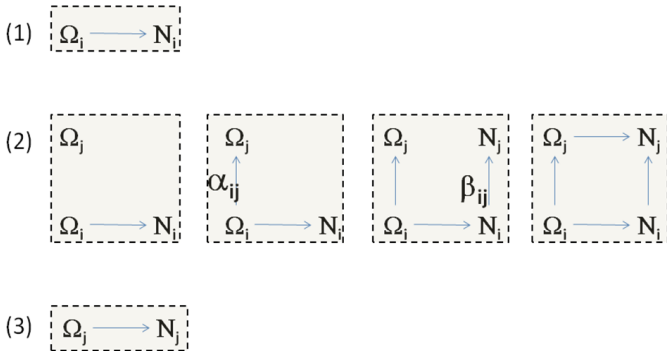


Fig. 4. Discovering in Case-based Reasoning

5 Privacy Issues

This section describes a technique that can preserve the privacy for particular personal information in specific rules inference. In Figure 1-Ⓞ, since rules are defined over specific information, e.g., personal information, and since generalization approach [9] is not satisfied, user accesses need to be controlled properly. The access control techniques have been developed successfully [1]. However, they are not efficient in the domain interoperating two or more repositories. The following employs PKI [4].

To request the server to reason about specific DNA gene sequence with specific personal information in specific rules, participants need asymmetric keys. The following can generate private and public keys for nutritionist Joyce who helps patient John.

```
keytool -genkey -alias CA1 -keystore herOnly.ks
-storepass ColumbiaUniversity -keypass nutrients.org
-dname "CN=Joyce, OU=Nutrition, O=Columbia, L=New York,
ST=NY, C=US" ;
```

 (5)

After the certificate authority CA1 signs on the request jar file B.jar containing DNA gene sequences for John, and to generate the signed jar file Bs.jar. The access policy is defined to grant the write permission to the jar code Bs.jar to a gene sequence database, gen-net to access John's gene sequence.

In this environment, the following java execution securely

```
java -Djava.security.manager -Djava.security.policy
= our.policy -cp Bs.jar John gen-net
```

 (6)

In this way, the gene sequences for John and the nutrition data matched with his gene can be inter-shared securely and privately.

6 Summary

In this paper, we introduced two biological structures: DNA gene-net and nutrition-net. For interoperability of DNA gene-net and nutrition-net, two-tiered nutrition assistant system is proposed. The nutrition assistant system uses rule-based and case-based reasoning about to promote humans wellness and health. This interoperability are performed securely and privately.

References

1. Ben Ghorbel-Talbi, M., Cuppens, F., Cuppens-Boulahia, N., Bouhoula, A.: Managing Delegation in Access Control Models. In: IEEE ADCOM (2007)
2. Ghoting, A., Makarychev, K.: Indexing Genomic Sequences on the IBM Gene. In: Proc. of the Conf. on High Performance Computing Networking, Storage and Analysis (2009)
3. Haslhofer, B., Klas, W.: A survey of techniques for achieving metadata interoperability. ACM Computing Surveys 42 (2010)
4. Huang, J., Nicol, D.: A calculus of trust and its application to PKI and identity management. In: ACM IDtrust 2009 (2009)
5. Igel, C., Glasmachers, T., Mersch, B., Pfeifer, N., Meinicke, P.: Gradient-based Optimization of Kernel-Target Alignment for Sequence Kernels Applied to Bacterial Gene Start Detection, pp. 216–226 (2007)
6. Kelley, P., Bresee, J., Cranor, L., Reeder, R.: A "Nutrition label" for privacy. In: Proc. of the 5th Symposium on Usable Privacy and Security (2009)
7. National Center for Biotechnology Information, <http://www.ncbi.nlm.nih.gov>

8. Neves, M., Chagoyen, M., Carazo, J., Pascual-Montano, A.: CBR-Tagger: a Case-based Reasoning Approach to the Gene/Protein Mention Problem. In: Proc. of the Workshop on Current Trends in Biomedical Natural Language Processing (2008)
9. Rowell, L.: Writing the rules of digital privacy. *ACM networker* 13(2) (2009)
10. Zhao, W., Serpedin, E., Dougherty, E.: Inferring Connectivity of Genetic Regulatory Networks Using Information-Theoretic Criteria. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 262–274 (2008)

Proposing a Novel Artificial Neural Network Prediction Model to Improve the Precision of Software Effort Estimation

Iman Attarzadeh and Siew Hock Ow

Department of Software Engineering
Faculty of Computer Science & Information Technology
University of Malaya, 50603 Kuala Lumpur, Malaysia
attarzadeh@siswa.um.edu.my, show@um.edu.my

Abstract. Nowadays, software companies have to manage different software development processes based on different time, cost, and number of staff sequentially, which is a very complex task and supports project planning and tracking. Software time, cost and manpower estimation for separate projects is one of the critical and crucial tasks for project managers. Accurate software estimation at an early stage of project planning is counted as a great challenge in software project management, in the last decade, as it allows considering project financial, controlling, and strategic planning. Software effort estimation refers to the estimations of the likely amount of cost, schedule, and manpower required to develop software. This paper proposes a novel artificial neural network prediction model incorporating Constructive Cost Model (COCOMO). The new model uses the desirable features of artificial neural networks such as learning ability, while maintaining the merits of the COCOMO model. This model deals efficiently with uncertainty of software metrics to improve the accuracy of estimates. The experimental results show that using the proposed model improves the accuracy of the estimates, 8.36% improvement, when the obtained result compared to the COCOMO model.

Keywords: Software engineering, software project management, software cost estimation models, COCOMO model, soft computing techniques, and artificial neural networks.

1 Introduction

Accurate and reliable software cost and time estimation is counted as one of the great problems and ongoing challenges in software engineering, in the last decades. Enhancing the precision of estimates would facilitate more effective time and budgets controlling for project managers during the development process. In order to make precise software estimates, several algorithmic and non-algorithmic cost estimation models have been proposed and developed. The Constructive Cost Model (COCOMO) is the most popular model in software companies due to its capabilities and characteristics to estimate software effort in person-month (PM) during

development process. This paper proposes a precise artificial neural network estimation model incorporating COCOMO model to overcome the vagueness and uncertainty of software estimates.

1.1 Software Effort Estimation Models

Software project managers and developers, always, interested to estimate the total software budget and schedule at the early stages of development process. These estimates can help them to make good decisions on project management and strategic planning. Software effort prediction approaches can be categorised into algorithmic and non-algorithmic methods. Algorithmic methods use regression techniques with statistical analysis of historical data. Software Life Cycle Management (SLIM) [1] and Constructive Cost Model (COCOMO) [2] are two common algorithmic estimation methods. Non-algorithmic methods are based on heuristic approaches such as Expert Judgment, Price-to-Win, and machine learning approaches [3]. The COCOMO model was proposed by Barry Boehm in 1981[2] and it is one of the most cited, best known, widely used and the most plausible of all proposed effort estimation methods. The COCOMO model uses for project effort, time, cost, and manpower estimations. The COCOMO II model includes three sub-models for software estimations as follows: Application Composition Level, Early Design Level, and Post-Architecture Level. The Post-Architecture Level of COCOMO II includes 17 project cost factors, 5 scale factors, and software size, which presents project attributes and characteristics [2]. The COCOMO II formula shown in equation “1” as follows:

$$\text{Effort} = A \times [\text{Size}]^B \times \prod_{i=1}^{17} \text{Effort Multiplier}_i \quad (1)$$

$$\text{where } B = 1.01 + 0.01 \times \sum_{j=1}^5 \text{Scale Factor}_j$$

In the equation“1”:

A: Multiplicative Constant

Size: Size of the software measures in terms of KSLOC (thousands of Source Lines of Code, Function Points or Object Points)

1.2 Artificial Neural Networks

Artificial Neural Networks (ANN) are simplified mathematical techniques of human brain. They are collection of neurons, Process Elements (PEs), with internal connection and their function is based on distributed computing networks. They can learn from previous project information and experiences to provide new data, rules, and experiences based on inference of learnt data. The main idea in ANN is to produce intelligent systems capable of sophisticated computations. It is similar to the biological neurons in human brain structures. In fact, each neuron is like a

mathematical function with some inputs, a mathematical formula, and outputs. Each ANN includes a specific architecture, layers, and nodes. Each node has a mathematical function, inputs, and outputs, which generates a non-linear function of its input [3, 4]. Using ANN starts by generating network architecture then selecting a proper learning technique for train, test, and validate the network based on a data set. The most widely used ANN training techniques are feed-forward and recurrent techniques. Figure 1 shows the functionality of a node in ANN.

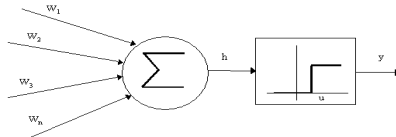


Fig. 1. The functionality of a node in ANN

Each node produce the weighted sum of its M inputs, x_j , where $j = 1, 2, \dots, m$, and generate an output of 1 if this result is above the defined threshold u . Otherwise, an output of 0 generates. The obtained formula is shown in equation “2”.

$$y = \theta \left(\sum_{j=1}^m W_j X_j - u \right) \tag{2}$$

In the equation “2”, the θ is a unit step function at 0 and w_j is the synapse weight associated with the j -th input. U is considered as another weight i.e. $w_0 = -u$ attached to the neuron with a constant input of $x_0 = 1$. Positive weights model excitatory synapses, while negative weights model inhibitory ones. The activation function in Figure 1 is known as a step function however, there are a number of functions that can be utilised such as Sigmoid, Gaussian, and Linear [5, 6].

2 Related Works

Software researchers attempt to improve software effort estimation models to overcome the uncertainty of results. Many software effort estimation models have been proposed and developed over the last decades. Using capabilities of artificial neural networks in software effort estimation, especially learning from historical project, can be as an alternative to achieve acceptable results. Using back propagation learning algorithm on a multilayer perceptron is one good application of soft computing techniques, which proposed by Witting and Finnie [7] to estimate software development effort. In another research, Karunanithi [8] suggested using artificial neural network techniques such as the feed-forward and Jordon-network with expert experiments to estimate software flexibility and reliability.

Samson [9] utilised another soft computing technique, Albus multiplayer perceptron, to estimate software development time and cost. He used the COCOMO data set, which includes 63 projects information. A different artificial neural network

with back propagation learning algorithm proposed by Tadayon [10], however, it is not clear how the applied dataset was divided to train and validate of his proposed system. Khoshgoftaar and Jingzhou [11, 12] considered a real time approach to estimate the usability of each software metrics such as source lines of code. Researchers still attempt to apply and use the advantages of artificial neural networks to propose an accurate, reliable and flexible software estimation model. The ANN has been successfully used for solving several problems in software engineering [13, 14].

3 The Proposed COCOMO Model Incorporating Artificial Neural Networks

In this research a new architecture of ANN proposes to accommodate the COCOMO II Post-architecture model. The COCOMO II includes five scale factors (SF), seventeen effort multipliers (EM), and software size. The proposed model has 23 inputs, which include software size, scale factors, effort multipliers and the system output is effort estimation in PM (person-months). Therefore, the proposed artificial neural network architecture includes 23 input nodes in the input layer, which corresponds to all SFs, EMs, and software size parameters. Applying the proposed ANN architecture on the COCOMO II post-architecture model needs to data pre-processing in the input layer using the Sigmoid Activation function. The proposed ANN architecture for the COCOMO model shown in Figure 2 and 3 and follows:



Fig. 2. Architecture of proposed artificial neural network

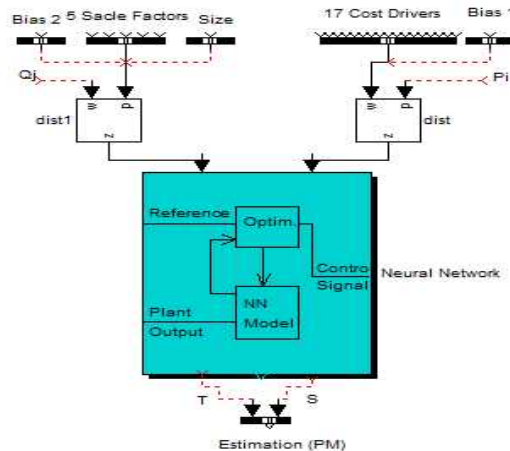


Fig. 3. The proposed artificial neural network based on COCOMO II

In the proposed ANN model, all effort multipliers values, EM_i , used in COCOMO model are pre-processed to $\log(EM_i)$ and the size of the product, in KSLOC, is not considered as one of the input parameters to the network but as a co-factor for the initial weights, initialisation parameter, for scale factors (SF). The sigmoid activation function in the hidden layer is defined by $f(x) = \frac{1}{1+e^{-x}}$. The related weights of input nodes connected to the hidden layer are defined by P_i for Bias1 and each input $\log(EM_i)$ for $1 \leq i \leq 17$. Besides, the related weights with each scale factor, SF_j , from input nodes to the hidden layer are $q_j + \log(\text{size})$ for $1 \leq j \leq 5$ and the bias defined by Bias2. The parameters ‘W’ and ‘b’ in Figure 2, show the related weights to the arcs from the hidden layer nodes to the output layer nodes. The weight parameters ‘W’ and ‘b’, are relevant to the values of the hidden layer nodes. The output nodes have the specific identity function.

One of the contributions of this research compared to other related works is the addition of $\log(\text{Input Size})$ to the weight Q_j of scale factors in system input, which adjusts the weights Q_j . Another important difference in this model compared to other works is the training and biasing approach of artificial neural network. Customisation of the COCOMO formula is done by adjusting the initial values of weights ‘W’ and ‘b’ to the offset of the values of the nodes in the ANN hidden layers. The back-propagation algorithm is used as the training method in the proposed model. The data sets that used for system training will discuss at the following section. However, if there is no appropriate data set for system training, the weights and biases parameters in the model results the estimation using random generated input/output. The proposed model output, the effort, would be derived from COCOMO equation, ‘1’, by considering the initial values of Bias1 as $\log(A)$ and Bias2 as 1.01. The ANN weights are initialised as $p_i = 1$ for $1 \leq i \leq 17$ and $q_j = 1$ for $1 \leq j \leq 5$. For propagating the input parameters the values of nodes in the hidden layer are considered as follow:

$$f(p_0 \text{ Bias1} + \sum_{i=1}^{17} p_i * \log(EM_i)) = \text{sigmoid} (\text{Bias1} + \sum_{i=1}^{17} p_i * \log(EM_i)) = \frac{A * \prod_{i=1}^{17} EM_i}{1 + A * \prod_{i=1}^{17} EM_i} = \alpha \tag{3}$$

$$f((q_0 + \log(\text{size})) * \text{Bias2} + \sum_{j=1}^5 (q_j + \log(\text{size}))(SF_j)) = \text{sigmoid} (\log(\text{size}) * (\text{Bias2} + \sum_{j=1}^5 SF_j)) = \frac{\text{Size}^{1.01 + \sum_{j=1}^5 SF_j}}{1 + \text{Size}^{1.01 + \sum_{j=1}^5 SF_j}} = \beta \tag{4}$$

Then initialisation of weights ‘W’ and ‘b’ as follow:

$$W = \frac{\beta}{2(1-\alpha)(1-\beta)} \quad \text{and} \quad b = \frac{\alpha}{2(1-\alpha)(1-\beta)} \tag{5}$$

The ANN output is calculated as:

$$PM = W * \alpha + b * \beta = \frac{\alpha\beta}{(1-\alpha)(1-\beta)} = A. \text{Size}^{1.01 + \sum_{j=1}^5 SF_j} * \prod_{i=1}^{17} EM_i \tag{6}$$

4 Training Algorithm

The training algorithm is done by iteration of forward and backward techniques until the terminating conditions are satisfied, for instance the changes in weights are less than or equal to a basic threshold or a basic number of iterations have been done.

The training algorithm includes follow steps:

- Selecting a training sample and propagate the input parameters across the ANN to compute the system output.
- Error detection in system output, and determining the amount of error gradient in all the other layers.
- Determining the amount of changes for the ANN weights and updating the ANN weights.
- Repeating the steps until the ANN error is sufficiently small, less than or equals a specific threshold, after an epoch is complete.

5 Results and Discussion

Experiments were done by using two different data sets: the original COCOMO data set, which includes 63 projects information and an artificial data set, which includes 100 projects data. The artificial data set inferred from the existing COCOMO dataset based on another ANN model using previous training algorithm.

5.1 Data Sets Description

The COCOMO data set, Data set #1, was the first attempt to evaluate the proposed ANN-COCOMO model, which is includes 63 historical projects and it is a public data set [1]. This data set retrieved form 63 historical projects from software companies and industries. The second data set created based on the COCOMO data set by proposing a new ANN architecture, Data set #2, which is trained by the COOCMO data set and characteristics of training algorithm explained in the previous section. Table 1 shows the structure of second data set.

Table 1. The artificial data set includes 100 projects information

No.	Mode	Size	Effort
1	1.1200	51.2500	246.5900
2	1.2000	12.5500	58.2800
3	1.0500	81.5200	550.4000
...
97	1.2000	56.5300	354.7300
98	1.0500	16.0400	67.1400
100	1.1200	54.1700	262.3800

5.2 Evaluation Method

The proposed ANN-COCOMO model evaluation and validation is done by using the most widely accepted evaluation methods: Mean Magnitude of Relative Error

(MMRE) and Pred (L). The Pred(L) method means probability of a project having a relative error of less than or equal to L (for instance, Pred(25%)). The Magnitude of Relative Error (MRE) is defined as follows:

$$MRE_i = \frac{|Actual\ Effort_i - Predicted\ Effort_i|}{Actual\ Effort_i} \tag{7}$$

The value of MRE is calculated for each observation i whose effort is estimated. The aggregation of MRE over multiple observations (for i=1 to N) can be achieved through the Mean MRE (MMRE) as follows:

$$MMRE = \frac{1}{N} \sum_i^N MRE_i \tag{8}$$

The Magnitude of Error Relative to the estimation (MER) is another measure similar to MRE. Intuitively, MRE seems preferable to MER since it measures the relative error to the estimate. MRE uses predicted effort as shows in equation “7”. The MMRE is used to the mean MER in equation “8”. However, the MMRE and MMER are sensitive to individual estimations with excessively large MREs or MERs. Therefore, the aggregate measure less sensitive to extreme values is also considered, namely the median of MRE and MER values for the N observations (MdMRE and MdMER respectively). A complementary condition is the prediction at level 1, Pred(l) = k/N, where k is the number of observations where MRE (or MER) is less than or equal to l, and N is the total number of observations. Thus, Pred(25%) gives the percentage of projects which were predicted with a MRE (or MER) less or equal than 0.25.

The proposed ANN-COCOMO model and the original COCOMO model are used for model evaluation. The two data sets, Data set #1 and Data set #2, separately applied to the new artificial neural network effort estimation model and original COCOMO model. For each project in the data set the estimated effort, MRE, and Pred (25%) calculated by applying on the proposed ANN model and the original COCOMO model. Finally, the MMRE for each data set is calculated to avoid any sensitivity to the calculated results. The results comparison of COCOMO data set, Date Set #1, and artificial dataset, Data Set #2, shown in Tables 2 and 3.

Table 2. Results comparison of the ANN proposed model and COCOMO model

Data set	Model	Evaluation	
		MRE	Pred (25%)
Data set #1	COCOMO II	0.542561832	45%
	Proposed Model	0.487257017	52%
Data set #2	COCOMO II	0.462579313	30%
	Proposed Model	0.428617366	39%
MMRE	COCOMO II	0.502570573	37.5%
	Proposed Model	0.457937192	45.5%

The results in the Table 2 shows: the values of MMRE in the proposed ANN model for Data set #1 and #2 are 0.487257017 and 0.428617366 and for the Pred(25%) are 52% and 39%. The aggregation of Data sets, #1 and #2 shows that the

MMRE is 0.457937192 and Pred(25%) equals 45.5%. The analysis of the results in Table 2 indicates:

- The proposed ANN effort estimation model has the MMRE, MMRE_ANN = 0.457937192, less than COCOMO model, MMRE_COCOMO = 0.502570573. Obviously, it means the accuracy of proposed ANN model is better than COCOMO model. Because, if the value of MMRE is closed to zero, it means the amount of error, difference of actual and estimated effort, is very low. In other words, the accuracy of estimation is high.
- In case of Pred(25%), the value of Pred(25%) for proposed ANN estimation model is 45.5% and for COCOMO model is 37.5%. Pred method presents the number of projects with the MRE less than 25%. Therefore, if the value of Pred(25%) is closed to 100%, it means the estimated value for the project is closed to the actual amount. So, the accuracy of estimation is high.

According to the analysis of the final results, the proposed ANN effort estimation model shows better accuracy than the COCOMO model. Table 3, compares the percentage of the accuracy improvement of ANN estimation model in compare to the COCOMO model.

Table 3. The accuracy of the estimation models

Model		Evaluation	
Proposed Model vs. COCOMO II	COCOMO II	MMRE	0.50257057
			3
	Proposed Model Improvement %		8.36%

The analysis of the final result indicates that the percentage of the accuracy improvement in the proposed artificial neural network model is 8.36%. In summary, the analysis of the experimental results shows that the proposed artificial neural network effort estimation model generates more accurate estimates than the COCOMO model and the system output provides a better performance due to the high granularity demanded from the results.

6 Conclusion

One on the crucial and challenging issues in software project management is accurate and reliable estimation of the required effort at the early stages of software development process. Software attributes essentially have properties of vagueness and uncertainty when they are measured by human judgment and they differ from software development environments. A software effort estimation incorporating artificial neural networks can overcome the vagueness and uncertainty of software attributes, which used in effort estimation. This approach can be a worthy attempt in the software project

management. This research work presented a new artificial neural network architecture to handle uncertainty and imprecision in software effort estimation.

This approach shows that by applying artificial neural network on the algorithmic effort estimation models, accurate estimates are achievable. The analysis of the results demonstrated that using artificial neural network approach for the software cost estimation is an applicable approach to address and overcome the vagueness and uncertainty of software attributes. Furthermore, the proposed artificial neural network estimation model presented better estimation accuracy when compared to the COCOMO model. The utilisation of soft computing approaches such as artificial neural networks, fuzzy logic, and neuro-fuzzy systems for other software engineering approaches can be considerable in the future.

References

- [1] Boehm, B.: *Software Engineering Economics*. Prentice-Hall, Englewood Cliffs (1981)
- [2] Boehm, B., Abts, C., Chulani, S.: *Software Development Cost Estimation Approaches – A Survey*. University of Southern California Center for Software Engineering, Technical Reports, USC-CSE-2000-505 (2000)
- [3] Putnam, L.H.: A General Empirical Solution to the Macro Software Sizing and Estimating Problem. *IEEE Transactions on Software Engineering* 4(4), 345–361 (1978)
- [4] Srinivasan, K., Fisher, D.: Machine Learning Approaches to Estimating Software Development Effort. *IEEE Transactions on Software Engineering* 21(2) (1995)
- [5] Molokken, K., Jorgensen, M.: A review of software surveys on software effort estimation. In: *IEEE International Symposium on Empirical Software Engineering, ISESE*, pp. 223–230 (October 2003)
- [6] Huang, S., Chiu, N.: Applying fuzzy neural network to estimate software development effort. *Applied Intelligence Journal* 30(2), 73–83 (2009)
- [7] Witting, G., Finnie, G.: Using Artificial Neural Networks and Function Points to Estimate 4GL Software Development Effort. *Journal of Information Systems* 1(2), 87–94 (1994)
- [8] Karunanithi, N., Whitely, D., Malaiya, Y.K.: Using Neural Networks in Reliability Prediction. *IEEE Software Engineering* 9(4), 53–59 (1992)
- [9] Samson, B.: Software cost estimation using an Albus perceptron. *Journal of Information and Software*, 55–60 (1997)
- [10] Tadion, N.: Neural Network Approach for Software Cost Estimation. In: *International Conference on Information Technology: Coding and Computing, ITCC*, pp. 116–123 (2005)
- [11] Khoshgoftar, T.M., Allen, E.B., Xu, Z.: Predicting testability of program modules using a neural network. In: *3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology*, pp. 57–62 (2000)
- [12] Jingzhou, L., Guenther, R.: Analysis of attribute weighting heuristics for analogy-based software effort estimation method AQUA+. *Empirical Software Engineering Journal* 13(1), 63–96 (2008)
- [13] Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering* 17(4), 491–502 (2005)
- [14] Chiu, N.H., Huang, S.J.: The adjusted analogy-based software effort estimation based on similarity distances. *Journal of Systems and Software* 25, 628–640 (2007)

Tracing Conformational Changes in Proteins Represented at a Coarse Level

Nurit Haspel*

Department of Computer Science, University of Massachusetts Boston, Boston MA 02125 USA
nurit.haspel@umb.edu

Abstract. Many large protein complexes undergo extensive conformational changes as part of their functionality. Tracing these changes is important for understanding the way these proteins function. It is not always possible to obtain a high resolution structure for very large complexes. Electron cryo-microscopy (Cryo-EM) enables the representation of large macromolecular structures at a medium resolution level (4–8Å). Traditional conformational search methods cannot be applied to medium resolution data where structural information may be partial or missing. Additionally, simulating large scale conformational changes in proteins require a massive amount of computational efforts. We apply a search method from robotics to structural information obtained from medium resolution Cryo-EM maps, modeled to approximate backbone trace level. The pathways obtained by this method can be useful in understanding protein motions, providing reliable results for the medium resolution data. To provide a baseline validation for our method, we tested it on Adenylate Kinase and Cyanovirin. To test the data on actual cryo-EM determined structures, we simulated the conformational opening of the GroEL single ring complex. We show that we can produce low energy conformational pathways which correspond to known structural data. The method presented here is a promising step towards exploring the conformational motion of even larger complexes.

1 Introduction

Proteins are flexible molecules that undergo conformational changes as part of their interactions with other proteins or drug molecules [1]. Changes in torsional angles may induce localized changes or large scale domain motions. Tracing these changes is crucial for understanding the way these proteins perform their function. Only a relatively small number of proteins have their structure determined experimentally at atomic resolution. In cases where the high resolution structure cannot be determined, as in the case of very large complexes, electron cryo-microscopy (Cryo-EM) enables the structural determination of large macromolecular complexes at sub-nanometer resolution (4–8Å) [2, 3]. At this resolution the backbone or sidechain positions cannot always be determined, and typically only partial information exists about the location and connectivity of the secondary structure elements. More recently, with the advance of Cryo-EM determination methods and data processing tools, an increasing number of medium resolution data can be traced to a backbone level with high accuracy. An example of a medium

* Part of the work was done when the author was at the Department of Computer Science at Rice University, Houston, TX 77005 USA.

resolution structure can be seen in Figure 1(a), where a backbone trace of the GroEL chaperonin monomer is modeled, based on Cryo-EM image. Several recent studies perform local refinement and fitting of high resolution structures to Cryo-EM models [4, 5], but to the best of our knowledge, no algorithm exists for tracing and analyzing the large scale conformational changes undergone by complexes represented at medium resolution. Hence, a gap of knowledge exists. Existing physics-based computational methods that trace and simulate conformational changes in proteins where full atomic resolution is available include Molecular Dynamics (MD) [6], Monte Carlo (MC) [7] and their variants. They normally require full atomic representation of the molecule, and most available force fields are not suitable to deal with coarse grained representation, especially when modeled from sub-nanometer resolution images. In the past, several efficient conformational search algorithms have been developed. Some use a coarse representation of the protein molecule [8, 9] and employ various efficient search methods such as Normal Mode Analysis (NMA) [4, 10], elastic network modeling (ENM) [11, 12, 13], or morphing [14]. ENM- and NMA-based methods are especially useful in sampling local motions, and it is not always clear whether large scale conformational sampling methods can be sampled correctly using normal mode analysis, and the normal modes have to be re-calculated for the sampling to be correct. Sampling based motion planning methods have been successfully applied towards an efficient exploration of the conformational space of macromolecules. Motion planning is an area in robotics concerned with finding a pathway for robot-like objects in constrained environments [15, 16]. When applied to biological problems, the protein is represented as an articulated body with the degrees of freedom in all or selected torsional angles. The physical constraints are implicitly encoded in a penalty function which approximates the potential energy of the molecule. The conformational space of the protein is explored so that high energy regions are avoided and feasible conformational pathways are obtained more efficiently than with traditional simulation methods. Among the many applications of motion planning to biology are the characterization of near-native protein conformational ensembles [17, 18, 19], the study of conformational flexibility in proteins [20, 21], protein folding and binding simulation [22, 23], modeling protein loops [24, 20], simulation of RNA folding kinetics [25, 26] and recently the elucidation of conformational pathways in proteins, subject to pre-specified constraints [27, 28].

Many of those methods are successful in sampling the conformational landscape of proteins but are often biased by the protein native conformation and some of them require additional, problem specific information.

In this work we present an efficient motion-planning based methodology to perform conformational search on complex macromolecular structures represented at medium resolution extracted from Cryo-EM imaging. The molecule is mapped into a reduced representation using a small number of parameters that represents its degrees of freedom. This allows for larger complexes to be explored efficiently. A similar method has been applied recently to higher resolution structures [28]. The results of the previous research encouraged us to take the methodology one step further and explore lower resolution structures with high degree of uncertainty. To the best of our knowledge, there are no large scale conformational sampling methods that accept lower than backbone

resolution and this is the main strength of this method and the main difference from our previous work at [28].

Problem Statement. Given two conformational states of a molecule, denoted by start and goal, and represented as two medium resolution Cryo-EM maps, our goal is to find a set of affine transformations that, when applied successively to the degrees of freedom of the start conformation, the start conformation will be brought within a tolerance range of the goal conformation under a defined distance metric. Furthermore, each intermediate conformation along the pathway must be feasible under a given scoring function, which approximates its potential energy. The degrees of freedom of the structures lie in the flexible parts connecting rigid structural elements. In this work we assume that secondary structure elements do not change significantly during domain motions and that the flexible parts are the loops connecting them. While this assumption is true in many cases, there are cases where secondary structure elements melt or change. In these cases, and if backbone or higher resolution modeling of these parts is available, it is possible to incorporate a more detailed modeling of the flexible parts into the general framework of the algorithm without limiting the proposed procedure. The flexible parts can be modeled using existing methods such as elastic network modeling [11]. We also assume that approximate backbone trace can be made using structural modeling tools and Cryo-EM data analysis methods [29, 30]. Extension to cases where only partial information is given about the location of secondary structures is a subject of on-going and future work. Figure 1 shows an illustration of the conformational transition of the GroEL monomer from the closed structure (Figure 1(a)) to the opened structure (GroEL-GroES-ADP7) (Figure 1(b)). Figure 1(c-d) show the single-ring 7 member complex. The closed conformation is a backbone trace model generated from a mid-resolution cryo-EM image.

It should be emphasized that the algorithm does not always produce the same conformational pathway, but rather a possible pathway. By repeating the procedure several times we produce a set of feasible pathways, thus limiting the huge search space to a manageable number of possibilities which can later be refined and filtered using information about the tested systems.

2 Methods

2.1 Data Representation

This part is the main difference from our previous work [28]. The data representation here is based upon a medium resolution Cryo-EM map rather than on a detailed atomic representation. Given a Cryo-EM map which contains electron density data, the data is encoded into a compact representation using the EMAN software package [31], developed to process Cryo-EM maps. A multi-level representation of the data allows us to conveniently manipulate different parts of the structure at will. The different levels of representation include:

1. Pseudo atoms [29], which are feature points of increased density, where most likely atoms are found. Notice that due to the resolution pseudo atoms do not correspond to exact atom locations.

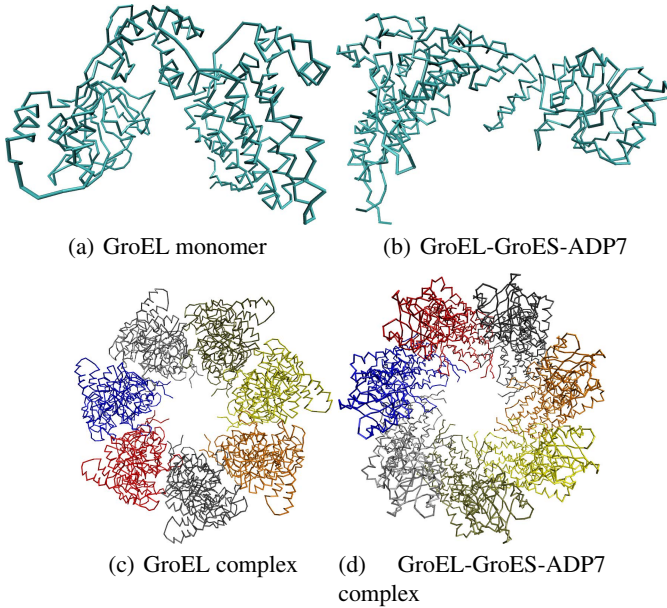


Fig. 1. Cryo-EM maps generated for GroEL monomer using EMAN [31]. (a) The GroEL monomer (Created from cryo-EM model). (b) The GroEL-GroES-ADP7 monomer taken from PDB structure 1SX4, chain A. (c) The GroEL complex (built as a symmetric complex of the monomer modeled in (a)). (d) The GroEL-GroES-ADP7 complex - taken from PDB structure 1SX4.

2. A skeleton [32], which captures the topology of the structure and helps detecting helix and sheet regions and their connectivity.
3. Secondary structures, which are assigned using either prior knowledge or the SSE-Hunter and SSEBuilder tools [29]. These tools assign secondary structure elements to a given Cryo-EM map using local topology information obtained from the pseudo atoms and connectivity information obtained from the skeleton.

Based on this data, the location of secondary structure elements and often even partial or full backbone and side chain information can be obtained with high degree of accuracy [33]. The high-level data structure that represents a conformation is a graph $G = (V, E)$ such that each secondary structure element is a node $v \in V$ in the graph. Two secondary structure elements v_1 and v_2 are connected by an edge $e \in E$ if there is at least one pair of adjacent amino acids r_1, r_2 , such that $r_1 \in v_1$ and $r_2 \in v_2$. The backbone angles in r_1, r_2 , and a small number of sequentially adjacent residues form the degrees of freedom of the protein. In other words, the protein motions consist of bond rotations in these residues while the remaining angles stay fixed.

Based on the graph we construct a spanning tree $T = (V, X)$ where X is a subset of E using a greedy approach. The root of the tree is specified as the structure that is expected to move the least during the search as determined by aligning the start and goal structures and measuring the least RMSD between corresponding secondary structure

elements. Each one of the root's neighbors forms a child node in the tree, and at each stage the selected node and its adjacent edges are removed from the graph. The process repeats iteratively until all the secondary structure elements are represented in the tree. There may be more than one correct topology to the structure. We picked the topology that follows the order of amino acids, which seems to give the best results. It should be noted that it is not always possible when no information about the exact location of the amino acids is available when the resolution is too low. In some cases we may know that the positions of certain secondary structure elements is likely to stay fixed. This allows us to speed up the search for a feasible pathway by restricting motions to the remaining secondary structure elements. Let $K \subseteq V$ be the set of secondary structures that is free to move. This set is used below in the definition of a distance metric for our representation.

It should be emphasized that this representation can be used even if there is no detailed structural information such as backbone location or even if only partial information about secondary structures breaking down, since the conformation representation assumes only secondary structure knowledge.

2.2 Distance between Structures

The search method requires a distance measure to estimate the progress in the conformational search. In the case of proteins and protein complexes the distance measure is not trivial to define due to the complexity of protein structures and the high dimensionality of the search space. Finding a good distance measure between two molecules is an active area of research [34]. This issue becomes especially challenging when the proteins are represented at a coarse resolution and traditional distance measures such as RMSD may not be accurate due to the approximate location of the α carbon representation that may cause inaccuracies to accumulate. In order to measure the distance between structures we use a method we developed previously and gave good result in a previous work [28]. Since it only requires knowledge of the location of secondary structure elements it is especially suitable for coarse grained molecular representations.

The distance measure is defined in terms of the relative positions between secondary structure elements. We compute for a conformation C a feature vector:

$$v_C = \langle \text{score}(C^1), \text{score}(C^2), \dots, \text{score}(C^k) \rangle \quad (1)$$

where the components of the vector are scores calculated for the K secondary structure elements of the conformation, based on their positioning with respect to one another. The distance between two conformations, C_1 and C_2 is defined as the Euclidean distance between their feature vectors, i.e., $\|v_{C_1} - v_{C_2}\|^2$. By definition, when C_2 is the goal structure, the *score* of C_1 is the magnitude of its vector representation. Therefore, the lower the score for a given conformation, the more similar it is to the goal structure.

2.3 Penalty Function

There are several potential functions that are suitable for C- α representation [8]. These potential functions take into account the hydrophobicity of the amino acids, their

interactions with the solvent and with one another. While coarse-grained energy models are an approximation of the protein potential energy and are often biased towards a folded state, they give good results given their simplicity. They allow for an efficient exploration of the protein conformational space where more detailed structural representations, which require a more accurate potential function, consume a vast amount of resources. We use the energy function developed by Brown and Head-Gordon [35]. This function has been shown to give good results while using only one bead per amino acid. The energy function classifies the 20 amino acids into three categories - Hydrophobic (H), Polar (P) and Neutral (N). The potential energy is given by the following equation:

$$E_{\text{total}} = \sum_{\text{angles}} K_{\theta}(\theta - \theta_{eq})^2 + \sum_{\text{dihedrals}} [A(1 + \cos\phi) + B(1 - \cos\phi) + C(1 + \cos 3\phi) + D(1 + \cos[\phi + \frac{\pi}{4}])] + \sum_{i,j \geq i+3} 4\epsilon_H S_1 [\frac{\sigma}{r_{ij}^{12}} - \frac{\sigma}{r_{ij}^6}] \quad (2)$$

Where $r_{i,j}$ is the distance between amino acids and the constants A, B, C, and D, S_1 and S_2 are determined by the secondary structure type and the amino acid type. See [35] for more details.

When no backbone trace is available, none of the above coarse grained potential functions can be used. Our structural representation and search algorithm are suitable for sub-nanometer resolution and do not assume knowledge about the locations of specific amino acids. However, there are only a few papers regarding coarser than backbone representation [36]. The use of these very coarse models in the context of our model, where no normal mode analysis is performed, should be tested carefully, since they may admit non-biological motions as well. However, it can be used as a filtering tool that greatly reduces the number of possible pathways. In cases where a more detailed structural model exists, a more realistic energy function can be used for filtering and refinement. This area is the subject of on-going and future research.

2.4 Search Methodology

The search is performed using a sampling-based motion planning algorithm. Motion planning algorithms have been applied extensively in the past to solve biological problems due to the analogy between protein chains and robotic articulated mechanisms [22, 23]. The search methodology applied in this paper is based on the Path-Directed Subdivision Tree (PDST) planner [37]. We chose this algorithm because of its good performance with articulated systems with complex dynamics moving in physically constrained environments. It has shown good results in our previous work [28]. We adapted the algorithm to model protein motions. In our adaptation, the planner iteratively constructs a tree of conformational pathways as the search progresses. The input to the algorithm consists of the start and end conformations of a molecule, represented as sets of articulated secondary structures as discussed in Data Representation above. The root of the search tree is a “pathway” of length 0 consisting only of the starting structure. At every iteration a previously generated pathway is selected for propagation using a deterministic scoring scheme described below. From a random conformation

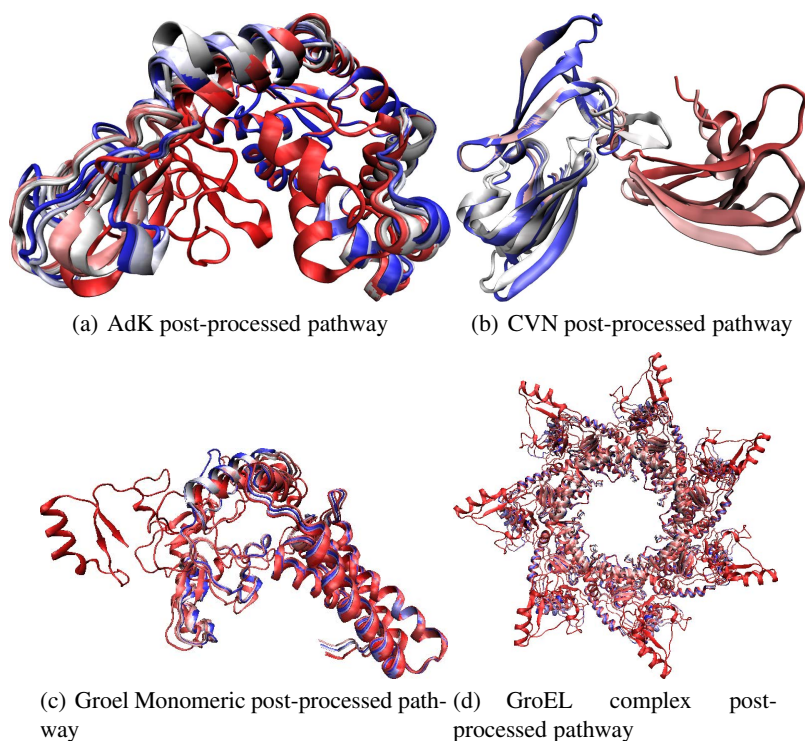


Fig. 2. Illustration of the results for AdK, CVN and GroEL: (a) An example of a conformational pathway obtained for AdK after resolution enhancement and basic energy minimization. (b) An example of a conformational pathway for CVN. (c-d) An example of a conformational path of GroEL. The monomeric path is shown in (c) for clarity, and the entire complex is illustrated in (d). The conformation colors are interpolated on the red (start) to blue (goal) scale.

along that pathway, a new pathway is propagated by applying a small random rotation to the ϕ or ψ backbone dihedral angle of a residue that resides on a loop connecting two randomly chosen secondary structure elements. A molecular motion is sampled by applying the rotation until a high energy conformation is reached. The coarse grained energy function described above is used to determine when a high energy conformation is encountered. A high energy conformation is defined as being more than 100 kcal/mol above the starting energy. The relatively high threshold aims to compensate for inaccuracies in the structure due to the low resolution. The algorithm maintains a subdivision of the low-dimensional projection of the conformational space (described in Distance Between Structures above) into cells, such that no sample spans more than one cell in the subdivision. The goal of the subdivision is to guarantee coverage of the search space [37]. After a sample is selected for propagation, the cell containing that sample is subdivided into two cells. The algorithm keeps track of how many samples are contained in each cell to estimate how dense the sampling is in different areas of the space. It maintains a scoring scheme that gives selection preference to samples residing

in large, empty cells, thus pushing the exploration towards unvisited areas in the conformational space. Probabilistic completeness is obtained via a scoring scheme that favors the selection of samples contained in larger cells and leads to unexplored areas of the search space. The sample scores are updated in a way that guarantees that every sample in the tree will eventually be selected for propagation and avoids over-sampling of parts of the space. Previous studies in path-directed motion planning algorithms [28, 38] showed that employing a biasing scheme in a small percentage of the iterations greatly improves the performance of the planner. We employed biasing at 10% of the iterations. During these iterations the scoring scheme described above is ignored and a sample is chosen out of a pool of conformations closest to the goal conformation, which gives the planner a better chance to successfully terminate the search. We found that the biasing improves the performance of the algorithm. Our top-level algorithm runs PDST iteratively. Each iteration runs until a generated conformation is closer to the goal conformation than a pre-specified intermediate distance threshold, where the distance threshold is determined by the distance measure described above. We found that a threshold of 0.8–0.9 of the distance between the start and goal conformations is usually sufficient to achieve good results and running further does not benefit the results much. The iterative runs of the PDST planner help reduce memory use and improve performance, as also shown in [39].

3 Results and Discussion

We ran the algorithm on three test cases, Adenylate Kinase (AdK), cyanovirin (CVN) and the GroEL single-ring complex. In this study we seek to provide a proof of concept and some real test-cases. AdK and CVN undergo extensive conformational transitions, they are well studied and have an abundance of data for testing and comparison. To provide test for real backbone-resolution data we tested the algorithm on the GroEL 7-member ring complex. A CryoEM model is available only for the closed form, and the open form was taken from the PDB, using accession code 1SX4. The PDB structures were resampled as C- α traces. Figure 2 shows an illustration of the AdK, CVN and GroEL examples.

We ran the algorithm 100 times per protein on the UMass Boston Supercomputing Cluster, where each machine runs at 2.2 Ghz and has 4 GB RAM. For comparison purposes, we produced conformational pathways using a random walk Monte Carlo like algorithm [7] with the same resolution, representation and penalty function described in this work. The random walk algorithm is similar to the one used for comparison in our previous work [28]. Using the same representation, similarity score and potential function described in our algorithm, the random walk algorithm differs from the common use of Monte Carlo in protein conformational search. Rather than optimizing the energy, it optimizes the above mentioned similarity score in order to simulate a conformational pathway from the start to the goal conformation. The energy, while not optimized, is used to filter out non-feasible conformations. The random walk implementation uses the Metropolis criterion for the selection of steps. At each iteration a random conformational pathway is generated from the current conformation by applying a small random transformation to one of the randomly chosen degrees of freedom

connecting secondary structure elements, in a similar way to the one used to generate new conformations described in the Search Methodology subsection above. If a step brings the similarity score of the generated conformation closer to the goal it will be accepted. Otherwise it is accepted with a probability proportional to $e^{\Delta S}$ where ΔS is the difference in the similarity score of the current step and the previous step. In practice, this criterion accepts all “good” steps while allowing a very small fraction of “bad” steps.

No comparison was made to any other conformational search method since random walk is very easy to implement, but no other method exists that models conformational changes in low resolution level and therefore such a comparison would be meaningless. In order to compare the performance of the two methods by an objective standard, each was run for a fixed amount of time and the least RMSD (IRMSD) of the closest conformation to the goal at that given time step was measured. Generally, IRMSD is not available for medium resolution structures. However, it was used for the sake of this initial baseline validation due to the fact that the full resolution structures are available. IRMSD was measured after 10 and 20 minutes for AdK. In the case of GroEL, which is a longer running example, measurements were taken after 15 and 30 minutes. For CVN, which was a shorter example, measurements were taken after 5 and 10 minutes. Table 3 summarizes the average IRMSD results over 80 test runs, where the top and bottom 10% outliers were removed.

Adenylate Kinase (AdK). AdK is a monomeric phosphotransferase enzyme that catalyzes reversible transfer of a phosphoryl group from ATP to AMP. AdK contains 214 amino acids and assumes an “open” conformation in the unligated structure and a bound, “closed” conformation. The IRMSD between the two structures is 6.95Å. Supposedly, during the transition from the “open” to “closed” form, the largest conformational change occurs in the LID (residues 118–167) and NMP (residues 30–67) domains with the rest of the protein – the CORE domain being relatively rigid. We modeled the closed state to open state motion using the C- α traces of PDB codes 1AKE and 4AKE for the closed and open states respectively. Our model contains 7 secondary structure elements where most of the CORE domain was modeled as one large segment and was considered fixed, since it does not undergo a large-scale motion. Figure 2(a) shows the overlapped resulting structure with the goal structure. The final set of transformations was applied to the backbone traces (which are known in this case) to generate the figure. The C- α RMSD from the goal structure is 1.622Å. As seen in table 3 the resulting average IRMSD was 1.93Å in the end of the runs. Random walk performed slightly worse compared to our planner with an average IRMSD of 2.26Å.

Cyanovirin-N (CVN). CVN is an anti-viral fusion inhibitor protein that binds to viral sugars, and is trialed for preventing sexual transmission of HIV. It comprises two repeat domains of 30% sequence identity which undergo swapping [40]. We simulated the unpacking of the repeat domains of a single chain from the intertwined monomeric conformation to an extended domain-swapped conformation. The swapped conformations deviate by approximately 16Å. CVN contains 101 amino acids and our model contains 6 rigid elements. The flexible rotation axis resides mainly between residues 48–55. The distance measure threshold for successful termination of the algorithm was

Performance statistics for the AdK, CVN and GroEL complex examples. The average \pm (standard deviation) IRMSD data were taken over 80 runs where the top and bottom 10% outliers were removed from the original set of 100 runs.

	AdK	AdK RW [†]	CVN	CVN RW [†]	GroEL	GroEL RW [†]
Initial IRMSD (Å)	6.95	6.95	16.01	16.01	14.64	14.64
#Residues	214	214	101	101	525 \ddagger	525 \ddagger
IRMSD at first measurement (Å)	2.49 \pm 0.41	2.621 \pm 0.54	4.85 \pm 1.55	4.46 \pm 1.84	7.06 \pm 0.38	7.07 \pm 0.44
IRMSD at second measurement (Å)	2.36 \pm 0.38	2.54 \pm 0.53	3.57 \pm 0.95	4.42 \pm 1.67	6.62 \pm 0.43	6.95 \pm 0.37
Final IRMSD (Å)	1.93 \pm 0.16	2.32 \pm 0.49	2.29 \pm 0.25	3.20 \pm 1.2	6.26 \pm 0.43	6.83 \pm 0.34

[†] Random walk. See Results section for details.

a normalized distance of 0.91 from the goal conformation. Figure 2(b) shows an example of a pathway from the start to the end conformation. The C_{α} RMSD from the goal structure is 1.67Å. As seen in Table 3, our algorithm significantly outperformed random walk with an average IRMSD of about 2.29Å comparing to 3.2Å for random walk. Many of our runs got as low as 1.5Å from the final conformation. The average run time was approximately 29 minutes.

GroEL Complex. The GroEL protein belongs to the chaperonin family and is found in a large number of bacteria [41]. It is required for the correct folding of many proteins. GroEL requires the lid-like cochaperonin protein complex GroES. Binding of substrate protein, in addition to binding of ATP, induces an extensive conformational change that allows association of the binary complex with GroES. We modeled the epical domain movement from the GroEL-GroES-ADP7 complex (modeled from chain A of PDB code 1SX4) to the GroEL monomer (modeled using a C_{α} trace extracted from Cryo-EM data, analogous to the closed GroEL monomer) The monomer contains 525 amino acids, and our model contains 8 secondary structure elements where most of the equatorial domain, whose structure does not change significantly, was modeled as one large segment and was considered fixed. The initial IRMSD between the C_{α} atoms of the two complexes is 14Å. The IRMSD from the goal structure was measured with respect to the C_{α} trace of the Cryo-EM model. Figure 2(c-d) shows an example of a pathway from the start to the end conformation. Table 3 shows that our method significantly outperforms random walk. The average IRMSD between the resulting structures and the goal structure was 6.2Å after 30 minutes compared to approximately 6.8Å for MC. The pathways obviously indicate a closing motion of the complex, but IRMSD was used here mainly for compatibility with the other examples. It is not a very good indicator of the quality of the pathway in this case, since the Cryo-EM model contains inaccuracies that makes it somewhat different than the atomic structure of GroEL which was used to model the open structure. In a previous work [28] we used PDB code 1SS8, which is close in structure to the Cryo-EM model used in this work. We then observed an IRMSD of less than 5Å, with some runs reaching as close as 2.5Å from the goal structure. When comparison between the two structures is made less accurate, other distance measurements should probably be considered [34].

Analysis of the Results. In order to provide initial validation for our results, we tested whether our algorithm produces biologically reasonable, low energy pathways when using an all-atom force field. Such an analysis was done in earlier works [27, 28],

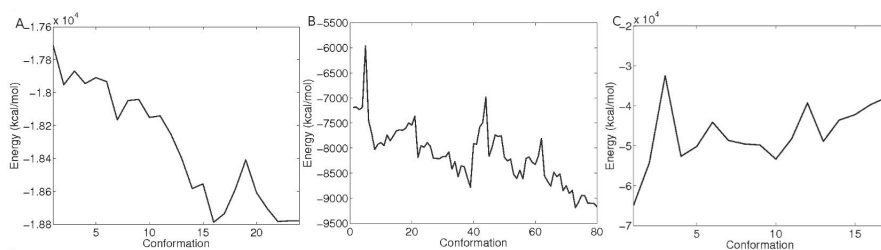


Fig. 3. Energetic profiles of the resulting pathways (a) A potential energy plot for ADK, CVN and GroEL slightly minimized conformational pathways. Notice the different potential energy scale.

where the authors used a similar method to show that their conformational search was reasonable. For this purpose we chose targets for which high resolution structures are available. Side chain information was completed for the resulting pathways using the algorithm described in [42]. The resulting full-atomic structures were minimized for 500 Steepest Descent steps using the AMBER energy minimization package [6]. The minimization was done for a relatively small number of steps and was restrained in order to resolve initial clashes but not cause large conformational changes to the structures. The purpose of this test is not to provide a fully minimized pathway, but to show that the algorithm produces pathways with reasonable conformations whose clashes can be resolved within a small number of minimization steps. Figure 3 shows the potential energy plot of a sampled conformational pathway for the three test cases. As seen, even with a small number of energy minimization steps all the intermediate structures exhibit low potential energies, below -20000 kcal/mol for AdK, below -7000 kcal/mol for CVN and around -40000 kcal/mol for GroEL, as measured by AMBER.

4 Conclusions

We present a novel method for exploring large scale conformational changes in proteins represented at medium resolution using Cryo-EM data and image processing techniques. The search methodology is based on robot motion planning, and it strikes a balance between an efficient coverage of the conformational space and fast exploration towards the goal structure. The input molecule is modeled using a coarse representation and a relatively simple potential function to guide the search. This representation does not require atomic details and thus makes the computation tractable and especially useful in cases where a detailed structural model is not available and facilitates dealing with partial or missing data. We tested our algorithm on the following well studied proteins: Adenylate Kinase and Cyanovirin. Additionally, we ran the algorithm on actual backbone-resolution models obtained from cryo-EM data, the GroEL complex (where one state was obtained from cryo-EM and the other state from the PDB). We show that our method performs significantly better than random walk by producing low energy pathways with resulting structures closer to the goal structure. We believe this is an important step towards a larger scale modeling of more complex biological systems such

as virus capsid shells. Additionally, since there is an abundance of cryo-EM models whose resolution do not allow backbone trace modeling, this method can be the basis to simulating conformational changes in even lower than backbone resolution models. The algorithmic framework is similar, and the main difference is selecting a potential function that handles lower-than-backbone resolution. This is an area of on-going work.

Acknowledgements. I thank Dr. Steve Ludtke for providing part of the data and the training on Cryo-EM processing software. I also thank Dr. Tao Ju for useful ideas and Dr. Erion Plaku and Ioan Şucan for providing code which was incorporated in this work. Special thanks to Dr. Lydia Kavrakı, Dr. Mark Moll, Dr. Matt Baker and Dr. Wah Chiu for their collaboration and valuable contribution to this work. Computational experiments were conducted on the supercomputing facilities in the College of Science and Mathematics at UMass Boston.

References

1. Perutz, M.F.: Mechanisms of cooperativity and allosteric regulation in proteins. *Quart. Rev. Biophys.* 22, 139–236 (1989)
2. Schmid, M.F., Sherman, M.B., Matsudaira, P., Chiu, W.: Structure of the acrosomal bundle. *Nature* 431, 104–107 (2004)
3. Jiang, W., Li, Z., Zhang, Z., Baker, M., Prevelige Jr., P.E., Chiu, W.: Coat protein fold and maturation transition of bacteriophage P22 seen at subnanometer resolutions. *Nature Structural Biology* 10(2), 131–135 (2003)
4. Schroeder, G., Brunger, A.T., Levitt, M.: Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution. *Structure* 15, 1630–1641 (2007)
5. Lasker, K., Dror, O., Shatsky, M., Nussinov, R., Wolfson, H.J.: EMatch: discovery of high resolution structural homologues of protein domains in intermediate resolution cryo-EM maps. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 4(1), 28–39 (2007)
6. Case, D.A., Cheatham, T., Darden, T., Gohlke, H., Luo, R., Merz Jr., K.M., Onufriev, A., Simmerling, C., Wang, B., Woods, R.: The Amber biomolecular simulation programs. *J. Computat. Chem.* 26, 1668–1688 (2005)
7. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* 220, 671–680 (1983)
8. Head-Gordon, T., Brown, S.: Minimalist models for protein folding and design. *Curr. Opin. Struct. Biol.* 13(2), 160–167 (2003)
9. Whitford, P.C., Miyashita, O., Levy, Y., Onucic, J.N.: Conformational transitions of adenylylate kinase: Switching by cracking. *Journal of Molecular Biology* 366(5), 1661–1671 (2007)
10. Schuyler, A., Jernigan, R., Qasba, P., Ramakrishnan, B., Chirikjian, G.: Iterative cluster-nma: A tool for generating conformational transitions in proteins. *Proteins* 74, 760–776 (2009)
11. Zheng, W., Brooks, B.: Identification of dynamical correlations within the myosin motor domain by the normal mode analysis of an elastic network model. *J. Mol. Biol.* 346(3), 745–759 (2005)
12. Temiz, N., Meirovitch, E., Bahar, I.: Escherichia coli adenylylate kinase dynamics: comparison of elastic network model modes with mode-coupling (15)n-nmr relaxation data. *Proteins* 57, 468–480 (2004)
13. Gohlke, H., Thorpe, M.: A natural coarse graining for simulating large biomolecular motion. *Biophysical Journal* 9, 2115–2120 (2006)

14. Weiss, D., Levitt, M.: Can morphing methods predict intermediate structures? *J. Mol. Biol.* 385, 665–674 (2009)
15. Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E., Thrun, S.: *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press (2005)
16. Kavraki, L.E., Švestka, P., Latombe, J.-C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* 12, 566–580 (1996)
17. Shehu, A., Kavraki, L., Clementi, C.: On the characterization of protein native state ensembles. *Biophysical Journal* 92(5), 1503–1511 (2007)
18. Shehu, A., Kavraki, L., Clementi, C.: Multiscale characterization of protein conformational ensembles. *Proteins: Structure, Function and Bioinformatics* 76(4), 837–851 (2009)
19. Haspel, N., Geisbrech, B., Lambris, J., Kavraki, L.E.: Multi-scale characterization of the energy landscape of proteins with application to the c3d/efb-c complex. *Proteins: Structure, Function and Bioinformatics* 78(4), 1004–1014 (2010)
20. Cortés, J., Siméon, T., de Angulo, V.R., Guieysse, D., Remauld-Siméon, M., Tran, V.: A path planning approach for computing large-amplitude motions of flexible molecules. *Bioinformatics* 21(suppl. 1), i116–i125 (2005)
21. Shehu, A., Clementi, C., Kavraki, L.E.: Sampling conformation space to model equilibrium fluctuations in proteins. *Algorithmica* 48, 303–327 (2007)
22. Thomas, S., Tang, X., Tapia, L., Amato, N.M.: Simulating protein motions with rigidity analysis. *J. Comp. Biol.* 14(6), 839–855 (2007)
23. Chiang, T.H., Apaydin, M.S., Brutlag, D.L., Hsu, D., Latombe, J.-C.: Using stochastic roadmap simulation to predict experimental quantities in protein folding kinetics. *J. Comp. Biol.* 14(5), 578–593 (2007)
24. Yao, P., Dhanik, A., Marz, N., Propper, R., Kou, C., Liu, G., van den Bedem, H., Latombe, J.-C., Halperin-Landsberg, I., Altman, R.B.: Efficient algorithms to explore conformation spaces of flexible protein loops. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 5(4), 534–545 (2008)
25. Tang, X., Thomas, S., Tapia, L., Amato, N.M.: Tools for simulating and analyzing rna folding kinetics. In: *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, San Francisco, CA, USA, pp. 268–282 (April 2007)
26. Shehu, A., Clementi, C., Kavraki, L.E.: Modeling protein conformational ensembles: From missing loops to equilibrium fluctuations. *Proteins: Structure, Function and Bioinformatics* 65, 164–179 (2006)
27. Raveh, B., Enosh, A., Furman-Schueler, O., Halperin, D.: Rapid sampling of molecular motions with prior information constraints. *Plos Comp. Biol.* (2009) (in press)
28. Haspel, N., Moll, M., Baker, M., Chiu, W., Kavraki, L.E.: Tracing conformational changes in proteins. *BMC Structural Biology* (2010) (in press)
29. Baker, M.L., Ju, T., Chiu, W.: Identification of secondary structure elements in intermediate resolution density maps. *Structure* 15, 7–19 (2007)
30. Abeysinghe, S.S., Ju, T., Baker, M., Chiu, W.: Shape modeling and matching in identifying protein structure from low-resolution images. In: *ACM Symposium on Solid and Physical Modeling*, pp. 223–232 (2007)
31. Ludtke, S.J., Baldwin, P.R., Chiu, W.: EMAN: semiautomated software for high-resolution single-particle reconstructions. *J. Struct. Biol.* 128, 82–97 (1999)
32. Ju, T., Baker, M.L., Chiu, W.: Computing a family of skeletons of volumetric models for shape description. *Computer Aided Design* 39(5), 352–360 (2007)
33. Zhang, J., Baker, M., Schroeder, G., Douglas, N., Reissman, S., Jakana, J., Dougherty, M., Fu, C., Levitt, M., Ludtke, S., Frydman, J., Chiu, W.: Mechanism of folding chamber closure in a group ii chaperonin. *Nature* 463, 379–383 (2010)

34. Ballester, P.J., Richards, W.G.: Ultrafast shape recognition to search compound databases for similar molecular shapes. *J. Comput. Chem.* 28(10), 1711–1723 (2007)
35. Brown, S., Fawzi, N., Head-Gordon, T.: Coarse grained sequences for protein folding and design. *Proc. Nat. Acad. USA* 100, 10 712–10 717 (2003)
36. Doruker, P., Jernigan, R., Bahar, I.: Dynamics of large proteins through hierarchical levels of coarse-grained structures. *J. Comput. Chem.* 23(1), 119–127 (2002)
37. Ladd, A.M.: Motion planning for physical simulation. Ph.D. dissertation, Dept. of Computer Science, Rice University, Houston, TX (Dec. 2006)
38. Şucan, I.A., Kruse, J.F., Yim, M., Kavraki, L.E.: Reconfiguration for modular robots using kinodynamic motion planning. In: *ASME Dynamic Systems and Control Conference*. Michigan, Ann Arbor (2008)
39. Tsianos, K., Kavraki, L.E.: Replanning: A powerful planning strategy for hard kinodynamic problems. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 1667–1672 (September 2008)
40. Botos, I., O’Keefe, B., Shenoy, S., Cartner, L., Ratner, D., et al.: Structures of the complexes of a potent anti-hiv protein cyanovirin-n and high mannose oligosaccharides. *J. Biol. Chem.* 277, 34336–34342 (2002)
41. Zeilstra-Ryalls, J., Fayet, O., Georgopoulos, C.: The universally conserved GroE (Hsp60) chaperonins. *Annu. Rev. Microbiol.* 45, 301–325 (1991)
42. Heath, A.P., Kavraki, L.E., Clementi, C.: From coarse-grain to all-atom: Toward multi-scale analysis of protein landscapes. *Proteins: Structure, Function and Bioinformatics* 68(3), 646–661 (2007)

An Event Graph Model for Discovering Trends from Text Streams

Chengli Zhao, Xue Zhang, and Dongyun Yi

Department of Mathematic and Systems Science, College of Science,
National University of Defense Technology, 410073 Changsha, Hunan, P.R. China
{chenglizhao, xuezhang, dongyunyi}@nudt.edu.cn

Abstract. In this paper, we formally define and study the event graph model based on set theory and multi-relations theory, and discuss the methods of modeling event and event relations in detail. The event graph model is mainly designed to extract the potential events and the relationships between events from massive text streams, and further discover the trends embodied in the contents in text streams. We also study the connectivity of the event graph model, and give the equivalent conditions to determine the connectivity of event graph.

Keywords: event graph, trend, text stream.

1 Introduction

With the rapid development of network and information technology, people always encounter a lot of text stream data, such as instant messaging chats, e-mail and online news and more. In such stream text data, there often exist interesting events and graph structures constructed by events and their relations. How to mine the useful information from these massive text streams has become an important problem.

In this paper, based on set theory and multi-relations theory, we formally define and study the event graph model to discover, extract and summarize the potential events and event graph structures from massive texts, and discuss the method how to model the node and edge of the event graph with the examples of probabilistic mixture model and relative entropy respectively. We also study the connectivity of the event graph model, and give the equivalent conditions to determine the connectivity of event graph. Connectivity of the graph shows us the way to find the trends in text streams.

The rest of the paper is organized as follows. In Section 2, we discuss the related work in detail. In Section 3, we formally introduce the event graph model. In Section 4, we discuss the connectivity of the event graph and give some proof. We draw the conclusions to summary the paper in Section 5. Finally, Section 6 is acknowledgment.

2 Related Work

General Systems is a common approach that abstracts and considers a system as a set of independent and interacting parts, and provide the general theory and methods of

study the systems. In the domain of general systems, Lin studies the concept of general systems mathematically. In his book [1], he defines a system as a set of objects and a set of relations between the objects. Also he introduces a model of general systems based on the methods of set theory, and studies some basic global properties of systems in detail, including layer structures, centralized systems, relations between systems, and a characterization of centralizable systems. The theory and method described in the book presents a general framework to study different kinds of systems.

In text mining, Zhai proposes a generative probabilistic mixture model [2] which models the semantic content in texts in order to perform cross-collection clustering. This model can be estimated using the Expectation Maximization algorithm, so it is a very useful tool in text mining. Based on the above probabilistic mixture model, Mei studies the evolutionary patterns of themes in text streams and explores the methods how to discover and summarize them, and an evolution graph of themes on the Asian tsunami disaster is constructed in his paper [3]. Because there usually exists an evolution relationship between two themes or topics with time stamps, it is necessary to find the evolution graph structure.

Lots of work to predict trends in texts has been presented in the past few years. Based on WWW, Gloor extracts and predicts long-term trends on the popularity of relevant concepts such as brands, movies, and politicians by calculation of betweenness of these concepts. He introduces a novel set of social network analysis based algorithms [4] for mining the Web, blogs, and online forums to identify trends and also find who launch these them. The Algorithms he used include the temporal computation of network centrality measures, mining and analyzing large amounts of text based on social network analysis, and sentiment analysis and information filtering methods.

3 Event Graph Model

Given a text stream, the objective of event graph model is to discover the trends from it through extract the event graph in the text stream. This graph model will give us a specific insight to the massive text stream. In order to explain the event graph model, we firstly define the following concepts.

Definition 1 (Event). An event $e = (\theta, t)$ is an ordered pair of sets $e = (\theta, t)$, such that θ is a semantically topic which normally is represented as a probabilistic distribution of words, and t the time when the event e happens, defined by a time function $t = t(e)$.

Definition 2 (Event Graph). An event graph is an ordered pair of sets $G = (E, R)$, such that E is the set of all vertices (events) of G , and R a set of edges (relations) defined on E . The sets E and R are respectively called the vertex (event) set and the edge (relation) set of the event graph G .

The event graph $G = (E, R)$ is trivial if $E = \emptyset$. It is easy to see that only nontrivial event graph is meaningful to study in practical application.

3.1 Event Modeling

Let $e_i = (\theta_i, t_i) (i = 1, 2, \dots, n)$ be n events in a text stream, and $C = \{d_1, d_2, \dots, d_T\}$ a text stream, where d_j refers to a text with time stamp $j (j = 1, 2, \dots, T)$. Probabilistic mixture model [2] can be introduced to describe and extract events in text stream C . In this model, words are regarded as data drawn from a mixture model with component models for the topic word distributions and a background word distribution. Words in the same text share the same mixing weights. The model can be estimated using the Expectation Maximization (EM) algorithm to obtain the topic word distributions.

Each text could be seen as a sequence of words from a vocabulary set $V = \{w_1, w_2, \dots, w_{|V|}\}$. The topic θ in the event $e = (\theta, t)$ can be defined by a unigram language model, such as a word distribution $\{p(w|\theta)\}_{w \in V}$. Naturally we have $\sum_{w \in V} p(w|\theta) = 1$.

Let $\theta_1, \theta_2, \dots, \theta_k$ be k unigram language models and θ_B be a background model for the whole text stream C . A text d is regarded as a sample of the following mixture model:

$$p(w:d) = \lambda_B p(w|\theta_B) + (1 - \lambda_B) \sum_{i=1}^k [\pi_{d,i} p(w|\theta_i)] \tag{1}$$

Where w is a word in text d , $\pi_{d,i}$ is the mixing weight for text d for choosing the event θ_i such that $\sum_{i=1}^k \pi_{d,i} = 1$, and λ_B is the mixing weight for θ_B . The purpose of using a background model θ_B is to make the event models more discriminative. More details to estimate parameters in (1) can be found in [3]

3.2 Event Relation Modeling

Definition 3 (Event Evolution). Let $e_1 = (\theta_1, t_1)$ and $e_2 = (\theta_2, t_2)$ be two events. If $t_1 \leq t_2$ and the similarity between events e_1 and e_2 is above a give threshold, we say that there is an evolutionary transition from e_1 to e_2 , which we denote by $e_1 \rightarrow e_2$. We also say that θ_2 is evolved from θ_1 , or θ_1 evolves to θ_2 .

Given two probability mass functions $p(x)$ and $q(x)$, the Kullback-Leibler divergence (or relative entropy) between p and q is defined as

$$D(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{2}$$

$D(p \parallel q)$ is always non-negative and is zero if and only if $p = q$. Because it is not symmetric and does not satisfy the triangle inequality, it is not a true distance between two distributions, but it is still useful to apply the KL-divergence to measure the distance between any two distributions [5].

Based on the property of KL-divergence, we use the KL-divergence to model the relation between any two events and measure their similarity. Let $e_1 = (\theta_1, t_1)$ and $e_2 = (\theta_2, t_2)$ be two events where $t_1 \leq t_2$. We assume that e_2 has a smaller evolution distance to e_1 if their unigram language models θ_2 and θ_1 are closer to each other. Since the KL-divergence $D(\theta_2 \parallel \theta_1)$ can model the additional new information in θ_2 as compared to θ_1 , it appears to be a natural measure of evolution distance between two events.

$$D(\theta_2 \parallel \theta_1) = \sum_{i=1}^{|V|} p(w_i \mid \theta_2) \log \frac{p(w_i \mid \theta_2)}{p(w_i \mid \theta_1)} \tag{3}$$

Formula (3) quantifies the relation between any two events in text streams.

4 Property of Model

In this part we will discuss the connectivity of event graph and give a complete proof in mathematics based on set theory and multi-relations theory [1]. Connectivity of the graph shows us the way to find the trends in text streams.

Let $G = (E, R)$ be an event graph and $r \in R$ a relation. The support of r , denoted $\text{Supp}(r)$, is defined by

$$\text{Supp}(r) = \{e \in E : e \text{ belongs to the event set on which } r \text{ is defined}\} \tag{4}$$

The event graph $G = (E, R)$ is connected if it cannot be represented in the form

$$G = G_1 \oplus G_2 \triangleq (E_1 \cup E_2, R_1 \cup R_2) \tag{5}$$

Where $G_i = (E_i, R_i) (i = 1, 2)$ such that $E_i \subset E$ and $R_i \subset R$ and $E_1 \cap E_2 = \emptyset$.

Theorem 1. An event graph G is connected if and only if for any two events $x, y \in E$, there exists n relations $r_i \in R$ such that

$$x \in \text{Supp}(r_1) \text{ and } y \in \text{Supp}(r_n) \text{ and } \text{Supp}(r_i) \cap \text{Supp}(r_{i+1}) \neq \emptyset, i = 1, 2, \dots, n-1 \tag{6}$$

Proof: Necessity. We prove by contradiction. Suppose that the event graph G is connected and there exist two events $x, y \in E$ such that there do not exist n relations $r_i \in R, i = 1, 2, \dots, n$, for any natural number $n \geq 1$, such that

$$x \in \text{Supp}(r_1) \text{ and } y \in \text{Supp}(r_n) \text{ and } \text{Supp}(r_i) \cap \text{Supp}(r_{i+1}) \neq \emptyset, i = 1, 2, \dots, n-1$$

From the hypothesis that G is connected, it follows that there must be relations $r_1, s_1 \in R$ such that $x \in \text{Supp}(r_1)$ and $y \in \text{Supp}(s_1)$. Then our hypothesis implies that

$$\text{Supp}(r_1) \cap \text{Supp}(s_1) = \emptyset \quad (7)$$

Let $U_0 = \text{Supp}(r_1)$ and $V_0 = \text{Supp}(s_1)$, and for each natural number $n \in \mathbb{N}$ let

$$U_n = \bigcup \{ \text{Supp}(r) : r \in R \text{ and } \text{Supp}(r) \cap U_{n-1} \neq \emptyset \} \quad (8)$$

$$V_n = \bigcup \{ \text{Supp}(s) : s \in R \text{ and } \text{Supp}(s) \cap V_{n-1} \neq \emptyset \} \quad (9)$$

Then $U_0 \subseteq U_1 \subseteq \dots \subseteq U_n \subseteq \dots, V_0 \subseteq V_1 \subseteq \dots \subseteq V_n \subseteq \dots$, and $U_n \cap V_m = \emptyset$ hold for all natural numbers $n, m \in \mathbb{N}$.

We now define two subsystems of $G : G_i = (E_i, R_i), (i=1, 2)$ such that

$$E_1 = \bigcup_{n=0}^{\infty} U_n \text{ and } E_2 = E - E_1 \quad (10)$$

$$R_1 = \{ r \in R : \text{Supp}(r) \cap E_1 \neq \emptyset \} \text{ and } R_2 = \{ r \in R : \text{Supp}(r) \subseteq E_2 \}$$

Then we have $R_1 \cup R_2 = R$ and $R_1 \cap R_2 = \emptyset$. In fact, for each relation $r \in R$, if $r \notin R_1$, then $\text{Supp}(r) \cap E_1 = \emptyset$ and so $\text{Supp}(r) \subseteq E_2$, thus $r \in R_2$.

Therefore, $G = (E, R) = (E_1 \cup E_2, R_1 \cup R_2) = G_1 \oplus G_2$, contradiction.

Sufficiency. The proof is again by contradiction. Suppose condition (ii) holds and G is disconnected. Thus, there exist nontrivial subgraphs G_1 and G_2 of G such that $G = G_1 \oplus G_2$.

Suppose that $G_i = (E_i, R_i), (i=1, 2)$. Pick an event $e_i \in E_i, (i=1, 2)$. Then there are no relations $r_j \in R, j=1, 2, \dots, n$, for any fixed $n \in \mathbb{N}$, such that

$$e_1 \in \text{Supp}(r_1) \text{ and } e_2 \in \text{Supp}(r_n) \text{ and } \text{Supp}(r_i) \cap \text{Supp}(r_{i+1}) \neq \emptyset, i=1, 2, \dots, n-1 \quad (11)$$

Contradiction.

Generally, a connected event graph indicates that some relative stable trends are forming. The event graph can tell us much more than a single event itself does.

5 Conclusions

In this paper, based on set theory and multi-relations theory, we formally define and study the event graph model to discover, extract and summarize the potential events and event graph structures from massive texts, and further discover the trends embodied in the contents in text streams. We also study the connectivity of the event graph model, and give the equivalent conditions to determine the connectivity of event graph. The event graph model can be applied to many fields, such as online news, classification of text data collection, and online text data organization and topic detection of the text streams. This model is still preliminary, but it can serve as a foundation for the analysis of relations between events in massive text streams.

Acknowledgments. At first, we would like to thank Professor Yi Lin for his supervision in set theory and multi-relation theory. Also, we thank the reviewers for their comments that help improve this paper.

References

1. Lin, Y.: General Systems Theory: A Mathematical Approach. Kluwer Academic and Plenum Publishers, New York (1999)
2. Zhai, C., Velivelli, A., Yu, B.: A cross-collection mixture model for comparative text mining. In: Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 743–748 (2004)
3. Mei, Q., Zhai, C.: Discovering Evolutionary Theme Patterns from Text - An Exploration of Temporal Text Mining. In: Proceedings of the 2005 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2005)
4. Gloor, P., Krauss, J., Nann, S., Fischbach, K., Schoder, D.: Web Science 2.0: Identifying Trends through Semantic Social Network Analysis. In: IEEE Conference on Social Computing (SocialCom 2009), Vancouver, August 29-31 (2009)
5. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley (1991)
6. Zhai, C., Lafferty, J.: Model-based feedback in the KL-divergence retrieval model. In: Tenth International Conference on Information and Knowledge Management (CIKM 2001), pp. 403–410 (2001)

Protein Structure Alignment in Subquadratic Time

Aleksandar Poleksic

Department of Computer Science, University of Northern Iowa
Cedar Falls, Iowa, USA
poleksic@cs.uni.edu

Abstract. The problem of finding an optimal structural alignment for a pair of superimposed proteins is often amenable to the Smith-Waterman dynamic programming algorithm, which runs in time proportional to the product of the lengths of sequences being aligned. While the quadratic running time is acceptable for computing a single alignment of two, spatially “fixed”, structures, the time complexity becomes a bottleneck when running the Smith-Waterman routine multiple times in order to find an optimal pairwise superposition. We present a subquadratic running time algorithm capable of computing an alignment that optimizes one of the most widely used measures of protein structure similarity, defined as the number of pairs of residues in two proteins that can be superimposed under a predefined distance cutoff. The algorithm presented in this article can be used to significantly improve the speed-accuracy tradeoff in a number of popular protein structure alignment methods.

Keywords: protein structure, structure comparison, alignment, dynamic programming.

1 Introduction

Automated methods for protein structure comparison are of critical importance in several fields, including protein three-dimensional structure prediction [1-5], functional site comparison [6,7,8], and protein structural and functional annotation [9,10,11]. Protein structure comparison problem is much more difficult than its closely related sequence alignment problem [12-15]. For methods that minimize the inter-atomic distances, such as STRUCTAL [16,17], TM-align [18], Fr-TM-align [19], CAALIGN [20], LOCK [21], or LGA [22], the sequence alignment problem can be viewed as a subproblem of the structure comparison problem, since the goal of the latter is to simultaneously find both, a superposition and an alignment that maximizes a given structure similarity measure. In fact, a common approach to finding an optimal structural superposition of two proteins is to solve multiple pairwise alignment problems, one for each inspected spatial superposition of the input protein structures.

One of the most intuitive and most widely used measures of pairwise structure similarity is the number of atoms in two proteins that can be superimposed under a specified distance cutoff. For now, we will denote this metric by $CA \leq d$, where d

denotes the distance threshold in Ångströms (and CA indicates that the structure of each protein is represented by its sequence of α -carbon atoms).

Many widely used protein structure similarity metrics build upon $CA \leq d$, including GDT_TS [22], MaxSub [23], AL0 [24], " CA -atoms $< 3\text{Å}$ " [25,26] and Q-score [25]. GDT_TS is the main measure used in the CASP benchmark of methods for protein structure modeling [1]. This measure is defined as the average value of GDT_P_i , $i \in \{1, 2, 4, 8\}$, where GDT_P_i represents the percentage of C_α atoms that can be superimposed under i Ångströms of the aligned atoms in the experimental structure. In the CAFASP experiment [27], the quality of a protein model is given by the model's MaxSub score, which represents the weighted fraction of the number of atoms in the model structure that can be fit under 3.5Å . The LiveBench experiment [28] uses " CA -atoms $< 3\text{Å}$ " (in our notation $CA < 3$) and Q-score, among other metrics, to evaluate the sensitivity and the specificity of protein structure prediction servers. The Q-score measure is defined as " CA -atoms $< 3\text{Å}$ " divided by the length of the model.

The widespread use of $CA \leq d$ establishes the need for an efficient algorithm for its optimization. For a pair of superimposed proteins, p and q , $CA \leq d$ can be maximized using a simplified version of the standard Smith-Waterman dynamic programming algorithm [29] with zero gap penalties. This algorithm first computes the score matrix

$$S(i, j) = \begin{cases} 1 & \text{if } \|p_i - q_j\| \leq d \\ 0 & \text{otherwise} \end{cases}$$

(where $\|p_i - q_j\|$ denotes the Euclidean distance between the p_i and q_j) and then fills out the dynamic programming matrix in order to find an optimal alignment of p and q . The cost of both procedures, i.e., the procedure for computing the score matrix and the procedure for filling out the dynamic programming matrix is $O(mn)$, where m and n denote the lengths of proteins p and q , respectively.

While the Smith-Waterman method is fast enough for computing a single alignment of two, fixed in space, proteins, the time complexity becomes a bottleneck when finding an optimal pairwise structural superposition by repeatedly running the Smith-Waterman procedure, once for each inspected spatial orientation of the input structures. To circumvent high computational cost, current methods for protein structure matching trade sensitivity for speed by utilizing heuristic techniques in search for a reasonable, suboptimal solution.

Here we present an $O(mn^{3/4})$ worst-case running time algorithm, guaranteed to maximize $CA \leq d$ for any pair of protein structures. Our benchmarking results show that, in typical protein structure matching applications, the speedup factor of our algorithm over the Smith-Waterman algorithm exceeds an order of magnitude. Hence, our algorithm can be readily applied to improve the tradeoff between the speed and the accuracy in a number of existing protein structure comparison methods, including some of the methods discussed above.

We emphasize that subquadratic alignment algorithms are only known for some special sequence alignment problems, such as the Longest Common Subsequence problem (LCS). Using the so-called "Four Russians Speedup" technique [30,31], LCS problem can be solved in $O(n^2/\log n)$ time.

2 Methods and Results

A protein p of length m can be viewed as a sequence of points in the three-dimensional space:

$$p = (p_1, \dots, p_m), \quad p_i \in R^3, \text{ for } i \in \{1, \dots, m\}.$$

In many applications, the p_i 's represent the protein's C_α atoms.

An alignment of proteins $p = (p_1, \dots, p_m)$ and $q = (q_1, \dots, q_n)$ is a sequence of pairs of points from p and q :

$$A(p, q) = ((p_{i_1}, q_{j_1}), \dots, (p_{i_k}, q_{j_k})),$$

where $1 \leq i_1 \leq \dots \leq i_k \leq m$ and $1 \leq j_1 \leq \dots \leq j_k \leq n$. We will use $A_d(p, q)$ to denote an alignment of p and q that maximizes $CA \leq d$, i.e. the number of aligned pairs (p_i, q_j) at distance $\leq d$.

The subquadratic running time algorithm, presented below, consists of two procedures: a procedure for computing the score matrix and a procedure for computing an optimal alignment. The total cost of our method is dominated by the cost of computing the score matrix, since our alignment routine runs on the order of $O(m \log n)$.

2.1 Computing the Score Matrix

Our algorithm first computes a "trim-down" version of the standard score matrix $S = S(i, j)$. More precisely, the algorithm, presented here, generates, for every point p_i from the protein p , a list $L(i) = (j_1, \dots, j_l)$, $j_1 < \dots < j_l$, of positions of all points from the protein q that are at distance $\leq d$ from p_i . It should be noted that the length of $L(i)$ cannot exceed K_d , where K_d represents an upper bound on the number of C_α atoms that can be packed inside a sphere of radius d in R^3 . This implies that the space requirement for storing the collection of all lists $L = (L(1), \dots, L(m))$ (one for each point p_i from p) does not exceed $K_d \cdot m = O(m)$.

The most straightforward way of computing $L(i)$ is to calculate the distances $\|p_i - q_j\|$ between p_i and each q_j and then append j to the end of the list $L(i)$ if $\|p_i - q_j\| \leq d$. The problem with this approach is that it requires $O(n)$ operations for

each p_i , resulting in $O(mn)$ total cost of the score matrix computation. To speed up the computation of $L(i)$, we first note that many distance calculations can be skipped due to the spacing of the protein's consecutive C_α atoms. Let $w > 0$ be the smallest integer such that $wd > c$, where c is an upper bound on the distance between two consecutive C_α atoms ($c \sim 3.8\text{\AA}$) and let $k = \lceil wd \rceil$. If $\|p_i - q_j\| > 2k$ then $j+1$ does not belong to $L(i)$, since

$$\|p_i - q_{j+1}\| > \|p_i - q_j\| - \|q_{j+1} - q_j\| > 2wd - c > wd \geq d.$$

In general, if $\|p_i - q_j\| > (t+1)k$, where $t > 0$ is an integer, then none of $j+1, \dots, j+t$ belongs to $L(i)$, rendering the calculations of distances between p_i and each q_{j+1}, \dots, q_{j+t} unnecessary.

Assuming the cubic lattice model of protein structures, we now prove that each list $L(i)$ can be computed in $O(n^{3/4})$ time.

Let B_t denotes the closed ball of radius $(t+1)k$ centered at p_i , where $t \geq 0$ is an integer (Fig. 1).

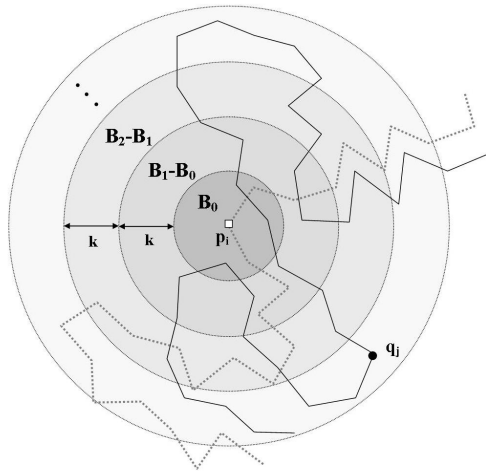


Fig. 1. A toy example of two protein structures, p and q , represented by dotted gray and black lines, respectively. If $q_j \in B_{t+1} - B_t$, then $S(i, j+l) = 0$ for every $l \in \{1, \dots, t\}$.

For every inspected point q_j from the spherical shell $B_{t+1} - B_t$, at least t points from the protein q can be skipped, because $\|p_i - q_j\| > (t+1)k$. Because we are interested in an upper bound on the algorithm's cost, we can assume that the visited points from the protein q are packed as tightly as possible around the point p_i (this scenario results in the least number of skipped points s from q). The key observation here is that the total number of inspected points from q can be represented as

$v = b_0 + b_1 + b_2 + \dots + b_h + a$, where b_i denotes the number of points from q that are packed inside $B_i - B_{i-1}$ (by definition B_{-1} is empty) and $0 \leq a < b_{h+1}$. The total number of skipped points from q is $s \geq 1b_2 + 2b_3 + \dots + (h-1)b_h$. According to the result of Chamizo and Iwaniec, $v = O(h^3)$ and $s = \Omega(h^4)$ (see Theorem 1.1, [3]) and, therefore, $v = O(s^{3/4})$. Since $s < n$, it follows that $v = O(n^{3/4})$.

The algorithm for computing the list L can be written as follows:

Algorithm SCORE_MATRIX

```
// Given the proteins p and q and the distance cutoff
// d, compute the score matrix L.
1.   for i ← 1 to m do
2.       j ← 1
3.       pos ← 1
4.       while j ≤ n do
5.           distance ←  $\|p_i - q_j\|$ 
6.           if distance ≤ d
7.               L[i, pos] ← j
8.               j ← j + 1
9.               pos ← pos + 1
10.          else
11.              j ←  $\lfloor (distance - d) / c \rfloor + j + 1$ 
```

It should be emphasized that the algorithm SCORE_MATRIX, is even more efficient than the general procedure we have just described, since it uses $\|p_i - q_j\| > d + tc$ as the criteria for skipping t points from q . While both $\|p_i - q_j\| > d + tc$ and $\|p_i - q_j\| > (t+1)k$ are sufficient conditions for skipping q_{j+1}, \dots, q_{j+t} , the former results in a more efficient algorithm while the latter makes our proof easier to follow.

2.2 Computing Optimal Alignment

In this section we present $O(m \log n)$ algorithm for computing an optimal alignment $A_d(p, q)$ of p and q . In contrast to the method described below, a standard $O(mn)$ dynamic programming algorithm for $A_d(p, q)$ implements the following recurrence relation to compute the score $C(i, j)$ of an optimal alignment of the sub-structures $p^i = (p_1, \dots, p_i)$ and $q^j = (q_1, \dots, q_j)$:

$$C(i, j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ C(i-1, j-1) + 1 & \text{if } \|p_i - q_j\| \leq d \\ \max\{C(i-1, j), C(i, j-1)\} & \text{otherwise} \end{cases}$$

As we will demonstrate shortly, the special binary form of the score matrix (Fig. 2) makes the protein structure alignment problem amenable to a much more efficient technique, a technique similar to one used for computing the longest common substring (LCS) of two strings over a finite alphabet [33,34]. It is interesting to note that our method has better (worst-case) running time than the corresponding $O(mn \log n)$ algorithm for LCS [33], due to a "sparse" score matrix for any given pair of protein structures. In order to describe the algorithm in more details, we first need some terminology.

We call a pair of indices (i, j) a *match* if $S(i, j) = 1$ (i.e. if $\|p_i - q_j\| \leq d$). It is not difficult to see that the collection M of all matches can be partitioned as

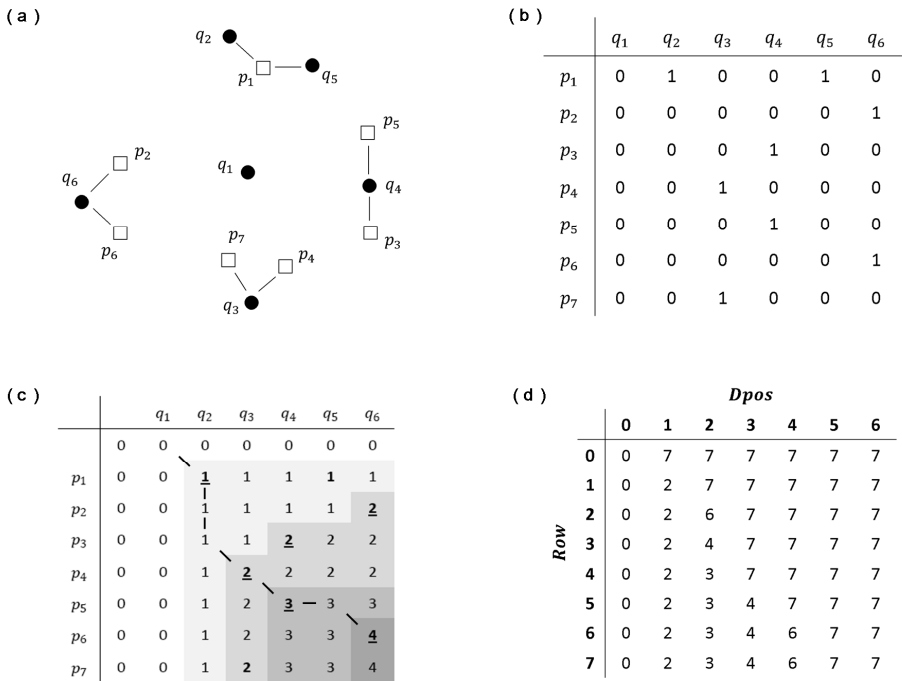


Fig. 2. (a) A toy example of a structure superposition of two proteins p and q . A line connecting p_i and q_j indicates that $\|p_i - q_j\| \leq d$ (b) The score matrix $S = S(i, j)$ (c) Dynamic programming matrix, with k -matches in bold and dominant k -matches underlined (d) Updating the array $Dpos$ of positions of dominant matches, row by row.

$M = \bigcup_{k>0} Q_k$, where Q_k is the set of k -matches, defined as $Q_k = \{(i, j) \mid (i, j) \in M \text{ and } C[i, j] = k\}$.

To find an optimal alignment $A_d(p, q)$, it is sufficient to focus on *dominant k -matches* [35], i.e. k -matches (i, j) such that $j = \min\{j' \mid (i, j') \in M\}$. A single row of the dynamic programming matrix contains at most one dominant k -match, for every $k > 0$. An optimal alignment $A_d(p, q)$ corresponds to a sequence of dominant matches, one for each pair of aligned residues (Fig. 2c). To quickly find this sequence, we scan the rows of the score matrix and update the array $Dpos$ of positions of dominant matches in q (Fig. 2d). Initially, $Dpos[0]$ is set to zero and all other values are set to $n+1$. The rows of the score matrix are then processed, one by one, from right to left. Whenever a dominant k -match (i, j) is found, $Dpos[k]$ is set to j . The array of positions of dominant matches can be efficiently updated using an $O(\log n)$ binary search algorithm. More specifically, for each match (i, j) , the binary search algorithm can be applied to determine whether there exists an open interval $(Dpos[k], Dpos[k+1])$ containing j . If such an interval exists, $Dpos[k+1]$ is set to j . Since the score matrix contains no more than $K_d \cdot m = O(m)$ matches, all of its rows can be processed in $O(m \log n)$ time. The pseudocode for processing the rows of the score matrix (FORWARD), performing the binary search (SEARCH), and tracing back an optimal alignment (TRACEBACK) are given below.

Algorithm FORWARD(L)

// Computes the optimal alignment score $bestScore$, the array of dominant
// positions $Dpos$ and the array $Back$ for tracing back an optimal alignment
// $A_d(p, q)$.

```

1.   bestScore ← 0; Dpos[0] ← 0
2.   for l ← 1 to n do
3.       Dpos[l] ← n+1
4.   for i ← 1 to m do
5.       len ← length of L[i]
6.       for pos ← len downto 1 do
7.           j ← L[i, pos]
8.           k ← SEARCH(j)
9.           if k ≠ -1
10.                Dpos[k+1] ← j
11.                Back[k+1] ← i
12.                if k+1 > bestScore
13.                    bestScore ← k+1
```

Algorithm SEARCH(*j*)

// On input *j*, returns the integer *k*, such that $D_{\text{pos}}[k] < j < D_{\text{pos}}[k+1]$, or -1
 // if such an integer does not exist.

```

1.   left ← 0
2.   right ← bestScore + 1
3.   while left ≤ right do
4.     mid ← ⌊(left + right) / 2⌋
5.     if j > Dpos[mid]
6.       left ← mid + 1
7.     else if j < Dpos[mid]
8.       right ← mid - 1
9.     else return -1
10.  if j < Dpos[mid]
11.    return mid - 1
12.  else return mid

```

Algorithm TRACEBACK

// Computes an optimal alignment $A = A_d(p, q)$.

```

1.   for k ← bestScore downto 1 do
2.     A[Back[k]] ← Dpos[k]

```

It is not difficult to see that the hidden constant factor in the running time of the above alignment routine is $\sim K_d$, where K_d is the number of C_α atoms that can be packed inside a sphere of radius d in R^3 . In practical applications, the hidden constant is small since the distance cutoff is usually set below 8\AA .

2.3 Benchmark

To test the efficiency of our algorithm in real applications, we compiled a test set consisting of 246 pairs of structurally related chains (at various structural levels) from the FSSP database [36]. Our test set is chosen so that the protein pairs can be grouped into three bins of equal size (82 pairs in each), according to the chain lengths: $m, n \leq 250$, $250 < m, n \leq 500$ and $m, n > 500$. The set of pairs of proteins used in our analysis can be downloaded from http://bioinformatics.cs.uni.edu/fast_align.html.

Since the efficiency of our method depends on the proteins' geometry and the spatial positions of the proteins relative to each other, we performed a head-to-head comparison of our algorithm and the standard Smith-Waterman algorithm in four different settings. In the first setting (Table 1), we compared the speed of the two methods on a set of pairs of structurally superimposed chains. The chains were optimally superimposed using the MAMMOTH program [37]. The remaining speed

tests, summarized in Tables 2-4, were performed using the same set of protein pairs, but with the chains from each pair positioned randomly in space, instead of being structurally aligned.

We estimated the factor of speedup of our method over the Smith-Waterman algorithm as a function of the distance between the centers of the proteins, c_1 and c_2 using the distance cutoff $d = 3$: $c_1 = c_2$ (Table 2), $\|c_1 - c_2\| = (r_1 + r_2)/2$ (Table 3) and $\|c_1 - c_2\| = r_1 + r_2$ (Table 4), where r_1 and r_2 denote the radiuses of the proteins' bounding spheres. We note that, for all practical purposes, the results presented in Tables 2-4 are most relevant, since the majority of superpositions inspected by a typical iterative methods for protein structure matching are far away from an optimal superposition [15,26].

Table 1. Observed factor of speedup of our method over the Smith-Waterman method on the set of structurally superimposed pairs

<i>Chain length:</i>	$m, n \leq 250$	$250 < m, n \leq 500$	$m, n > 500$
Score matrix	2	5	7
Alignment	25	105	176
Total	4	10	13

Table 2. Speedup factor when the structures are randomly oriented but have the same center of mass

<i>Chain length:</i>	$m, n \leq 250$	$250 < m, n \leq 500$	$m, n > 500$
Score matrix	5	12	16
Alignment	88	756	1654
Total	8	22	30

Table 3. Speedup factor on the set of randomly oriented pairs of structures satisfying $\|c_1 - c_2\| = (r_1 + r_2)/2$

<i>Chain length:</i>	$m, n \leq 250$	$250 < m, n \leq 500$	$m, n > 500$
Score matrix	6	14	18
Alignment	156	860	1709
Total	10	24	34

Table 4. Speedup factor on the set of randomly oriented pairs of structures such that $\|c_1 - c_2\| = r_1 + r_2$

<i>Chain length:</i>	$m, n \leq 250$	$250 < m, n \leq 500$	$m, n > 500$
Score matrix	8	17	24
Alignment	260	1070	2134
Total	13	33	46

As seen in Table 1, when applied to optimally superimposed chains of lengths $250 < m, n \leq 500$, our alignment method is about 105 times faster than the corresponding Smith-Waterman dynamic programming algorithm. If the cost of computing the score matrix is taken into account, our method is about an order of magnitude faster than the Smith-Waterman algorithm.

We observed a significant increase in the efficiency of our method on structurally unaligned chains, in particular when the structures are far away from each other. For instance, on the set of pairs of proteins of moderate lengths ($250 < m, n \leq 500$), with the same center of mass, the speedup factor is 22 (12 for the score matrix computation and 756 for the alignment). On the other hand, if the bounding spheres of the two structures are only touching each other ($\|c_1 - c_2\| = r_1 + r_2$), the speedup factor is 33 (17 for the score matrix computation and 1070 for the alignment).

3 Conclusion

Many pairwise structure comparison algorithms minimize the proteins' inter-atomic distances by inspecting many different superpositions of the input structures, keeping track of the best superposition and the alignment found so far. In order to find a solution reasonably close to optimum, these methods must search the space of all superpositions with a fine-tooth comb, performing an alignment procedure each time a new superposition is generated. For this task, even an $O(n^2)$ Smith-Waterman alignment algorithm is computationally too expensive.

We present a much faster algorithm for computing an alignment that maximizes one of the most widely used measures of protein structure similarity, defined as the number of pairs of atoms in two structures that can be fit under a specified distance cutoff. Our algorithm can be readily applied to improve the speed-accuracy tradeoff of many popular protein structure similarity methods, including the methods commonly used in protein structure prediction benchmarks.

References

1. Moult, J., Fidelis, K., Kryshtafovych, A., Rost, B., Hubbard, T., Tramontano, A.: Critical assessment of methods of protein structure prediction Round VII. *Proteins* 69(S8), 3–9 (2007)
2. Debe, D.A., Danzer, J.F., Goddard, W.A., Poleksic, A.: STRUCTFAST: protein sequence remote homology detection and alignment using novel dynamic programming and profile-profile scoring. *Proteins* 64, 960–967 (2006)
3. Kim, D.E., Chivian, D., Baker, D.: Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Res.* 32(suppl. 2), W526–W5331 (2004)
4. Teodorescu, O., Galor, T., Pillardy, J., Elber, R.: Enriching the sequence substitution matrix by structural information. *Proteins* 54, 41–48 (2004)
5. Zhou, H., Zhou, Y.: Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments. *Proteins* 58, 321–328 (2005)

6. Xie, L., Bourne, P.E.: Detecting evolutionary relationships across existing fold space, using sequence order-independent profile-profile alignments. *Proc. Natl. Acad. Sci. USA.* 105, 5441–5446 (2008)
7. Gold, N.D., Jackson, R.M.: SitesBase: a database for structure-based protein–ligand binding site comparisons. *Nucleic Acids Res.* 34, D231–D234 (2006)
8. Poleksic, A., Fienup, M., Danzer, J.F., Debe, D.A.: A different look at the quality of modeled three-dimensional protein structures. *J. Bioinform. Comput. Biol.* 6, 335–345 (2008)
9. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 247, 536–540 (1995)
10. Orengo, C.A., Michie, A.D., Jones, D.T., Swindells, M.B., Thornton, J.M.: CATH—a hierarchic classification of protein domain structures. *Structure* 5, 1093–1108 (1997)
11. Wu, C.H., Huang, H., Yeh, L.S., Barker, W.C.: Protein family classification and functional annotation. *Comput. Biol. Chem.* 27, 37–47 (2003)
12. Goldman, D., Papadimitriou, C.H., Istrail, S.: Algorithmic Aspects of Protein Structure Similarity. In: Proceedings of the 40th Annual Symposium on Foundations of Computer Science, pp. 512–522. IEEE Computer Science, Washington, DC (1999)
13. Caprara, A., Carr, R., Istrail, S., Lancia, G., Walenz, B.: 1001 optimal PDB structure alignments: integer programming methods for finding the maximum contact map overlap. *J. Comput. Biol.* 11, 27–52 (2004)
14. Xu, J., Jiao, F., Berger, B.: A Parameterized Algorithm for Protein Structure Alignment. In: RECOMB, pp. 488–499 (2006)
15. Kolodny, R., Linial, N.: Approximate protein structural alignment in polynomial time. *Proc. Natl. Acad. Sci. USA.* 101, 12201–12206 (2003)
16. Gerstein, M., Levitt, M.: Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. In: Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology, pp. 59–67. AAAI Press, Menlo Park (1996)
17. Levitt, M., Gerstein, M.: A unified statistical framework for sequence comparison and structure comparison. *Proc. Natl. Acad. Sci.* 95, 5913–5920 (1998)
18. Zhang, Y., Skolnick, J.: TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* 33, 2302–2309 (2005)
19. Pandit, S.B., Skolnick, J.: Fr-TM-align: A new protein structural alignment method based on fragment alignments and the TM-score. *BMC Bioinformatics* 9, 531 (2008)
20. Oldfield, T.J.: CAALIGN: a program for pairwise and multiple protein structure alignment. *Acta Crystallogr. D Biol. Crystallogr.* 63, 514–525 (2007)
21. Singh, A.P., Brutlag, D.L.: Hierarchical protein structure superposition using both secondary structure and atomic representations. In: Proceedings of the International Conference of Intelligent Systems in Molecular Biology, vol. 5, pp. 284–293 (1997)
22. Zemla, A.: LGA - a Method for Finding 3D Similarities in Protein Structures. *Nucleic Acids Res.* 31, 3370–3374 (2003)
23. Siew, N., Elofsson, A., Rychlewski, L., Fischer, D.: MaxSub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics* 16, 776–785 (2000)
24. Sali, A., Blundell, T.L.: Comparative protein modeling by satisfaction of spatial restraints. *J. Mol. Biol.* 234, 779–815 (1993)
25. Ginalski, K., Grishin, N.V., Godzik, A., Rychlewski, L.: Practical lessons from protein structure prediction. *Nucleic Acids Res.* 33, 1874–1891 (2005)

26. Poleksic, A.: Algorithms for optimal protein structure alignment. *Bioinformatics* 25, 2751–2756 (2009)
27. Fischer, D., Rychlewski, L., Dunbrack Jr., R.L., Ortiz, A.R., Elofsson, A.: CAFASP3: the third critical assessment of fully automated structure prediction methods. *Proteins* 53(S6), 503–516 (2003)
28. Rychlewski, L., Fischer, D.: LiveBench-8: the large-scale, continuous assessment of automated protein structure prediction. *Protein Sci.* 14, 240–245 (2005)
29. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. *J. Mol. Biol.* 147, 195–197 (1981)
30. Arlazarov, V.L., Dinic, E.A., Kronrod, M.A., Faradzev, I.A.: On economic construction of the transitive closure of a directed graph. *Soviet Math. Dokl.* 11, 1209–1210 (1970)
31. Masek, W.J., Paterson, M.S.: A faster algorithm for computing string-edit distances. *J. Computer and System Science* 20, 18–31 (1980)
32. Chamizo, F., Iwaniec, H.: On the sphere problem. *Revista Matemática Iberoamericana* 11, 417–429 (1995)
33. Hunt, J.W., Szymanski, T.G.: A fast algorithm for computing longest common subsequences. *Communications of the ACM* 20, 350–353 (1997)
34. Mukhopadhyay, A.: A fast algorithm for the longest-common-subsequence problem. *Information Sciences* 20, 69–82 (1980)
35. Hirshberg, D.S.: Algorithms for the longest common subsequence problem. *JACM* 24, 664–675 (1977)
36. Holm, L., Ouzounis, C., Sander, C., Tuparev, G., Vriend, G.: A database of protein structure families with common folding motifs. *Protein Sci.* 1, 1691–1698 (1992)
37. Ortiz, A.R., Strauss, C.E., Olmea, O.: MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci.* 11, 2606–2621 (2002)

Bio-inspired Self-organized Public Key Authentication Mechanism for Mobile Ad-hoc Networks

Parisa Memarmoshrefi, Roman Seibel, and Dieter Hogrefe

Institute for Computer Science, Telematics Group, University of Göttingen
Goldschmidtstrasse 7, 37077 Göttingen, Germany
{memarmoshrefi, seibel, hogrefe}@cs.uni-goettingen.de

Abstract. In mobile ad-hoc networks (MANETs), where there is no centralized authority to provide security, trust and reputation mechanisms are applied to maintain security by identifying trustworthy and untrustworthy nodes. However, traditional authentication mechanisms are infeasible for MANETs due to the lack of infrastructure and frequent topology changes. In this paper, we propose a self-organized and localized public key authentication mechanism based on ant colony systems. Every node generates its own public-private key pair, issues certificates to neighboring nodes and provides on-demand authentication services by means of gathering certificate chains towards a target node. Pheromone concentration left by ants along the path of the certificate chains represents the trust level of a node towards other nodes. This model is able to authenticate public keys by selecting the most trustworthy path in certificate chains gathered by ants and can identify and prevent certificate chains with individual or colluding malicious nodes.

Keywords: public key authentication, security threat in trust and reputation systems, ant colony optimization, MANETs.

1 Introduction

Mobile ad-hoc networks are multi-hop wireless networks without any infrastructure which are used in different applications, such as civilian or military applications and emergency rescues. In environments where cooperation is unavoidable providing security services for communication is an essential issue. Since authentication is the most important and the basic part of any secure communication, in this work we consider the authenticity of a node as the context of trust in the authentication process. In order to provide secure network communication a key distribution procedure between nodes is necessary, in which the keys are transmitted in a secure way over basically insecure channels. A framework of trust relationships is required to be built for authentication purposes in the key distribution procedure.

A classification of authentication mechanisms in MANETs is presented in [1], identifying three different key management schemes: 1-central certification authority (CA) systems, which are not suitable for dynamic environments; 2-distributed CA systems, where n nodes in a MANETs collectively perform the task of a CA; 3- self

CA systems which are based on web of trust. This model allows nodes to become an individual CA, generate their own keying material and issue public key certificates for their own and for others base on their knowledge. Through a certificate, the binding of a node's identity to its corresponding public key is proven by a digital signature of the issuer. Each node maintains a local certificate repository, and performs the public key authentication via chain of certificates.

However there still exist security threats in this trust model. A number of security threats are presented in [3] which in general could be applied in trust and reputation systems. Therefore one of the most important subjects is identifying and coping with dishonest misbehaving nodes along the certificate chains who try to cheat other nodes into believing in false node-public key bindings.

To mitigate the problem, we propose on-demand, trust-based public key management based on ant colony systems [4]. The dynamic nature of ad hoc networks, caused by the mobility of nodes and the changing behavior of nodes, makes ant colony optimization an appropriate choice for a trust model. In our proposed scheme each node creates its own public-private key pair, issues certificates to neighboring nodes and stores the trust level of nodes in its repository. Reactively a node performs public key authentication by sending out ants toward the target node. The responsibility of the ants is to find the most trustworthy certificate chain. At the same time, through building a certificate chain, the ants leave traces of pheromone on the path representing a trust level of the path. Despite of misbehaving nodes each node can make a suitable decision about obtaining the public key of a target node.

The rest of the paper is organized as follows: Section 2 represents some related works. Section 3 includes trust model and security threats of our model. The ant colony system is described in section 4 and a description of our proposed model is presented in section 5. Some experimental results are presented in section 6 and finally section 7 provides conclusion and future works.

2 Related Work

A public key certificate is a data structure in which a public key is bound to an identity and signed by the issuer of the certificate. In PGP [5] certificates are mainly stored in a centralized certificate repositories. [6] proposes a self-organized public key management where certificates are stored and distributed by the nodes. The main problem of this scheme is large overhead for storing the approximate global certificate graph. To solve this problem, authors in [7] proposed a solution. They designed an on-demand public key management. In this scheme all certificates need to be issued and trusted locally. A certificate chain can be obtained hop-by-hop, as long as a route discovered between source node and destination node. Recently another solution is proposed [2] which is based on the existence of a web of trust.

On the other hand, many trust and reputation systems have been proposed, for dealing with malicious behavior, in many different domains such as human social networks, e-commerce [8], peer-to-peer networks [9][10][11], mobile ad-hoc networks CONFIDANT [12] and CORE [13] and sensor networks [14][15].

TACS [11], which was helpful for our model, AntRep [16] and [17] are trust models using a bio-inspired algorithm of the ant colony systems in order to provide guarantee of network resources availability and trustworthiness. In these systems the main principle behind the interaction is called stigmergy, which means that the trace left in the environment by an action encourages the performance of the next action, by the same or different agent (ant).

3 Trust Model and Security Threats

The trust model of our scheme is based on a web of trust of public key certificates that guarantees the bindings of the public keys to their related user identities. As an example $Cert_{i \rightarrow j}$ denotes the certificate that i signed with its private key, Sig_i , to show the binding of node j 's identity, ID_j , and its corresponding public key PK_j . In addition to ID_j and PK_j , the data structure $Cert_{i \rightarrow j}$ contains trust value or confidentiality, C , and validity time, T . We consider this web of trust a certificate graph $G(V, E)$, whose set of vertices, V , represents public keys and the set of edges, E , represents certificates.

Each node periodically, depending on the expiration time of certificates, creates direct edges to its neighbors and issues certificates to them, if there is an acceptable confidentiality level for binding neighbors' ID to their corresponding PK. When a node, S , wants to authenticate the public key of another node, D , which is not located in radio range of S , a chain of valid certificates from S to D is required, fig. 1.

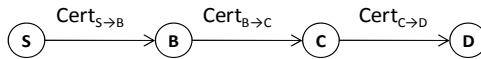


Fig. 1. Certificate Chain

In our example the certificate chain from S to D is $\{Cert_{S \rightarrow B}, Cert_{B \rightarrow C}, Cert_{C \rightarrow D}\}$. Every certificate in the chain will be verified with the public key of the previous certificate in the chain. But how to verify the certificate chain and how to choose the certificate chain composed of trustworthy nodes is still a problem that we discuss in following parts.

3.1 Trust Metrics

Trust metrics is a measure that represents the assurance that a requesting node can obtain the public key of the destination node correctly, through the certificate chain. In this chain fashion, trust transitivity plays a great role which is based on recommendation between entities. However, there is a difference between trusting an entity to provide a specific service and trusting an entity that recommends someone who can provide the service [18]. Trust in the service object is functional trust, while trust in recommending agents is referral trust. In our model we consider the functional trust as the honest binding rate i.e. the number of correct binding signs over all trials.

On the other hand referral trust is the dissemination of these scores to the relying nodes that can be considered as recommendations.

Any node in the network can calculate the trust value of another node's public key if there is a physical communication and consequently a certificate chain between the two nodes using formula 1.

$$t_{SD} = \prod_{k=1}^{k=n} t_k \quad (1)$$

t_k is the trust value between two directly connected nodes on the certificate chain from node S to node D. n is the number of hops between source and destination. It is obvious that the trust in another's public key fades along the path of recommendation.

3.2 Security Threats

There is no guarantee in such decentralized public key management systems that all nodes act correctly and honestly. In general two types of functional and referral misbehavior threatens the security of our trust-based system.

Functional misbehavior occurs when a node or a group of nodes refuses to act correctly in service provision. In our authentication model it raises by not participating in the authentication process or issuing a false certificates with an incorrect binding of a key to an identity. Impersonating another node is an example of functional misbehavior. A malicious node i may issue a certificate that binds the identity of another node, ID_j , to its public key, PK_i , and signs it with its private key Pr_i . The aim of the malicious node is eavesdropping a messages sent to j. Another example is binding the public key of node k, PK_k , to ID_j ; although it should be bound to ID_k .

In the second type of misbehavior, referral misbehavior, a malicious node tries to trick other nodes by providing dishonest recommendations by manipulating the confidence in the authenticity of a given key. One of the important threats of this kind is the Sybil attack [19], where a malicious node generates several keys and identities, binds the IDs to corresponding public keys and issues certificates for them. In this case the malicious node can use these nodes to issue false certificates. As the false certificate is signed by many Sybil nodes, it could be considered as a correct certificate to non Sybil nodes.

The aim of our proposed model is a self-organized authentication mechanism which enables defense against these two types of misbehavior. In this paper we concentrate on misbehaving nodes who try to defect the authentication service by disseminating false information.

4 Ant Colony Optimization

In a system based on ant colony optimization, mobile agents, called artificial ants spread through the network from source to destination in order to find the most

trustworthy path towards a destination node. They remember the visited nodes they pass, and deposit ‘pheromone’ on them. Ants are attracted to paths with higher pheromone concentration. When an ant wants to move from starting node S toward a destination it chooses one of the neighboring nodes of S , i , with the probability defined by following transition rule:

$$p(S, i) = \frac{[\tau_{Si}]^\alpha \cdot [\eta_{Si}]^\beta}{\sum_{j \in N(S)} [\tau_{Sj}]^\alpha \cdot [\eta_{Sj}]^\beta} \quad ; \quad \sum_{i \in N(S)} p(S, i) = 1 \quad (2)$$

where τ_{Si} is the pheromone deposit on the edge between S and i , η_{Si} is the goodness value of the link between S and its neighbor node, $N(S)$ is the list of neighboring nodes of S and α and β are the weights for balancing between deposited pheromone and goodness value of the edge respectively.

The following transition rule is used to provide a pseudo-aleatory path choice:

$$r = \begin{cases} \operatorname{argmax}_{j \in N(S)} [\tau_{Sj}]^\alpha \cdot [\eta_{Sj}]^\beta & \text{if } q \leq q_0 \\ R & \text{otherwise} \end{cases} \quad (3)$$

where r is the next chosen node by an ant in its next movement, q_0 is the probability of choosing deterministically the most promising edge, q is a measure in range of $[0, 1]$ and R is a randomly selected neighbor node.

Once a forward ant finds the required destination, a return ant is generated which retraces the path of the forward ant back to the source. The return ant then updates the value of pheromone at each intermediate node according to following reinforcement learning rule:

$$\tau_{ij} = (1 - e) \cdot \tau_{ij} + \Delta\tau_{ij} \quad (4)$$

where the backward ant came from neighbor j to node i , e is the rate of pheromone evaporation. Pheromone evaporation is a function of time and allows the system to forget the old information, search new paths and also avoid convergence to premature-optimal solutions by encouraging exploration of edges not yet visited. $\Delta\tau_{ij}$ is the amount of pheromone deposited with typically $\Delta\tau_{ij} = K/f(c)$. $K > 0$ is a constant. $f(c)$ is the cost function which serves as a metric of hop counts from current node to destination, the delay of finding a destination, the available bandwidth of the link or the energy consumption of each node along the way. The security metrics are explained in the system description part.

5 System Descriptions

We consider an ad hoc environment, in which all nodes perform five main processes: public-private key generation and certificate issuing, certificate chains discovery, public key authentication by certificate verification, certificate chains trust updating and certificate revocation. The following shows the details of each process.

5.1 Public-Private Key Generation and Certificate Issuing

First each node creates its public key and corresponding private key locally. Then all neighboring nodes issue public key certificates for each other. If node A, based on its knowledge believes that a given public key PK_B belongs to a given node B, node A has to issue a certificate for node B and sign it with its private key Pr_A , to show its assurance of the binding of identity B, ID_B , to its related public key PK_B . Each node saves the public key certificates it issues for others and certificates issued to it in its repository. For every node in the certificate repository there is a confidence level of trust that shows to which extent that node issues correct and not mismatched certificates. Figure 2 is an example of public key certificate generation for the nodes who are in radio range of each other.

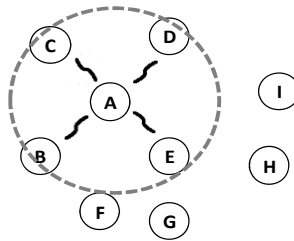


Fig. 2. Certificate issuing for neighboring nodes located in the radio range

Table 1 shows the certificate table (CT) in which every node stores the certificates issued by neighboring nodes. Each entry in CT corresponds to one certificate and each column shows the belief of each neighboring node to a certain certificate.

Table 1. Certificate Table of Node A

Certificates	Neighbors			
	B	C	D	E
$Cert_i$	$Cert_{B \rightarrow i}$	$Cert_{C \rightarrow i}$	$Cert_{D \rightarrow i}$	$Cert_{E \rightarrow i}$
...				

Each node also has a table to store trust value of its neighboring nodes. Since this value presents the pheromone we name the table a trust-pheromone table (table2).

Table 2. Trust-Pheromone Table of Node A

Trust-Pheromone	Neighbors			
	B	C	D	E
Pheromone	t_{AB}	t_{AC}	t_{AD}	t_{AE}

5.2 Certificate Chain Discovery

Our model is a reactive evidence distribution scheme. Ants are sent out only when a certain certificate is required. We assume that during the key generation and certificate issuing step, trust relationships have been established between nodes and their neighbors. We also assume that source node S wants to obtain the public key of destination node, D (Figure 3). S sends out several ants to explore the path to the required certificate. Ants in each node, e.g. i , choose their next state, e.g. j , according to formula (2), in which τ_{ij} is the phomone and represents the referral trust that node i has on relaying node j on obtaining the public key of destination node, D. η_{ij} is the functional trust that node i has in general toward node j in providing authentication service. In this work we assume that all nodes are trustable in making the authentication service available.

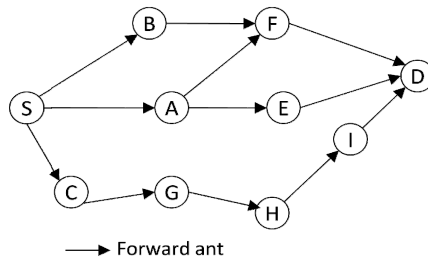


Fig. 3. Forward ants carry certificate request

Forward ant stores the certificate path. Once a forward ant finds the required certificate, a backward ant is generated who retraces exactly the path of the forward ant back to the source. Through returning from destination to source, the backward ant adds the certificates of the intermediate nodes in its packet and provides a chain of certificates (figure 4).

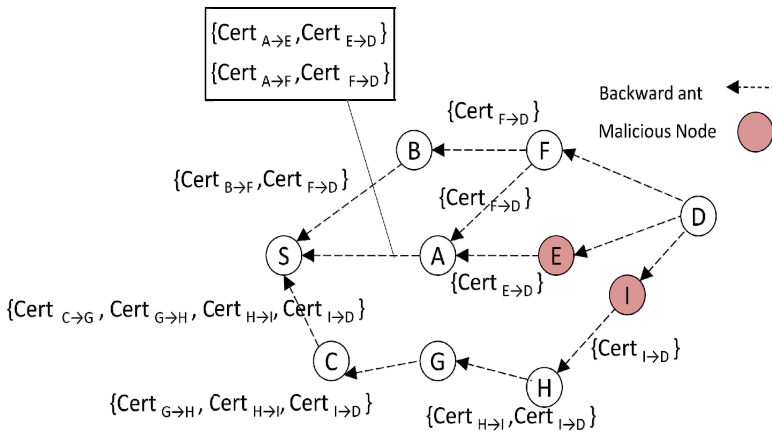


Fig. 4. Backward ants carry certificate reply

Finally when the source node receives the backward ants from destination, it computes the trust value of the chain, using formula 1, and inserts the certificate chains and their corresponding trust values into the certificate chain table.

Table 3. Certificate chain table of a node

Certificate chain	Destination	Trust value
SBFD	D	t_1
SAED	D	t_2
SAFD	D	t_3
SCGHID	D	t_4

In case of any topology changes during a backward ant tries to find the way back to the source, source node should send out some other ants toward target node.

5.3 Public Key Authentication by Certificate Verification

When the source node receives different chains of certificates, it verifies the ID-key binding of the destination which is contained in certificate chains. If S detects no conflicting certificates (e.g. certificates contain the same identity of D but with different public keys), it regards the maximum received trust value as trust value of the D's certificate:

$$t_{SD} = \text{Max}_{i \in CCh(S \rightarrow D)}(t_i) \tag{5}$$

where $CCh(S \rightarrow D)$ is the list of certificate chains which S receives by requesting for certificate of D. However, considering the existence of misbehaving nodes, S may receive mismatched destination certificates through different certificate chains.

5.4 Certificate Chains Trust Update

Trust updates will occur in three following situations:

5.4.1 General Local Updating: In each intermediate node, if there is no mismatch information received by backward ants from its neighbors, the pheromone entry of the neighbor node, from where backward ant came from, will be updated as following:

$$t_{ij} = (1 - e) \cdot t_{ij} + dt_{ij} \tag{6}$$

$$dt_{ij} = e \cdot (1 - t_{ij} \cdot \eta_{ij}) \tag{7}$$

where e is the pheromone evaporation value. By dt_{ij} in formula (7) we give the opportunity to edges with lower values of pheromone to recover faster.

5.4.2 Punishing: In case of observing any mismatch in certificate chains, node S analyzes the received certificate chain to identify Sybil nodes. S classifies the malicious nodes that offer confidentiality values for a certificate which is far from the opinion of the norm of the nodes in certificate chains. As this observation analysis is out of the scope of this paper, we suppose that the malicious nodes are already identified by the source node (e.g. node E and I in figure 4). In this case, as punishment, the source node reduces the trust level of its neighbors who led to the malicious node in the certificate chain by evaporating the pheromone of the edge between S and those neighbors (e.g. A and C in figure 4). Node S also has the responsibility of notifying A and C about the malicious nodes.

$$t_{ij} = t_{ij} - \omega \cdot e \cdot t_{ij} \cdot dp \quad , \quad dp = \frac{1}{D_{pm}} \quad (8)$$

Weight ω is the trust value of a node toward its notifier neighboring node (e.g. t_{AS} in figure 4). This weight is equal to 1 if the notifier itself is also the punisher. D_{pm} is the distance factor (hop count) between punished (p) node and malicious node (m), which can be obtained through the certificate chain. The longer this distance the less is the punishing amount. The punished node who continues to act maliciously as a consequence will be isolated and not further used in the authentication process.

Nodes A and C also have to punish their next neighbors in the path leading to the malicious nodes; Otherwise they will be classified as potentially malicious nodes.

5.4.3 Rewarding: Source node S updates the trust value of every node along the most reliable certificate chain as follows:

$$t_{ij} = (1 - e) \cdot t_{ij} + e \cdot (1 + t_{Max} \cdot \eta_{ij}) \cdot t_{ij} \quad (9)$$

where t_{Max} is the highest trust value corresponding to the most reliable certificate chain. It shows the edges with higher pheromone value are more rewarded than those with lower value.

5.5 Certificate Revocation

If a node believes that in a certificate it issued, the binding of ID to public key of the target node is no longer valid or the trust value is less than the trust threshold, t_{th} , it can revoke that certificate. When a node receives a certificate revocation, it compares the trust value of sender of the message, t_{is} , and the trust value of the nodes its certificate is claimed to be revoked, t_{it} :

$$t_{it} = t_{it} - t_{is} \quad (10)$$

If t_{it} is lower than the trust threshold, it deletes the revoked certificate, otherwise the certificate will still be used.

6 Simulation and Results

We assume that trust relationships have been established between each node and its neighbors. Matlab is used for simulation. The simulation parameters are as follows. 30 nodes are randomly placed in the 100 x 100 meter square form area. Nodes are variables and the radio range of each node is 30 meter. Also we consider the following values for parameters: $\alpha = \beta = 1$, $e = 0.1$, $q_0 = 0.9$, $\eta_{ij} = 1$ for $\forall i, j$.

The simulation proceeds in rounds. Each round the updated pheromone values lunched into the network. In each round five requests are made. In each request one of the four randomly chosen nodes requests the public key certificate of one of two random destination nodes. The numbers presented in each iteration are averaged over all five requests.

First we consider all requests are to be the same and between a fixed pair of source and destination nodes. Figure 5 shows the trust value of received certificate chains in case of increasing the percentage of malicious nodes in the network. It is shown that after some iteration the network learns the chains of trustable nodes that lead to the certificate of the destination node. The result shows that the model is capable of choosing the trustworthy nodes to get the public key certificate of the destination node despite of up to 60% malicious nodes. The reduction of the reliability of the certificate chain in case of having 30% malicious nodes is because of the location of some interconnected nodes. If these nodes, which connect one part of the network to another part, are malicious no reliable path could be found between these two parts of the network.

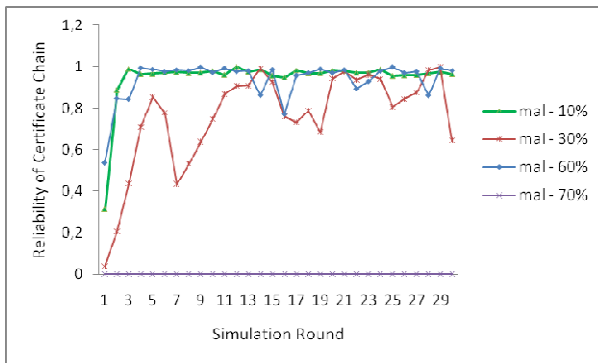


Fig. 5. Reliability of certificate chain with different percentage of malicious nodes

In the second experiment we made five different requests and compare the success rate with different amounts of malicious nodes. The success rate shows the percentage of requests for which the requester successfully obtains the public key certificate. It is the number of correct certificates obtained over the total number of requests each round.

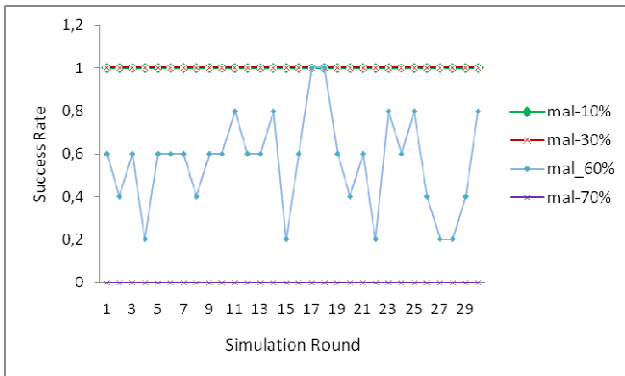


Fig. 6. Comparison of success rate with different percentage of malicious nodes

Figure 6 shows success rate in case of up to 30% malicious nodes in network, is constantly 1. By increasing the number of malicious nodes to 60% the average success rate is nevertheless more than fifty percent. All requests are completely failing if the amount of misbehaving nodes reaches 70%, which means 21 nodes out of 30 nodes.

7 Conclusions and Future Works

In this work, we proposed a robust self-organized public key management scheme for ad-hoc environments based on ant colony systems. This scheme is able to authenticate and obtain the public key of a target node successfully in spite of malicious relay nodes. Traces of pheromones represent the trust level of nodes throughout the certificate chain.

As our future work we aim at clustering data gathered by ants in order to identify Sybil nodes. Using ants, the source node has the opportunity to gather information about the public key of the destination through different chains containing different recommending nodes. Therefore by analyzing the recommendations, the source node can identify Sybil nodes who try to pretend to be different nodes and disseminate false information. We intend to implement our proposed scheme by a network simulator and explore its performance and consider further parameters such as node mobility and its effects.

References

1. Hashmi, S., Brooke, J.: Authentication Mechanisms for Mobile Ad-Hoc Networks and Resistance to Sybil Attack. In: Proceedings of the Second International Conference on Emerging Security Information, Systems and Technologies. IEEE Computer Society (2008)
2. Dahshan, H., Irvine, J.: A Robust Self-Organized Public Key Management for Mobile Ad hoc Networks. Security and Communication Networks, 16–30 (2010)

3. Mármol, F.G., Pérez, G.M.: Security Threats Scenarios in Trust and Reputation Models for Distributed Systems. *Computers & Security* 28(7), 545–556 (2009)
4. Cordon, O., Herrera, F., Stützle, T.: A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends. *Mathware & Soft Computing* (2002)
5. Zimmermann, P.: *The Official PGP User's Guide*. MIT Press (1995)
6. Capkun, S., Buttyan, L., Hubaux, J.-P.: Self-Organized Public-Key Management for Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 52–64 (2003)
7. Li, R., Li, J., Liu, P., Chen, H.: On-Demand Public-Key Management for Mobile Ad hoc Networks: Research Articles. *Wireless Communications & Mobile Computing* 6, 295–306 (2006)
8. Jøsang, A., Ismail, R., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems*, 618–644 (2007)
9. Marti, S., Garciamolina, H.: Taxonomy of Trust: Categorizing P2P Reputation Systems. *Computer Networks*, 472–484 (2006)
10. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The Eigentrust Algorithm for Reputation Management in P2P networks. In: *Proceedings of the 12th International Conference on World Wide Web (WWW 2003)*. ACM, Budapest (2003)
11. Mármol, F.G., Pérez, G.M., Skarmeta, A.F.G.: TACS, a Trust Model for P2P Networks. *Wireless Personal Communications* 1, 153–164 (2009)
12. Buchegger, S.: Le Boudec. J.Y.: A Robust Reputation System for P2P and Mobile Ad-hoc Networks. In: *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems* (2004)
13. Michiardi, P., Molva, R.: Core: a Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad hoc Networks. In: *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*, pp. 107–121 (2002)
14. Ganeriwal, S., Srivastava, M.B.: Reputation-based Framework for High Integrity Sensor Networks. In: *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 66–77. ACM, Washington DC (2004)
15. Boukerch, A., Xu, L., Khatib, E.: Trust-based Security for Wireless Ad hoc and Sensor Networks. *Computer Communications* 30, 2413–2427 (2007)
16. Wang, W., Zeng, G., Yuan, L.: Ant-based Reputation Evidence Distribution in P2P Networks. In: *Proceedings of the Fifth International Conference on Grid and Cooperative Computing*, pp. 129–132. IEEE Computer Society (2006)
17. Jiang, T., Baras, J.S.: Ant-Based Adaptive Trust Evidence Distribution in MANET. In: *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops*, vol. 7, pp. 588–593. IEEE Computer Society (2004)
18. Jøsang, A., Golbeck, J.: Challenges for Robust of Trust and Reputation Systems. In: *Proceedings of the 5th International Workshop on Security and Trust Management*, Saint Malo, France (2009)
19. Douceur, J.R.: The Sybil Attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) *IPTPS 2002*. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)

A Method for the Detection of Meaningful and Reproducible Group Signatures from Gene Expression Profiles

Louis Licamele^{1,2} and Lise Getoor¹

¹ University of Maryland, College Park MD 20740, USA,
licamele@cs.umd.edu

² Vanda Pharmaceuticals, Inc., Rockville MD 20850, USA

Abstract. Gene expression microarrays are commonly used to detect the biological signature of a disease or to gain a better understanding of the underlying mechanism of how a group of drugs treat a specific disease. The outcome of such experiments, e.g., the signature, is a list of differentially expressed genes. Reproducibility across independent experiments remains a challenge. We are interested in creating a method that can detect the shared signature of a group of expression profiles, e.g., a group of samples from individuals with the same disease or a group of drugs that treat the same therapeutic indication. We have developed a novel Weighted Influence - Rank of Ranks (WIMRR) method, and we demonstrate its ability to produce both meaningful and reproducible group signatures.

Keywords: gene expression analysis, gene expression profiles, drug discovery, bioinformatics, data mining.

1 Background

Microarray technology is often credited with leading the advancement in the field of modern biological research and was coined as an Array of Hope shortly after its introduction [1]. As microarrays have become commonplace in the laboratory, the amount of gene expression data available in the public domain continues to grow at a rapid pace. Microarray experiments, whether they set out to discover biomarkers for a particular disease or to characterize a group of similar tissue samples, tend to have the same outcome: a list of differentially expressed genes (DEGs). In recent years, a growing debate has developed surrounding the scientific validity of microarrays in respect to their reliability [2-3]. Low reproducibility of DEGs across independent experiments testing the same hypothesis has become the norm [4]. Novel methods to detect robust group signatures from gene expression experiments are needed.

Gene expression profiling has traditionally been used to detect genetic differences between various types of groups including detecting gender differences [5], predicting cancer prognoses [6], segmenting and explaining diseases and their

subtypes [7], and understanding the underlying mechanism of biological processes and pathways [8]. Gene expression data are good sources for investigating and predicting the potential therapeutic effects of a drug because they characterize the response of the cell to external stimuli. A method that generates more reliable and reproducible results (e.g., lists of DEGs) from gene expression data is well positioned to become the core predictive model of a drug discovery system.

It is important to note, however, that many factors complicate analysis of gene expression experiments, including assumptions about the biological processing of mRNA and confounding factors inherent in mRNA expression data. Furthermore, reproducibility has remained low among these types of experiments, calling into doubt the validity of the detected signatures. For example, using an identical set of RNA samples across several different commercial platforms, Tan et al. [9] found only four common DEGs. Both Ramalho-Santos [10] and Ivanova [11] independently found only six DEGs in common among roughly 200 that had been identified in each study (even though they had a similar study design using the same platform). In another study by Miller et al. [12], who compared the effect of varying platforms on the same samples, there were only 11 DEGs in common of 425 DEGs that were found by CodeLink and 138 DEGs found by the Affymetrix platform. These are all examples of studies that exhibit how current methods are producing irreproducible signatures. This lack of reproducible findings indicates that false positives are being detected, and that these methods may be overfitting the data. Furthermore, many methods are complex and only explain a group in a piecewise fashion (e.g., a decision tree-type model). We believe that the ideal method does not require such strict filtering and instead dynamically weights the influence of each probe based on the relative rank of that probe within each member of the group.

We propose the creation of a group profile that will serve as the representative profile for a given group of interest. A gene expression profile is the representation of the activity of thousands of genes at once for a given sample. A group profile represents the shared activity of these thousands of genes across all of the member samples belonging to the group. For example, we can create a group profile consisting of all available antipsychotic drugs; we refer to this as an antipsychotic profile. Traditionally, researchers attempt to find probes or genes that form the signature for a group by evaluating probes above a certain fold-change threshold. These methods will detect the signature common to the group in the rare case that the shared effect is incredibly strong (and there are no large experimental biases between the expression profiles). However, the majority of the time, the true signal is missed because it is not significantly up- or down-expressed in every one of the instances that make up a group (we refer to this as the full group). These methods preferentially detect very big changes within a subgroup of samples and then merge all of these differentially expressed genes with a combination function. Unfortunately, this approach does not find true signatures common to the full group and allows the method to overfit the data. Our method differs from most previous methods by focusing on detecting

signatures common to the full group, signatures that are normally overlooked by other methods, e.g., decision trees and support vector machines, which can explain a group as a combination of rules defining unknown subgroups.

The representation of a group profile is a ranked list of all probesets on the microarray. A benefit of our approach is that this is the same representation as a single profile. This representation allows any current and future methods for non-parametric gene expression data to be used with our group profiles. We can focus on the most up- and down-expressed probesets from the profile, which we refer to as the signature of the group (separately they are the up and down signatures respectively). For example, we can make use of methods developed by others (e.g., Connectivity Map (CMAP) [13]) to use this antipsychotic group profile to search a database for drugs sharing the same signature. Alternatively, we can use still other methods (e.g., the L2L Microarray Analysis Tool [14]) to evaluate if any particular biological process is overrepresented within this signature, an approach that would provide additional insight into the common mechanism of antipsychotic therapies.

In this paper, we introduce and describe our rank of ranks method for group profile creation. We evaluate the utility of this group analysis method using a pilot study in which we focus on the antipsychotic group from the original CMAP build 01 dataset. Our evaluation consists of both understanding the group profiles biologically and demonstrating the ability to use a signature from these profiles as a predictive model of therapeutic use. We conclude with a full analysis of the newer, and larger CMAP build 02 dataset, including a sensitivity evaluation of each group as well as the validation of the most robust profiles within an independent dataset. All the results are available at GEPedia.org.

2 Problem Definition

Given a database D of treatments (i.e., drugs or other compounds), $D = t_1, \dots, t_n$, we are interested in creating a set of group profiles. A group can be defined as a set of instances (e.g., cells treated with a particular drug) that share something of interest in common (e.g., the same therapeutic use, mechanism of action, side effect, chemical structure). We are interested in understanding what is biologically common for a given group profile as well as evaluating the ability to query the database with the group profile to predict new members of the group. Our goal is to discover other drugs or treatments, perhaps originally developed for a different therapeutic purpose, which are likely to also share the same therapeutic properties as the query group. These therapeutic agents are thus good candidates for which new uses can then be evaluated.

For each treatment instance t in the database, there is both general information about the experimental conditions of the sample as well as the actual experiment data from the microarray itself. The gene expression profile is represented as a ranked list (amplitude of the treatment as compared to the control). Information specific to the treatment (i.e., the name of the drug, the therapeutic class [class] and subclass [subclass] as defined by the chemicals Anatomical

Therapeutic Chemical [ATC] code) is represented. There is also information that describes the experimental conditions of the sample, specifically the molar amount of substance (mol), the vehicle used for delivery of the drug (e.g., water, EtOH, MeOH, DMSO), and the batch or round in which the sample was run. A group, and therefore a group profile, can be created from any of these meta-labels associated with the samples.

3 Group Profile Creation (Weighted Influence Model - Rank of Ranks Method)

Previous methods have demonstrated that weighted distribution-based statistics can be more robust in detecting similarity in the pairwise comparison of gene expression data [13]; therefore, we propose a method for determining what is common among a group by also using a weighted method. This dynamic weighting of probes allows us to avoid strictly filtering any probes as is done with a fold-change threshold approach. We calculate the average rank of each probe across the members of the group and then re-rank the probes based on this average rank. We refer to this as the Weighted Influence Model, Rank of Ranks (WIMRR) method. The rank of each probe within each treatment t is known: $\text{rank}(p, \text{probes}(t))$. Let us assume we have a binary membership function, $\text{member}(t, g)$, that returns 1 if treatment instance t is a member of group g and returns 0 otherwise. The size of the group is equal to the number of treatment instances that are members of the group. The average rank for each probe is then calculated. Given this set of average ranks across the members of a particular group, the probes are now re-ranked according to how consistently they are up- or down-expressed across the group. We define $\text{Profile}(g)$ as the probes in $\text{probes}(g)$ sorted by their average rank across all members of the group.

4 Group Profile Evaluation - A Pilot Study

We make use of the original CMAP dataset (build 01) from the Broad Institute to evaluate our group profile method as part of a pilot study. We refer to this as the CMAP 1.0 dataset. We use this smaller, simpler dataset to characterize our method. Later, we analyze the newer CMAP build 02 dataset (CMAP 2.0), which contains many more treatments. For each treatment instance in the CMAP dataset, probe sets are first ranked based on their level of expression relative to the vehicle control in a fashion similar to the method described by Lamb et al. [13]. A group profile is then created for each therapeutic use according to the ChemBank annotation for the instances using our novel WIMRR method. The signature of each group profile is created by selecting the top and bottom k probes. For this evaluation, we set $k = 50$.

4.1 Antipsychotics from Pilot Study

We focus on the antipsychotic profile from the CMAP 1.0 dataset as an example by which to analyze the WIMMR group profile creation method. The

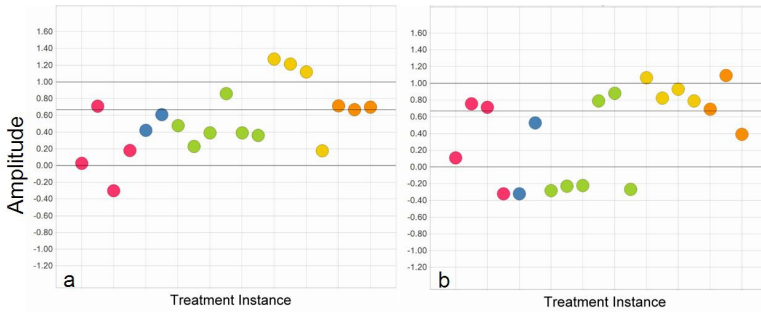


Fig. 1. The amplitude values for a) the top probe found by the group profile method is from the BHLHB2 gene and b) the top probe by the fold-change method that is greater than 2. The lines correspond to a fold-change of 2 and 3, respectively.

antipsychotic group is selected as the example because it includes a large number of unique drugs. The instances from the CMAP 1.0 dataset that are labeled as antipsychotic agents according to ChemBank are used to create this group. The antipsychotics profiled in this dataset include chlorpromazine, clozapine, haloperidol, thioridazine, and trifluoperazine. There are 19 profiles total for this group, consisting of replicates across different concentrations. The group profile is created and the top and bottom 50 probes are selected to serve as the signature for this group.

The top and bottom probes can both provide valuable insight. We focus on the top 50 probes, but the same analysis can be performed with the bottom 50 probes in an analogous way. The amplitude value for the top probe (Affymetrix probe id 201170 s at) is shown in Fig. 1A. This probe, which corresponds to the basic helix-loop-helix domain containing, class B, 2 (BHLHB2) gene, is almost exclusively up-expressed in all of the antipsychotic instances. We evaluate the specificity of this probe by determining how this probe behaves across the whole database (Fig. 2A). All but one of the antipsychotic instances (pink dots in first column) show a clear increase in expression levels. The next set of groups all contain drugs that are known to also act as antipsychotics; this is expected if this probe is predictive of antipsychotic activity. The second group is the tranquilizers (includes prochlorperazine, fluphenazine, and trifluoperazine), the third group is antiemetics (includes prochlorpromazine and trifluoperazine), and the fourth group is the antineoplastics (includes prochlorpromazine). There is a clear pattern of antipsychotic activity related to the up-expression of this probe across the database.

We now compare what we have seen with the top probe from our method with a probe selected using more conventional methods. A potential alternative method for selecting probes (and genes) of interest that has been used extensively in the field has been to select probes that are commonly up-, or down-, expressed above a particular threshold. The most common thresholds used in the literature are fold-changes greater than or equal to either 2 or 3, which correspond to amplitude values of 0.67 and 1.0, respectively. We select the best probe from this

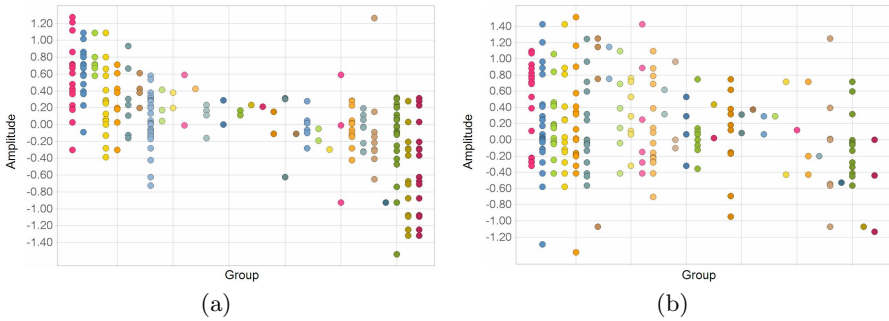


Fig. 2. (a) Specificity of top probe, BHLHB2, from the group profile method. From left to right, the first group is the antipsychotics, the second is the tranquilizers (includes prochlorperazine, fluphenazine, and trifluoperazine), the third group is antiemetics (includes prochlorpromazine and trifluoperazine), and the fourth group is the antineoplastics (includes prochlorpromazine). (b) The top probe from the fold-change greater than 2 method is not specific to antipsychotics. The first group is the antipsychotics, the second is anti-inflammatory, the third is antineoplastics and the fourth is analgesics.

alternative method, determining the probe that exhibits a fold-change greater than 2 in the most antipsychotic instances. The best probe found by this method was for the SEMA3B gene. The amplitude values across all of the antipsychotics for this probe are shown in Fig. 1B. Note that even though some of the individual instances have a very high amplitude value, roughly one-third of the instances have the opposite effect. Again, we determine the specificity of this probe to the antipsychotics by evaluating how it behaves across the rest of the database (Fig. 2B). Visually, we can see that this probe is not specific to the antipsychotics at all.

As validation of our group profile method, we examine BDNF. BDNF (Brain-Derived Neurotrophic Factor) has long been a candidate gene for both schizophrenia and bipolar disorder [15-17]. This additional information demonstrates how this method can give insight into the etiology of the disease that these drugs treat. It also demonstrates how the method extends beyond solely learning about the mechanism of action of drugs. Turning back to the best result from the alternative (fold-change threshold) method, there is no known link between SEMA3B and antipsychotics, schizophrenia, bipolar disorder or other topics expected to be related to antipsychotic agents.

5 Understanding Group Signatures

As mentioned earlier, one of the major benefits of our group profile method is that we can easily plug our group profile results into many algorithms and tools developed to analyze (individual) gene expression data. The probe sets in the group profile signatures can be evaluated for significant overrepresentation of gene ontology (GO) terms, e.g., GO Biological Processes, using the L2L analysis

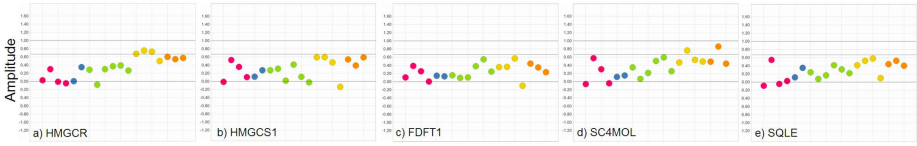


Fig. 3. The amplitude values for the probes in the most significantly up-expressed GO term for the antipsychotic group: sterol biosynthetic process. The probes correspond to the a) HMGR, b) HMGCS1, c) FDFT1, d) SC4MOL, and e) SQLE genes.

tool [14]. Given a list of probe sets, e.g., DEGS, and a list to match them to, e.g. GO:BiolProc, L2L calculates the expected number of matches given the probes found on the microarray. From the actual and expected matches, an enrichment score and the corresponding P value for each GO term is then calculated [18]. Additional lists of published probe sets are also evaluated, including GO Cellular Component, GO Molecular Function, reactome protein-protein interactions [19], predicted human MicroRNA targets [20], and cancer gene expression modules [21].

We use the L2L method to evaluate the example group profile of the antipsychotics. The top 50 probes are evaluated for significant overrepresentation of GO Biological Process terms. The most significant terms are all related to lipid homeostasis (Table 1). There are five genes involved in the sterol biosynthetic process (GO:0016126) within the top 50 probes. Out of over 22,000 probes, only 41 are annotated as belonging to this GO term, so 0.11 probes for this term are expected by chance. This GO term, along with the next three in Table 1, pass Bonferroni correction for multiple testing ($p \leq 1.11E-05$ after correction for all four GO terms). The amplitude values for the five genes that are involved in this pathway are shown in Fig. 3. There is an obvious trend that the expression of these probes is increased in almost every antipsychotic instance in our database. However, even though they are always up-expressed, the amplitude value is normally below the common threshold used by other researchers (fold-change of 2 or 3). This is a good example of how the group profile method is able to detect consistent, and therefore more robust, signals in gene expression data; signals that are normally overlooked by current methods.

Table 1. The most significantly overrepresented GO Biological Process terms from the up-expressed antipsychotic signature

GO Term	GO ID	Probes	Expected	Actual	Enrichment	P Value
sterol biosynthetic process	GO:0016126	41	0.11	5	44.73	1.04E-07*
steroid biosynthetic process	GO:0006694	88	0.24	5	20.84	4.89E-06*
alcohol metabolic process	GO:0006066	371	1.01	8	7.91	1.05E-05*
sterol metabolic process	GO:0016125	104	0.28	5	17.63	1.11E-05*
steroid metabolic process	GO:0008202	211	0.58	6	10.43	2.91E-05
cholesterol biosynthetic process	GO:0006695	31	0.08	3	35.50	8.60E-05
lipid biosynthetic process	GO:0008610	281	0.77	6	7.83	1.40E-04

Support for these GO Biological Process findings comes from the work of other researchers aimed at understanding the molecular origin of the known metabolic side effects of antipsychotics that include increased weight gain and propensity to adiposity and insulin resistance [22]. Our observation is consistent with literature reports of an antipsychotic drug effect on the same or overlapping sets of genes involved in lipid homeostasis. Interestingly, a genome-wide screen of *Saccharomyces cerevisiae* heterozygotes had previously revealed that the antipsychotics haloperidol, chlorpromazine, and trifluoperazine had a strong effect on genes involved in yeast fatty acid biosynthesis (OLE1, the ortholog of the human SCD), sterol biosynthesis or phospholipid transport [23].

6 Querying with Group Signatures

The WIMRR method is able to create a specific representative profile for a group of gene expression profiles. We have demonstrated the ability to gain insight into the mechanism of action of a drug class (as well as the disease that it is used to treat) using WIMRR group profiles. Now we utilize the strength of a group profile to detect and predict the therapeutic use of a drug based on an individual gene expression profile.

We use the truncated KS statistics described previously for pairwise (instance-to-instance) similarity calculations [13] to detect instances that are similar to a group profile of interest (instance-to-group). Using the same antipsychotic group profile, we query the database of instances using $k = 50$ (i.e., the signature discussed previously). The instances most similar to this group profile are shown in Table 2, along with their KS score. The last column in Table 2 represents membership in the group of interest, i.e., if a given treatment is a member of the antipsychotic group used in creating the profile. Scanning the list, we see that prochlorperazine (Instance ID = 995) is the most similar non-antipsychotic drug. It turns out that prochlorperazine is in fact a phenothiazine antipsychotic; however, it is more commonly used for the treatment of nausea and vertigo. Prochlorperazine is a highly potent neuroleptic, which is considered a typical antipsychotic. The next non-antipsychotic is fluphenazine, for which two replicates show up as extremely similar to the antipsychotic profile. Fluphenazine is a typical antipsychotic drug used for the treatment of psychosis, e.g., schizophrenia and bipolar disorder. Fluphenazine is also an extremely potent phenothiazine. The next novel compound is calmidazolium, which is a calmodulin inhibitor. Though it is not used as an antipsychotic, it is validated because many of the antipsychotic drugs are potent inhibitors of calmodulin [24].

In fact, it turns out that many of the most significant results are already used as an antipsychotic agent even though they are not labeled in ChemBank as such. These examples are a validation of our method and increase the confidence in the other results that are not already supported by the literature, as these are potentially the important and still unknown alternative uses for these therapeutic agents.

Table 2. The database was queried with the antipsychotic signature (up and down together) and the most similar

Rank	Instance ID	Name	KS Score	Antipsychotic Member
1	1010	thioridazine[INN]	1.58	X
2	1068	thioridazine[INN]	1.483	X
3	1004	trifluoperazine[INN]	1.469	X
4	995	prochlorperazine[INN]	1.435	
5	910	trifluoperazine[INN]	1.408	X
6	417	thioridazine[INN]	1.387	X
7	983	haloperidol[INN]	1.352	X
8	1024	haloperidol[INN]	1.346	X
9	1017	fluphenazine[INN]	1.317	
10	1075	fluphenazine[INN]	1.293	
11	421	trifluoperazine[INN]	1.256	X
12	906	calmidazolium	1.223	
13	870	pyrvinium	1.209	
14	1053	prochlorperazine[INN]	1.201	
15	418	haloperidol[INN]	1.167	X
16	1009	clozapine[INN]	1.162	X
17	419	chlorpromazine[INN]	1.138	X
18	1003	nordihydroguaiareticacid	1.1	
19	416	clozapine[INN]	1.09	X
20	1105	monensin[INN]	1.077	
21	978	pyrvinium	1.065	
22	893	pararosaniline	1.051	
23	882	ionomycin	1.027	
24	941	rottlerin	1.023	
25	1012	troglitazone[INN]	1.018	
26	1082	haloperidol[INN]	1.009	X
27	1055	chlorpromazine[INN]	0.997	X
28	1041	haloperidol[INN]	0.992	X
29	997	chlorpromazine[INN]	0.99	X

7 Analysis of CMAP V2.0

We have introduced our method for creating group profiles from gene expression data. For this, we have used the original version of the CMAP dataset as our motivating example. We have seen how we can gain biological insight from these profiles as well as how to predict new members by querying the group signature. Here we present our analysis of the newly released CMAP 2.0 dataset with our method and describe the results. Groups are defined according to the compounds ATC code. We have analyzed all the groups at ATC level 3 and level 4. ATC level 3 defines the therapeutic/pharmacological subgroup, e.g., N05A = Antipsychotics. ATC level 4 further defines a subgroup based on chemical properties, e.g., N05AE = Indole Derivative Antipsychotics. We focus on groups with three or more compounds, resulting in 117 ATC level 3 groups and 148 level 4 groups.

7.1 GEPedia.org

We have compiled all of the results from our analysis of CMAP 2.0 and have made them available online at GEPedia.org. In this manuscript, we focus on evaluating our group profile method and only highlight a few interesting results from this analysis. We assume that there are many undiscovered biological insights within this dataset. We are releasing all of the data allowing researchers to examine the results for further discoveries and to compare with their own datasets.

Currently, the organization of GEPedia.org is based around the analysis presented in this paper. We include the output of the complete analysis of all groups. For every group, i.e., for all ATC groups, we have made available a) the profile itself, including the up- and down-expressed signatures, b) the analysis of the profile according to the L2L tool, c) the sensitivity analysis of the profile, and d) the results of searching across the database with the signature. In the future, we plan to modify the website to allow more interactive analysis of the data in addition to allowing scientists to upload, analyze, and share their own gene expression data.

7.2 Sensitivity Analysis and Independent Validation

A sensitivity analysis is performed in order to prioritize the evaluation of the most promising group profiles. To do this, we randomly divide the group into two equal-sized subgroups: a training group that contains half of the treatment instances from the group and a test group composed of the remainder of the group. A group profile is created for both subgroups, and the top (up-tags) and bottom (down-tags) 100 probes are selected. The number of probes in common between the two subgroups is calculated for both the up- and down-tags respectively. The treatment instances are re-randomized and this process is repeated for a total of 10 iterations. The average number of probes in common across the 10 iterations is calculated for the up- and down-tags. The higher the average number of probes in common (for the up-tags, down-tags, or both up- and down-tags), the more robust we consider the group profile. From this value, i.e., the average number of probes in common, we estimate the probability assuming a binomial distribution.

The most robust ATC level 3 (therapeutic/pharmacological) group profiles are shown in Table 3 for both the up and down signatures together (full results in Supplemental Table 1 and Supplemental Table 2 for the up and down signatures, respectively). The full results for the level 4 ATC (chemical/therapeutic/pharmacological) group profiles for the up and down signatures are shown in Supplemental Table 3 and Supplemental Table 4, respectively. The associated probability for each of these profiles is also listed. The observed probabilities indicate that some of these profiles are not random. Corrections for multiple testing are performed, and the Bonferroni-corrected P values are also included in each of the tables.

At the onset of this paper, we mention that we are interested in creating a gene expression profile for groups sharing a therapeutic use, and so we focus our analysis on the ATC level 3 groups. There are 36 groups with significant (Bonferroni-corrected $P^* < 0.05$) up-expressed signatures and 28 for the down-expressed signatures. Out of these groups, 25 groups are robust for both up- and down-expressed signatures. While a robust up- or down-expressed signature can independently give novel insight into the underlying shared biological function of a group, we focus on groups that are significant for both because we also want to use these profiles to help predict novel uses of the drugs in our database. The similarity metric that we have adopted requires both the up and down signatures

Table 3. The most robust group profiles across the whole database are presented here

Group	Drugs	Up	P Up*	Down	P Down*	Label
N05A	28	70.6	3.09E-139	49.4	9.59E-86	Antipsychotics
R06A	27	23.7	1.07E-31	12	5.70E-12	Antihistamines for Systemic Use
N06A	25	29.6	5.70E-43	12.1	4.04E-12	Antidepressants
D07A	19	49.8	1.11E-86	19.7	1.64E-24	Corticosteroids, Plain
G01A	18	12.1	4.04E-12	6.5	1.98E-04	Antiinfectives and Antiseptics
D01A	16	10.2	2.42E-09	11.7	1.59E-11	Antifungals for Topical Use
S01B	16	7.9	3.36E-06	8.9	1.56E-07	Antiinflammatory Agents
N03A	11	13.1	1.22E-13	17.8	3.01E-21	Antiepileptics
H02A	11	18.5	1.94E-22	5.2	6.72E-03	Corticosteroids for Systemic Use
R03B	10	15.6	1.35E-17	4.6	3.09E-02	Drugs for Obstructive Airway Diseases, Inhalents
D10A	9	18.7	8.80E-23	6.5	1.98E-04	Anti-Acne Preparations (Topical)
L04A	8	39.9	2.46E-64	31.4	1.47E-46	Immunosuppressants
D07X	8	25.3	1.12E-34	6.7	1.13E-04	Corticosteroids, (Dermatologicals)
G03D	8	11.1	1.22E-10	4.7	2.41E-02	Progestogens
L01X	7	19.9	7.31E-25	11.5	3.16E-11	Other Antineoplastic Agents
L02B	6	19.3	8.12E-24	13.5	2.93E-14	Hormone Antagonists (and related)
R03A	6	9.8	8.89E-09	5.3	5.17E-03	Adrenergics, Inhalents
C08C	6	5.6	2.34E-03	4.5	3.95E-02	Selective Calcium Channel Blockers
G03C	5	30.5	9.35E-45	10.1	3.36E-09	Estrogens
S01C	5	10.3	1.74E-09	5.3	5.17E-03	Anti-inflammatory -infective (Combo)
C08E	4	11.7	1.59E-11	7.3	1.99E-05	Non-selective Calcium Channel Blockers
C01A	3	61.1	2.94E-114	62.2	4.60E-117	Cardiac Glycosides
L01D	3	6.4	2.62E-04	22.9	3.16E-30	Cytotoxic Antibiotics (and related)
L01B	3	7.9	3.36E-06	19.8	1.09E-24	Antimetabolites

to be used together. We now present a deeper analysis of the most robust profiles. The larger the set of unique drugs that compose a group, the more evidence we have that the therapeutic mechanism is what is being detected in the profile. For this reason, we focus on the significant groups with the largest number of unique drugs. We compare our results to those from an independent dataset using the same method (Table 4).

7.3 Antipsychotic Group (N05A)

We start our analysis with the largest group that meets our significance threshold: the antipsychotic group with 28 unique drugs. The ATC level 3 code for this group is N05A. The antipsychotic profile is the most robust result from the ATC level 3 groups when evaluating the up-expressed signature (Bonferroni-corrected P value: $P^*=3.10E-139$). This corresponds to an average of 70.6 probes that are shared between the top 100 probes of two random subgroups. Interestingly, this same group is the second most significant when evaluating the robustness of the down-expressed signature ($P^*=9.59E-86$; Average probes in common = 49.4). In an attempt to discover what the underlying shared biological process is within these antipsychotic agents, we turn to the L2L analysis. The most overrepresented GO Biological Process term is Sterol Biosynthetic Process (GO:0016126; $P^*=6.45E-20$). This is the same term that was found over-expressed within the smaller pilot study and demonstrates that our group profile method can detect the true signature with a small set of samples.

We have the ability to compare this profile with the antipsychotic profile recently published by Polymeropoulos et al. [25]. It is important to note that

these two profiles were created by two independent laboratories, with different cell lines and with a different, but overlapping, set of antipsychotics. These two profiles are very similar, and they share 34 probes in common among their top 100 probes ($P=6.42E-54$). The most significant GO Biological Process term from the Polymeropoulos et al. antipsychotic group profile is Lipid Biosynthesis. Given the significant overlap of the profiles, it is not surprising that this term is actually a grandparent of Sterol Biosynthetic Process (connected through the GO term Steroid Biosynthetic Process). The GO term Lipid Biosynthesis is also highly significant within the CMAP v2.0 antipsychotic group ($P^*=2.70E-13$).

The down-expressed signatures also share several probes in common (Probes=6; $P=6.79E-06$). The GO Biological Process analysis points to a significant down-regulation of the DNA regulation process (GO:0006260; $P^*=3.61E-07$). Barochovsky et al. have demonstrated in vivo that compounds acting on the central nervous system, specifically those that affect noradrenergic, dopaminergic, and serotonergic neurotransmitters, reduce brain cell replication [26]. This observation of compounds acting on the CNS was a dose-dependent effect and was seen for both agonists and antagonists. This down-expressed signature, like the up-expressed signature, is well supported by the literature. The antipsychotic profile that we have discovered is robust, both in and across datasets. Furthermore, we have demonstrated the ability of our group profile method to give biological insights into the potentially unknown shared biological process exhibited by a group of drugs.

Table 4. The most robust profiles were evaluated against an independent dataset (Polymeropoulos et al)

Group	Vanda PDR Group	Probes In Common	P
N05A	CNS:Antipsychotics	34	6.42E-54
R06A	Respiratory Agent:Histamine Antagonist	4	1.13E-03
N06A	CNS:Antidepressants	15	1.07E-18
D07A	Dermatological:Corticosteroids	30	7.88E-46

7.4 Antihistamine Group (R06A)

The second-largest group that meets our significance criteria is the antihistamines (full annotation: Antihistamines for Systemic Use; ATC Code: R06A). This group contains 27 unique drugs. The sensitivity analysis reveals 23.7 probes on average shared within the up-expressed signature and 12 for the down-expressed ($P^*=1.07E-31$ and $P^*=5.70E-12$, respectively). The up-expressed signature exhibits a common underlying theme related to negative regulation of I-kappaB kinase / NF-kappaB cascade (GO:0043124; $P=6.08E-05$). This GO signature is not as strong as some of the other profiles and is not significant when corrected for multiple testing. However, it is interesting to note that this signature is consistent with the known effect of antihistamines on NF-kappaB. Roumestan et al. have shown that antihistamines inhibit NF-kappaB through both H1 receptor-dependent and independent mechanisms [27]. This profile does

not replicate when compared to the equivalent group (Respiratory Agent: Histamine Antagonist) from the dataset presented by Polymeropoulos et al., though a similar trend is seen. The average number of probes in common is four and one respectively, for the up- and down-expressed signatures ($P = 1.13\text{E-}03$ and $P = 3.60\text{E-}01$).

7.5 Antidepressant Group (N06A)

Next, we discuss the third-largest group: the antidepressants (ATC Code: N06A). There are 25 unique drugs within this group. The sensitivity analysis results in an average of 29.6 and 12.1 probes in common for the up- and down-expressed signatures ($P^*=5.70\text{E-}43$ and $P^*=4.04\text{E-}12$, respectively). Evaluating the up-expressed signature, the most overrepresented GO Biological Process term is Sterol Biosynthetic Process (GO:0016126; $P^*=1.19\text{E-}09$). This is the same core mechanism seen within the antipsychotic group, but this signature is seen on a smaller scale. Polymeropoulos et al. demonstrated the same relationship between the expression profile of antipsychotic and antidepressant drugs [25]. When we compare our antidepressant profile to the antidepressant profile from the dataset from Polymeropoulos et al., we find 15 probes in common ($P=1.07\text{E-}18$). The down-expressed signature does not reproduce within the Polymeropoulos et. al. dataset, sharing only one probe in common.

7.6 Corticosteroid Group (D07A)

The last group that we evaluate in depth is the corticosteroids ($N=19$; ATC Code: D07A). This profile is also robust according to the sensitivity analysis. The average number of probes in common for the up-expressed signature is 49.8 ($P^*=1.11\text{E-}86$). The down-expressed signature has an average of 19.7 probes in common ($P^*=1.64\text{E-}24$). Individually, the up- and down-expressed signatures do not exhibit a significant result for any GO Biological Process, but evaluated together they demonstrate an effect on the regulation of the interleukin-6 biosynthetic process ($P^*=1.38\text{E-}02$). Corticosteroids are involved in a wide range of physiological systems such as stress response, immune response and regulation of inflammation. Interleukin-6 acts as both a pro-inflammatory and anti-inflammatory cytokine that can be secreted to stimulate response to trauma [28]. There is a significant overlap between this profile and the corresponding profile (Dermatological: Corticosteroids) from Polymeropoulos et al. The up-expressed signatures share 30 probes in common while the down-expressed share nine probes, corresponding to probabilities of $P=7.88\text{E-}46$ and $9.72\text{E-}10$, respectively.

8 Conclusions

We have introduced and evaluated our method for creating group profiles from gene expression data. The ability to have reproducible sets of differentially expressed genes from microarray experiments has been a big challenge, and we have demonstrated how our method is able to overcome this obstacle. Furthermore, we

have illustrated how to gain biological insight from such group profiles as well as the ability to use them as a signature to query a database. In our example domain of a drug discovery system, this biological insight allows researchers to potentially learn about the etiology of the disease that these compounds are being used to treat and gives them a predictive tool to find novel uses for other drugs.

Though a major focus of this work has been to introduce our method and validate it across independent datasets, we are also releasing all group profiles from the full CMAP 2.0. This includes all corresponding meta-analysis that has been performed: L2L analysis, similarity searching results, etc. We this resource contains of hidden biological insight into many groups of drugs and their target diseases, and for further in-depth research.

There are many possible avenues of further improvements and research. Thus far, we have assumed that explicit groups are given a priori. Our sensitivity analysis validates how coherent a group is; however, it does not dictate what to do if the outcome is not positive. For example, a leave-one-out analysis can be done to exclude members that do not fit well within a group. Lastly, it is important to note that our method is focused on determining a reproducible genetic profile for a group of samples; in this case, drugs of a particular class. We provide no guarantee as to the uniqueness of such profiles and instead claim that these profiles can be used to compare groups. We have kept the full ranked list as the profile, and so it is straightforward for extensions to this method to be developed to further refine and learn what genetic components make up a more unique signature if that was the end goal. In keeping the full profile, i.e., the re-ranked list of probesets, we allow further research methods, which are developed for individual expression profiles, e.g., the L2L method, to also be applicable to our group profiles.

References

1. Lander, E.S.: Array of hope. *Nat. Genet.* 21, 3–4 (1999)
2. Frantz, S.: An array of problems. *Nature Reviews Drug Discovery* 4, 362–363 (2005)
3. Marshall, E.: Getting the noise out of gene arrays. *Science* 306, 630–631 (2004)
4. Ein-Dor, L., Zuk, O., Domany, E.: Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proc. Natl. Acad. Sci. USA* 103, 5923–5928 (2006)
5. Zhang, W., Huang, R.S., Duan, S., Dolan, M.E.: Gene set enrichment analyses revealed differences in gene expression patterns between males and females. *Silico. Biol.* 9, 55–63 (2009)
6. van t Veer, L.J., Dai, H., van de Vijver, M.J., He, Y.D., et al.: Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415, 530–536 (2002)
7. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537 (1999)
8. DeRisi, J.L., Iyer, V.R., Brown, P.O.: Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* 278, 680–686 (1997)
9. Tan, P.K., Downey, T.J., Spitznagel, E.L., Xu, P., et al.: Evaluation of gene expression measurements from commercial microarray platforms. *Nucleic Acids Res.* 31, 5676–5684 (2003)

10. Ramalho-Santos, M., Yoon, S., Matsuzaki, Y., Mulligan, R.C., et al.: Stemness: transcriptional profiling of embryonic and adult stem cells. *Science* 298, 597–600 (2002)
11. Ivanova, N.B., Dimos, J.T., Schaniel, C., Hackney, J.A., et al.: A stem cell molecular signature. *Science* 298, 601–604 (2002)
12. Miller, R.M., Callahan, L.M., Casaceli, C., Chen, L., et al.: Dysregulation of gene expression in the 1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine-lesioned mouse substantia nigra. *J. Neurosci.* 24, 7445–7454 (2004)
13. Lamb, J., Crawford, E.D., Peck, D., Modell, J.W., et al.: The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313, 1929–1935 (2006)
14. Newman, J.C., Weiner, A.M.: L2L: a simple tool for discovering the hidden significance in microarray expression data. *Genome Biol.* 6, R81 (2005)
15. Gupta, M., Chauhan, C., Bhatnagar, P., Gupta, S., et al.: Genetic susceptibility to schizophrenia: role of dopaminergic pathway gene polymorphisms. *Pharmacogenomics* 10, 277–291 (2009)
16. Gerard, S.: Reviewing medications for bipolar disorder: understanding the mechanisms of action. *The Journal of Clinical Psychiatry* 70, e02 (2009)
17. Xiu, M., Hui, L., Dang, Y., Hou, T.D., et al.: Decreased serum BDNF levels in chronic institutionalized schizophrenia on long-term treatment with typical and atypical antipsychotics. *Prog. Neuropsychopharmacol. Biol. Psychiatry* 33, 1508–1512 (2009)
18. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., et al.: Gene ontology: tool for the unification of biology. *The Gene Ontology Consortium. Nat. Genet.* 25, 25–29 (2000)
19. Vastrik, I., DEustachio, P., Schmidt, E., Joshi-Tope, G., et al.: Reactome: a knowledge base of biologic pathways and processes. *Genome Biol.* 8, R39 (2007)
20. John, B., Enright, A.J., Aravin, A., Tuschl, T., et al.: Human MicroRNA targets. *PLoS Biol.* 2, e363 (2004)
21. Segal, E., Friedman, N., Koller, D., Regev, A.: A module map showing conditional activity of expression modules in cancer. *Nat. Genet.* 36, 1090–1098 (2004)
22. Newcomer, J.W., Sernyak, M.J.: Identifying metabolic risks with antipsychotics and monitoring and management strategies. *J. Clin. Psychiatry* 68, e17 (2007)
23. Lum, P.Y., Armour, C.D., Stepaniants, S.B., Cavet, G., et al.: Discovering modes of action for therapeutic compounds using a genome-wide screen of yeast heterozygotes. *Cell* 116, 121–137 (2004)
24. Donohoe, D.R., Aamodt, E.J., Osborn, E., Dwyer, D.S.: Antipsychotic drugs disrupt normal development in *Caenorhabditis elegans* via additional mechanisms besides dopamine and serotonin receptors. *Pharmacol. Res.* 54, 361–372 (2006)
25. Polymeropoulos, M.H., Licamele, L., Volpi, S., Mack, K., Mitkus, S.N., et al.: Common effect of antipsychotics on the biosynthesis and regulation of fatty acids and cholesterol supports a key role of lipid homeostasis in schizophrenia. *Schizophr Res.* 108, 134–142 (2009)
26. Barochovsky, O., Patel, A.J.: Effect of central nervous system acting drugs on brain cell replication in vitro. *Neurochem. Res.* 7, 1059–1074 (1982)
27. Roumestan, C., Henriquet, C., Gougat, C., Michel, A., et al.: Histamine H1-receptor antagonists inhibit nuclear factor-kappaB and activator protein-1 activities via H1-receptor-dependent and -independent mechanisms. *Clin. Exp. Allergy* 38, 947–956 (2008)
28. Heinrich, P.C., Behrmann, I., Haan, S., Hermanns, H.M., et al.: Principles of interleukin (IL)-6-type cytokine signalling and its regulation. *Biochem. J.* 374, 1–20 (2003)

Load Balancing Using Hybrid ACO – Random Walk Approach

Neha Bhatia, Rohan Kundra, Anurag Chaurasia, and Satish Chandra

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology
Waknaghat, Solan, Himachal Pradesh, India
{neha.juit07,urinkzone,anurag200788}@gmail.com,
satish.chandra@juit.ac.in

Abstract. Telecommunication and network systems have become more complex in recent years. Routing and optimal path finding are some of the important network problems. Traditional routing methods are not capable to satisfy new routing demands. Swarm intelligence is a relatively new approach to problem solving which provides a basis with which it explores problem solving without providing a global model. A Random Walk approach is similar to a drunkard moving along a sidewalk from one lamp post to another where each step is either backwards or forwards based on some probability. In this paper a hybrid algorithm is proposed that combines Ant Colony Optimization algorithm and Random Walk. The overall time complexity of the proposed model is compared with the existing approaches like distance vector routing and Link State Routing. The new method is found to be better than the existing routing methods in terms of complexity and consistency.

Keywords: Load Balancing, Ant Colony Optimization, Random Walk Routing, Traditional Routing, cover and access time.

1 Introduction

Telecommunication and network systems have become more complex in recent years and the network problems have also been increased. Some of the important network problems are routing and finding the optimal path. Several routing algorithms have been developed to improve routing and cope up with the problems related to networks. Vast complexity of networking problems such as load balancing, routing and congestion requires more efficient methods and techniques to solve these problems [1].

The traditional routing methods are not capable enough to satisfy new routing demands. Increase in number of users and network services force improvement in throughput to satisfy all of the services. This situation has led to development of new routing methods to increase efficiency. Such is the case, for example, of a routing method known as LBR (Load Balancing Routing) based on a load-balancing scheme. This method addresses routing by equally distributing load over all possible paths.

This diminishes the congestion probability in more inexpensive links, improving the overall network performance.

Swarm intelligence is a relatively new approach that solves problem with the help of social behaviour of insects and animals. In particular, ants are the one of the insects which have inspired a number of methods and techniques among which the most successful is the general purpose optimization technique known as ant colony optimization. Despite ACO being a successful algorithm but there are many improvements which can make it more efficient and less complex.

Yet another well known algorithm is Random Walk which is a successful method used for routing that uses random steps to move forward based on transition matrix and Markov chain. Random Walks arise in many models in Mathematics and Physics. It can be used to reach obscure parts of large sets, to generate random elements in large sets, etc. The aim of this work is to use ACO algorithm efficiently with reduced complexity. The proposed algorithm combines two famous algorithms of ACO and Random walk into a hybrid to be used in distributed systems.

2 Load Balancing

Load balancing is a technique to spread work between two or more computers, network links, CPUs, hard drives or other resources, in order to get optimal resource utilization, maximize throughput, and minimize response time. It is useful while dealing with multiple communication links. For example, a company having more than one internet connections always ensures access even if one of the connections fails. It means an arrangement is set that second should start its work as first fails. Instead of using only one connection at a time load balancing can help both the connections to work at same time by handling the load and prevent failure due to overload. A program can be designed to select between the connections and can be used. Generally there are multiple routes through the networks in all telecommunication companies. This may result in network congestion leading to the need of load balancing to shift traffic from one path to another and to improve network efficiency.

In such networks, there are number of nodes through which calls are routed. Load balancing is distributing the load over all the nodes and minimise lost calls. Shortest routes of calls can be determined in many possible ways.

3 ACO Algorithm

The ant colony optimization (ACO) algorithm is based on the behaviour of ants that control their movement on basis of corresponding pheromone level in their path. These ants deposit pheromone on the ground in order to leave their footsteps as favourable path that should be followed by other ants of the colony. Ants cooperate using an indirect form of communication through pheromone they deposit on the edges of the graph while moving [3]. Many Special Cases of ACO algorithm have been proposed. The three most successful being Ant System, Ant Colony System and

Max-Min Ant System. The scheduled activities and the general algorithm for ACO can be given as follows:

1.Initialization:

Set routing tables and initialize pheromone levels

2.Loop:

While destination is not reached

Construct Routing Tables

Optional Search

Update Pheromones

end while

3.1 Construct Routing Tables (Pheromone Tables)

In this case Routing tables are replaced by Pheromone tables which tell us the probability of choosing a node in form of pheromone strengths. An n- node network uses n different kinds of pheromones. The entries in the tables are probabilities which influence the ants' decision of choosing the next node on the way to their destination.

3.2 Optional Search

Some actions may be needed before updation of pheromones. Problem specific and/or centralized actions can be implemented by such actions, which cannot be performed by single ants. The most prevalent action in the application of optional search to the constructed solutions is the optimization of solutions which can further be used for updating.

3.3 Update Pheromones

This phase of algorithm is implemented to increase or decrease pheromone values based on the impact of the solution whether its good or bad. The process of updating pheromones is quite simple: when an ant arrives at a node, the probability (pheromone) corresponding to that node in the table is increased by following formula:

$$p_{new} = \frac{p_{old} + \Delta p}{1 + \Delta p} \quad (1)$$

Here p_{new} is the updated and increased probability and Δp is the increase in probability .For the evaporating ants the other entries in the table of this node are decreased according to:

$$P = \frac{p_{old}}{1 + \Delta p} \quad (2)$$

Since the new values sum up to 1, they can again be interpreted as probabilities.

4 Random Walk Algorithm

The random sequence of points which is a result of selecting a neighbour of a node at random , followed by moving to another selected node again at random is a random walk on the graph. It is actually a finite Markov chain [10] that is time reversible.

1.Initialization:

*Set parameters, calculate the transition probabilities
Construct Transition Matrix*

2. Loop:

*While termination conditions not met do
Move to neighbouring nodes based on Matrix
end while*

Consider a directed graph $G = (V,E)$, $|V| = n$, $|E| = m$. Each edge (i,j) of the graph has a weight $P(i,j) > 0$. $P(i,j)$ denotes the probability to reach j from i in one step. A Markov chain is a sequence of random variables $X_0, X_1, \dots, X_t \in \Omega$ that obey the "Markovian property", that is,

$$\Pr[X_{t+1} = y | X_0 = x_0, X_1 = x_1, \dots, X_t = x_t] = \Pr[X_{t+1} = y | X_t = x_t] \tag{3}$$

The transition matrix is used to show the transition probabilities $P(i,j)$ which is computed as follows for a random walk:-

$$P(i, j) = \begin{cases} \frac{1}{deg_{out}(i)} & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Suppose we have the following graph:

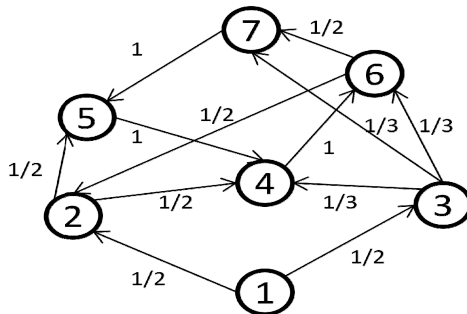


Fig. 1. Sample Graph

Then the transition probabilities and transition matrix is as follows:

Table 1. Transition matrix

		Destination nodes						
		1	2	3	4	5	6	7
Source nodes	1	0	1/2	1/2	0	0	0	0
	2	0	0	0	1/2	1/2	0	0
	3	0	0	0	1	0	1/3	1/3
	4	0	0	0	0	0	1	0
	5	0	0	0	1	0	0	0
	6	0	1/2	0	0	0	0	1/2
	7	0	0	0	0	1	0	0

Random Walk can be done as follows: (if we need to reach 7)

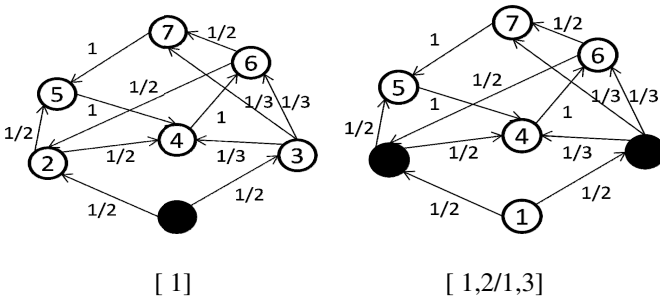


Fig. 2a Walk at t=0,1

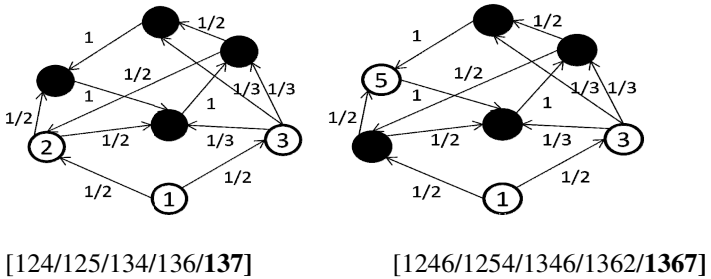


Fig. 2b Walk at t=2,3

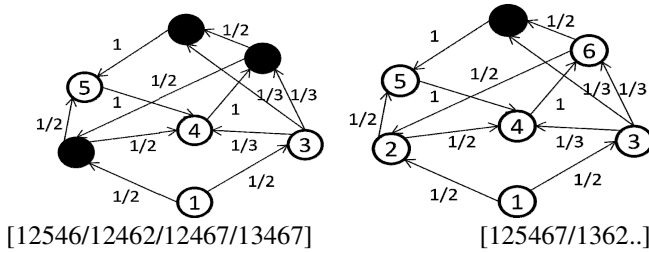


Fig. 2. Random Walk at time instants t=5,6

4.1 Access Time or Hitting Time

It is the expected number of steps before node j is visited, starting from node i. The sum

$$\kappa(i, j) = H(i, j) + H(j, i) \tag{5}$$

is called commute time: this is the expected number of steps in a random walk starting at i, before node j is visited and then node i is reached again. There is a way to express access time in terms of commute times.

$$H(i, j) = \frac{1}{2} (\kappa(i, j) + \sum_u \Pi(u) [\kappa(u, j) - \kappa(u, i)]) \tag{6}$$

4.2 Cover Time

The expected number of steps from each node to reach every other node is called the cover time.

$C_u(G) = E[\# \text{ steps to reach all nodes in } G \text{ on walk that starts at } u]$; and $C(G) = \max C_u(G)$:

Cover times for simple graphs:-

- $C(K_n) = \theta(n \log n)$, where K_n is the complete graph on n nodes that included self loops. The bound follows from the coupon collector.

- $C(L_n) = \theta(n^2)$, where L_n is the line graph on n nodes.

- $C(n\text{-node lollipop}) = \theta(n^3)$, where an n-node lollipop is a L_{n-2} with a K_2 at one of the ends.

It turns out that the $\theta(n^3)$ bound is the worst possible for cover time.

5 Proposed Algorithm

The Proposed Algorithm combines ACO and Random Walks in following way:

1.Initialization:

*Construct the pheromone tables and transition matrix
Take a pheromone level as threshold*

2.Loop:

*While ant doesn't reach the destination
If the probability to go to some node is less than threshold then use transition matrix probability to move else use pheromone table to move
Update pheromone tables
End while*

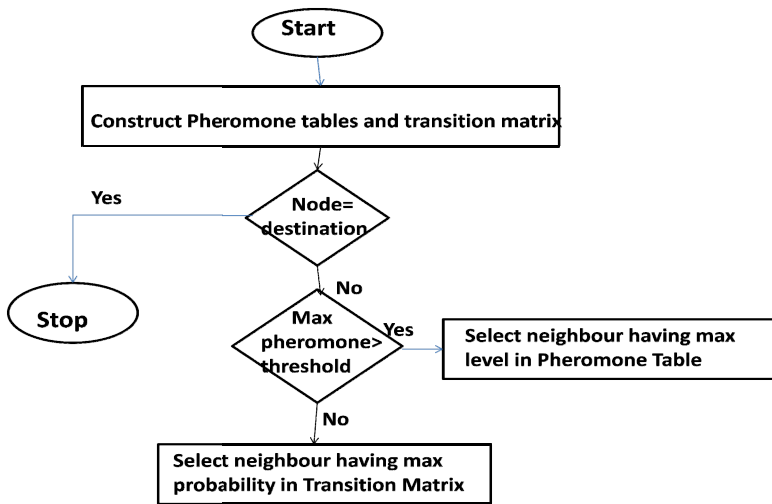


Fig. 3. Flowchart of proposed algorithm

5.1 Proposed Algorithm versus Traditional Routing

The Traditional Routing methods [1] algorithms such as the distance vector routing or RIP and link state routing or OSPF are discussed. Of particular interest are the issues of

- 1) Dependency on routing table
- 2) Moving routing table

1. Dependency on routing table

In both RIP and OSPF, a node N_i depends on the routing table created by interaction between all the neighbouring nodes. Further the neighbouring nodes of N_i depend on the routing information of their neighbouring nodes which in turn depend on other neighbouring nodes.

But in our proposed algorithm, the paths are independent of each other and can be chosen without interference and we can move from any source to a destination.

2. Moving routing table

Routing in RIP involves the overhead of moving routing tables of each node N_i to every one of its neighbours. For a large network N , the routing table consisting of costs between different nodes of the graph will be too large. It will increase the overhead to a very high extent due to large size of the routing table to be passed from one node to its neighbours at every step.

In OSPF, routing is achieved by flooding process that is passing a link-state-packet (LSP) to every other node in a network. Although an LSP carrying the cost information is smaller than routing table but the flooding process passes a copy of LSP to all the nodes in the network. Moving through different paths can lead to transmission of multiple identical copies of LSP to the same node.

Routing in Proposed algorithm involves transmission of ants rather than routing tables or by flooding LSPs.

6 Load Balancing in Telecommunication Network

The proposed algorithm can be used in telecommunication network for routing of calls. In addition to calls, the network also supports a population of simple mobile agents with behaviours modelled on the trail laying abilities of ants. The ants move across the network by randomly choosing its neighbours based on the pheromone level and the transition matrix.[13] The distribution of pheromones at each node help decide the path to be followed by any node. Calls between nodes are routed as a function of the pheromone distributions and random walk at each intermediate node.

To determine the route for a call from particular node to a destination, check the largest probability in pheromone table for this destination. If it is greater than threshold value then the neighbour node corresponding to this probability will be the next node on the route to this destination. else the neighbour is selected based on transition matrix. The route is valid if destination is reached and then the call is placed on the network.

In this way, calls and ants dynamically interact with each other. Newly arriving calls influence the load on nodes, which will influence the ants by means of delay mechanism. Ants influence the routes represented by pheromone table and transition matrix which in turn determine the routing of new calls.

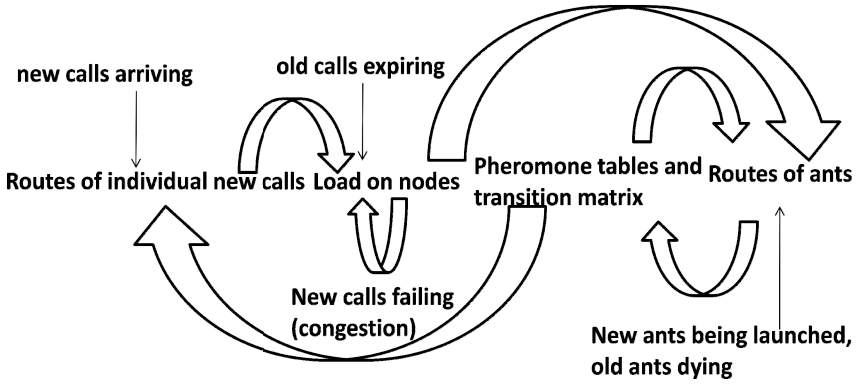


Fig. 4. Routing paths of the calls

7 Performance

The Proposed algorithm is better than traditional routing methods and has less complexity. It is described as follows using the concept of convergence time which is proved for the ACO before.[17].The convergence time is average iteration time to reach an optimal solution. The convergence time of the proposed algorithm is less than only applying ACO. Discussing about the convergence time of ACO alone as follows using the proved results:

$S(S, \Omega, f)$ is an optimization problem where S is the discrete solution space, Ω is the constraint set, $f: S \rightarrow R^+$ is the objective function, and R^+ is the positive real domain. The ACO algorithm finds the optimal solution s^* such that $f(s^*) \leq f(s)$ for $\forall s \in S$. T is the pheromone matrix, of which the element $\tau(i, j)$ is the pheromone trail value of edge (i, j) . E_γ is a measurement of the time complexity of ACO algorithms.

$q(i, t, s^*) = P(s_i = c_i^* | T, x_{i-1})$ is the crucial probability that an artificial ant selects c_i^* as the i th component, where $s_{i-1}, c_i^* \in S$ and s^* is the optimal solution ($i = 1, 2, 3, \dots, n$). Given K ants in an ACO algorithm, the probability that s^* is attained at the t th iteration time can be calculated according to the following:

$$\begin{aligned}
 p^t &= P\{\xi(t) \in Y^* | \xi(t-1) \notin Y^*\} \\
 &= 1 - (1 - P(t, s^*))^K \\
 &= (1 - (1 - \prod_{i=1}^n q(i, t, s^*)))^K
 \end{aligned}
 \tag{7}$$

where,

Y^* is the optimal state set if there is at least a solution s^*

$\xi(t)$ is the state of the stochastic process for the ACO algorithm per iteration.

The Time complexity can be calculated as [17]

$$E_\gamma \leq (1 - \lambda(0)) [1 - (1 - p_{low}^n)^K]^{-1}
 \tag{8}$$

is held if $q(i, t, s^*) \geq p_{low} > 0$ for $t = 1, 2, \dots$ and $i = (1, \dots, n)$ where s^* is the only optimal solution.

The Pheromone value is an important aspect of ACO algorithm so if we want to use it to calculate probability we can do as follows [17]:

$$P(s_{i+1} = c_j | T, x_i) = \begin{cases} \frac{\tau(x_i, x_j, t)}{\sum_{y \in C \wedge (x_i, y) \in J(x_i)} \tau(i, y, t)}, & \text{if } \langle x_i, x_j \rangle \in J(x_i) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Where

$J(x_i)$ is the set of feasible neighbour for s_i .

$C(i, t)$ is the pheromone rate of the path $\langle a_{i-1}^*, a_i^* \rangle \in s^*$

The Updation Of pheromones runs $K.n$ times per iteration if there are K artificial ants.

So we know how probability is dependent upon the pheromone status and in turn the convergence time. Generally the ACO algorithms took exponential time $\approx n^n$ when analyzed. The Probability is better if pheromone level is high. Taking the threshold prevents this exponential time to choose the node. Instead random walk can reduce the time to $n \log(n), n^2$ or at max n^3 and then further optimality depends upon the ACO algorithm to determine the optimal solution.

8 Summary

In this paper a new method based on hybrid of ACO and Random Walk was proposed. ACO is definitely better than the traditional methods used for routing and load balancing. The proposed algorithm is designed in a way to improve its performance further. The integration of Random Walk increases the probability of always finding the shortest path. Evaporation of Pheromone will decrease the level that can take ants to a wrong neighbor. That might result in choosing the wrong path as the probability of finding optimal solution depends on the pheromone level. At this instant, when pheromone level is less than the threshold, random walk is one of the optimal algorithms that can help decide the next neighbor and increase consistency. Regarding the complexity integration of random walk, we have seen it reduces the complexity as discussed in performance section earlier. So the proposed method is better than using traditional approaches.

References

- [1] Sim, K.M., Sun, W.H.: Ant colony optimization for routing and load-balancing: survey and new directions. IEEE Transactions on Systems Man, and Cybernetics, Part A 33(5), 560–572 (2003)

- [2] Dorigo, M., Gambardella, L.M.: Ant Colony System: A cooperative learning approach to the traveling salesman problem. *Journal of IEEE Transactions on Evolutionary Computation* (1997)
- [3] Randall, M., Lewis, A.: A Parallel Implementation of Ant Colony Optimization. *Journal of Parallel and Distributed Computing* 62(9), 1421–1432 (2002)
- [4] Dorigo, M., Birattari, M., Stutzle, T.: Ant Colony Optimization. *IEEE Computational Intelligence Magazine* (2009)
- [5] Mukherjee, D., Sriyank: Ant Colony Optimization Technique Applied in network Routing Problem. *International Journal of Computer Application* 15, Article 13 (2010)
- [6] Hung, K.-S., Su, S.-F., Lee, Z.-J.: Improving Ant Colony Optimization Algorithms for Solving Traveling Salesman Problems. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 11(4) (2007)
- [7] Liu, H., Li, P., Wen, Y.: Parallel Ant colony Optimization Algorithm. In: *Proceedings of the 6th World Congress on Intelligent Control and Automation, China* (2006)
- [8] Zhao, D., Luo, L., Zhang, K.: An Improved Ant Colony Optimization for communication network routing problem. In: *Bio-Inspired Computing, BIC-TA* (2009)
- [9] Burioni, R., Cassi, D.: Random walks on graphs: ideas, techniques and results. *Journal of Physics A: Mathematical and General* 38(8) (2005)
- [10] Lawrence, R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE* 77 (1989)
- [11] Aldous, D.: An introduction to covering problems for random walks on graphs. *Journal of Theoretical Probability* 2(1), 87–89 (1989)
- [12] Hildebrand, M.V.: A survey of results on random random walks on finite groups. *Probability Surveys* 2, 33–63 (2005)
- [13] Tian, H., Shen, H., Matsuzawa, T.: Random Walk Routing for Wireless Sensor networks. In: *Proceedings of the Sixth International Conference on Parallel Computing, Applications and Technologies* (2005)
- [14] Kim, S.-S., Smith, A.E., Hong, S.-J.: Dynamic Load Balancing Using Ant Colony Approach in Micro-cellular Mobile Communication Systems. In: *Advances in Metaheuristics for Hard Optimization*, pp. 137–152 (2008)
- [15] Schoonderwoerd, R., Holland, O., Bruten, J., Rothkrantz, L.: Ant-Based Load Balancing in Telecommunications Networks. *Adaptive Behaviour* 5(2), 169–207 (1996)
- [16] Blum, C.: Ant Colony Optimization: Introduction and Hybridizations. In: *7th International conference on Hybrid Intelligent Systems, HIS 2007*, pp. 24–29 (2007)
- [17] Gutjahr, W.J.: First steps to the runtime complexity analysis of ant colony optimization. *Computers and Operations Research* 35(9), 2711–2727 (2008)
- [18] Huang, H., Wu, C.-G., Hao, Z.-F.: A Pheromone-Rate-Based Analysis on the convergence Time of ACO Algorithm. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics* 39(4) (2009)

An Ant-Colony Algorithm to Transform Jobshops into Flowshops: A Case of Shortest-Common-Supersequence Stringology Problem

Suchithra Rajendran¹, Chandrasekharan Rajendran², and Hans Ziegler³

¹ Department of Industrial Engineering, College of Engineering, Guindy, Anna University, Chennai 600025, India

² Department of Management Studies, Indian Institute of Technology Madras, Chennai 600036, India

³ Department of Production and Logistics, University of Passau, 94032 Passau, Germany
snowy_410@yahoo.co.in, craj@iitm.ac.in, ziegler@uni-passau.de

Abstract. In this work we address the problem of transforming a jobshop layout into a flowshop layout with the objective of minimizing the length of the resulting flowline. This problem is a special case of the well-known classical Shortest Common Supersequence (SCS) stringology problem. In view of the problem being NP-hard, an ant-colony algorithm, called PACO-SFR, is proposed. A new scheme of forming an initial supersequence of machines (i.e., flowline) is derived from a permutation of jobs, followed by the reduction in the length of the flowline by using a concatenation of forward reduction and inverse reduction techniques, machine elimination technique and finally an adjacent pair-wise interchange of machines in the flowline. The proposed ant-colony algorithm's performance is relatively evaluated against the best known results from the existing methods by considering many benchmark jobshop scheduling problem instances.

Keywords: Jobshop, Flowshop, Shortest Common Supersequence, Ant-colony algorithm.

1 Introduction

A jobshop is a manufacturing system that has a process layout with machines capable of performing similar operations located together, while a flowshop is a manufacturing system that has a flowline-based (i.e., product-based) layout such that products or jobs move in the shop with a uni-directional flow in the order of their processes. In other words, the flow of all jobs through the shop for processing on machines is in the same forward direction with no back-tracking, but it is possible that a job may skip some machines for its processing in the flowline. According to Knolmayer et al. [1], a jobshop layout is a common configuration in many manufacturing systems, and the transformation of jobshops into flowshops is vital and relevant in the context of an efficient supply chain management. Kimms [2] observed that while transforming a jobshop layout into a flowshop, the key objective is to minimize the length of the

resultant flowshop because the minimization of the length of the flowshop serves to minimize the lead times of production of jobs in the resultant flowshop, thereby leading to reduced inventory levels.

The problem of minimizing the length of the resultant flowline is a special case of the well-known classical Shortest Common Supersequence (SCS) stringology problem [2-4]. Framinan [4] observed that even though the SCS problem and the problem of transforming a jobshop into a flowshop appear identical in terms of complexity and problem statement, they are not, in general, equivalent with respect to some of the assumptions in jobshop problems (e.g. in a jobshop it is assumed that the consecutive operations of a job are not performed on the same machine; mostly it is assumed that each job is processed on each machine only once and a job is processed on all machines, and the sequence of processing a job is independent of the sequence of processing another job). The general SCS problem is known to be NP-hard [5-6] and so is the problem under consideration [2]. Some attempts towards transforming a jobshop into a resultant flowshop are due to Kimms [2], Framinan and Ruiz-Usano [3] and Framinan [4, 7]. Framinan [4] made a thorough analysis of the existing algorithms for the SCS problem (e.g. genetic algorithms by Branke et al. [8]), ant colony algorithm by Michel and Middendorf [9] and beam search by Framinan and Ruiz-Usano [3] adapted to the jobshop-transformation problem under consideration and the proposed tabu search by considering seventy well-known jobshop problem instances, and reported the findings (see Table 6 of that paper) in order to be used as a benchmark for future researchers. To our knowledge, the work by Framinan [4] is the most exhaustive study till date.

In the present work we propose an ant-colony algorithm with some new features for transforming a jobshop into a flowshop. First we construct an initial supersequence of machines on the basis of a sequence of jobs and the associated machine ordering, followed by two concatenations of forward and inverse reduction procedures, machine elimination technique and finally a local search scheme involving an adjacent pair-wise interchange of machines in the flowline, applied twice. Our work employs an approach of obtaining the initial supersequence of machines that is different from the previous attempts (e.g. Framinan and Ruiz-Usano [3]; Framinan [4]) and it also differs by employing two concatenations of forward and inverse reduction techniques and finally the local search involving an adjacent pair-wise interchange of machines. As for the generation of sequences of jobs for obtaining the supersequences of machines, we employ the PACO, the ant-colony algorithm proposed by Rajendran and Ziegler [10] for the permutation flowshop scheduling problem. This ant-colony algorithm is found to be one of the best algorithms for permutation flowshop scheduling (see Ruiz et al. [11] and Ruiz and Stuetzle [12]). It is to be noted that our proposed ant-colony algorithm constructs a sequence of jobs from which a feasible supersequence of machines is generated, whereas the ant-colony algorithm by Michel and Middendorf [9] and the genetic algorithm by Branke et al. [8] construct a full feasible supersequence of machines. Hence we find our ant-colony algorithm computationally simple; moreover, our string reduction techniques are also computationally simple in the sense that every reduced supersequence is checked for feasibility with respect to the machine routing of every job and this check is computationally straightforward and simple.

2 Proposed Ant-Colony Algorithm Integrated with String Formation and Reduction Techniques (PACO-SFR)

The salient features of the proposed ant-colony algorithm integrated with the novel features of string formation and reduction techniques (called PACO-SFR) are now presented, followed by a detailed discussion of the algorithm. The PACO-SFR first generates a sequence or string of jobs by following the procedure related to the generation of an ant-sequence described in the PACO (see Rajendran and Ziegler [10]) that uses the pheromone intensity or trail matrix $[\tau_{ij}]$ and hence a supersequence of machines is formed; on this supersequence or flowline of machines, it then employs two concatenations of forward and inverse reduction procedures, a machine reduction technique, and finally a local search involving an adjacent pair-wise interchange of machines in the flowline. The resultant flowline thus obtained and its length constitute the solution corresponding to the ant-sequence or string of jobs generated. The PACO-SFR employs the job-index based insertion scheme (called JIS) as a local search scheme to generate improved job sequences. The job sequence thus obtained with respect to the resultant supersequence of machines with the minimum string or flowline length is used to update the pheromone intensity or trail matrix $[\tau_{ij}]$. This algorithm is carried out over 40 times or iterations, and the best job sequence thus obtained (in terms of the minimum flowline length) is returned.

We discuss the complete algorithm with numerical illustrations through which we explain the mechanism of the PACO-SFR. The following jobshop problem with $n = 6$ jobs and with $m = 4$ machines, and with the following sequence of machine-routings is considered throughout in this paper (note that the sequence of visits of jobs on machines is as per the order given below):

job 1	: 1-2-3-4
job 2	: 1-3-2-4
job 3	: 1-4-2-3
job 4	: 2-3-1-4
job 5	: 2-4-1-3
job 6	: 3-4-1-2

2.1 Generation of a Supersequence or String of Machines

In our study we obtain a supersequence of machines by ordering jobs in a specific order and thereafter laying out machines in that order. For example, if we order the jobs in the order $\{1-2-3-4-5-6\}$, the corresponding initial supersequence of machines or flowline is $\{1-2-3-4-1-3-2-4-1-4-2-3-2-3-1-4-2-4-1-3-3-4-1-2\}$ obtained by arranging machines for processing jobs in the given order $\{1-2-3-4-5-6\}$. This supersequence of machines is then reduced by employing the string reduction techniques presented in this study. Note that previous research attempts (e.g. those by Framinan and Ruiz-Usano [3]; Framinan, [4]) obtained the initial supersequence of machines or flowline $\{1-1-1-2-2-3-2-3-4-3-4-4-3-2-2-1-1-1-4-4-3-4-3-2\}$ by

arranging the machines of the first operations of jobs in the given order {1-2-3-4-5-6}, then the machines of the second operations of jobs in that order, and so on up to the machines of the last operations of jobs taken in the given order {1-2-3-4-5-6}. However, from our computational experiments, we found that our arrangement of initial supersequence of machines has resulted in yielding flowlines with less lengths than those obtained from the arrangement of initial supersequence of machines suggested by previous researchers.

As an example, consider a two-job two-machine SCS problem given by Framinan [4]:

job 1	: 1-1-1-2-2-2
job 2	: 2-2-2-1-1-1

When we follow our approach of forming the initial supersequence of machines with machines laid out as per the job order {1-2}, we have {1-1-1-2-2-2-2-2-2-1-1-1} and after the application of the forward reduction technique or the inverse reduction technique (see their details in the text to follow), we have the resultant supersequence {1-1-1-2-2-2-1-1-1}. However, when the approach by Framinan [4] is followed, we have the initial supersequence {1-2-1-2-1-2-2-1-2-1-2-1}, and after the application of the forward reduction technique, we have the supersequence {1-2-1-2-1-2-1-2-1-2-1} which is longer than that obtained from our approach.

2.2 Implementation of the Set of String Reduction Techniques on a Job Sequence

The initial job sequence in our PACO-SFR is obtained by ordering jobs as {1-2-3-...-n} and hence we form a supersequence of machines. We then employ the string reduction techniques, namely, two concatenations of forward and inverse reduction techniques, machine elimination technique and a local search involving an adjacent pair-wise interchange of machines on the job sequence to get a reduced supersequence of machines. We first present the forward reduction procedure (see Framinan [4]). Considering one job at a time, we scan the given supersequence of machines from the left to see what machines are required to process the chosen job and mark the machines accordingly. After all jobs are considered, the marked machines survive in the supersequence of machines. For the supersequence of machines or flowline {1-2-3-4-1-3-2-4-1-4-2-3-2-3-1-4-2-4-1-3-3-4-1-2}, we get the reduced supersequence or flowline {1-2-3-4-1-3-2-4-3}, after this forward reduction of scanning from the left to the right of the given supersequence and satisfying the machine-routings with respect to every job.

As for the inverse reduction (also see Framinan [4]), considering one job at a time, we scan the supersequence of machines from the right to the left to see what machines are required to process the chosen job in the reverse sequence of operations and mark the machines accordingly. After all jobs are considered, the marked machines survive in the supersequence of machines. For the supersequence of machines {1-2-3-4-1-3-2-4-1-4-2-3-2-3-1-4-2-4-1-3-3-4-1-2}, we get the reduced flowline {1-2-3-1-4-2-4-1-3-4-1-2}.

It is evident that the forward reduction procedure need not yield the same resultant reduced supersequence of machines as that from the application of the inverse reduction. For this reason, Framinan [4] applied the forward reduction on the initial supersequence, and once again considering the initial supersequence, Framinan applied the inverse reduction procedure. The better of the two resultant supersequences is chosen by Framinan. However, according to our proposal, we explore two concatenations: apply first the forward reduction procedure on the initial supersequence and then apply the inverse reduction on the resultant reduced supersequence (called concatenation (forward + inverse)) to possibly reduce the supersequence of machines; apply first the inverse reduction procedure on the initial supersequence and then apply the forward reduction on the resultant reduced supersequence (called concatenation (inverse + forward)) to reduce the supersequence of machines. Then we take the better of these two reduced supersequences. For example, when we apply the concatenation (forward + inverse) on the supersequence {1-2-3-4-1-3-2-4-1-4-2-3-2-3-1-4-2-4-1-3-3-4-1-2}, we get the resultant reduced supersequence {1-2-3-4-1-3-2-4-3}; when we apply the concatenation (inverse + forward) on the supersequence {1-2-3-4-1-3-2-4-1-4-2-3-2-3-1-4-2-4-1-3-3-4-1-2}, we get the resultant reduced supersequence {1-2-3-1-4-2-4-1-3-2}. The better of these two supersequences is chosen by us for possible further reduction, i.e., {1-2-3-4-1-3-2-4-3}.

It is therefore evident that the two concatenations of forward reduction and inverse reduction serve to reduce the length of the supersequence of machines than the application of only one reduction technique. As a further example, we can show the effectiveness of these two concatenations with the same numerical illustration with the machine routing of job 4 changed to (2-1-4) (instead of (2-3-1-4)) and that of job 6 changed to (4-1-2) from (3-4-1-2). The concatenation (forward + inverse) yields the resultant supersequence of machines {1-2-4-1-3-2-4-3} (with the forward reduction first yielding {1-2-3-4-1-3-2-4-3} and the inverse reduction thereafter yielding {1-2-4-1-3-2-4-3}) and the concatenation (inverse + forward) yields {1-3-4-2-4-1-3-4-2} (with the backward reduction first yielding {1-3-1-4-2-4-1-3-4-1-2} and the forward reduction thereafter yielding {1-3-4-2-4-1-3-4-2}). The two resultant supersequences obtained by the concatenation of the forward and backward reduction techniques are of less length than those yielded by the single application of either the forward reduction technique or the inverse reduction technique. Reverting to our original example, we consider two supersequences {1-2-3-4-1-3-2-4-3} and {1-2-3-1-4-2-4-1-3-2} (obtained from two concatenations of forward and inverse reduction techniques), and choose the supersequence that has less length.

Let us see how we can reduce its string length further. Now we employ the machine elimination technique (also attempted by Framinan and Ruiz-Usano [3]) on the supersequence {1-2-3-4-1-3-2-4-3}, called S . We remove only one machine at a time from S and see if the resultant supersequence is feasible with respect to satisfying the machine routing of every job; if so, the last such reduced (by one element) supersequence is chosen. As an example, suppose we remove machine 1 (found first in the supersequence) and we have the supersequence {2-3-4-1-3-2-4-3}. This supersequence does not satisfy the machine routings of all jobs. Then we remove

machine 2 found in the second position of S with no success. However if we remove machine 3 found in position 6 in S , we have the resultant supersequence $\{1-2-3-4-1-2-4-3\}$ and this supersequence satisfies all machine routings, thereby resulting in a reduced string length. We continue with this process of removing a machine from S until the last machine in S is considered for elimination. The last such reduced supersequence, if it exists, with the string length $|S|-1$ is chosen; otherwise S is retained. It is interesting to note that while the machine elimination technique serves to reduce the supersequence $\{1-2-3-4-1-3-2-4-3\}$, it does not reduce the supersequence $\{1-2-3-1-4-2-4-1-3-2\}$.

For the purpose of exploring a further reduction in string length, we employ a local search involving an adjacent interchange of machines in the supersequence, after the machine elimination technique. This local search works as follows. We swap the machines found in positions i and $i+1$, where $i = 1, 2, \dots, |S|-1$. Every such resultant supersequence is subjected to the concatenation (forward + inverse), and the best among the resultant feasible supersequences (feasible in terms of all jobs' machine routings being present in the supersequence) and S with the least string length is chosen. In the supersequence $\{1-2-3-1-4-2-4-1-3-2\}$, when we swap machines 1 and 4 found in adjacent positions, we have the resultant supersequence reduced to $\{1-2-3-4-1-2-4-3\}$ and this is chosen because it is feasible with respect to machine routings of all jobs being present in it and it has a less string length than the original supersequence. This local search of adjacent pairwise interchange of machines is implemented twice successively to possibly reduce the string length to the extent possible.

Thus we obtain a supersequence of machines or flowline with possibly minimum string length obtained from the given sequence of jobs, and this string reduction is achieved through two concatenations of forward and inverse reduction techniques, followed by a machine elimination technique and a local search involving two-time application of the adjacent pair-wise interchange of machines in a supersequence.

2.3 Improvement of a Job Sequence Using the JIS

The effectiveness or performance of a job sequence is measured in terms of the length of the resultant supersequence of machines. The initial job sequence $\{1-2-\dots-n\}$ is taken as the current seed sequence and subjected to the JIS three times successively for a possible improvement in its performance, and this sequence is taken as the seed sequence to the PACO-SFR to begin with. Let $[k]$ denote the index of the job in position k of the current seed sequence of jobs and let i refer to the index of jobs.

Do the following:

```

for  $i = 1(1) n$ :
{
for  $k = 1(1) n$ :
{
if  $[k] \neq i$ 

```

then

insert job i in position k of the current seed sequence and adjust the sequence accordingly by not changing the relative positions of other jobs; determine the resultant supersequence of machines for this job sequence, apply the proposed set of string reduction techniques on the supersequence of jobs and hence compute the performance of the job sequence in terms of the length of the reduced supersequence of machines.

}

choose the best sequence among the generated $(n-1)$ job sequences;

if the performance of this sequence (in terms of the length of the corresponding reduced supersequence of machines) is better than or equal to the performance of the current seed sequence, then the current seed sequence is replaced by the best sequence found above.

}

The current seed sequence finally returned by the three-time application of the JIS is in fact the possibly improved job sequence by the JIS in relation to the seed sequence to the JIS. This final sequence is the seed sequence to the PACO-SFR (called the best sequence of jobs as of now) and let the string length of this sequence be denoted by Z_{best} .

2.4 Initialization of Parameters in the PACO-SFR

We initialize the pheromone intensity or trail matrix as follows:

set $\tau_{ik} = (1/Z_{best})$,
 if (position of job i in the seed sequence to the PACO-SFR - $k|+1) \leq n/4$;
 $(1/(2 \times Z_{best}))$,
 if $n/4 < (\text{position of job } i \text{ in the seed sequence to the PACO-SFR} - k|+1) \leq n/2$;
 $(1/(4 \times Z_{best}))$, otherwise.

The rationale behind this setting of τ_{iks} is that the seed solution to the PACO-SFR being good, those positions that are close to the position of job i in the seed sequence should be associated with larger values of τ_{iks} than those that are away from the position of job i in the seed sequence. ρ is set to 0.75 in our study.

2.5 Construction of an Ant Sequence and Its Improvement by the JIS

In order to build a complete ant sequence of jobs, the following procedure is used to choose an unscheduled job i for position k , starting from a null sequence, for $k = 1, 2, \dots, n$.

Set $T_{ik} = \sum_{q=1}^k \tau_{iq}$ and sample a uniform random number u in the range $[0, 1]$.

If $u \leq 0.4$ then the first unscheduled job as present in the best sequence of jobs obtained so far is chosen;

else

if $u \leq 0.8$ then

among the set of the first five unscheduled jobs, as present in the best sequence of jobs obtained so far, choose the job with the maximum value of T_{ik} ;

else

job i is selected from the same set of five unscheduled jobs for position k as a result of sampling from the following probability distribution:

$$P_{ik} = \left(\frac{T_{ik}}{\sum_l T_{lk}} \right),$$

where job l belongs to the set of the first five unscheduled jobs, as present in the best sequence obtained so far (note that when there are less than five jobs unscheduled, then all such unscheduled jobs are considered).

A complete ant sequence of n jobs is constructed accordingly and thereafter the set of proposed string reduction techniques is applied for obtaining the reduced supersequence of machines corresponding to this ant sequence of jobs. This ant sequence of jobs is then subjected to the JIS three times and the final resultant sequence of jobs (called the current sequence) with the length of the corresponding reduced supersequence of machines denoted by Z_{current} . If this current sequence's Z_{current} is same as or better than Z_{best} , then set this sequence and Z_{current} as the best sequence and Z_{current} respectively.

2.6 Updating of Pheromone Trails or Intensities

In the PACO-SFR, updating of the trail intensities is based not only on the resultant sequence of jobs obtained after the three-time application of the JIS on the ant sequence, but also on the relative distance between a given position and the position of job i in the resultant sequence, and also related to the best sequence obtained so far. The trails are updated as follows.

Let h be the position of job i in the resultant sequence;

$$\text{set } (\tau_{ik})^{\text{updated}} = \rho \times (\tau_{ik})^{\text{old}} + (1/(\text{diff} \times Z_{\text{current}})), \text{ if } |h - k| \leq 1;$$

$$\rho \times (\tau_{ik})^{\text{old}}, \text{ otherwise,}$$

where $\text{diff} = (|\text{position of job } i \text{ in the best sequence obtained so far} - k| + 1)^{1/2}$. This differential setting is based on the premise that the jobs occupying positions in the current sequence closer to their respective positions in the best sequence obtained so far should get their corresponding trail intensities increased by larger values.

2.7 Termination Condition

The PACO-SFR is terminated after 40 iterations (i.e., after the generation of 40 ant-sequences followed by the three-time application of the JIS). We use the PACO-SFR to generate a total of about $123n^2$ job sequences.

3 Computational Evaluation of the PACO-SFR

We consider the seventy benchmark jobshop instances considered by Framinan [4] and execute our ant-colony algorithm to solve the jobshop transformation problem instances. Framinan had given the best string length achieved with respect to every problem instance as a result of the implementation of algorithms such as H2 and H3 due to Branke et al. [8], BS due to Framinan and Ruiz-Usano [3] and TS due to Framinan [4] (see Table 6 in Framinan [4]). It is to be noted that the length of the resultant flowshop for every jobshop problem instance, as reported by Framinan, is through a consolidation of all the mentioned algorithms and that Framinan had reported the CPU time requirements and not the number of transformations or job sequences enumerated in the process of obtaining the results reported in the paper before the final transformation was obtained from all algorithms considered. For the sake of standardizing the computational effort requirement independent of the computer, its operating system and the programming language, in our work we have noted the number of job sequences enumerated to get the best flowline-length so that future researchers would find it easy to relatively evaluate our work. Note that a job sequence leads to the generation of a supersequence of machines, followed by the application of string reduction techniques. The computational of string reduction techniques is quite small and is the same across all job sequences. We have also noted the final job sequence and the corresponding supersequence of machines obtained from the PACO-SFR for the sake of completely reporting our results for possible future reference for researchers, apart from noting the details regarding the supersequence of machines obtained from the initial job sequence improved by the three-time application of the JIS and the number of job sequences enumerated to obtain the best string length for every problem instance in the PACO-SFR. The results are presented in Table 1.

It is found that the proposed ant-colony algorithm yields better results than those reported by Framinan [4] for 32 jobshop problem instances (i.e., for 46% of the problem instances), the best known solutions for 32 problem instances (i.e., for 46% of the problem instances) and worse solutions only in six problem instances (i.e., in only 8% of the problem instances). It is also seen from the computational experiments that the initial supersequences of machines (obtained after the three-time application of the JIS on the job sequence $\{1-2-\dots-n\}$) are the same as the string lengths reported by the previous researchers in 25 problem instances, less in 8 problem instances than those reported so far and quite close in most problem instances, thereby demonstrating the effectiveness of the proposed string reduction techniques. The best solutions for the benchmark problem instances are shown in bold in Table 1.

4 Summary

In this work we have dealt with the problem of transforming jobshops into flowshops with the objective of minimizing the length of the flowshop. An ant-colony algorithm,

Table 1. Computational results

Jobshop problem instance	n	m	String length derived from the initial job sequence	Best string length obtained by the ant-colony algorithm	Best string length reported by Framinan [4]
La01	10	5	13	13	13
La02	10	5	12	12	12
La03	10	5	12	12	12
La04	10	5	13	13	13
La05	10	5	12	12	12
La06	15	5	14	14	14
La07	15	5	14	14	14
La08	15	5	13	13	13
La09	15	5	14	14	14
La10	15	5	14	14	14
La11	20	5	14	14	14
La12	20	5	14	14	14
La13	20	5	15	15	15
La14	20	5	15	15	15
La15	20	5	14	14	14
La16	10	10	35	33	35
La17	10	10	36	34	35
La18	10	10	37	36	38
La19	10	10	36	31	35
La20	10	10	36	34	35
La21	15	10	42	39	42
La22	15	10	44	39	41
La23	15	10	41	40	42
La24	15	10	42	40	43
La25	15	10	43	41	42
La26	20	10	47	44	43
La27	20	10	44	44	44
La28	20	10	49	45	46
La29	20	10	49	45	45
La30	20	10	47	44	45
La31	30	10	51	49	49
La32	30	10	51	49	49
La33	30	10	53	49	49
La34	30	10	52	50	49
La35	30	10	53	50	49
La36	15	15	83	78	79
La37	15	15	80	74	79
La38	15	15	83	77	80
La39	15	15	79	75	80
La40	15	15	81	75	78
Orb01	10	10	29	27	28
Orb02	10	10	33	32	34
Orb03	10	10	20	20	20
Orb04	10	10	34	33	34
Orb05	10	10	28	26	26

Table 1. (continued)

Orb06	10	10	29	27	28
Orb07	10	10	33	32	34
Orb08	10	10	20	20	20
Orb09	10	10	34	33	34
Orb10	10	10	28	26	26
swv01	20	10	29	29	29
swv02	20	10	29	27	28
swv03	20	10	28	27	28
swv04	20	10	30	29	29
swv05	20	10	29	29	30
swv06	20	15	62	59	61
swv07	20	15	63	57	59
swv08	20	15	59	56	60
swv09	20	15	62	56	59
swv10	20	15	60	57	57
swv11	50	10	34	32	32
swv12	50	10	34	33	33
swv13	50	10	34	32	32
swv14	50	10	33	32	33
swv15	50	10	34	33	33
swv16	50	10	59	54	52
swv17	50	10	58	53	54
swv18	50	10	58	56	54
swv19	50	10	59	55	53
swv20	50	10	58	55	55

called PACO-SFR, is proposed with the integration of string reduction techniques in the ant-colony algorithm. The performance of the ant-colony algorithm is relatively evaluated by considering the best reported work and with the consideration of benchmark jobshop problem instances. It is found that the proposed ant-colony algorithm obtains better solutions and the best known solutions in most problem instances.

Acknowledgments. The first author gratefully acknowledges the support from DAAD for her research stay in the University of Passau during May-July, 2010. The second author thanks Alexander von Humboldt Stiftung for the financial support and C R Chandrasekaran for his initial involvement in this work. The authors thank the three reviewers for their comments and suggestions to improve the earlier version of the paper.

References

1. Knolmayer, G., Mertens, P., Zeier, A.: Supply Chain Management based on SAP Systems. Springer, Berlin (2002)
2. Kimms, A.: Minimal investment budgets for flow line configuration. IIE Transactions 32, 287–298 (2000)

3. Framinan, J.M., Ruiz-Usano, R.: On transforming job-shops into flow-shops. *Production Planning and Control* 13, 166–174 (2002)
4. Framinan, J.M.: Efficient heuristic approaches to transform job shops into flow shops. *IIE Transactions* 37, 441–451 (2005)
5. Raiha, K.J., Ukkonen, E.: The shortest common supersequence problem over binary alphabet is NP-complete. *Theoretical Computer Science* 16, 187–198 (1981)
6. Timkovsky, V.G.: Complexity of common subsequences and supersequences problems and related problems. *Cybernetics* 25, 565–580 (1990)
7. Framinan, J.M.: An adaptive branch and bound approach for transforming job shops into flow shops. *Computers & Industrial Engineering* 52, 1–10 (2007)
8. Branke, J., Middendorf, M., Schneider, F.: Improved heuristics and a genetic algorithm for finding short supersequences. *OR Spektrum* 20, 39–46 (1998)
9. Michel, R., Middendorf, M.: An ACO algorithm for the shortest common supersequence problem. In: *New Ideas in Optimization*. McGraw-Hill, Maidenhead (1999)
10. Rajendran, C., Ziegler, H.: Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research* 155, 426–438 (2004)
11. Ruiz, R., Maroto, C., Alcaraz, J.: Two new robust genetic algorithms for the flowshop scheduling problem. *Omega* 34, 461–476 (2006)
12. Ruiz, R., Stuetzle, T.: A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operations Research* 177, 2033–2049 (2007)

On the Ambiguity and Complexity Measures of Insertion-Deletion Systems

Kamala Krithivasan¹, Lakshmanan Kuppusamy²,
Anand Mahendran², and Khalid M.²

¹ Department of Computer Science and Engineering,
IIT Madras,
Chennai-600 036, India
kamala@iitm.ac.in

² School of Computing Science and Engineering,
VIT University,
Vellore-632 014, India
{klakshma,manand,mkhalid}@vit.ac.in

Abstract. In DNA processing and RNA editing, gene insertion and deletion are considered as the basic operations. Based on the above evolutionary transformations, a computing model has been formulated in formal language theory known as *insertion-deletion* systems. In this paper we study about ambiguity and complexity measures of these systems. First, we define the various levels of ambiguity ($i = 0, 1, 2, 3, 4, 5$) for insertion-deletion systems. Next, we show that there are inherently i -ambiguous insertion-deletion languages which are j -unambiguous for the combinations $(i, j) \in \{(5, 4), (4, 2), (3, 1), (3, 2), (2, 1), (0, 1)\}$. Further, We prove an important result that the ambiguity problem of insertion-deletion system is undecidable. Finally, we define three new complexity measures $TLength-Con$, $TLength-Ins$, $TLength-Del$ for insertion-deletion systems and analyze the trade-off between the newly defined ambiguity levels and complexity measures.

Keywords: DNA processing, insertion-deletion systems, inherently ambiguous languages, unambiguous grammar, complexity measures.

1 Introduction

In the last few years, Natural Computing which includes biologically inspired computing is an area which is pursued with interest. It includes DNA computing, membrane computing and evolutionary computing among other topics. The developments which have taken place in DNA computing have inspired the definition and study of new theoretical models in formal language theory, *sticker systems*, *splicing systems*, *Watson-Crick automata*, *insertion-deletion systems* [2], [6] are some of the theoretical models inspired by the behaviour of DNA strands in biology. In [4] insertion operation was only considered and in [14] insertion-deletion systems were introduced. They have opened a new avenue in formal language theory as a new model for generating languages. Informally, the

insertion and deletion operations of an insertion-deletion system is defined as follows: If a string α is inserted between two parts w_1 and w_2 of a string w_1w_2 to get $w_1\alpha w_2$, we call the operation as insertion, whereas if a substring β is deleted from a string $w_1\beta w_2$ to get w_1w_2 , we call the operation as deletion.

Given an insertion-deletion system, the weight of the system is based on the maximal length of insertion, maximal length of the context used for insertion, maximal length of deletion, maximal length of the context used for deletion and they are (respectively) denoted as $(n, m; p, q)$. The total weight is defined as the sum of n, m, p, q . There have been many attempts to characterize the recursively enumerable languages (i.e., computational completeness) using insertion-deletion systems with less weights. In [13] the universality results were obtained with weight 5 (of the combinations $(1, 2; 1, 1)$, $(2, 1; 2, 0)$, $(1, 2; 2, 0)$). In [1] and [6], this result was improved with weight 4 (of the combinations $(1, 1; 1, 1)$ and $(1, 1; 2, 0)$ respectively).

Insertion-deletion operations have some relevances to some phenomena in human genetics. In the following Fig.1. we try to show how the insertion-deletion systems are applied in the field of genetics. Consider a single strand DNA sequence $S_1 = xuvyz$, where x, u, v, y, z are all strings. Add a single stranded DNA sequence $u'w'v'$ to the sequence $xuvyz$, where $u'v'$ are the Watson-Crick complements of the strings u, v and w' is the complement of some string w , see Fig.1(a). First, annealing will take place such that u' will stick to u and v' to v , thus we obtain the scenario as in Fig.1(b). Then a cut by a restriction enzyme to the double stranded DNA sequence w in Fig.1(c) and by adding a primer z' , we obtain a double stranded sequence as in Fig.1(d). Finally, by melting the double stranded sequence the two strands will be separated, hence we obtain two strings. One string will be of the form $S_2 = xuwvyz$ (as in Fig.1(e)). The string S_2 implies that the string w is inserted between u and v . Thus, the string S_2 obtained from S_1 shows the use of insertion operation in DNA sequences. A similar annealing can be theoretically performed for deletion operation.

Ambiguity is considered as one of the fundamental problems in formal language theory. A grammar is said to be ambiguous, if there exists more than one distinct derivation of the words in the generated language. As we have seen above that the insertion-deletion system can be applied theoretically in DNA processing, the ambiguity in DNA processing (which uses the insertion-deletion system) can happen in the following manner. Let W_1W_2 be a DNA strand and suppose we want to insert $W_3W_4W_5$ between W_1 and W_2 to obtain another DNA strand $W_1W_3W_4W_5W_2$. This can be done first by inserting W_3 between W_1 and W_2 , followed by inserting W_4 between W_3 and W_2 , followed by inserting W_5 between W_4 and W_2 . The other sequence would be first by inserting W_5 between W_1 and W_2 , followed by inserting W_4 between W_1 and W_5 , followed by inserting W_3 between W_1 and W_4 . This shows that ambiguity in gene sequences is also possible (i.e., starting from one sequence we are able to get another sequence in more than one way such that the intermediate sequences are different). This motivates us to define formally the ambiguity for insertion-deletion systems. Based on the components used for insertion-deletion system, different levels of ambiguity are

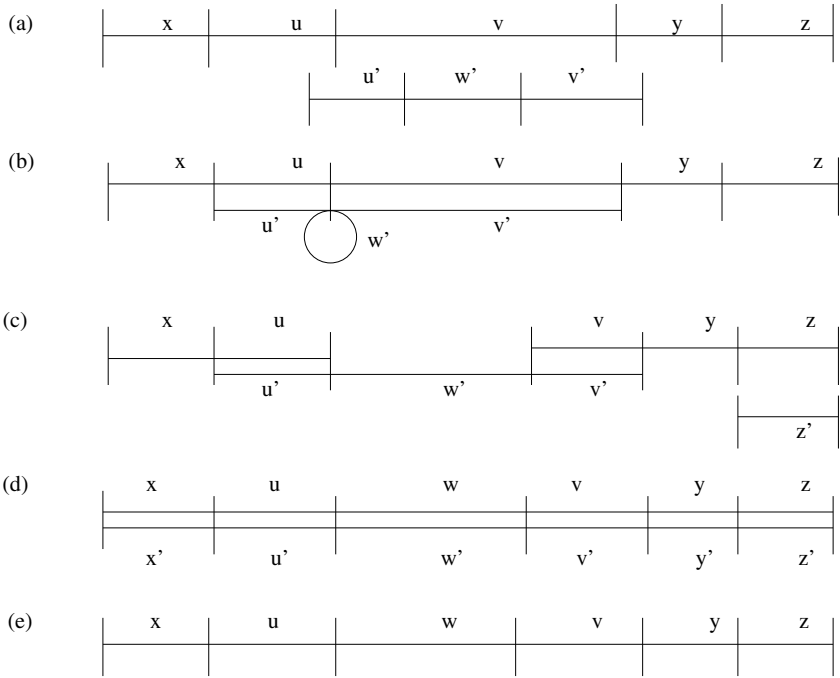


Fig. 1. Insertion by annealing

defined. Study of this concept of ambiguity may be useful in considering inheritance properties and phylogenetic trees [17]. More specifically, when these intermediate sequences are represented as phylogenetic trees, we can see that the trees are different and thus it might help us to identify the inheritance properties.

As insertion-deletion system is a counterpart for *contextual grammars* defined by S. Marcus [15] in 1969, here we briefly recall about the work done on ambiguity in contextual grammars. Unlike context-free grammars, defining ambiguity for contextual grammars is not so obvious since the derivation of contextual grammars consists of many components such as *axioms*, *contexts* and *selectors*. In [7], the notion of ambiguity was considered for the first time in this field, for external contextual grammars. Then, in [10,16], several levels of ambiguity were defined for internal contextual grammars by considering the components used in the derivation. There were many open problems formulated in [10,16] on different aspects of ambiguity and some of them have been solved in [11]. For more details on contextual grammars, we refer to [8].

As Fig.1. shows the applicability of insertion-deletion systems in gene sequences, storing of such sequences would be economical if the corresponding insertion-deletion system is economical. Such an economical system can be identified by means of descriptonal complexity measures. The basic measures for insertion systems are defined in [8]: $Ax, MAx, TAx, Prod, Symbol$. As the insertion-deletion systems is the slight modification of insertion systems, the

measures Ax , MAx , TAx , $Prod$ are even applicable to insertion-deletion systems. Here economical system means the corresponding system should be minimal with respect to some measure. Once the ambiguity and complexity measures are defined the trade-off between them can be analyzed by identifying languages whose corresponding grammars are ambiguous and minimal in measure M , but unambiguous and not minimal in measure M .

Since insertion-deletion systems is an intermediate model between contextual grammars and context sensitive grammars, analyzing the ambiguity and developing new complexity measures of such systems would be more interesting. In Section 3, we define the various ambiguity levels ($i = 0, 1, 2, 3, 4, 5$) for insertion-deletion systems. Next, we show that there are inherently i -ambiguous insertion-deletion languages which are j -unambiguous for the combinations $(i, j) \in \{(5, 4), (4, 2), (3, 1), (3, 2), (2, 1), (0, 1)\}$. Next, we discuss an example that shows how the ambiguity level defined for insertion-deletion systems can be interpreted in gene sequences. Further, we also prove that there is no solution (i.e., no algorithmic procedure) to decide whether a given arbitrary insertion-deletion system is ambiguous or not. In other words, the ambiguity problem for an insertion-deletion system is undecidable. Finally, in Section 4, we introduce three new complexity measures $TLength - Con$ (total length of contexts used in insertion/deletion rules), $TLength - Ins$ (total length of the contexts used in insertion rules plus the length of the strings to be inserted), $TLength - Del$ (total length of the contexts used in deletion rules plus the length of the strings to be deleted) and we analyze the trade-off between the newly defined ambiguity levels and the above complexity measures of insertion-deletion systems.

2 Preliminaries

We assume that the readers are familiar with the notions of formal language theory. However, we recall the basic notions which are used in the paper. A finite non-empty set V is called an alphabet. We denote by V^* , the free monoid generated by V , by λ it identity or the empty string, and by $V^+ - \{\lambda\}$. The elements of V^* are called *words* or *strings*. For any word $w \in V^*$, we denote the length of w by $|w|$.

Next, we will look into the basic definitions of insertion-deletion systems. Given an insertion-deletion (in short ins-del) system $\gamma = (V, T, A, R)$, where V is an alphabet, $T \subseteq V$, A is a finite language over V , R is a finite triples of the form $(u, \alpha/\beta, v)$, where $(u, v) \in V^*$, $(\alpha, \beta) \in (V^+ \times \{\lambda\}) \cup (\{\lambda\} \times V^+)$. The pair (u, v) is called as contexts. Insertion rule will be of the form $(u, \lambda/\alpha, v)$ which means that α is inserted between u and v . Deletion rule will be of the form $(u, \beta/\lambda, v)$, which means that β is deleted between u and v . In other words, $(u, \lambda/\alpha, v)$ corresponds to the rewriting rule $uv \rightarrow u\alpha v$, and $(u, \beta/\lambda, v)$ corresponds to the rewriting rule $u\beta v \rightarrow uv$.

Consequently, for $x, y \in V^*$ we can write $x \implies y$, if y can be obtained from x by using either an insertion rule or a deletion rule which is given as follows: (the down arrow \downarrow indicates the position where the string is inserted/deleted and the underlined string indicates the string inserted/deleted)

1. $x = x_1u^\lambda vx_2, y = x_1u\alpha vx_2$, for some $x_1, x_2 \in V^*$ and $(u, \lambda/\alpha, v) \in R$.
2. $x = x_1u\beta vx_2, y = x_1u^\lambda vx_2$, for some $x_1, x_2 \in V^*$ and $(u, \beta/\lambda, v) \in R$.

The language generated by γ is defined by

$$L(\gamma) = \{w \in T^* \mid x \implies^* w, \text{ for some } x \in A\}$$

where \implies^* is the reflexive and transitive closure of the relation \implies .

Next, we will introduce the various descriptonal complexity measures of ins-del systems. Given a ins-del system $\gamma = (V, T, A, R)$, the basic measures of ins-del systems are defined as follows:

$$Ax(\gamma) = \text{card}(A), MAx(\gamma) = \max_{w \in A} |w|, TAx(\gamma) = \sum_{w \in A} |w|,$$

$$Prod(\gamma) = \text{card}(R),$$

where Ax denotes the number of axioms, MAx denotes the maximum length of an axiom, TAx denotes total length of all axioms, $Prod$ denotes the number of insertion-deletion rules. For $M \in \{Ax, MAx, TAx, Prod\}$ and for a language L , we define $M(L) = \min\{M(\gamma) \mid L = L(\gamma)\}$. We call γ as a minimal system for L with respect to the measure M or we simply say γ is minimal in M for L . For a measure M and a language L , we define $M^{-1}(L) = \{\gamma \mid L(\gamma) = L \text{ and } M(\gamma) = M(L)\}$. Here, $M^{-1}(L)$ denotes the set of all optimal systems for L with respect to the measure M . Two measures M_1, M_2 are said to be *incompatible* if there exists a language L such that $M_1^{-1}(L) \cap M_2^{-1}(L) = \emptyset$. Otherwise, M_1 and M_2 are said to be *compatible*. For more details on complexity measures, we refer to [8].

3 Various Ambiguity Levels

In this section, we define various ambiguity levels for ins-del system based on the components used in the derivation.

Consider the following derivation step in a ins-del system $\gamma, \delta : w_1 \implies w_2 \implies \dots \implies w_m, m \geq 1$, such that $w_1 \in A$ and the following scenarios can happen (1) $w_j = x_{1,j}u_jv_jx_{2,j}$ and $w_{j+1} = x_{1,j}u_j\alpha_jv_jx_{2,j}$, when an insertion rule $(u_j, \lambda/\alpha_j, v_j)$ is used, where $x_{1,j}, u_j, v_j, x_{2,j} \in V^*$. (2) $w_j = x_{1,j}u_j\beta_jv_jx_{2,j}$ and $w_{j+1} = x_{1,j}u_jv_jx_{2,j}$, when a deletion rule $(u_j, \beta_j/\lambda, v_j)$ is used, where $x_{1,j}, u_j, v_j, x_{2,j} \in V^*$. The sequence which consists of used axiom, strings to be inserted/deleted is called as *Control Sequence* which is given as follows: $w_1, \alpha_1/\beta_1, \alpha_2/\beta_2, \alpha_3/\beta_3, \dots, \alpha_{m-1}/\beta_{m-1}$. The sequence which consists of used axiom, the string (α_j/β_j) to be inserted/deleted and the used contexts (u_j, v_j) is called *Complete Control Sequence* which is given as follows: $w_1, (u_1, \alpha_1/\beta_1, v_1), (u_2, \alpha_2/\beta_2, v_2), (u_3, \alpha_3/\beta_3, v_3), \dots, (u_{m-1}, \alpha_{m-1}/\beta_{m-1}, v_{m-1})$. The position where insertion (α) /deletion (β) takes place can be given by the *description* of δ , as follows: $w_1, x_{1,1}u_1(\alpha_1/\beta_1)v_1, x_{2,1}, x_{1,2}u_2(\alpha_2/\beta_2)v_2x_{2,2}, x_{1,3}u_3(\alpha_3/\beta_3)v_3x_{2,3}, \dots, x_{1,m-1}u_{m-1}(\alpha_{m-1}/\beta_{m-1})v_{m-1}x_{2,m-1}$.

- Definition 1.**
1. A *ins-del* system γ , is said to be 0-ambiguous, if there exist at least two different axioms, $w_1, w_2 \in A$, $w_1 \neq w_2$, such that they both derive the same word z , i.e., $w_1 \implies^+ z$, $w_2 \implies^+ z$.
 2. A *ins-del* system γ , is said to be 1-ambiguous, if there are two different unordered control sequences which derives the same word.
 3. A *ins-del* system γ , is said to be 2-ambiguous, if there are two different unordered complete control sequences which derives the same word.
 4. A *ins-del* system γ , is said to be 3-ambiguous, if there are two different ordered control sequences which derives the same word.
 5. A *ins-del* system γ , is said to be 4-ambiguous, if there are two different ordered complete control sequences which derives the same word.
 6. A *ins-del* system γ , is said to be 5-ambiguous, if there are two different descriptions which derives the same word.

More precisely, an *ins-del* system γ to be 2 or 4 ambiguous, it should have at least two distinct contexts and to be 1 or 3 ambiguous it should have at least two distinct strings used for insertion/deletion. A system which is not i -ambiguous, for some $i = 0, 1, 2, 3, 4, 5$, is said to be i -unambiguous. A language L is *inherently i -ambiguous* if every system γ generating L is i -ambiguous. A language for which a i -unambiguous system exists is called i -unambiguous. From the above definitions, it is easy to see that each inherently i -ambiguous language is inherently j -ambiguous for $(i, j) \in \{(1, 2), (1, 3), (2, 4), (3, 4), (4, 5)\}$. However, the converse is not true. Therefore, whenever we show that a language is inherently j -ambiguous, we want to make sure that the result is not followed by the fact that the language is inherently i -ambiguous for the above i and j . In fact, in the following theorems, we do the same.

3.1 Inherently Ambiguous Ins-del Languages

In this section, we show that there exist inherently i -ambiguous *ins-del* languages for $i = 0, 2, 3, 4, 5$.

Theorem 1. *There are inherently 5-ambiguous ins-del languages which are 4-unambiguous.*

Proof. Consider the language $L_1 = \{a^n b b a^m \mid n, m \geq 1\}$. The language L_1 can be generated by the following *ins-del* system γ_1 .

$$\gamma_1 = (\{a, b\}, \{a, b\}, \{abba\}, \{(a, \lambda/a, \lambda)\}).$$

As, the system γ_1 uses only string a for both insertion/deletion and only one context (a, λ) obviously, the system γ_1 is 4-unambiguous. Hence, the language L_1 is 4-unambiguous.

To prove the language L_1 is inherently 5-ambiguous, consider an arbitrary system γ'_1 which generates L_1 . The axiom in the system γ'_1 should be of the form $a^r b b a^s$, $r, s \geq 1$. The contexts used in insertion/deletion rule could be in one of the forms (a^{r_1}, b) , (a^{r_1}, bb) , (a^{r_1}, a^{r_2}) , (a^{r_1}, λ) , (λ, a^{r_2}) , (b, a^{r_2}) , (bb, a^{r_2}) , $r_1, r_2 \geq 1$. The string which will be inserted/deleted should be of the form a^p , $p \geq 1$. To

prove the system γ'_1 is ambiguous, consider a word $a^k b b a^l \in L_1$. From this word, the word $a^{k'} b b a^{l'}$ (for large values of k' and l') can be obtained by two different descriptions which differs by the position where the string a is inserted/deleted. First by getting $a^{k'}$ from a^k and then $a^{l'}$ from a^l or the other way round. Hence, the system γ'_1 is ambiguous. Any ins-del system generating this language will have this property and hence it is ambiguous. Therefore, the language L_1 is inherently 5-ambiguous. \square

Theorem 2. *There are inherently 4-ambiguous ins-del languages, which are 2-unambiguous.*

Proof. Consider the language $L_2 = \{a^n b^n c^m d^m \mid n, m \geq 1\}$. The language L_2 can be generated by the following ins-del system γ_2 .

$$\gamma_2 = (\{a, b, c, d\}, \{a, b, c, d\}, \{abcd\}, \{(a, \lambda/ab, b), (c, \lambda/cd, d)\}).$$

From the system γ_2 it is easy to see that both the contexts (a, b) and (c, d) are required to the generate the language. In order to generate equal number of a 's and b 's in the language L_2 the system has to use the context (a, b) . Similarly, to generate equal number of c 's and d 's in the language L_2 the system has to use the context (c, d) . Since no other alternate contexts are available in the system γ_2 to generate equal number of a 's and b 's and equal number of c 's and d 's the system γ_2 is 2-unambiguous. Therefore, the language L_2 is 2-unambiguous.

To prove the language L_2 is inherently 4-ambiguous, consider an arbitrary system γ'_2 which generates L_2 . The system γ'_2 should have the axiom of the form $a^i b^j c^j d^i, i, j \geq 1$. In order to generate the strings of the form $a^n b^n c d, n \geq 1$, the system should have the context of the form $(a^{r_1}, b^{r_2}), r_1, r_2 \geq 1$ and the string to be inserted/deleted should be of the form $a^{t_1} b^{t_1}, t_1 \geq 1$. Similarly, to generate the strings of the form $abc^m d^m, m \geq 1$, the system should have the context of the form $(c^{s_1}, d^{s_2}), s_1, s_2 \geq 1$ and the string to be inserted/deleted should be of the form $c^{u_1} d^{u_1}, u_1 \geq 1$. To prove the system γ'_2 is ambiguous, consider a word $a^{r_1} b^{r_1} c^{s_1} d^{s_1}, r_1, s_1 \geq 1 \in L_2$. From this word, the word $a^p b^p c^q d^q$ can be obtained by two different ordered complete control sequences. In one sequence the required number of a 's and b 's can be inserted/deleted by using the context of the form (a^{r_1}, b^{r_2}) , followed by the insertion/deletion of c 's and d 's by using the context of the form (c^{s_1}, d^{s_2}) . In another sequence, first the required number of c 's and d 's can be inserted/deleted by using the context of the form (c^{s_1}, d^{s_2}) , followed by the insertion/deletion of a 's and b 's by using the context of the form (a^{r_1}, b^{r_2}) . Hence the system γ'_2 is ambiguous. Any ins-del system generating this language can have at least two different complete control sequences. Therefore, the language L_2 is inherently 4-ambiguous. \square

Theorem 3. *There are inherently 3-ambiguous ins-del languages which are 1 and 2 unambiguous.*

Proof. Consider a language $L_3 = \{ca^n b^m d \mid n, m \geq 1\}$. The language L_3 can be generated by the following ins-del system γ_3 .

$$\gamma_3 = (\{a, b, c, d\}, \{a, b, c, d\}, \{cabd\}, \{(a, \lambda/a, \lambda), (b, \lambda/b, \lambda)\}).$$

From the system γ_3 , it is easy to see that both the contexts and strings are required to generate the language. In order to generate the language L_3 , the context (a, λ) and the string a has to be used $n - 1$ times and similarly the context (b, λ) and the string b has to be used $m - 1$ times. Therefore, the system γ_3 is 1 and 2-unambiguous. Hence, the language L_3 is 1 and 2-unambiguous.

To prove the language L_3 is inherently 3-ambiguous, consider an arbitrary system γ'_3 which generates L_3 . The system γ'_3 should have an axiom of the form $ca^i b^j d, i, j \geq 1$. In order to generate L_3 , insertion rules should be of the form, $(c^i, \lambda/a^j, \lambda), (a^i, \lambda/a^j, \lambda), (a^i, \lambda/a^j, a^k), (c^i, \lambda/a^j, a^k), (\lambda, \lambda/a^j, a^k), (a^i, \lambda/a^j, b^k), (b^i, \lambda/b^j, b^k), (b^i, \lambda/b^j, \lambda), (\lambda, \lambda/b^j, b^k), (a^i, \lambda/b^j, b^k), (\lambda, \lambda/b^j, d^k), (b^i, \lambda/b^j, d^k)$, where $i, j, k \geq 1$. Similarly, the system should have deletion rules of the form, $(c^i, a^j/\lambda, \lambda), (a^i, a^j/\lambda, \lambda), (a^i, a^j/\lambda, a^k), (c^i, a^j/\lambda, a^k), (\lambda, a^j/\lambda, a^k), (a^i, a^j/\lambda, b^k), (b^i, b^j/\lambda, b^k), (b^i, b^j/\lambda, \lambda), (\lambda, b^j/\lambda, b^k), (a^i, b^j/\lambda, b^k), (\lambda, b^j/\lambda, d^k), (b^i, b^j/\lambda, d^k)$, where $i, j, k \geq 1$. The string used for the insertion/deletion should be of the form $a^i, i \geq 1, b^j, j \geq 1$. From this, we can derive two different ordered control sequences which derives the same word. Consider an arbitrary word $ca^i b^j d \in L_3$. From this word, the word $ca^t b^s d$ can be derived in two different ordered control sequences. In one sequence, the required number of a 's can be inserted/deleted, followed by the insertion/deletion of required number of b 's. In another sequence, first the required number of b 's can be inserted/deleted, followed by the insertion/deletion of required number of a 's. Therefore, the system γ'_3 is ambiguous. Any ins-del system generating L_3 will have this property. Hence, the language L_3 is inherently 3-ambiguous. \square

Theorem 4. *There are inherently 2-ambiguous ins-del languages which are 1-unambiguous.*

Proof. Consider the language $L_4 = \{ba^n \mid n \geq 0\} \cup \{a^n c \mid n \geq 0\} \cup \{ba^n c \mid n \geq 0\}$. The language L_4 can be generated by the following ins-del system γ_4 .

$$\gamma_4 = (\{a, b, c\}, \{a, b, c\}, \{b, c, bc\}, \{(b, \lambda/a, \lambda), (\lambda, \lambda/a, c)\}).$$

As, the system γ_4 is having only one string which is used for insertion, the system γ_4 is 1-unambiguous. Therefore, the language L_4 is 1-unambiguous.

To prove the language L_4 is inherently 2-ambiguous, consider an arbitrary system γ'_4 which generates L_4 . The system γ'_4 must have three axioms of the form $ba^i, a^j c, ba^k c, i, j, k \geq 0$. If the system γ'_4 is having less than three axioms, it will generate strings not in the language. In order to generate the first part of the language the system must have an insertion rule of the form $(b, \lambda/a^i, \lambda), i \geq 1$. Similarly, the deletion rule must be of the form $(b, a^i/\lambda, \lambda), i \geq 1$. To generate the second part of the language the system must have an insertion rule of the form $(\lambda, \lambda/a^j, c), j \geq 1$. Similarly, the deletion rule must be of the form $(\lambda, a^j/\lambda, c), j \geq 1$. The string to be inserted/deleted should be of the form $a^p, p \geq 1$. To prove the system γ'_4 is ambiguous, consider a word $ba^l c \in L_4$, where $l = k + ij$. This word can be derived from the axiom $ba^k c$ by two different unordered complete control sequences. In one sequence the context (b, λ) can be used j times and in another sequence the context (λ, c) can be used i times. Thus obtaining two different

unordered complete control sequences. Hence, the system γ'_4 is 2-ambiguous. Any ins-del system generating the language L_4 will have this property. Therefore, the language L_4 is inherently 2-ambiguous. \square

Theorem 5. *There are inherently 0-ambiguous ins-del languages without context, which are 1-unambiguous with finite context.*

Proof. Consider the language $L_5 = \{a, b\}^+$. The language L_5 can be generated by the following ins-del system γ_5 .

$$\gamma_5 = (\{a, b\}, \{a, b\}, \{a, b\}, \{(a, \lambda/a, \lambda), (a, \lambda/b, \lambda), (b, \lambda/b, \lambda), (b, \lambda/a, \lambda)\}).$$

Since, the string a or b is inserted only to the right of the axiom by using the contexts (a, λ) or (b, λ) , any word in the language can be generated in a unique manner. Hence, the system γ_5 is 1-unambiguous. Therefore, the language L_5 is 1-unambiguous. Moreover, any word in L_5 can be generated from only one axiom, therefore γ_5 is 0-unambiguous.

To prove the language L_5 is inherently 0-ambiguous if no contexts is used, consider an arbitrary system γ'_5 which generates L_5 . The system must have two axioms of the form $a^{r_1}, r_1 \geq 1$ and $b^{r_2}, r_2 \geq 1$. The string to be inserted/deleted should be of the form a^{p_1} and $b^{p_2}, p_1, p_2 \geq 1$. To prove γ'_5 is ambiguous, consider a word $b^i a^j b^k \in L_5$ for large values of i, j, k . This word can be derived from two axioms a^{r_1} and b^{r_2} as follows:

$$\begin{aligned} a^{r_1} &\xrightarrow{*}_{ins} b^i a^{r_1} \xrightarrow{*}_{del} b^i a^j \xrightarrow{*}_{ins} b^i a^j b^{s_1} \xrightarrow{*}_{del} b^i a^j b^k \\ b^{r_2} &\xrightarrow{*}_{ins} b^{r_2} a^j \xrightarrow{*}_{del} b^{r_2} a^j b^{s_2} \xrightarrow{*}_{ins} b^{r_2} a^j b^k \xrightarrow{*}_{del} b^i a^j b^k \end{aligned}$$

Since the system is without contexts, insertion/deletion can happen at any place. Any system without contexts which generates the language L_5 will have this property and hence it is ambiguous. Therefore, the language L_5 is inherently 0-ambiguous if no contexts is considered. \square

3.2 Ambiguity Issues in Gene Sequences

In this subsection, we show an example for how the level 5-ambiguity discussed for ins-del systems can be interpreted in gene sequences. Consider an *orthodox string* available in gene sequences. A string w over a complementary alphabet Σ is called orthodox iff it is (i) the empty string ϵ , or (ii) the result of inserting two adjacent complementary element $b\bar{b}$, for some $b \in \Sigma$, anywhere in an orthodox string [3]. A language is orthodox iff it contains only orthodox strings. The orthodox language L_{od} can be generated by the ins-del system $\gamma_{od} = (\{b, \bar{b}\}, \{b, \bar{b}\}, \{\lambda\}, R)$, where $b \in \{a, t, g, c\}$, \bar{b} is complement of b (i.e $\bar{a} = t, \bar{g} = c, \bar{t} = a, \bar{c} = g$) and R is given as $R = [(\lambda, \lambda/b\bar{b}, \lambda)]$. Consider the following string in orthodox language $gctagcat$. This string can be derived in two different descriptions by γ_{od} . The two different descriptions are given as follows:

$$\begin{aligned} \text{Description 1 : } &\downarrow ta \implies \underline{gcta}^\downarrow \implies \underline{gctagc}^\downarrow \implies \underline{gctagcat} \\ \text{Description 2 : } &ta^\downarrow \implies \downarrow \underline{tagc} \implies \underline{gctagc}^\downarrow \implies \underline{gctagcat} \end{aligned}$$

Note that the axiom, order of insertion of strings, order of contexts (here (λ, λ)) all are same in both derivations, but the position of insertion is different in each derivation. Therefore, the grammar γ_{od} is 5-ambiguous. Thus, starting from same (gene) sequence, we are able to get the same sequence, but the inter-mediate gene sequences are different. This suggests that there may be more than one way that a gene sequence can be processed.

3.3 Decidability of Ambiguity for Ins-del Systems

First, we will slightly brief about Post Correspondence Problem(PCP). Let Σ be an alphabet set containing at least two symbols. An instance of PCP has two ordered sets of strings $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ over Σ , we say that this instance of PCP has a solution if there is a sequence i_1, \dots, i_m , $m \geq 1$, with $1 \leq i_j \leq n$ for each $1 \leq j \leq m$, such that $x_{i_1} \dots x_{i_m} = y_{i_1} \dots y_{i_m}$. It has been proved that the PCP problem is not decidable in the sense that there cannot exist an algorithm which will take an arbitrary instance of PCP as input and say whether this instance of PCP has a solution or not.

Decidability is one of the basic questions to be answered in formal language theory. *Emptiness, Finiteness* are some of the examples for decidability problems. For more details on decidability problems, we refer to [9]. Once the ambiguity is defined for the ins-del system, one question naturally arises on the ambiguity of ins-del systems: *Does there exist an algorithm to decide whether a given arbitrary ins-del system γ is ambiguous or not?* In the next theorem, we prove that ambiguity problem of ins-del systems is undecidable by using the Post Correspondence Problem (this is called reducing PCP to the ambiguity problem).

Theorem 6. *The ambiguity (of any i , $i = 0, 1, 2, 3, 4, 5$) of ins-del systems is undecidable.*

Proof. Let $\Sigma = \{a', b'\}$. Consider two arbitrary words (x_1, \dots, x_n) and (y_1, \dots, y_n) over $\Sigma = \{a', b'\}$. Construct an ins-del system γ .

$$\gamma = (\{a, a', b, b', c, d, e\}, \{a, a', b, b', c, d, e\}, \{cccccecd, ccdcccd\}, I)$$

where the insertion rules (I) is given as follows:

$$\begin{aligned} I_1 &= (cc, \lambda/d, cce) \\ I_2 &= (cccccec, \lambda/x_{i_1} a^{i_1} b, d) \\ I_3 &= (x_{i_k}, \lambda/x_{i_{k+1}} a^{i_{k+1}} b, a^{i_k} b), \quad 1 \leq i_k \leq n \\ I_4 &= (dcc, \lambda/e, c) \\ I_5 &= (ccdccc, \lambda/y_{j_1} a^{j_1} b, d) \\ I_6 &= (y_{j_l}, \lambda/y_{j_{l+1}} a^{j_{l+1}} b, a^{j_l} b), \quad 1 \leq j_l \leq n \end{aligned}$$

Let w_1 be the word derived from the axiom $cccccecd$ (The \downarrow denotes the position where the string is inserted in the next derivation step and the number

at the suffix in each derivation symbol ‘ \implies ’ indicates which insertion rule is applied).

$$\begin{aligned} ccccec^\downarrow d &\implies_{I_2} ccccec x_{i_1}^\downarrow a^{i_1} bd \implies_{I_3}^* cc^\downarrow cccex_{i_1} \dots x_{i_k} a^{i_k} b \dots a^{i_1} bd \\ &\implies_{I_1} ccdccex_{i_1} \dots x_{i_k} a^{i_k} b \dots a^{i_1} bd. \end{aligned}$$

Let w_2 be the word derived from the axiom $ccdcccd$.

$$\begin{aligned} ccdccc^\downarrow d &\implies_{I_5} ccdcccy_{j_1}^\downarrow a^{j_1} bd \implies_{I_6}^* ccdcc^\downarrow cy_{j_1} \dots y_{j_l} a^{j_l} b \dots a^{j_1} bd \\ &\implies_{I_1} ccdcccecy_{j_1} \dots y_{j_l} a^{j_l} b \dots a^{j_1} bd. \end{aligned}$$

By comparing w_1 and w_2 , they are both equal iff $k = l$ and $i_1 = j_1, \dots, i_k = j_k$ and $x_{i_1} \dots x_{i_k} = y_{i_1} \dots y_{i_k}$. So, there exists a Post Correspondence Solution (PCP) for the instance i_1, \dots, i_k for the strings (x_1, \dots, x_n) and (y_1, \dots, y_n) . We can see that, the system is ambiguous as it derives the same word from two different axioms.

Conversely, if PCP for (x_1, \dots, x_n) and (y_1, \dots, y_n) has a solution for the instance i_1, \dots, i_k , such that $x_{i_1} \dots x_{i_k} = y_{i_1} \dots y_{i_k}$, then clearly the system γ is 0-ambiguous (i.e., from two different axioms, we are able to get two words w_1 and w_2 , such that $w_1 = w_2$).

Therefore, if the ambiguity problem of ins-del systems is decidable, then PCP is said to be decidable by the above reduction method which is not the case. Therefore, 0-ambiguity problem for ins-del system is not decidable. In a similar fashion, we see that i -ambiguity problem ($i = 1, 2, 3, 4, 5$) for ins-del system is undecidable. □

4 New Complexity Measures

In this section, we define some new complexity measures for ins-del systems. Given a ins-del system $\gamma = (V, T, A, R)$, the basic complexity measures have been discussed in the preliminary section. Let the system γ contains the insertion rules of the form $(u_1, \lambda/\alpha_1, v_1), \dots, (u_m, \lambda/\alpha_m, v_m)$ and similarly the deletion rules of the form $(u_1, \beta_1/\lambda, v_1), \dots, (u_m, \beta_m/\lambda, v_m)$. Based on the above rules, the new measures are defined as follows:

$$\begin{aligned} TLength - Con(\gamma) &= \sum_{(u,v) \in R} |uv|, \\ TLength - Ins(\gamma) &= \sum_{(u,\alpha,v) \in R} |u\alpha v|, \\ TLength - Del(\gamma) &= \sum_{(u,\beta,v) \in R} |u\beta v|. \end{aligned}$$

The measure $TLength - Con$ specifies the total length of the contexts used in insertion/deletion rules, $TLength - Ins$ specifies the total length of the contexts

used in insertion rules plus the length of the strings (α) to be inserted and $TLength - Del$ specifies the total length of the contexts used in deletion rules plus the length of the strings (β) to be deleted.

4.1 Trade-off between Ambiguity and Measures

In this section, we analyze the trade-off between the newly defined ambiguity levels and various complexity measures. We show that when a minimal measure M is chosen for a language L , then all the corresponding systems which generates wwL are ambiguous. On the other hand, when an unambiguous system is chosen for L the system is not minimal in M . The corollary is the consequences of the theorem and by the fact that the measures Ax, MAx, TAx are pairwise compatible [5]. Before, we proceed to the results, let us adapt the notion of ‘pseudo inherently ambiguous’ introduced in [12]. A system γ is said to be restricted if γ satisfies some conditions imposed on it. Here, we impose the condition minimal measure (with respect to a measure M) to systems and we say that γ is restricted with respect to M .

Definition 2. *A language L is said to be pseudo inherently ambiguous (when considered minimal with respect to M) if every restricted system (with respect to M) generating L is ambiguous.*

In the next theorem, we show that if we want to store the language L_6 with the corresponding system which is minimal in terms of $Prod$, then any such system is (5-) ambiguous. On the other hand, if we want to store L_6 with the corresponding system which is unambiguous, then any such system is not minimal with respect to $Prod$. A similar result is discussed in Theorem 8. Thus, in the following section we make a trade-off between ambiguity and complexity measures in choosing a system for a language.

Theorem 7. *There are pseudo inherently 5-ambiguous ins-del languages when $Prod$ is considered minimal, but there are 5-unambiguous systems which are not minimal in $Prod$.*

Proof. Consider the language $L_6 = \{d(abb)^n \mid n \geq 0\} \cup \{(abb)^n c \mid n \geq 0\}$. The language L_6 can be generated by the following 5-ambiguous ins-del system γ_6 .

$$\gamma_6 = (\{a, b, c, d\}, \{a, b, c, d\}, \{d, dabb, c, abc\}, \{(abb, \lambda/abb, \lambda)\}).$$

The system γ_6 is minimal with respect to $Prod$. From γ_6 , it is easy to see that $Prod(L_6) = 1$. Any system which generates L_6 must have at least one insertion/deletion rule. Hence, the language L_6 is minimal in $Prod$.

Consider any system γ'_6 which generates L_6 for which $Prod(\gamma'_6) = 1$. Since different abb is chosen for insertion/deletion in deriving the words $d(abb)^i$ and $(abb)^j c$ for large value of i and j , the position where the string abb is inserted/deleted differs, but derives the same word. Hence, the system γ'_6 is ambiguous. Note that the system γ'_6 is 4-unambiguous.

However, the language L_6 is 5-unambiguous as we can give a 5-unambiguous $\gamma''_6 = (\{a, b, c, d\}, \{a, b, c, d\}, \{d, c\}, \{(d, \lambda/abb, \lambda), (\lambda, \lambda/abb, c)\})$ which generates L_6 . The first part of the language can be generated by inserting the string abb to the right of d . Similarly, the second part of the language can be generated by inserting the string abb to the left of c . As the position of inserting the string never changes in both parts of the language, the system γ''_6 is unambiguous. Note that, the system γ''_6 is not minimal in $Prod$, since $Prod(\gamma''_6) = 2$. \square

Theorem 8. *There are pseudo inherently 4-ambiguous ins-del languages when Ax is considered minimal, but there are 4-unambiguous systems which are not minimal in Ax .*

Proof. Consider the language $L_7 = \{b(a)^{3n} \mid n \geq 0\} \cup \{c(a)^{3n} \mid n \geq 0\} \cup \{ba^{3n}ca^{3m} \mid n, m \geq 0\}$. The language L_7 can be generated by the following 4-ambiguous ins-del system γ_7 .

$$\gamma_7 = (\{a, b, c\}, \{a, b, c\}, \{b, c, bc\}, \{(b, \lambda/aaa, \lambda), (c, \lambda/aaa, \lambda)\}).$$

The system γ_7 is minimal with respect to Ax . From γ_7 , it is easy to see that $Ax(L_7) \leq 3$. Any system which generates L_7 must have at least three axioms corresponding to the three parts of the language. If the system γ_7 is having less than three axioms, it will generate strings not in the language. Hence, $Ax(L_7) \geq 3$, which implies $Ax(L_7) = 3$.

Consider any system γ'_7 which generates L_7 for which $Ax(\gamma'_7) = 3$. Since the string $ba^{3i}, i \geq 0 \in L_7$, the system should have insertion/deletion rule of the form $(b, \lambda/aaa, \lambda) / (b, aaa/\lambda, \lambda)$. Similarly, since the string $ca^{3j}, j \geq 0 \in L_7$, the system should have insertion/deletion rule of the form $(c, \lambda/aaa, \lambda) / (c, aaa/\lambda, \lambda)$. Consider a word $ba^{3r}ca^{3s}, r, s \geq 0$, this word can be derived from the axiom of the form $ba^{3p}ca^{3q}, p, q \geq 0$ by two different ordered complete control sequences. In one sequence the insertion/deletion $(b, \lambda/aaa, \lambda) / (b, aaa/\lambda, \lambda)$ rules can be used first, followed by the insertion/deletion rules $(c, \lambda/aaa, \lambda) / (c, aaa/\lambda, \lambda)$. In another derivation, the insertion/deletion rules $(c, \lambda/aaa, \lambda) / (c, aaa/\lambda, \lambda)$ can be used first, followed by the insertion/deletion rules $(b, \lambda/aaa, \lambda) / (b, aaa/\lambda, \lambda)$. Note that since the string aaa to be inserted/deleted is same in both the derivations, the system is 1 and 3-unambiguous.

However, the language L_7 is 4-unambiguous as we can give a 4-unambiguous $\gamma''_7 = (\{a, b, c\}, \{a, b, c\}, \{b, baaa, c, caaa, bc, baaac, bcaaa, baaacaaa\}, \{(a, \lambda/aaa, \lambda)\})$ which generates L_7 . Note that the system γ''_7 is not minimal in Ax . \square

Corollary 1. *There are pseudo inherently 4-ambiguous ins-del languages when Ax, MAx, TAx are minimal, but there are 4-unambiguous systems which are not minimal with respect Ax, MAx, TAx .*

5 Conclusion

Insertion-deletion systems were defined and motivated by the way DNA strands are inserted and deleted. In this paper, we defined various ambiguity levels

($i = 0, 1, 2, 3, 4, 5$) for insertion-deletion systems. Next, we showed that there are inherently i -ambiguous insertion-deletion languages which are j -unambiguous for the following combinations $(i, j) \in \{(5, 4), (4, 2), (3, 1), (3, 2), (2, 1), (0, 1)\}$. We have not proved the result for the pair $(1, 0)$ and is left as open. We have also shown an example that how the ambiguity levels defined for ins-del systems can be interpreted in gene sequences. Then, we proved that the ambiguity problem for insertion-deletion systems is undecidable. Further, we defined three new complexity measures $TLength - Con$, $TLength - Ins$, $TLength - Del$. We discussed the trade-off between the newly defined ambiguity levels and complexity measures in insertion-deletion systems. We aimed to show that there are pseudo inherently i -ambiguous insertion-deletion languages which are i -unambiguous. A few more complexity measures like maximal length of an inserted and deleted strings can be defined and the trade-off between the ambiguity levels and new complexity measures can be analyzed as a future work.

Acknowledgment. The second author would like to acknowledge the project SR/S3/EECE/054/2010, Department of Science and Technology, New Delhi, India. The third author would like to acknowledge the scheme “Mentoring of Engineering Teachers by INAE Fellows” of Indian National Academy of Engineering, New Delhi, India and his work was partially carried out during the author’s visit to Indian Institute of Technology Madras, Chennai, India.

References

1. Takaharai, A., Yokomori, T.: On the computational power of insertion-deletion systems. In: Natural Computing, vol. 2, pp. 321–336. Kluwer Academic Publishers (2003)
2. Calude, C.S., Paun, G.: Computing with cells and atoms. An introduction to Quantum, DNA and Membrane Computing. Taylor and Francis, London (2001)
3. Searls, D.B.: The computational linguistics of biological sequences. In: Hunter, L.(ed.) Artificial Intelligence and Molecular Biology, pp. 47–120. AAAI Press (1993)
4. Galiukschov, B.S.: Semicontextual grammars. Mat. Logical. Mat. Ling., Talinin University, 38–50 (1981) (in Russian)
5. Georgescu, G.: The Syntactic Complexity of Internal Contextual Grammars and Languages. In: Martin-Vide, C. (ed.) II Intern. Conf. Mathematical Linguistics, Amsterdam, pp. 17–28 (1997)
6. Paun, G., Rozenberg, G., Salomaa, A.: DNA Computing, New Computing Paradigms. Springer (1998)
7. Paun, G.: Contextual Grammars. The Publ. House of the Romanian Academy of Sciences, Bucuresti (1982)
8. Paun, G.: Marcus Contextual Grammars. Kluwer Academic Publishers (1997)
9. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley (2006)
10. Ilie, L.: On Ambiguity in Internal Contextual Languages. In: Martin-Vide, C. (ed.) II Intern. Conf. Mathematical Linguistics, Tarragona, 1996, pp. 29–45. John Benjamins, Amsterdam (1997)

11. Lakshmanan, K.: A note on Ambiguity of Internal Contextual Grammars. *Theoretical Computer Science*, 436–441 (2006)
12. Lakshmanan, K., Anand, M., Krithivasan, K.: On the Trade-off Between Ambiguity and Measures in Internal Contextual Grammars. In: Cămpeanu, C., Pighizinni, G., (eds.) *DCFS 2008*, Charlottetown, Canada, pp. 216–223 (2008)
13. Kari, L., Păun, G., Thierrin, G., Yu, S.: At the crossroads of DNA computing and formal languages: Characterizing RE using insertion-deletion systems. In: *Proc. of 3rd DIMACS Workshop on DNA Based Computing*, Philadelphia, pp. 318–333 (1997)
14. Kari, L.: On insertion/deletion in formal languages. Ph.D Thesis, University of Turku (1991)
15. Marcus, S.: Contextual Grammars. *Rev. Roum. Pures. Appl.* (1969)
16. Martin-Vide, C., Miguel-Verges, J., Păun, G., Salomaa, A.: Attempting to Define the Ambiguity in Internal Contextual Languages. In: Martin-Vide, C. (ed.) *II Intern. Conf. Mathematical Linguistics*, Tarragona 1996, pp. 59–81. John Benjamins, Amsterdam (1997)
17. Setubal, Meidanis: *Introduction to Computational Molecular Biology*. PWS Publishing Company (1997)

A Multiobjective Phenomic Algorithm for Inference of Gene Networks

Rio G.L. D'Souza¹, K. Chandra Sekaran², and A. Kandasamy²

¹ St Joseph Engineering College, Mangalore, India

² National Institute of Technology Karnataka, Surathkal, Mangalore, India
{rio,kchnitk}@ieee.org, kandy@nitk.ac.in

Abstract. Reconstruction of gene networks has become an important activity in Systems Biology. The potential for better methods of drug discovery and of disease diagnosis hinge upon our understanding of the interaction networks between the genes. Evolutionary methods are proving to be successful in such problems and a number of such methods have been proposed. However, all these methods are based on processing of genotypic information. We have presented an evolutionary algorithm for reconstructing gene networks from expression data using phenotypic interactions, thereby avoiding the need for an explicit objective function. Specifically, we have also extended the basic phenomic algorithm to perform multiobjective optimization for gene network reconstruction. We have applied this novel algorithm to the yeast sporulation dataset and validated it by comparing the results to the links found between genes of the yeast genome at the SGD database.

Keywords: Gene networks, Phenomic algorithm, Multiobjective optimization, Evolutionary algorithms, Yeast Sporulation, Microarray data analysis.

1 Introduction

Advances in methods of gene expression measurement have heralded the advent of high throughput methods such as microarray technology. Biologists now can study hundreds of genes at a time, and such studies lead to the elucidation of relationships between genes which ultimately lead to a better understanding of the cellular processes that form the basis of life. However the datasets that result from such studies have high dimensionality. The challenge is to analyze such datasets without compromising their information content. Several researchers have developed methods of analysis which can determine useful patterns from the datasets without compromising the dimensionality [1].

Gene networks represent relationships between genes, based on observations of how the expression level of each gene affects the expression levels of the others [2]. The determination of these relationships from gene expression measurements is a reverse engineering or reconstruction activity [3]. Evolutionary methods have been found to be useful [4] to analyze and capture the relationships between hundreds of genes. The application of new ideas of evolutionary optimization to the inference of

gene networks is an ongoing process and many non-conventional methods have shown remarkable success [5]. The Phenomic Algorithm, introduced in [6], and further studied in [7], is one such method. It presents an evolutionary approach based on phenotypic interactions rather than genotypic mechanisms which are used in traditional evolutionary algorithms.

In this paper, we have modified the basic phenomic algorithm to handle multiple objectives. It is possible to employ multiobjective optimization to elucidate gene networks which are more biologically plausible [8]. We have used non-dominated sorting in order to determine the pareto-optimal solutions that best represent the balance between the objectives that we have chosen to optimize. We have applied the multiobjective phenomic algorithm to the yeast sporulation dataset [9] and results show a marked improvement in the quality of networks discovered.

The rest of this paper is organized as follows: In Section 2, we review the related work done by others. We devote Section 3 to a discussion about the methodology adopted by the basic phenomic algorithm and its implementation. We discuss the rationale for modification of the basic phenomic algorithm in Section 4 and its actual implementation in Section 5. Finally, Section 6 presents the results and validation, followed by Section 7 which concludes the paper.

2 Related Work

While early methods for reconstruction of gene networks focused on inferring Boolean networks [10] others have used differential equations [11], [12], [13] and Bayesian networks [14], [15] to infer qualitative, as well as quantitative models of gene networks. Given that gene networks are intrinsically nonlinear and dynamic systems, some researchers [8], [16] have used the S-system proposed by Savageau [17] in order to formulate an objective function for the evolutionary algorithm that they use to reverse engineer gene networks.

State space models [18] and information theoretic approaches [19], [20] have also been successfully applied to the problem of inferring gene networks from microarray data. In recent years machine intelligence based approaches [21], [22], [23] are becoming popular in this area due to their relative ease of application. A number of multiobjective evolutionary algorithms (MOEAs) have been applied to the problem of reconstructing gene networks from expression data [24], [25]. Notable among these algorithms is the non-dominated sorting genetic algorithm (NSGA) and its variations which have been applied to the problem of classification of cancer based on gene expression data [26], [27], [28].

The application of MOEAs to the elucidation of gene networks is an area which is receiving a large amount of focus from researchers due to the perceived benefits in applications such as drug discovery and the diagnosis of chronic diseases. This has been the motivation for the development of the phenomic algorithm [6], [7] which attempts to solve the problem of requiring an explicit fitness function for the optimization process. In this paper, we extend this algorithm to perform multiobjective optimization.

3 Basic Phenomic Algorithm

The basic phenomic algorithm is initialized with a population of individuals. Each individual has genetic information embedded within it. In the phenomic algorithm, we embed the expression of a gene within the individual. When constructing gene networks, we study the relationship between genes. If g_i and g_j are objects representing two such genes, their expression patterns across m samples may be written as $g_i = \{w_{ik} | 1 \leq k \leq m\}$ and $g_j = \{w_{jk} | 1 \leq k \leq m\}$.

When the microarray dataset contains records which represent the expression of each gene at m time-steps (instead of m samples) of an experiment, it is possible to verify whether the expression pattern of a gene g_i at a time-step $(t-1)$ has any correlation with the expression pattern of a gene g_j at time t . For this, we define the Pearson correlation coefficient across time-steps (from gene g_i at time-step $t = (k-1)$, to gene g_j at time-step $t = k$), as given in Eqn. (1).

$$Pear(g_i, g_j) = \frac{\sum_{k=2}^m (w_{i(k-1)} - \mu_{g_i})(w_{jk} - \mu_{g_j})}{\sqrt{\sum_{k=2}^m (w_{i(k-1)} - \mu_{g_i})^2} \sqrt{\sum_{k=2}^m (w_{jk} - \mu_{g_j})^2}} . \quad (1)$$

In the basic phenomic algorithm random pairs of genes are considered at a time and the proximity measure between them is calculated. Once the proximity measure is calculated, typical gene interactions such as “meet”, “know”, “like”, “dislike”, etc. are defined as operations on genes g_i and g_j , as shown in Eqns. (2) to (4).

$$meet(g_i, g_j) \text{ returns TRUE iff } g_i \text{ and } g_j \text{ were partners, at least once} . \quad (2)$$

$$know(g_i, g_j) \text{ returns TRUE iff } g_i \text{ and } g_j \text{ are part of the same subnetwork} . \quad (3)$$

$$like(g_i, g_j) \text{ returns TRUE iff } Pear(g_i, g_j) \leq D . \quad (4)$$

These operations determine the character of the phenotypic interactions that take place between gene objects. By storing links between genes that “like” each other it is possible to elucidate the relationships that are required for reconstructing the gene network. A brief description of the main features of the basic phenomic algorithm is given below:

1. Modeling Genes as Individuals: While modeling the genes as individuals, the expression profile of the gene is embedded within the object itself. Also the relationships with other genes which are discovered during the interaction phase are stored within the individual itself.

2. Simulating Gene Interaction: The stage is set for the survival-of-the-fittest by letting individuals to meet randomly. Eqns. (2) to (4) define the typical nature of these interactions between partners that meet.

Procedures for implementing the typical interaction criteria given in Eqns. (2) to (4).

```

meet( $g_i, g_j$ )
(
  if( $g_i$ .MET[ $g_j$ ])
    return TRUE;
  else
    {
      if(!know( $g_i, g_j$ ) and like( $g_i, g_j$ ))
        link( $g_i, g_j$ );
      set  $g_i$ .MET[ $g_j$ ] = TRUE;
    }
}

know( $g_i, g_j$ )
(
  if( $g_i$ .LINK[ $g_j$ ])
    return TRUE;
}

like( $g_i, g_j$ )
(
  if(Pear( $g_i, g_j$ )  $\leq D$ )
    return TRUE;
}

```

3. Enforcing Natural Processes: From time to time the population is consolidated by eliminating individuals which are replicates and have not acquired any links with other individuals. At the end of the process, the links between the genes, which are stored in the individuals, are used to construct the gene networks.

The structure of the basic phenomic algorithm is very similar to a genetic algorithm since phenotypic processing is encountered in every generation. Interested readers may refer to D'Souza et al. [6], [7] for further details of this algorithm. In the following sections, we have modified the basic phenomic algorithm to handle multiple objectives for optimizing the inference of gene networks.

4 Multiobjective Optimization

The inference of gene networks from microarray data is a problem where multiple, and often conflicting, objectives come into play. In this section, the fundamental concepts of MOEAs are presented and thereafter, multiple objectives are chosen for optimizing the inference of gene networks.

4.1 Multiobjective Evolutionary Algorithms

Early evolutionary algorithms were focused on optimizing single objectives. However, most optimization problems have multiple objectives. Optimization of multiple objectives requires that the relative importance of each objective be specified in advance which requires a prior knowledge of the possible solutions. But, by using the concept of Pareto-dominance, it is possible to avoid the need to know the possible solutions in advance. This is one of the reasons for the popularity of such Pareto-based approaches.

Before applying Pareto-based multiobjective optimization, some of the relevant concepts are defined and discussed. Consider the following m -objective minimization problem [29]:

$$\begin{aligned} \min F(X), \\ F = \{f_1(X), f_2(X), \dots, f_m(X)\} \end{aligned} \quad (5)$$

Where f_1, f_2, \dots, f_m are the m objectives. $F(X)$ could as well have been a maximization problem, but it is arbitrarily chosen to discuss from a minimization perspective.

1. Dominance: A solution X is said to dominate a solution Y if $\forall j = 1, 2, \dots, m$, $f_j(X) \leq f_j(Y)$, and there exists $k \in \{1, 2, \dots, m\}$ such that $f_k(X) < f_k(Y)$.

2. Pareto-Optimal Solutions: Solution X is called Pareto-optimal if it is not dominated by any other feasible solutions. Pareto-optimal, or non-dominated, solutions are those solutions which do not dominate each other, i.e., neither of them is better than the other in all the objective function evaluations.

3. Pareto-Front: The locus that is formed by a set of solutions that are equally good when compared to other solutions of that set is called as a Pareto-front. The solutions on each pareto-front are pareto-optimal with respect to each other.

In the next section, suitable objectives are selected based on the current knowledge of the biological properties of gene networks.

4.2 Multiple Objectives for Optimization

It is well known that most biological networks display the small-world network property that predicates sparseness between key nodes and dense local connections around each key node. This definition of small-world networks was offered by Spieth et al. [8]. In a conventional multiobjective evolutionary algorithm, the similarity of the target network to small-world networks could be used as an objective in order to determine the network that has the optimal number of links. Also, since the intention is to find as many links as possible, the number of links discovered could be used as the other objective. The two objectives are formally defined as Number of Links (NOL) in Eqn. (6) and Small-World Similarity Factor (SWSF) in Eqn. (7):

$$NOL = \sum_{i=1}^N l_{ij} . \quad (6)$$

$$SWSF = \frac{1}{C} \sum_{k=1}^C k.n_k . \quad (7)$$

where $l_{ij} = 1$ if gene g_i is linked to gene g_j , else $l_{ij} = 0$ (taken from the adjacency matrix of the network), N is the total number of genes in the target network, n_k is the number of nodes with out-degree of k , and C is the maximum cardinality of the set of genes that can influence any given gene. While optimizing, the objective NOL is maximized, whereas the objective SWSF is minimized. It should be noted here that the algorithms based on the phenomic approach do not directly evaluate solutions using Eqns. (6) and (7). These equations are given so that they can be used in other MOEAs whose results will be compared with the results of the phenomic algorithms. In the multiobjective phenomic approach the two objectives are realized indirectly as follows:

1. Multiobjective Screening: The objective NOL is implemented here by screening out duplicates without losing captured links. In multiobjective screening, the two individuals that meet check if their gene ID is the same. If so, the links captured by both the individuals up to that point are all copied into one of the individuals and the other is deleted. Thus the average number of links-per-gene will go on improving from one generation to the next.

The multiobjective screening procedure.

```
multiobjective_screening(gi, gj)
(
  if(gi.ID = gj.ID)
    appendLinks(gi, gj);
    delete(gj);
)
```

2. Multiobjective Phasing: The SWSF objective is implemented here by introducing a two-phased process. Initially, when two genes with dissimilar gene IDs meet, they are allowed to link without restriction, based on typical interaction criteria given in Eqns. (2) to (4). However, after a certain number of interactions (which is a parameter set at the beginning of the run) a pair of genes is allowed to link only if the first one has more links. Since, at any given time, all genes will not have undergone the same number of interactions; there will be many genes which have more links than the others. The net effect is that some key nodes will capture many more links than the others and most nodes will have very few links.

The multiobjective phasing procedure.

```

multiobjective_phasing( $g_i, g_j$ )
(
if(PHASE = 1)
    meet( $g_i, g_j$ );
else
    if(PHASE = 2)
        if( $g_i.LINKS > g_j.LINKS$ )
            meet( $g_i, g_j$ );
}
    
```

There is no need of fitness functions since there is no explicit fitness evaluation and only individuals that are fitter than others survive into the next generation.

5 The Multiobjective Phenomic Algorithm

The multiobjective phenomic algorithm is initialized in the same manner as the basic phenomic algorithm.

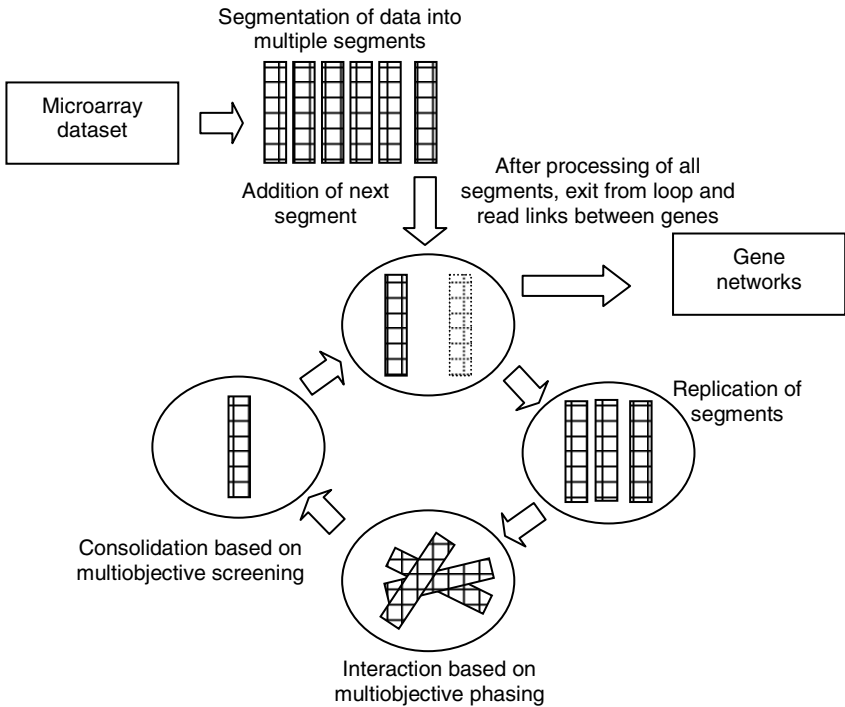


Fig. 1. Sequence of processing in the Multiobjective Phenomic Algorithm

As shown in Fig. 1, most of the sequence of processing and functions are the same as in the basic version, except for the changes that introduce multiobjective considerations. As explained in the previous section, the objective SWSF is achieved by applying multiobjective phasing criteria at the interaction phase of the algorithm. Also, the objective NOL is achieved by introducing multiobjective screening (described in the previous section) in the consolidation phase of the algorithm. The features of the basic algorithm, such as segmentation, interaction and consolidation, which contribute to the scalability and robustness of the algorithm, are retained in this multiobjective version.

The consensus network that is formed in the consolidation phase is just the simple union of all the links in the two individuals being consolidated at that point. It should be pointed out here that in the basic version of the algorithm, the links in one of the individuals were lost when one of them was randomly deleted. Retaining all the links in that algorithm would have led to a proliferation of spurious links since there was no mechanism to restrict the growth of links. In the current version, the interaction phase employs multiobjective phasing which, as explained in the previous section, limits the growth of links after a certain stage.

The multiobjective phenomic algorithm and its main functions.

```
multiobjective_phenomic_algorithm( )
(
  divide gene expression data into segments;
  initialize population with first segment replicated;
  set segment count to 0;

  while population has not reduced to size of single
  segment and there are more segments to process
  {
    interact_population;
    consolidate_population;
    replicate and add next segment;
    increment segment count;
  }

  read gene-links stored in the final population;
  display gene networks constructed from links;
}

interact_population( )
{
  for a preset number of iterations
  {
    randomly select two individuals from population;
    apply multiobjective phasing interaction criteria;
  }
}
```

```

}

consolidate_population( )
{
for a preset number of iterations
{
randomly select two individuals from population;
apply multiobjective screening criteria;
}
}

```

The results obtained from this algorithm and its performance are discussed in the next section.

6 Results and Discussion

The Multiobjective Phenomic Algorithm (MPA) was run on the Yeast sporulation dataset [9]. The expression profiles of 6118 genes are included in this dataset. From these profiles, only those that show a 2.2-fold increase in mRNA levels were extracted by Chu et al. [9]. Among them, finally, only the 45 genes were found to be significant by Kupiec et al. [30].

In Fig. 2, a typical gene network inferred by the MPA is shown. It is only one of the networks that were inferred in that run. Each run of the MPA infers anywhere between 10 to 15 networks. In the network of Fig. 2, for example, node 17 is shown to have a large number of links. This node represents the gene RFA1, which indeed is a crucial yeast gene. As per the SGD database [31], RFA1 is a “subunit of heterotrimeric Replication Protein A (RPA), which is a highly conserved single-stranded DNA binding protein involved in DNA replication, repair, and recombination.” Upon verification with the 220 interactions given in the SGD database, it was found that all the links shown in the network are true links.

In Fig. 3, the value of D was set at 0.02. The number of links can be seen to be much higher. The links were verified by looking up the SGD database and most links were found to be valid. However, it was noticed that a few false positives had crept it at this stage. The comparison of inference methods developed by others was restricted to the following multiobjective evolutionary algorithms:

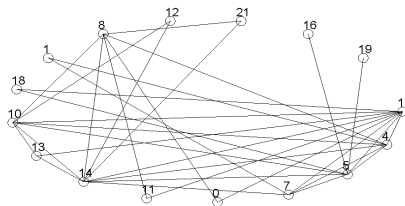


Fig. 2. A gene network inferred by MPA when $D = 0.015$

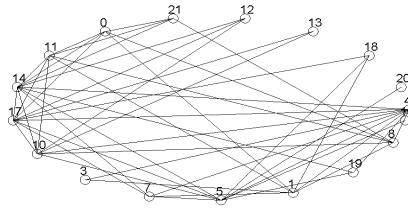


Fig. 3. A Gene Network Inferred by MPA when $D = 0.02$

1. Non-Dominated Sorting Genetic Algorithm Based Method (NSGA-Based):

The NSGA due to Deb [25] was further improved by Deb et al. [26], [27]. In this algorithm, non-dominated sorting is performed on combined parent and offspring population to assign ranks to all the solutions. Based on these ranks solutions are copied over to the next generation. The NSGA is one of the standard MOEAs and therefore it was incorporated into a gene network inference algorithm developed by Spieth et al. [32]. This gene inference algorithm is based on the S-system model and uses Relative Squared Error (RSE) to evaluate the goodness-of-fit between model and the underlying data.

2. Memetic Algorithm (MA-Based): The memetic algorithm developed by Spieth et al. [33] uses the S-systems model and a memetic search procedure for inference of gene networks. A genetic algorithm evolves the topology of the gene networks, while the S-system parameters are evolved through the memetic search procedure.

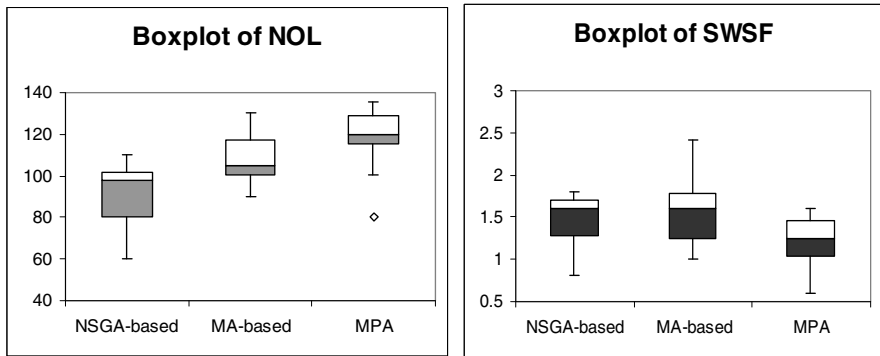


Fig. 4. Boxplots comparing NOL and SWSF of networks inferred by three algorithms being compared

In both the algorithms described above, in order to increase the relevance of gene networks inferred, the objectives defined in the previous section are used. The Number Of Links (NOL) and Small-World Similarity Factor (SWSF), which were defined in Eqn. (6) and Eqn. (7) are used as objectives, in addition to RSE, to perform multiobjective optimization. As seen from the boxplots in Fig. 4, MPA infers better networks than both the NSGA-based and MA-based methods.

7 Conclusion

We have presented the reconstruction of gene networks using the multiobjective phenomic algorithm and also presented results obtained when running the algorithm on the yeast sporulation dataset. The phenomic nature of the algorithm is manifested in its focus on the phenotypic, rather than genetic, information of an individual. Due to the implicit survival-of-the-fittest mechanisms the need for an explicit objective function was avoided.

Currently we are working on applying this algorithm to other datasets in order to study its effectiveness as optimization tool for inference of gene networks.

References

1. Schulze, A., Downward, J.: Navigating gene expression using microarrays - a technology review. *Nature Cell Biology* 3, E190–E195 (2001)
2. Soinov, L.A., Krestyaninova, M.A., Brazma, A.: Towards reconstruction of gene networks from expression data by supervised learning. *Genome Biology* 4(1), R6 (2003)
3. Bansal, M., Belcastro, V., Impiombato, A.A., di Bernardo, D.: How to infer gene networks from expression profiles. *Mol. Syst. Biol.* 3, 78 (2007), doi:10.1038/msb4100120
4. D'haeseleer, P., Liang, S., Somogyi, R.: Gene expression analysis and genetic network modelling: Tutorial. In: *Pacific Symposium on Biocomputing* (1999)
5. Siegal, M.L., Promislow, D.E.L., Bergman, A.: Functional and evolutionary inference in gene networks: does topology matter? *Genetica* 129(1), 83–103 (2007)
6. D'Souza, R.G.L., Chandra Sekaran, K., Kandasamy, A.: A phenomic algorithm for reconstruction of gene networks. In: *IV International Conference on Computational Intelligence and Cognitive Informatics, CICI 2007*, pp. 53–58. WASET, Venice (2007)
7. D'Souza, R.G.L., Chandra Sekaran, K., Kandasamy, A.: Reconstruction of gene networks using phenomic algorithms. *Intl. Journal of Artificial Intelligence Applications (IJAIA)* 1(2), 1–11 (2010), doi:10.5121/ijaia.2010.1201, ISSN: 0976-2191
8. Spieth, C., Streichert, F., Speer, N., Zell, A.: Optimizing Topology and Parameters of Gene Regulatory Network Models from Time-Series Experiments. In: *Deb, K., Tari, Z. (eds.) GECCO 2004, Part I. LNCS, vol. 3102*, pp. 461–470. Springer, Heidelberg (2004)
9. Chu, S., DeRisi, J., Eisen, M., et al.: The transcriptional program of sporulation in budding yeast. *Science* 282, 699–705 (1998)
10. Somogyi, R., Fuhrman, S., Askenazi, M., Wuensche, A.: The gene expression matrix: towards the extraction of genetic network architectures. In: *Proc. of Second World Cong. of Nonlinear Analysts (WCNA 1996)*, vol. 30(3), pp. 1815–1824 (1997)
11. Christley, S., Nie, Q., Xie, X.: Incorporating existing network information into gene network inference. *PLoS ONE* 4(8), e6799 (2009), doi:10.1371/journal.pone.0006799
12. Liu, B., de la Fuente, A., Hoeschele, I.: Gene network inference via structural equation modeling in genetical genomics experiments. *Genetics* 178, 1763–1776 (2008)
13. Qian, L., Wang, H., Dougherty, E.R.: Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering. *IEEE Trans. on Signal Processing* 56(7), 3327–3339 (2008)
14. Numata, K., Imoto, S., Miyano, S.: A structure learning algorithm for inference of gene networks from microarray gene expression data using Bayesian networks. In: *Proc. of the 7th IEEE Intl. Conf. on Bioinfo. and Bioengg. 2007 (BIBE 2007)*, pp. 1280–1284 (2007)

15. Ko, Y., Zhai, C., Rodriguez-Zas, S.: Inference of gene pathways using mixture Bayesian networks. *BMC Systems Biology* 3, 54 (2009), doi:10.1186/1752-0509-3-54
16. Noman, N., Iba, H.: Reverse engineering genetic networks using evolutionary computation. *Genome Informatics* 16(2), 205–214 (2005)
17. Savageau, M.A.: Power-law formalism: a canonical nonlinear approach to modelling and analysis. In: *Proc. of the World Congress of Nonlinear Analysts 1992*, pp. 3323–3334 (1995)
18. Hirose, O., Yoshida, R., Imoto, S., Yamaguchi, R., Higuchi, T., Charnock-Jones, D.S., Print, C., Miyano, S.: Statistical inference of transcriptional module-based gene networks from time course gene expression profiles by using state space models. *Bioinformatics* 24(7), 932–942 (2008), doi:10.1093/bioinformatics/btm639
19. Dougherty, J., Tabus, I., Astola, J.: Inference of gene regulatory networks based on a universal minimum description length. *EURASIP Journal on Bioinformatics and Systems Biology* (2008), doi:10.1155/2008/482090
20. Chaitankar, V., Ghosh, P., Perkins, E.J., Gong, P., Deng, Y., Zhang, C.: A novel gene network inference algorithm using predictive minimum description length approach. *BMC Syst. Biol.* 4(suppl. 1) (2010), doi:10.1186/1752-0509-4-S1-S7
21. Kentzoglanakis, K., Poole, M.: Gene network inference using a swarm intelligence framework. In: *Proc. of the 11th Annual Conf. Companion on Genetic and Evolutionary Computation Conference (GECCO 2009)*, pp. 2709–2712 (2009)
22. Xu, R., Wunsch, D.C., Frank, R.L.: Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. *IEEE/ACM Trans. on Computational Biology and Bioinformatics* 4(4), 681–692 (2007)
23. Zarnegar, A., Vamplew, P., Stranieri, A.: Inference of gene expression networks using memetic gene expression programming. In: Mans, B. (ed.) *Proc. of the 32nd Australasian Computer Science Conf. (ACSC 2009)*, *Conferences in Research and Practice in Information Technology (CRPIT)*, vol. 91 (2009)
24. Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective evolutionary algorithms: analyzing the state-of-the-art. *Evolutionary Computation* 8(2), 125–147 (2000)
25. Deb, K.: *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester (2001)
26. Deb, K., Reddy, A.R.: Classification of two-class cancer data reliably using evolutionary algorithms. *Publ. of Kanpur Genetic Algorithms Lab., India*, Report No. 2003001 (2003)
27. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Computation* 6(2), 182–197 (2002)
28. Kumar, P.K., Sharath, S., D'Souza, R.G., Chandra Sekaran, K.: Memetic NSGA—A multi-objective genetic algorithm for classification of microarray data. In: *15th Intl. Conf. on Advanced Computing and Communications, ADCOM*, pp. 75–80. IEEE (2007)
29. Jin, Y., Sendhoff, B.: Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Trans. on Systems, Man, and Cybernetics* 38(3), 397–415 (2008)
30. Kupiec, M., Ayers, B., Esposito, R.E., Mitchell, A.P.: The molecular and cellular biology of the yeast *Saccharomyces*. *Cold Spring Harbour*, 889–1036 (1997)
31. SGD project: *Saccharomyces genome database* (2007), <http://www.yeastgenome.org/> (September 15, 2007)
32. Spieth, C., Streichert, F., Speer, N., Zell, A.: Multi-Objective Model Optimization for Inferring Gene Regulatory Networks. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) *EMO 2005. LNCS*, vol. 3410, pp. 607–620. Springer, Heidelberg (2005)
33. Spieth, C., Streichert, F., Speer, N., Zell, A.: A memetic inference method for gene regulatory networks based on s-systems. In: *Proc. of Congress on Evolutionary Computation (CEC 2004)*, *Proc. Part I*, pp. 152–157. IEEE Press (2004)

Breaking the Box: Simulated Protein Computing

Christopher N. Eichelberger and Mirsad Hadzikadic

UNC Charlotte, Charlotte, NC 28223 USA
christopher@uncc.edu

Abstract. Computers since the 1940s have shared the same basic architecture described by Turing and von Neumann, in which one central processor has access to one contiguous block of main memory. This architecture is challenged by modern applications that require greater parallelism, distribution, coordination, and complexity. Here we show that a model of protein interactions can serve as a new architecture, performing useful calculations in a way that provides for much greater scalability, flexibility, adaptation, and power than does the traditional von Neumann architecture. We found that even this simple simulation of protein interactions is universal, being able to replicate the calculation performed on a digital computer, yet without relying upon a central processor or main memory. We anticipate that the convergence of information- and life-sciences is poised to deliver a platform that invigorates computing as it provides insight into the complexity of living systems.

Keywords: simulation, protein, parallel, distributed, complex adaptive system, architecture.

1 Introduction

For years, our computers have relied on the von Neumann architecture [40], the endless repetition of “fetch and execute.” The advantage of this serial design is that it is deterministic: The process is repeatable and predictable. Increases in performance have arisen from two main sources: increasing density on integrated circuits [29, 18], and distributing computing tasks across multiple processors [4]. Unfortunately, we are beginning to approach some hard limits with respect to circuit density, and adapting software to take advantage of multiple processors remains very difficult [17]. One approach to overcoming these limitations is to pursue different architectures. Biologically-inspired computing is one such alternative.

Life processes occur with frantic parallelism that is more widely distributed than our silicon-based computing, because there is no dependence upon a single core memory. This distribution comes at a cost: By working outside of the innately serial von Neumann architecture, chemical and biological systems give up strict determinism, and enjoy repeatability and predictability only probabilistically. Furthermore, the mechanics of neither cellular biology nor proteomics are sufficiently well understood to allow us to craft a large-scale, general purpose

computing machine that takes advantage of all of the parallelism implicit within single-celled organisms.

Previous work has focused on two fronts: approaching information science from within the lab; and incorporating laboratory science ideas into programming systems.

1.1 Approaching Information Science from within the Lab

Chemically-inspired computers, such as BZ machines, use a central oscillator as a synchronization method for molecular computations [1, 5]. Unfortunately, the central oscillator tends to constrain the speed of the entire system (to approximately 10 cycles per second [10]), negating many of the advantages of rapidly reacting chemical species. In sum, the billions of interacting molecules are doing significantly less work than they are capable of doing, because of the way they are used in aggregate. This architecture may provide benefits in terms of being applicable in settings where silicon processing not well suited, but it does not currently appear to be a viable candidate to provide greater computing throughput.

DNA computing, in contrast, originally relied upon DNA molecules as inert data elements operated upon by lab protocol qua software [3, 14], principally taking advantage of the ability to explore a huge number of combinations of data solutions simultaneously, albeit in a very manually-driven process. Since then, there have been projects that have used DNA in a more active manner, allowing it to participate in chemical reactions that produce behaviors that are recognizable as logical functions [35, 11, 34, 37]. MAYA-II was a system, built from more than 100 different DNA gates, that could play tic-tac-toe [23]. The team has extended this work, creating a simulation tool that helps to design and debug these networks of biological circuits [25]; this is important, because it reflects the impact of emergent complexity on even relatively small bio-chemical computers.

In vivo computing involves creating information-processing units out of chemical species that occur naturally within organisms (though not in the precise forms, configurations, or concentrations used). The goal is to use these live computing units to influence one or more of the processes within the organism. This function could include, for example, disease diagnosis, treatment, and drug delivery [2, 24]. Not only does emergence continue to play an important role in *in vivo* computing, but it is arguable that its role becomes paramount, as unintended side-effects of a computation carried on inside a living organism could be disastrous.

Where laboratory-based models excel is in making performing simple computations in settings that are not accessible to traditional processors. What they lack is an effective way to model the unintended consequences of introducing interacting chemical species into a complex environment.

1.2 Incorporating Laboratory Science Ideas into Programming Systems

One of the first significant proposals for couching computation in terms of chemical reactions was Banâtre and Le Métayer's Γ (alternately, GAMMA) [6, 7, 8, 9]. Γ portrays programming as a series of multiset transformations, in which the resident species are both data and rules. While Γ is parallelizable and probabilistic (in terms of which reactions are run on what data elements), there is one important accommodation made for halting: Each rule that fires is consumed as it runs. It is also important that reactions and data species do not support wild-cards, meaning that all inputs and rules must be enumerated explicitly. From early on, Banâtre and Le Métayer provided plenty of example programs demonstrating how Γ could be used to solve general computing tasks such as identifying an extreme value in a collection.

Following Γ came the chemical abstract machine, or CHAM [12, 13]. CHAM describes a language derived from Γ , but one that is treated in much greater, and more formal, detail. Whereas Γ served to highlight the utility of chemistry as a computing metaphor, CHAM highlights expresses the expectations and bounds on the formal language of one chemistry-inspired computing approach. CHAM was extended to include membranes, a mechanism that is used to provide for the isolation and localization of computations. Membrane computing is a CHAM concept.

Giavitto and Michel's MGS — (*encore*) un *Modèle Général de Simulation (de système dynamique)* — superclasses both Γ and CHAM (along with cellular automata, Lindenmayer systems, and Paun systems) [19]. Though it is weakly typed, MGS is a functional language that has support for a number of programmer-friendly constructs such as sets, sequences, records, and arrays. In contrast to Γ and CHAM, though, MGS allows transformation rules to include wild-cards. The rules have such a rich syntax, in fact, that they represent a rather wide departure from real chemistry: Rather than having simply $A + B \rightarrow C$, MGS allows A and B to inspect each other, evaluate independent expressions, and incorporate evaluations into C . One consequence of this flexibility is that the number of chemical species that MGS must track can become astronomically large, depending on the program being run: An implementation of a 100-city TSP problem, for example, would likely exhaust the resources of the local machine. As an additional aid to the programmer, but what is arguably another departure from verisimilitude, MGS allows the coder to specify ordered execution within a rule via statement priority. MGS is a programming environment available for public download, but it is constrained to operate on only one computer at a time; there is no cluster-aware version of MGS available.

COPASI — *CO*mplex *PA*thway *SI*mulator — is a joint project of three universities, and is designed to perform stoichiometric analysis and simulation [22, 31]. That is, given a set of chemical reactions and a starting set of reagent concentrations, COPASI has multiple methods of predicting how the solution will change over time, from ordinary differential equations to stochastic simulations based on Gillespie's earlier work. COPASI is not a programming tool *per se* so much

as it is a tool for (bio)chemists, but it is one of the tools that some of the other research projects employ while working on chemistry-inspired computing.

Matsumaru *et al.*, for example, have used both MGS and COPASI to explore chemical organization theory [26, 27, 28]. Their work centers on how to create standard computer science constructs — such as flip-flops and oscillators — using simulated chemical reactions, and how to use graph theory on the stoichiometric description of a system to help bridge the micro-level behaviors with the macro-level outcomes. Like most of the methods within this family, chemical organization theory is not constrained by conservation of mass; in fact, violating this conservation is key to the success of the oscillator they create, as they rely upon a constant influx of new reagents to drive the oscillator. Matsumaru and colleagues echo Müller-Schloer's concerns about organic computing: Emergence is a key property of chemical systems (and simulations), but the bottom-up approach to programming is difficult to program (and to control) effectively [30].

Where chemically- and biologically-inspired computing systems excel is in exploring new methods of parallelization as well as providing platforms for modeling and controlling complexity and emergence as they can be exhibited in real systems. What they lack is suitable verisimilitude to real molecules (and any continuous path to improve this) to allow their results to be more generally applicable to either massively-parallel programming systems or chemically-embedded systems.

1.3 Simulated Protein Computing

The model presented here is meant to be a hybrid approach between computing-inspired laboratory methods and lab-inspired computing methods. Its purpose is to help explore, simultaneously, and in a breadth-first manner, the following:

1. How might an abstract model of chemical interactions be used as the basis for a new computing architecture? Given that silicon-based models of molecular interactions are bound to be crude for now, how can we abstract the chemical model so that it can easily be upgraded over time?
2. How might we develop a (relatively) inexpensive, software-based simulation that provides the opportunity to explore, quantify, control, and re-use complexity in bottom-up systems such as we find in real, living cells?

Simulated protein computing has no more to do with real proteins [15] — at least for now — than a genetic algorithm has to do with real DNA molecules [21]. When we speak of crafting protein programs, we are still talking about writing text files that the computer will interpret and execute. Our intent, though, is not merely to drape clever coding tricks in superficial biological metaphor, but is to begin to explore the real capabilities and the real limitations that computer programs will exhibit when they are expressed as protein molecules. (See Table 1 for a comparison of traditional computing with simulated protein computing.) Neither computer science nor proteomics is yet ready for this convergence, but our experience suggests that the conjunction will be profitable to both disciplines.

Table 1. Contrasting traditional computing and protein computing

traditional computing	simulated protein computing
<p>Advantages:</p> <ul style="list-style-type: none"> – determinism – global memory: there is only one authoritative value for each variable – familiarity, and the amount of investment in the current architecture – simplicity and directness: this method is well suited for writing operating systems and word processors 	<p>Advantages:</p> <ul style="list-style-type: none"> – distributed memory: data proteins are scattered across the simulation, providing for concurrent access – parallelism: functional proteins are independent enzymes, all copies of which can execute simultaneously – distribution: coding enzymes can be introduced to any location, and can diffuse to new locations – localization: proteins can have become differentially concentrated across locations, providing for location-specific computing – separability: each function is an independent particle, so computation is innately separable – emergence: differential computation supports experimentation with self-modifying approaches to solving difficult problems, such as artificial intelligence or drug design and delivery – hybridization: protein computing is inspired by, and can inform, both information science and life science
<p>Disadvantages:</p> <ul style="list-style-type: none"> – serial execution – difficulty scaling: concurrency must be handled by the programmer – inseparability: not all problems are equally separable 	<p>Disadvantages:</p> <ul style="list-style-type: none"> – non-determinism – distributed memory: variable values can only be established by assay – novelty: writing code for this architecture requires a different mindset

2 Method

Programming within a living organism will mean fracturing the building blocks with which we build software. It is tempting to think of a cell as if it were one processing unit, but this would be a mistake. Though the nucleus may help to direct cellular activities, these processes occur throughout the cell, albeit in specialized forms depending on the location, and most often through the interaction of proteins. Our simulation creates a virtual cell as a collection of small, uniform volumes within which independent logical proteins interact. In a real cell, each of these spaces would contain the proteins that react; in our

simulation, each compartment maintains its own estimate of the proteins that may be present, treating each compartment as if it were a tiny reaction vessel.

There are two types of biological molecules that the simulation supports: inert, structural proteins, that are analogues to plain data in traditional programming, such as the integer 5; and enzymes, that can bind to other species, and interact with them, that are analogues to subroutines in traditional functional programming languages, such as “IF(condition, true-result, false-result)”.

Each new architecture entails new assumptions. Approaching programs as if they were proteins means assuming that many of the standard tools of digital computing are no longer available. There is, for instance, no longer a CPU. Every enzyme (active protein) is its own processor, working at the same time as all others, but without any knowledge of them except through their influence on the local, shared environment. There is also no main memory. The variable “X” no longer is a unique location in memory, but may exist in thousands of copies — each with its own value — across multiple locations. This means that though performing individual calculations may be very fast, determining an consensus output value may be very time-consuming, requiring an assay to establish the distribution of values. In traditional programming, the code directly manipulates a variable’s value; in simulated protein computing, the code shapes a variable’s probability density function. Arguments are no longer passed into functions, but functions have binding sites into which available proteins of the right shape and pattern may bind when needed. Subroutines are written as substrates for computing. This means that there are times when an enzyme gets a chance to become active, but cannot do so, because no suitable inputs are available to satisfy the binding sites. Lastly, intermediate computations inside of any subroutine (enzyme) no longer matter. The only state changes the system recognizes involve: denaturing a protein, thereby removing it from the local environment; assembling a new particle (or new conformation) from one or more existing particles, and introducing it into the local environment; and moving chemical species from one location to another (diffusion). See Table [II](#) for a comparison of these two architectures.

The main event loop in this type of system changes from the von Neumann architecture’s “fetch-execute” to one in which every cellular compartment does the following every time step: the compartment accepts species that infuse from neighbouring locations; generates a sample of proteins present at the physical location being simulated (from the probabilistic profile of the proteins that may be present); allows each protein the opportunity, if it has a functional form, to bind and react with the other proteins present in the sample window; takes the resulting list of proteins created, destroyed, and modified in the previous step, and updates the reaction vessel’s profile of proteins likely present; computes gradients against its neighbours, and prepares a list of diffused proteins to export to each neighbour at the beginning of the next time step.

Proteins are large molecules, but small processors, so protein programs look very much like low-level routines in which every smallest step must be represented explicitly. Our simulated enzymatic programs assume that there are

functional motifs — blocks of amino acid residues — that serve as the equivalent to machine instructions. These motifs, treated as if they were indivisible units, are assembled into simple tree-shaped programs that resemble to abstract syntax trees for a functional language. (See Figure 1 and Figure 3 for example programs used in the experiments described later in this work.) Every motif, when it is evaluated, returns a single value; these values are passed up the tree, reaching the head where the final value is discarded, because — as pointed out earlier — all intermediate results are meaningless once they have been used. The only motifs that change the environment are EMIT, responsible for introducing a new protein into the local environment, and DENATURE, which is responsible for removing an existing protein from the local environment. Table 2 shows the same function implemented twice, once in traditional pseudo-code and once in the form that might be used in this biologically inspired architecture.

Table 2. Contrasting two implementations of the same function

traditional pseudo-code	simulated protein computing
<pre>function get_minimum(A, B) { if (A < B) return A; return B; }</pre>	<pre>(if (and (exists (match value (.*))) (exists (match value (.*)))) (if lt(\$1, \$2) (complex (emit \$1) (denature \$2)) (complex (emit \$2) (denature \$1))))</pre>

One important difference between these two code samples is that the traditional version creates a new value that is a copy of whichever input parameter represents a lesser value; the protein version binds two values from the local environment, and replaces the greater value into a copy of the lesser. Whereas the former method creates a single, definitive answer, the latter method merely alters the distribution of values in the local environment. Another important difference is that the traditional version will always return a value when it is invoked; the protein version can only run when its local environment includes two VALUE proteins that can bind into its activation regions. This means that protein programs have reaction rates that are influenced by the concentration of other proteins nearby. This is a concern (and an opportunity) that programmers working in real biological systems will have, but that traditional programmers will not have.

To implement the method fully requires substantially more information about how the system is defined, constrained, and run, including: the language model,

with both syntax and operators; the modular abstraction of chemistry, including which reactions are allowed and how to compute chemical gradients; the (lossy) data compression used to handle the large volume of data about what protein species may occupy a given reaction vessel; the encoding that allows the secondary conformation of the protein programs, their tree shape, to be inferred from their primary conformation; the macro facility that allows certain proteins to be stored without loss of fidelity; the geometry of simulated cells and their constituent compartments; the extensible monitoring that allows us to probe any compartment; the graphical display of results as the simulation progresses; the XML pre-processor that simplifies writing source code for this platform; etc. These details exceed the scope of the current argument, and so are omitted.

3 Experiments and Results

Foregoing the programmers' canonical, "Hello, world!", this paper focuses on three separate experiments: two NAND experiments and a decomposition experiment.

The logical NOT AND (NAND) function compares two boolean values: If both inputs are TRUE, the result of the function is FALSE; otherwise, the function returns TRUE. To explore correctness, we present two versions of NAND. The first operates directly on raw numbers, and demonstrates the accuracy of the computation. The second version operates on labeled complexes, and demonstrates how networks of cascading NANDs can be executed reliably by a simulated protein computer. Jointly, the NAND experiments are important, because any universal binary computer can be simulated using nothing more than NAND functions.

The last experiment is a simple decomposition reaction that we use to explore the performance implications of protein programming.

3.1 Unlabelled NAND and Correctness

Assume that the simulation is roughly analogous to a flask containing various reagents. In the unlabelled NAND case, the flask contains only these reagents in equal proportions: integer value 0, an inert data species; integer value 1, an inert data species; and NAND, an enzyme that can bind to two inert data species. Once it binds to two data elements successfully, it then (and only then) applies the NAND function to its two bound inputs, and transforms one of them (selected randomly) from its bound input value to the result of the function. Once the evaluation is complete, the two bound species — one as given, and one transformed to the function result — are released back into the environment. Figure 1 illustrates this program.

The total number of particles in the system is fixed. (While the system does not enforce conservation of mass, it is recommended.) The only change in the system over time is the relative proportion of zeros and ones as the data proteins are transformed by the enzyme. Figure 2 is a scatter plot that displays the sampled concentrations of ones and zeros in the simulation over time.

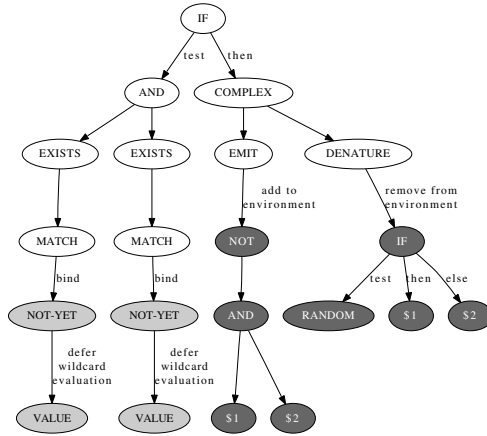


Fig. 1. Unlabeled NAND

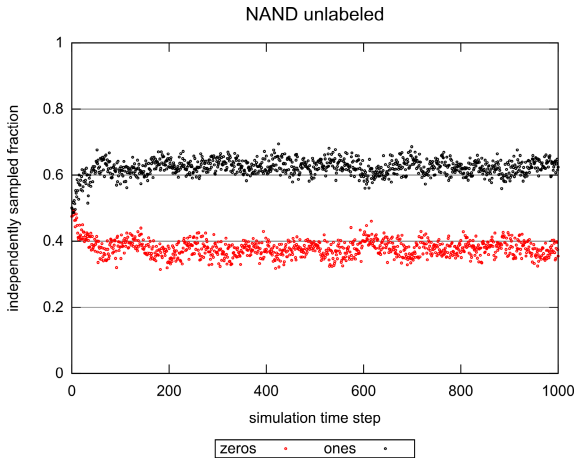


Fig. 2. Unlabeled NAND output. The black points represent the sampled concentration of ones; the red points represent the sampled fraction of zeros.

Within approximately 100 simulated time steps, the system reaches a rough equilibrium, in which the increase in the concentration of ones is offset by the increasing likelihood of the NAND function to return 0 in an environment dominated by ones. Assuming that we adopt the convention that the concentration of integer ones is $[\#1]$, and the concentration of integer zeros is $[\#0]$, we can express this equilibrium in its algebraic form as:

$$\frac{d[\#1]}{dt} = -\frac{d[\#0]}{dt} = -[\#1]^2 + [\#1][\#0] + [\#0]^2 \tag{1}$$

Given the additional constraint that, because we are preserving mass, the total number of integers (data proteins) remains fixed, the system becomes solvable:

$$[\#0] = \frac{3 - \sqrt{5}}{2} \approx 0.38, \quad [\#1] = \frac{\sqrt{5} - 1}{2} \approx 0.62 \quad (2)$$

The algebraic solution to the system matches the equilibrium on which the simulation fairly quickly settles, suggesting that the sampling and computation are being performed correctly.

3.2 Labeled NAND and Completeness

The initial test was artificially simple: There are very few useful applications that consist of a single, isolated calculation. It is more important to investigate whether serial computations can be performed reliably. To explore whether biologically-inspired computing can satisfy this requirement, we introduced labeled complexes (akin to tagging biological chemicals) into the system.

This second test uses labels to identify each piece of data as specific to one stage in a multi-stage computation. Each stage is tagged with a different label, and the stages together constitute a network of cascading NANDs. Being able to construct and coherently run such networks is relevant to the reach of the computing system, because NAND networks are sufficient to emulate any other function on a universal computer.

The reaction vessel is initialized with these species: complex (A,0), an inert data species, in which “A” is the label, and “0” is the value; complex (A,1), an inert data species, in which “A” is the label, and “1” is the value; $\text{NAND}(A,A) \rightarrow B$, a function that binds to two integer values that share the “A” label, computes the NOT AND result on these two operands, and then converts one of these bound inputs to the output value, changing its label from “A” to “B” (see Figure 3); $\text{NAND}(B,B) \rightarrow C$, similar to the function described previously, but using different labels.

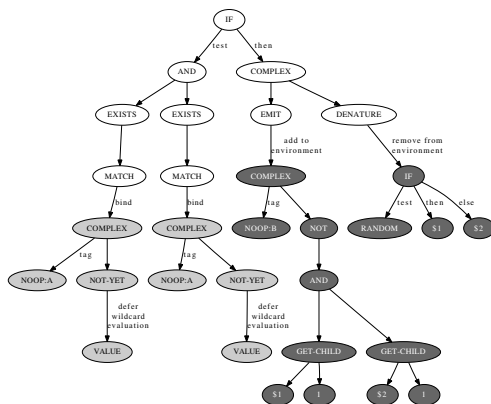


Fig. 3. Labeled NAND

Because we have already established that a single NAND performs as expected, we monitored only the total number of data elements that were labeled for each stage in the computing chain: A, B, and C. The results appear in the scatter plot in Figure 4.

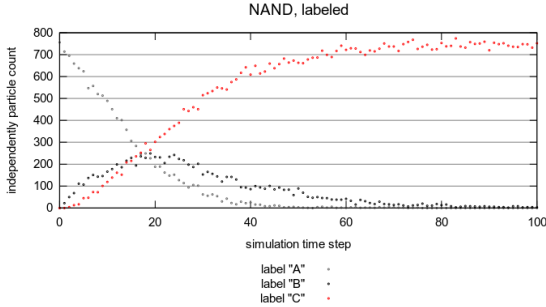


Fig. 4. Labeled NAND output. These are the particle counts, determined by assay, from the labeled NAND at each time step. The light gray points are the concentration of data element A; the black points are the concentration of data element B; and the red points are the concentration of data element C.

Note that the unreplenished inputs, labeled A, diminish over time as expected. Species B begins at a concentration of zero — because none were introduced into the reaction vessel, but must arise as a result of $\text{NAND}(A,A)$ producing them — and increases. Species B, however, grows more slowly than species A decreases, because $\text{NAND}(B,B)$ is consuming B to produce species C. These progressions validate an important property: the nominally serial computation is honouring the serial dependencies as expected, even while the reactions themselves are occurring in parallel. Additionally, Figure 5 shows that the simulated interactions of proteins are behaving as would be predicted by the differential equations that we would expect using Runge Kutta 4:

$$\frac{dA}{dt} = -k_A \cdot A, \quad \frac{dB}{dt} = -\frac{dA}{dt} - k_B \cdot B, \quad \frac{dC}{dt} = k_B \cdot B \quad (3)$$

As in the unlabelled case, the labeled NAND has produced behaviours that are consistent with what would have been predicted of a real, wet-lab system.

3.3 Decomposition and Performance

If protein computing is ever to be useful, it ought to provide performance advantages over traditional computing. To evaluate the expected performance increase of wet-lab computing over simulated protein computing, we focused on a simple decomposition reaction: $E + S \rightarrow ES \rightarrow E + P$. “E” is the enzyme catalyst; “S” is the substrate; and “P” is the product. As a simulated protein program, E binds to S, emits P back into the environment, and denatures S. The activity

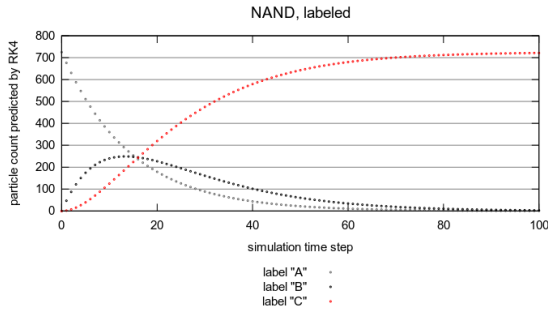


Fig. 5. Predicted output from the labeled NAND. Given the equations as written in (3), these are the curves that Runge Kutta 4 would predict. The light gray points are the concentration of data element A; the black points are the concentration of data element B; and the red points are the concentration of data element C.

of a real, yet fairly simple, enzyme can be characterized as a function of the amount of substrate present. We did this for the artificial case, fixing the enzyme at 1000 particles, and allowing the amount of substrate to range from 0 to twice the amount of enzyme present; each test case was replicated 20 times. The resulting average activity is plotted in Figure 6; the double-reciprocal of these data appear in Figure 7.

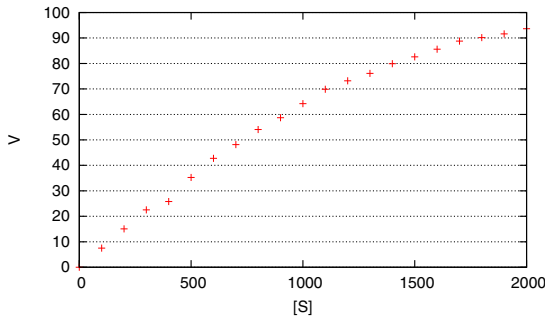


Fig. 6. The Michaelis-Menten plot of enzyme velocity (activity), V , versus substrate concentration, $[S]$

This double-reciprocal plot is only interesting, because the best-fit line has an R^2 value of 0.998, and a good linear fit of data on this plot happens to be typical of simple real-world enzymes. It is also simple, if naive, to estimate the Michaelis-Menten constants from this plot using Equation (4), concluding that $K_m \approx 4111$ particles, and $v_{max} \approx 318$ activations per second.

$$\frac{1}{v} = \frac{K_m}{v_{max}} \cdot \frac{1}{x} + \frac{1}{v_{max}} \tag{4}$$

Inspecting Figure 6 suggests that the analysis is off, because the curve appears to be nearing a plateau much faster than the expected v_{max} of 318 would imply;

furthermore, since the K_m maps to the concentration of substrate at which the enzyme performs at half its maximum velocity, it seems unlikely that a K_m of greater than 4000 is warranted.

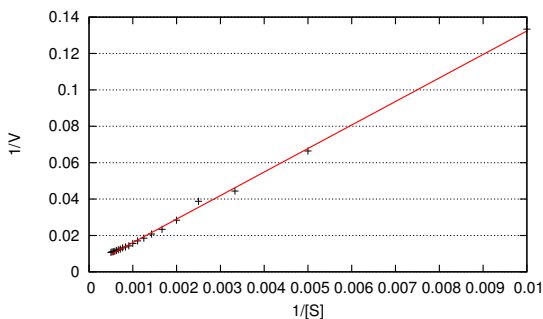


Fig. 7. The Lineweaver-Burk double-reciprocal plot of enzyme velocity (activity), $1/V$, versus substrate concentration, $1/[S]$

To estimate the performance of this simple decomposition if it were conducted entirely through real proteins rather than through simulation, temporarily assume that the complexity of the simulated deconstruction enzyme, E, is roughly comparable to the complexity of acetylcholinesterase (AChE). This likeness is motivated by the fact that AChE also decomposes its substrate (acetylcholine); in the experiments conducted here, E also performs only a single decomposition. Because the simulation used 1000 enzyme particles, the theoretical maximum number of times that any single E enzyme could activate per second is 0.318; AChE can react approximately 12,500 per second [20]. The implication is that real AChE outperforms the simulated E by a factor of roughly 39,000. Given the overhead of the simulation, that performance increase does not appear to be substantial, but the cursory analysis is misleading.

There are many difficulties in comparing the simulated decomposition enzyme to real AChE: there is no evidence to support the supposition that AChE's enzymatic efficiency is anywhere near that of E; there is, in fact, no basis for establishing an enzymatic efficiency for E at all; the mechanics of the artificial protein chemistry are so simple that there is almost no way to draw realistic comparisons to real proteins. Despite these difficulties, it is possible to appreciate better the increase in performance that protein computing represents by conducting a simple thought-experiment: One drop of a 0.1 mM solution of AChE, provided with a surfeit of substrate, would produce approximately 3.75×10^{19} reactions in a single second. The Jaguar cluster at Oak Ridge National Laboratories is a super-computer that, as of December 31, 2009, was at the top of the list of high-performance computers in the world [38]. Jaguar contains just over 2.7 million computing cores [32], each no more than 100 times more powerful than the desktop machine on which these tests were run. If this simulation were written so that it could occupy 100% of the Jaguar's resources, and there were

(magically) no inefficiencies introduced from the parallelization, it would still take more than 13 years to complete the equivalent computation performed by that one drop of AChE in one second.

Clearly, this is the kind of spurious extrapolation that invites ridicule, because it is clear that a considerable portion of the chemical reactions taking place represent redundant work. The obvious question to ask is: How much of the work being done is useful? Think of the computations in terms of breadth and depth, where breadth represents similar reactions occurring at an early stage in a much longer chain of computations, and depth represents progress down any single (potentially very deep) computational chain. If, out of one million enzymes, 90% are executing the equivalent of instruction #1, then they represent the breadth of computing; the 10% that are executing the equivalent of some later calculation in the larger effort represent the depth of the computing that is taking place. If one assumes that each unit time results in some (average) fraction, p , of all enzymes reacting, then the expected distribution of protein reactions will peak at $\lfloor p \cdot t \rfloor$, where t is the number of time units that have transpired. This means that the depth of computation increases linearly with the time elapsed, allowing one to conclude that — even when much of the computation in a protein system is duplicative, and seemingly wasteful — any single computational path is proceeding forward very rapidly.

This exaggerated extrapolation of expected throughput also serves another real purpose: It highlights two important properties of protein computing, both of which motivate continuing this research to find better ways to learn how to program within this paradigm:

1. Compactness: Real proteins, whether they are enzymes or inert data, can fill a small volume with large numbers, providing both for fast execution as well as significant exploration. Silicon computing, in contrast, occurs in only two dimensions.
2. Frictionless scaling: Given adequate substrate, two drops of enzyme solution will yield twice the product that one yields, because the activation of each molecule is entirely independent. Traditional computing, in contrast, imposes an increasing communication inefficiency as the number of computing units increases.

The core opportunity that real protein computing represents is to take advantage of the three-dimensional density of real containers in a way that is subject to fewer diminishing returns. The core opportunity that simulated protein computing represents is a way to explore the vagaries of programming in an entirely new way.

3.4 Conclusions

The two successful NAND experiments provide evidence that the simulated protein computing method described here is functionally complete, meaning that it can compute any binary-valued function [33]. This property makes it possible for protein computing to reproduce any function that is expressed within

a modern silicon processor. Because we assume that our traditional computing devices are universal, this same assumption now extends to include the model of protein computing presented here [39], [37], [16], even though traditional notions of recursive enumeration do not appear to apply to protein programs.

Of greater impact than its universality is the system's ability to scale up. Breaking calculations into pieces that can be sent to multiple computers in parallel is difficult, time-consuming, and prone to introduce error. Programming in the simulated protein environment presents a learning curve, because it is very different than traditional programming, yet all of the programs that get written can immediately be run across an arbitrarily large number of processors. Simulating a non-von Neumann architecture on a traditional computer is not particularly efficient, but it allows us to become familiar with the capabilities and pitfalls – including emergent properties arising from this bottom-up approach – of such a programming method until such time as a true in vivo implementation is available when it is conceivable that a solution of proteins — planned and refined in a software simulation — is introduced to a colony of generic cells; as the solution washes across the millions of living organisms, the foreign proteins invoke a chain of responses that culminate in an assay that provides the distribution of problem results. In such a system, it is not merely each of the millions of cells that is a processor, but each of the cells is a collection of millions of processing units, freeing us at last from the fetch-execute bottle-neck [36].

Notes and Comments. The germ from which this project grew — “What would it look like if we could use proteins to write programs?” — began as a project jointly conceived with Dr. Kayvan Najarian, now at Virginia Commonwealth University.

Dr. Seok-Won Lee of UNC Charlotte was instrumental in reviewing early drafts of this text. Dr. William Tolone and Dr. Zbigniew Ras also served as reviewers.

Full source code for this project (licensed under the GPLv3) is available via <http://www.simulatedproteincomputing.org>.

References

1. Adamatzky, A., De Lacy Costello, B.: Experimental logical gates in a reaction-diffusion medium: The XOR gate and beyond. *Phys. Rev. E* 66(4), 46112 (2002)
2. Adar, R., Benenson, Y., Linshiz, G., Rosner, A., Tishby, N., Shapiro, E.: Stochastic computing with biomolecular automata. *Proceedings of the National Academy of Sciences* 101(27), 9960–9965 (2004)
3. Adleman, L.M.: Molecular computation of solutions to combinatorial problems. *Science* 266(11), 1021–1024 (1994)
4. Anderson, D.P.: Public computing: Reconnecting people to science. In: *Conference on Shared Knowledge and the Web*, Residencia de Estudiantes (2003)
5. Bull, L., Adamatzky, A., De Lacy Costello, B.: On polymorphic logical gates in sub-excitable chemical medium (June 2010)

6. Banâtre, J.-P., Le Métayer, D.: A parallel machine for multiset transformation and its programming style. *Future Generation Computer Systems* 4(2), 133–144 (1988)
7. Banâtre, J.-P., Le Métayer, D.: The GAMMA model and its discipline of programming. *Sci. Comput. Program.* 15(1), 55–77 (1990)
8. Banâtre, J.-P., Le Métayer, D.: Programming by multiset transformation. *Commun. ACM* 36(1), 98–111 (1993)
9. Banâtre, J.-P., Priol, T.: Chemical programming of future serviceoriented architectures
10. Bánsági, T., Leda, M., Toiya, M., Zhabotinsky, A.M., Epstein, I.R.: High-frequency oscillations in the Belousov-Zhabotinsky reaction. *The Journal of Physical Chemistry A* 113(19), 5644–5648 (2009)
11. Benenson, Y., Adar, R., Paz-Elizur, T., Livneh, Z., Shapiro, E.: Dna molecule provides a computing machine with both data and fuel. *Proc. Natl. Acad. Sci. U.S.A.* 100(5), 2191–2196 (2003)
12. Berry, G., Boudol, G.: The chemical abstract machine. *Theor. Comput. Sci.* 96(1), 217–248 (1992)
13. Boudol, G.: A Generic Membrane Model, Global Computing Workshop, Rovereto. In: Priami, C., Quaglia, P. (eds.) *GC 2004. LNCS*, vol. 3267, pp. 208–222. Springer, Heidelberg (2005)
14. Braich, R.S., Chelyapov, N., Johnson, C., Rothemund, P.W.K., Adleman, L.: Solution of a 20-variable 3-sat problem on a dna computer. *Science* 296(5567), 499–502 (2002)
15. Creighton, T.E.: *Proteins: Structures and Molecular Properties*. W. H. Freeman and Company (1993)
16. Davis, M., Sigal, R., Weyuker, E.J.: *Computability, Complexity, and Languages, Fundamentals of Theoretical Computer Science*, 2nd edn (1994)
17. Dijkstra, E.W.: Solution of a problem in concurrent programming control. *Communications of the ACM* 8(9) (1965)
18. Dubash, M.: Moore’s Law is dead, says Gordon Moore. *Techworld* (2005)
19. Giavitto, J.-L., Michel, O.: Mgs: a rule-based programming language for complex objects and collections. *Electronic Notes in Theoretical Computer Science* 59(4) (2001)
20. Goodsell, D.S.: Acetylcholinesterase: molecule of the month. In: *RCSB Protein Data Bank* (June 2004)
21. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. The University of Michigan (1975)
22. Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., Kummer, U.: COPASI — a COMplex PATHway SIMulator. *Bioinformatics* 22(24), 3067–3074 (2006)
23. Sutovic, M., Lederman, H., Pendri, K., Andrews, W.L.B.L., Stefanovic, D., Macdonald, J., Li, Y., Stojanovic, M.N.: Medium scale integration of molecular logic gates in an automaton. *Nano Letters* 6(11), 2598–2603 (2006)
24. Kahan, M.: Towards molecular computers that operate in a biological environment. *Physica D: Nonlinear Phenomena* 237(9), 1165–1172 (2008)
25. Fanning, M.L., Macdonald, J., Stefanovic, D.: Advancing the Deoxyribozyme-Based Logic Gate Design Process. In: Deaton, R., Suyama, A. (eds.) *DNA 15. LNCS*, vol. 5877, pp. 45–54. Springer, Heidelberg (2009)
26. Matsumaru, N., Centler, F., P.S.: Chemical organization theory as a theoretical base for chemical computing. In: Teuscher, C., Adamatzky, A. (eds.) *Unconventional Computing 2005: From Cellular Automata to Wetware* (2005)

27. Matsumaru, N., Dittrich, P.: Organization-oriented chemical programming for the organic design of distributed computing systems. In: BIONETICS 2006: Proceedings of the 1st International Conference on Bio Inspired Models of Network, Information and Computing Systems, p. 14. ACM, New York (2006)
28. Mendes, P., Hoops, S., Sahle, S., Gauges, R., Dada, J., Kummer, U.: Computational modeling of biochemical networks using copasi. In: Maly, V. (ed.) *Methods in Molecular Biology, Systems Biology*, vol. 500, Humana Press, a part of Springer Science + Business Media, LLC (2009)
29. Moore, G.E.: Cramping more components onto integrated circuits. *Electronics* 38(8) (1965)
30. Müller-Schloer, C.: Organic computing: on the feasibility of controlled emergence. In: CODES+ISSS 2004: Proceedings of the 2nd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, pp. 2–5. ACM, New York (2004)
31. di Speroni, P., Matsumaru, F.N., Centler, F., Dittrich, P.: Chemical organization theory as a theoretical base for chemical computing. *International Journal of Unconventional Computing* 3(4), 285–309 (2007)
32. National Center for Computational Sciences. Jaguar (2009), <http://www.nccs.gov/computing-resources/jaguar/>
33. Pelletier, F.J., Martin, N.M.: Post's functional completeness theorem. *Notre Dame Journal of Formal Logic* 31(2) (1990)
34. Sayut, D.J., Niu, Y., Sun, L.: Construction and enhancement of a minimal genetic and logic gate. *Applied and Environmental Microbiology* 75(3), 637–642 (2009)
35. Simmel, F.C., Yurke, B., Sanyal, R.J.: Operation kinetics of a dna-based molecular switch. *J. Nanosci Nanotechnol* 2, 383–390 (2002)
36. Stepney, S. (et al.) Journeys in non-classical computation : A grand challenge for computing research (2004), www.nesc.ac.uk/esi/events/Grand_Challenges/proposals/stepney.pdf
37. Su, X., Smith, L.M.: Demonstration of a universal surface DNA computer. *Nucl. Acids Res.* 32, 3115–3123 (2004)
38. Top500 Supercomputer Sites. ORNLs Jaguar claws its way to number one, leaving reconfigured Roadrunner behind in newest TOP500 list of fastest supercomputers (2009), <http://www.top500.org/lists/2009/11/press-release>
39. Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* 2(42), 230–265 (1937)
40. von Neumann, J.: First draft of a report on the edvac (1945)

Contribution of Spatio-temporal Intensity Variation to Bottom-Up Saliency

Eleonora Vig¹, Michael Dorr², and Erhardt Barth¹

¹ Institute for Neuro- and Bioinformatics, University of Lübeck,
Ratzeburger Allee 160, 23538 Lübeck, Germany
{vig,barth}@inb.uni-luebeck.de

² Schepens Eye Research Institute, Dept. of Ophthalmology, Harvard Medical School,
20 Staniford Street, Boston, MA 02114, USA
michael.dorr@schepens.harvard.edu

Abstract. We investigate the contribution of local spatio-temporal variation of image intensity to saliency. To measure different types of variation, we use the geometrical invariants of the structure tensor. With a video represented in spatial axes x and y and temporal axis t , the n -dimensional structure tensor can be evaluated for different combinations of axes (2D and 3D) and also for the (degenerate) case of only one axis. The resulting features are evaluated on several spatio-temporal scales in terms of how well they can predict eye movements on complex videos. We find that a 3D structure tensor is optimal: the most predictive regions of a movie are those where intensity changes along all spatial and temporal directions. Among two-dimensional variations, the axis pair yt , which is sensitive to horizontal translation, outperforms xy and xt by a large margin, and is even superior in prediction to two baseline models of bottom-up saliency.

Keywords: video saliency, eye movements, intrinsic dimension, structure tensor, natural dynamic scenes.

1 Introduction

Visual attention, the selective processing of visual information, is an important component of biologically-inspired machine vision systems. Computational models of attention have proven to be invaluable in identifying points of interest within a scene and e.g., through that, enabling the otherwise time- and resource-consuming image processing to focus only on these potentially relevant scene locations.

In human vision, the extent to which a certain scene region captures the viewers' attention, i.e. its level of *saliency*, is determined by two different kinds of mechanisms. On the one hand, basic visual properties, such as motion, contrast, and colour influence where we direct our gaze. On the other hand, top-down knowledge, i.e. our goals and interests, also modulate attentional selection. The relative

contribution of the two mechanisms is still under debate; however, due to the involuntary and computationally more tractable nature of stimulus-driven attention, much work has focused on the bottom-up factors that drive eye movements.

Of major importance was the recognition that scene statistics at the centre of fixation differ significantly from those at random, control locations. Studies have shown that attended regions have high luminance-contrast [6,7], and found regularities in the higher-order statistics (e.g. high edge density [5]). Knowledge about such distinct image properties has been then used to build models of saliency that successfully predict human fixations in natural scenes, e.g. [4,2,9].

Another key finding is related to the region’s degree of spatial (and temporal) variance. It shows for images that intrinsically two-dimensional scene structures (i.e. of higher spatial variance), such as edges and curved lines, have a higher probability to be fixated [5]. In previous work, we could demonstrate for videos that features that change over space and in time also tend to be more salient. We found that the predictability of eye movements correlates with the intrinsic dimension: the higher the intrinsic dimension the higher the predictive power [8,1].

In the present study, we extend our previous analysis by quantifying the contribution of local spatio-temporal variation of image intensity to saliency. To measure different kinds of variation, we compute, for a set of natural outdoor videos, invariants of the n -dimensional structure tensor ($1 \leq n \leq 3$). Considering a video to be represented in spatial axes (x, y) and temporal axis t , the nD structure tensor is evaluated for different combinations of axes (2D and 3D) and also for the (degenerate) case of only one axis. To obtain a simple measure of bottom-up saliency, we use the symmetric invariants of the structure tensors, which we compute on several spatio-temporal scales. Finally, the resulting simple representations are evaluated and compared with two prototypical saliency models of dynamic scenes in terms of how well they can predict eye movements on videos.

2 Invariants of the n -Dimensional Structure Tensor

It has been previously shown that eye movement predictability correlates with the *intrinsic dimension* (iD), i.e. with the number of spatio-temporal directions in which the video changes locally. A classical method to estimate the intrinsic dimension is to consider the rank of the *structure tensor*. Given a grayscale video $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, the structure tensor captures signal variations based on the spatial and temporal derivatives at each pixel. For three-dimensional data, i.e. the spatio-temporal volume of the video, usually a three-dimensional structure tensor is defined. However, on subspaces of the video volume (e.g. combinations of two axes, or even considering the degenerate case of a single axis only) 1D or 2D structure tensors can be constructed.

Here, we formalize the problem for the two-dimensional structure tensor \mathbf{J}_2 , considering only the vertical spatial dimension y and the temporal dimension t . The generalization of the formulas for the n -dimensional case ($1 \leq n \leq 3$) is given in Table III. For the axis combination yt \mathbf{J}_2 is defined as

$$\mathbf{J}_2 = \omega(y, t) * \begin{pmatrix} f_y^2 & f_y f_t \\ f_y f_t & f_t^2 \end{pmatrix}, \tag{1}$$

where $\omega(y, t)$ is a Gaussian smoothing function and f_y and f_t stand for the first order partial derivatives $\delta f / \delta y$ and $\delta f / \delta t$. The scale on which the structure tensor is evaluated depends on the bandwidth of the filter kernel $\omega(y, t)$ and the derivative operators. Therefore, computations are performed on a spatio-temporal multiresolution pyramid.

The intrinsic dimension is, in practice, obtained from the symmetric invariants of the structure tensor:

$$\begin{aligned} H &= 1/2 \text{ trace}(\mathbf{J}_2) = \lambda_1 + \lambda_2 \\ K &= |\mathbf{J}_2| = \lambda_1 \lambda_2 \end{aligned} \tag{2}$$

where λ_i denote the eigenvalues of \mathbf{J}_2 . Regions where $H > 0$ are at least intrinsically one-dimensional ($iD \geq 1$), e.g. non-vertical stationary edges, vertically translating edges, and uniform regions that change in time, whereas $K > 0$ indicates an $i2D$ feature such as yt corners (changing motion) and structures that appear or disappear in yt , which correspond to non-vertical translation.

Table 1. n -dimensional structure tensors and their invariants, which correspond to the minimum intrinsic dimension (iD) of a region. Invariants that encode features of higher iD are in general better predictors of eye movements; therefore, they are used for further analysis (these are marked with a box).

n	nD Structure Tensor	Invariants (eigendecomposition of \mathbf{J}_n)
1	$\mathbf{J}_1 = \omega(u) * f_u^2$ $u \in \{x, y, t\}$	<div style="border: 1px solid black; display: inline-block; padding: 2px;">$H = \lambda_1$</div> ($iD = 1$)
2	$\mathbf{J}_2 = \omega(u, v) * \begin{pmatrix} f_u^2 & f_u f_v \\ f_u f_v & f_v^2 \end{pmatrix}$ $u, v \in \{x, y, t\}, u \neq v$	$H = \lambda_1 + \lambda_2$ ($iD \geq 1$) <div style="border: 1px solid black; display: inline-block; padding: 2px;">$K = \lambda_1 \lambda_2$</div> ($iD = 2$)
3	$\mathbf{J}_3 = \omega(x, y, t) * \begin{pmatrix} f_x^2 & f_x f_y & f_x f_t \\ f_x f_y & f_y^2 & f_y f_t \\ f_x f_t & f_y f_t & f_t^2 \end{pmatrix}$	$H = \lambda_1 + \lambda_2 + \lambda_3$ ($iD \geq 1$) $S = \lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3$ ($iD \geq 2$) <div style="border: 1px solid black; display: inline-block; padding: 2px;">$K = \lambda_1 \lambda_2 \lambda_3$</div> ($iD = 3$)

3 Prediction of Eye Movements with Tensor-Based Approaches

Having reviewed simple tensor-based video representations that characterize different types of spatio-temporal changes, we now quantitatively compare their power in predicting eye movements on complex natural videos.

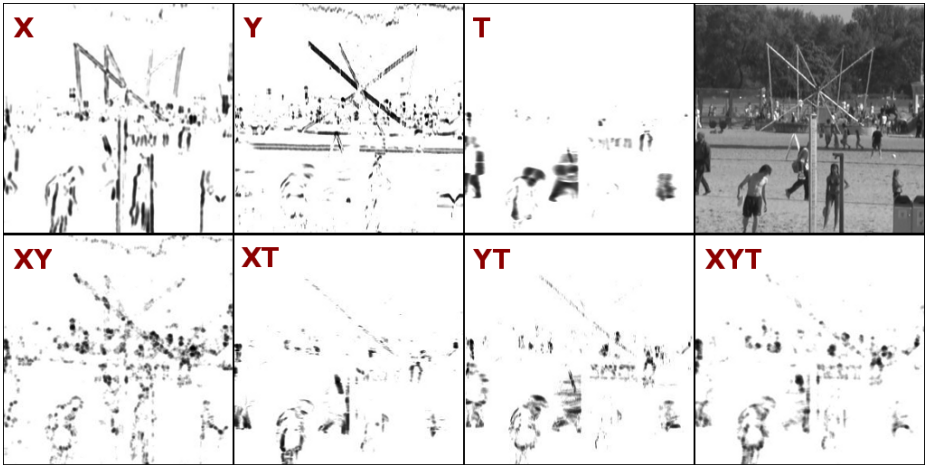


Fig. 1. Top row (from left): H of \mathbf{J}_1 computed along the individual axes x , y , and t ; original frame also shown. Bottom row (from left): K of \mathbf{J}_2 computed along the axes xy , xt , and yt ; below the original image: K of \mathbf{J}_3 along all three axes.

For our evaluation, we used a public data set¹ [3] that consists of 18 high-resolution movie clips (1280 by 720 pixels, 29.97 fps, about 20 s duration each) of natural outdoor scenes, and the gaze data of 54 human subjects freely viewing these videos. From the raw gaze data, collected with an Eye Link II eye tracker at 250 Hz, about 40,000 saccades were extracted. All movies were cropped to the same size along the spatial axes (preserving the central 600 by 600 pixels), to make the resulting space-time cubes rotation-invariant with regard to size (because movies had 600 frames). The total number of saccades that remained after the cropping was 24,370.

Invariants that encode features of higher intrinsic dimensionality were shown to be better predictors of eye movements; therefore, here only these were considered (see Table 1). For each video, we computed the invariants of the tensors \mathbf{J}_1 , \mathbf{J}_2 , and \mathbf{J}_3 along all possible dimensions/combinations of dimensions. See Figure 1 for still shots from a movie and the corresponding invariants. The above invariants were computed on each scale of an anisotropic spatio-temporal multiresolution pyramid with $S = 2$ spatial and $T = 2$ temporal scales, in which each spatial pyramid was decomposed further into its temporal bands.

To determine how well the different representations can predict the saliency level of video regions, next, we labelled areas in the videos as salient and non-salient. The class of salient locations is well defined by the human fixations (more precisely by the saccade landing points). To obtain the non-salient class, a number of biases need to be addressed (e.g. the central fixation bias, the tendency of observers to fixate more in the centre of the display). A common approach in the human vision literature, which we also follow here, is to shuffle

¹ <http://www.inb.uni-luebeck.de/tools-demos/gaze/>

scanpaths: the non-attended locations of a movie are chosen using randomly selected scanpaths from the other movies.

For eye movement prediction, instead of using directly the feature response at these locations, one must consider a spatio-temporal neighbourhood centred around fixations. This is partly accounted for by the image pyramid; however, we further consider a spatial window (of 32 pixels, i.e. about 1.2deg, on the highest pyramid level), as uncertainty in the measurements is higher in the spatial domain. On each pyramid level, we compute the window’s *energy*, i.e. the root-mean-square of the feature values (i.e. invariants) in the window. Thus, we obtain for each salient and non-salient video location a low-dimensional vector of *feature energies* computed on the different pyramid levels (procedure detailed in [8]).

Finally, the predictive power of the different representations is assessed by evaluating (through ROC analysis) the performance of one-dimensional maximum-likelihood classifiers when the feature energies from the single pyramid levels are used as inputs to the decision algorithm. In Table 2, we report average ROC scores (over the 18 movies) obtained for the “most predictive” scale (i.e. the pyramid level with the highest average ROC score).

For comparison, the saliency maps computed by two state-of-the-art algorithms for dynamic scenes (Itti & Koch and SUNDAY [4,9]) are treated as maximum-likelihood classifiers for discriminating between fixated and not fixated video regions. By thresholding these maps, movie regions above the threshold are classified as salient. A systematic variation of the threshold parameter gives us a single ROC curve per movie and model. The averaged ROC scores over all videos are reported in Table 2.

Table 2. Average ROC scores of the different models and representations

Model	ROC score	Model	ROC score	Model	ROC score
	x 0.621		xy 0.639	$\mathbf{J}_3(xyt)$	0.673
\mathbf{J}_1	y 0.617	\mathbf{J}_2	xt 0.637	Itti & Koch	0.644
	t 0.623		yt 0.656	SUNDAY	0.635

4 Discussion and Conclusion

With an average ROC score of 0.673 the three-dimensional structure tensor \mathbf{J}_3 is optimal, suggesting that the most predictive regions of a movie are indeed those where intensity varies along all spatial and temporal dimensions. Surprisingly, the second best predictor operates on the axis pair yt ; this predictor is sensitive for horizontal translations, which are most common in typical natural scenes. \mathbf{J}_2 evaluated on the axes yt outperforms xy and xt by a large margin (with an ROC score of 0.656 compared to 0.639 and 0.637, respectively), and is even superior to the two baseline models with ROC scores 0.644 (Itti & Koch) and 0.635 (SUNDAY), which incorporate a number of different features such as colour,

contrast, and orientation. Although one-dimensional variations perform worst (with \mathbf{J}_1 along the vertical axis giving the lowest score -0.617), their average prediction rate is significantly higher than chance (ROC score of 0.5).

Our results can be used to choose efficient active vision strategies. Under the assumption that the human visual system is near-perfectly optimized for natural environments, the spatio-temporal structure tensor J_3 thus picks the most informative regions. However, with our data, it is now also possible to choose which dimension should be sacrificed for faster computation in resource-limited systems, e.g. in an embedded real-time module of a robot with active vision sensors: for natural environments, the axis pair yt is more informative than xy or xt .

Future work will investigate the predictive power of other tensor representations, such as the Hessian matrix or the energy tensor, and implement the proposed simple saliency models for a real-time system attached to a camera.

Acknowledgments. We would like to thank Karl Gegenfurtner: data were collected in his lab at the Dept. of Psychology of Giessen University. Our research has received funding from the European Commission within the project GazeCom (contract no. IST-C-033816, www.gazecom.eu) of the 6th Framework Programme. All views expressed herein are those of the authors alone; the European Community is not liable for any use made of the information.

References

1. Barth, E., Dorr, M., Vig, E., Pomarjanski, L., Mota, C.: Efficient Coding and Multiple Motions. *Vision Research* (2010), doi:10.1016/j.visres.2010.08.011
2. Bruce, N., Tsotsos, J.K.: Saliency, Attention, and Visual Search: An Information Theoretic Approach. *Journal of Vision* 9(3), 1–24 (2009)
3. Dorr, M., Martinetz, T., Gegenfurtner, K., Barth, E.: Variability of Eye Movements when Viewing Dynamic Natural Scenes. *Journal of Vision* 10(10), 1–17 (2010)
4. Itti, L., Koch, C., Niebur, E.: A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(11), 1254–1259 (1998)
5. Krieger, G., Rentschler, I., Hauske, G., Schill, K., Zetsche, C.: Object and Scene Analysis by Saccadic Eye-Movements: An Investigation with Higher-Order Statistics. *Spatial Vision* 3, 201–214 (2000)
6. Reinagel, P., Zador, A.M.: Natural Scene Statistics at the Centre of Gaze. *Network* 10, 341–350 (1999)
7. Tatler, B.W., Baddeley, R.J., Gilchrist, I.D.: Visual Correlates of Fixation Selection: Effects of Scale and Time. *Vision Research* 45, 643–659 (2005)
8. Vig, E., Dorr, M., Barth, E.: Efficient Visual Coding and the Predictability of Eye Movements on Natural Movies. *Spatial Vision* 22(5), 397–408 (2009)
9. Zhang, L., Tong, M.H., Cottrell, G.W.: SUNDAY: Saliency Using Natural Statistics for Dynamic Analysis of Scenes. In: *Proceedings of the 31st Annual Cognitive Science Conference, Amsterdam, Netherlands* (2009)

State Clustering and Declustering of 3-Regular Graphs with Structural Rewriting

Kohji Tomita and Haruhisa Kurokawa

National Institute of Advanced Industrial Science and Technology (AIST)
Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568 Japan
{k.tomita,kurokawa-h}@aist.go.jp
<http://staff.aist.go.jp/k.tomita>

Abstract. We consider a variant of graph developing system and show various behavior with one type of graph-rewriting. This system is based on rewriting of 3-regular graphs with two possible states per node. We focus on a simple case of fixed number of nodes. The development processes include interaction among rather stable subgraphs. Some simple behaviors, such as clustering and declustering of states, are shown by simulation.

Keywords: cellular automata, graph-rewriting, graph automata, adaptive network.

1 Introduction

Recently, there has been increasing attention on adaptive networks [2]. There, co-evolution of topology (network) and states has been studied in various contexts such as complex networks. In modeling, analyzing or designing systems comprising many elements in full detail, it is important to clarify the dynamics of each element and the relations among elements. In most cases, structures and states are coupled closely in the sense that the global structure constrains the behavior of each element and the behaviors of the elements affect on the structure. One of the interesting behavior of such graph development in our concern is emergence of hierarchy such that graph is decomposed appropriately into subgraphs, and the overall dynamics is described by the interaction among subgraphs and dynamics in subgraphs.

We have been studying a particular class of graph dynamics called graph-rewriting automata [4]. It is an extension of cellular automata to dynamic graph structures. As in cellular automata, the number of neighbors of each cell is unchanging. Graphs are rewritten synchronously according to several types of local rules with changing capabilities of the number of nodes. Though our framework only treats 3-regular graphs [1], it is sufficiently general to represent various behaviors of network evolution such as self-organization or self-reconfiguration including self-replication.

In this paper, we restrict our attention to the behavior of planar graphs with fixed number of nodes and focus on cluster formation. The model is based on

rewriting of 3-regular planar graphs with two possible states per node. As for structural rewriting, we adopt a variant of a rule called commutation in the previous study [4]. It changes local connective relation of a pair of connected nodes. As a simple cluster, we adopt a connected subgraph all of whose nodes have the same state, connected to other subgraphs with different states. We see clusters are formed by a rewriting rule through simulation. In the development process, interaction occurs among clusters whose structure is rather stable.

In the following, formulation of our framework is given, and then simulation setting and results are shown. Finally, the conclusion follows.

2 Graph-Rewriting Automata

Let us introduce the framework that we consider in this paper. It is a variant of graph-rewriting systems, called a graph-rewriting automaton. (Evolution of networks based on similar rewriting rules is discussed in [7].) We assume that the base graph structure is a 3-regular planar graph: each node has three neighbor nodes. Different from ordinary graphs, a cyclic order of links is defined on each node. Each node has an internal state chosen from a finite set. More formally, it is defined as follows. The set of all two element subsets of a set A is denoted by $\mathcal{P}_2(A)$, i.e., $\mathcal{P}_2(A) = \{\{x, y\} | x, y \in A \text{ and } x \neq y\}$. Let $I = \{0, 1, 2\}$.

Definition 1. A base graph G is a triplet $\langle V, E, \xi \rangle$, where V is a finite set of vertices, E is a set of edges defined in the following, and $\xi : V \rightarrow S$ is a function that assigns a state to each vertex. Each edge specifies two incident vertices with link indices I ; more formally, E is a subset of $\mathcal{P}_2(V \times I)$ such that for every $\langle u, i \rangle \in V \times I$ there exists just one $\langle v, j \rangle \in V \times I$ such that $\{\langle u, i \rangle, \langle v, j \rangle\} \in E$.

This definition permits multiple edges or self-edges (loops). Hereafter, base graphs are also called graphs for simplicity. In this paper, we use only two states 0 and 1, i.e., $S = \{0, 1\}$, denoted by white and black nodes respectively in the figures. Also, three incident links of a node are drawn clockwise on this order around the node in sections 2 and 3.

Isomorphism between two graphs are defined by, in addition to the usual condition, that states of the nodes are the same and that cyclic order of the links defined on each node are the same. (As for link indices, only the order rather than the number is in concern.) In the following, isomorphic graphs are identified. An example of non-isomorphic graphs due to the link order is shown in Fig. 1.

We allow two types of rewriting, *cw* and *ccw*, as shown in Fig. 2. The above cyclic order is used to uniquely determine the nodes that are connected to the involved nodes in each structural rewriting.

A link is called applicable if its incident nodes have different states. At each time step, one applicable link is chosen randomly and one rewriting is applied. When the target link is one of double links, a loop is generated as in Fig. 3, in which rewriting is applied to the bold link of the left graph.

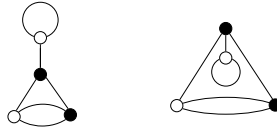


Fig. 1. Non-isomorphic graphs by link order

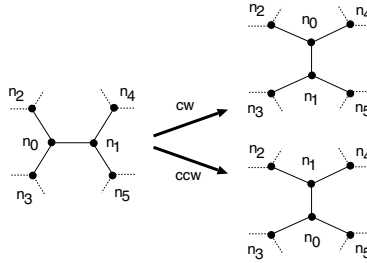


Fig. 2. Structural rewriting

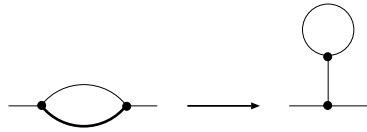


Fig. 3. Application to one of double links

This rewriting is complete for planar connected graphs with the same number of nodes in the following sense. If we omit the states and focus only on the connective relation among nodes, we can rewrite any connected graphs into any connected graph with the same number of nodes by choosing an appropriate sequence of links to which the rewriting is applied.

Depending on the states of target and neighbor nodes, a rewriting rule specifies which rewriting is executed. We use two kinds of rewriting rules, called *phobic* and *philic*, as defined below. In the following, $s(n)$ is the state of node n . Let $s_u = s(n_2) + s(n_4)$ and $s_l = s(n_3) + s(n_5)$.

– *phobic rule*:

$$\begin{cases} \text{cw, if } s(n_0) = 0 \text{ and } s_u < s_l \text{ or } s(n_0) = 1 \text{ and } s_u > s_l, \\ \text{ccw, if } s(n_0) = 0 \text{ and } s_u > s_l \text{ or } s(n_0) = 1 \text{ and } s_u < s_l, \\ \text{randomly chosen, if } s_u = s_l. \end{cases}$$

– *philic rule*:

$$\begin{cases} \text{cw, if } s(n_0) = 0 \text{ and } s_u > s_l \text{ or } s(n_0) = 1 \text{ and } s_u < s_l, \\ \text{ccw, if } s(n_0) = 0 \text{ and } s_u < s_l \text{ or } s(n_0) = 1 \text{ and } s_u > s_l, \\ \text{randomly chosen, if } s_u = s_l. \end{cases}$$

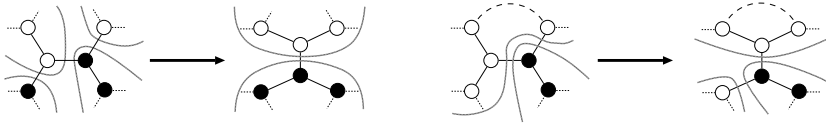


Fig. 4. Cluster merging and separation by phobic rule

The phobic rule tends to phase separation (like oil and water), whereas the philic rule tends to mixing.

3 Mesoscopic View

Graph development processes are generally complicated when a large number of nodes and intricate connective relation among nodes are involved. Instead of viewing the whole graph at the level of individual nodes, viewing it at an appropriate level of abstraction might be useful, if possible. One desirable description would be such that the whole graph is divided into parts, i.e., subgraphs, so that the dynamics of the whole graph is determined by the interaction among subgraphs, and that the interaction is described by a kind of states of the subgraphs. Then the whole behavior could be described at the level of subgraphs. Such will lead to hierarchical description of the graph development processes. This is suitable when graphs are developed in the hierarchical manner like living things with some organizing principle. Similar idea was partly explored as a meta-node in [5], but it was complex and given a priori.

In this paper, we try the simplest case for such description by using two kinds of rules: phobic and philic. As a simple cluster, we adopt a connected subgraph all of whose nodes have the same state, connected to other subgraphs with different states. In the development process, nodes within clusters have the same state, and hence are rather stable because structural rewriting is not applied between the nodes with the same node state. On the other hand, at the interface of two clusters, structural rewriting is performed, which leads to cluster merging and separation (see Fig. 4). The rewriting has different effects to the clusters depending on the global connective relation. In the following, the effect of these rules is examined through simulation.

4 Simulation

We conduct simulation for 10 initial graphs G_1, \dots, G_{10} with varying number of nodes.¹ Each graph includes equal number of nodes with state 0 and 1. G_1 and G_2 are shown, together with G_0 , in Fig. 5 at the individual node level. The

¹ Each graph G_i is obtained by the system in [4] using a rule set 'div 0, (0, 1, 1), 0', 'div 1, (0, 0, 1), 1', and 'com 0, 1' from an initial graph G_0 (in Fig. 5), composed of four nodes, at i -th step.

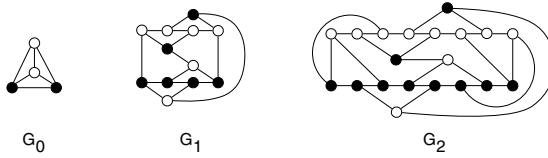


Fig. 5. Graphs for simulation

Table 1. The number of nodes and clusters in initial graphs

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
#nodes	12	20	28	52	84	148	260	428	692	1140
#clusters	6	6	14	18	26	42	66	102	174	282

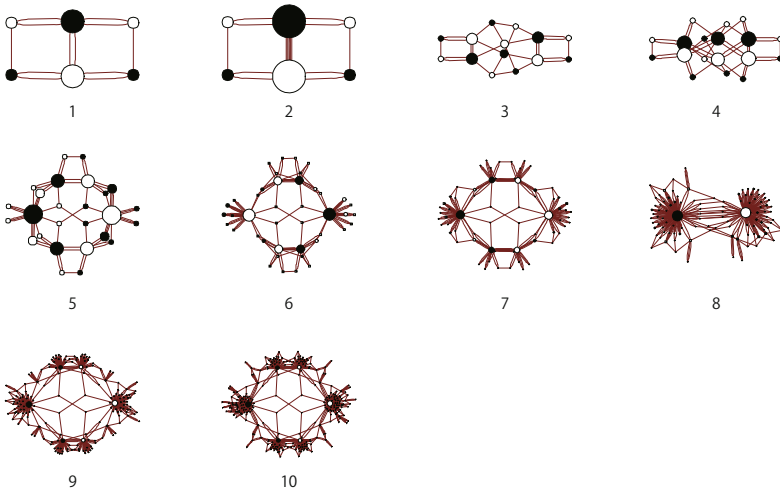


Fig. 6. Initial graphs G_1, \dots, G_{10} at the cluster level

number of nodes and clusters are shown in Table 1. Figure 6 shows the graphs in the cluster level. Each black or white circle indicates one cluster, and its size is proportional to the number of its member nodes.

We conducted simulation of phobic and philic rules for 100 trials from these initial graphs in 10,000,000 steps. Figure 7 indicates the change of average number of clusters by phobic rule. In the development process by phobic update, clustering is performed, i.e., the number of clusters tend to decrease. Depending on the total number of nodes, the number is different in the simulated steps.

The above figure only shows the average number for 100 trials, and the number deviates by each trial. Figure 8 details the result for G_5 with error bars. The bars indicate minimum and maximum numbers. Figure 9 shows an example of cluster

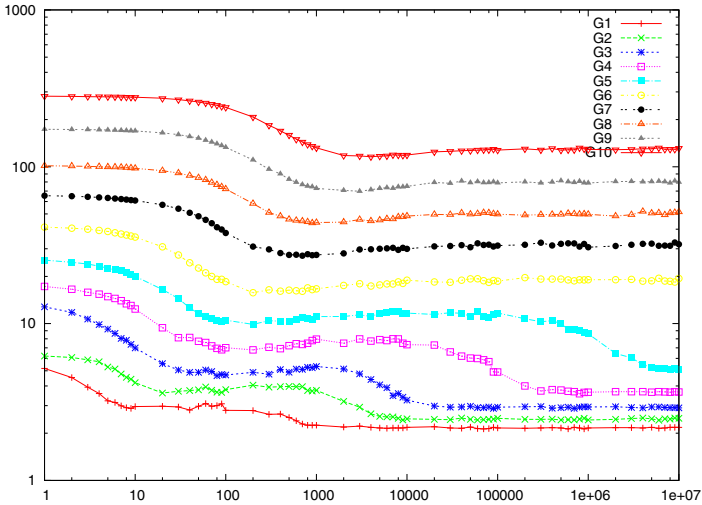


Fig. 7. The average number of clusters by phobic rule

change in one execution at steps 10^i for $i = 1, \dots, 6$. The number of clusters is shown in parentheses. We can observe by these results in Figs. 7, 8 and 9 that the number of clusters once grows before reducing.

Fig. 10 shows the change of average number of clusters for 100 trials by the phobic rule. The number of clusters tend to increase by the phobic rule. Depending on the total number of nodes, the number is different. In this simulation setting, the final number of applicable links was about 65–70% of the total number of links. Some of the obtained graphs from G_5 by phobic rule are shown as clusters in Fig. 11

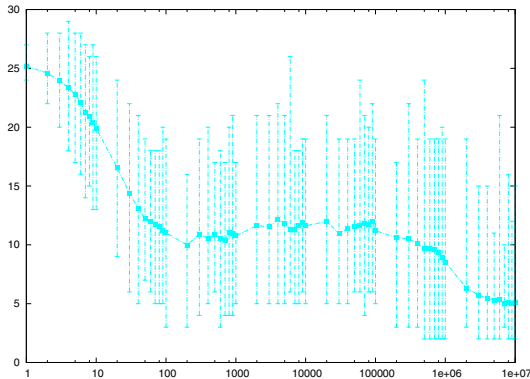


Fig. 8. The number of clusters obtained by phobic rule from G_5 . Error bars indicate minimum and maximum.

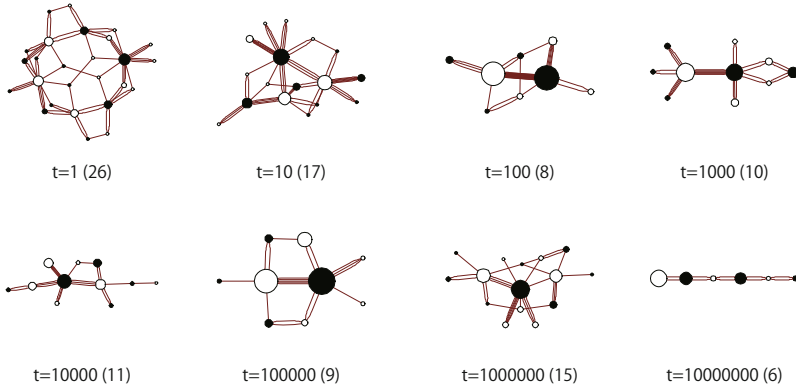


Fig. 9. Clusters obtained by phobic rule from G_5

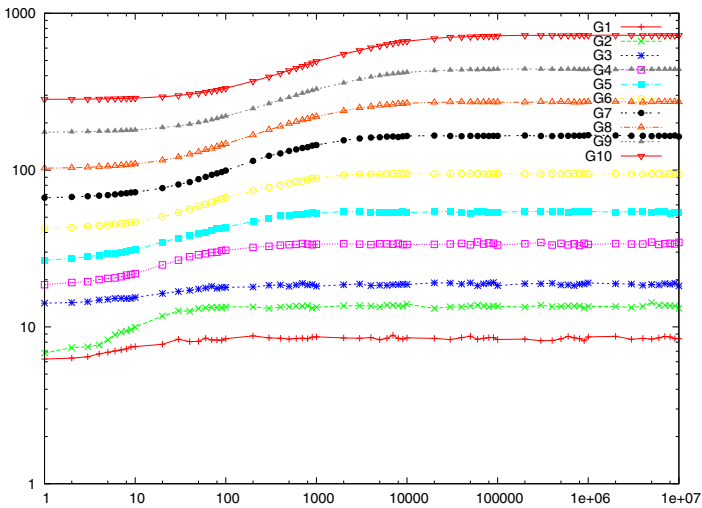


Fig. 10. The average number of clusters by philic rule

5 Conclusion

We have considered a variant of graph developing system and showed behaviors with phobic and philic graph-rewriting rules. This system is based on rewriting of 3-regular graphs with two possible state per node. We focused on a simple case of fixed number of nodes. Simple behaviors, such as clustering and declustering of graphs, are shown by simulation. The phobic rule appears to resulting in clustering, and the philic rule in mixing, regardless of graph and starting state.

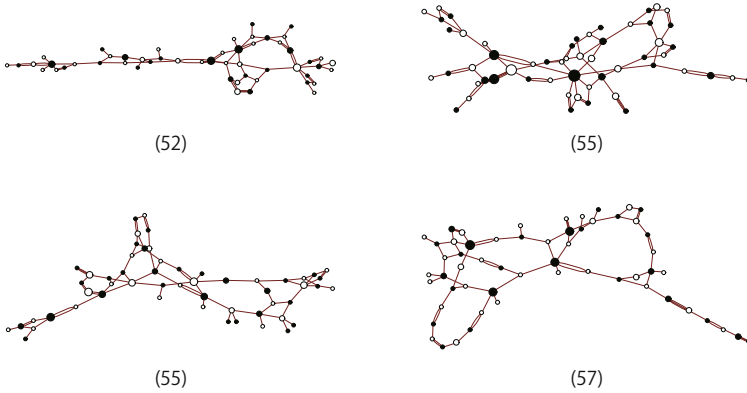


Fig. 11. Examples of obtained clusters by philic rule from G_5

There is much future work. We need to investigate more complex processes. In many cases the phobic rule led to two giant clusters in the simulations, but effects of the initial graphs on the behavior are not examined enough. Also, we need more analysis to clearly describe the dynamic behavior of clusters. Development processes were depicted mainly as the change of the number of clusters. This reflects only a limited aspect of the dynamics. The process of clustering was not straightforward and affected by the global structure. We need to develop better methods for describing the process including dynamic structural changes. In addition, analysis of structures and dynamics in a cluster are also necessary.

Acknowledgment. This work was supported by JSPS KAKENHI (21500231).

References

1. Greenlaw, R., Petreschi, R.: Cubic graphs. *ACM Computing Surveys* 27(4), 471–495 (1995)
2. Gross, T., Sayama, H. (eds.): *Adaptive Networks: Theory, Models and Applications*. Springer (2009)
3. Rozenberg, R. (ed.): *Handbook of Graph Grammars and Computing by Graph Transformation, Foundations*, vol. 1. World Scientific (1997)
4. Tomita, K., Kurokawa, H., Murata, S.: Graph Automata: Natural Expression of Self-Reproduction. *Physica D* 171(4), 197–210 (2002)
5. Tomita, K., Murata, S., Kurokawa, H.: Self-description for construction and computation on graph-rewriting automata. *Artificial Life* 13(4), 383–396 (2007)
6. von Neumann, J.: *Theory of Self-Reproducing Automata*. Univ. of Illinois Press (1966)
7. Wolfram, S.: *A New Kind of Science*. Wolfram Media (2002)

Evaluating Motion Estimation Models from Behavioural and Psychophysical Data

Émilien Tlapale¹, Pierre Kornprobst¹, Jan D. Bouecke²,
Heiko Neumann², and Guillaume S. Masson³

¹ Neuromathcomp Lab, INRIA Sophia Antipolis – Méditerranée
<http://www-sop.inria.fr/neuromathcomp>

² Vision and Perception Science Lab, University of Ulm

<http://www.informatik.uni-ulm.de/ni/mitarbeiter/HNeumann/>

³ DyVA Lab, INCM, UMR 6193 CNRS – Université de la Méditerranée
<http://www.incm.cnrs-mrs.fr/equipedyva.php>

Abstract. Offering proper evaluation methodology is essential to continue progress in modelling the neural mechanisms involved in vision information processing. Currently the evaluation of biologically inspired motion estimation models lacks a proper methodology for comparing their performance against behavioural and psychophysical data. Here we set the basis for such a new benchmark methodology based on human visual performance and designed a database of image sequences taken from neuroscience and psychophysics literature. In this article we focused on two fundamental aspects of motion estimation, which are the respective influence between 1D versus 2D cues and the dynamics of motion integration. Since motion models deal with many kinds of motion representations and scales, we defined two general readouts based on a global motion estimation. Such readouts, namely eye movements and perceived motion, will serve as a reference to compare simulated and experimental data. Baseline results are provided for biologically inspired artificial vision models but also for computer vision models. As a whole we provide here the basis for a valuable evaluation methodology to unravel the fundamental mechanisms of motion perception in the visual cortex. Our database is freely available on the web together with scoring instructions and results at: <http://www-sop.inria.fr/neuromathcomp/psymotionbench>

Keywords: evaluation methodology, biologically inspired artificial vision models, motion estimation, optical flow, motion perception, eye movements.

1 Introduction

Offering proper evaluation methodology is essential to continue progress in modelling the neural mechanisms involved in vision information processing. This general idea has been very well understood and applied in computer vision where

challenging benchmarks are now available for several key problems allowing models to be compared and further improved. For example, motion estimation performance in computer vision increased significantly thanks to several classical benchmarks, which pointed out strength and weaknesses of state of the art approaches [3, 2].

The benchmark for optical flow introduced by Baker and co workers [2] defines a set of challenging image sequences with associated ground truth. The choice of sequences was guided by the needs to evaluate models performance on key difficulties encountered by modellers (motion at objects boundaries, occlusions, non-rigid motions, large displacements). The proposed evaluation methodology consisted of several quantified criteria based on local comparisons between ground truth and output from computer vision models.

Herein we define a motion evaluation methodology where the visual system performance acts as ground truth. Since models aim at elucidating both the computational principles and the computing architectures involved in motion processing, comparing their outputs to biological responses is therefore a strong requirement. Such an evaluation methodology is very different from classical computer vision benchmarks where flow fields are compared together. In our context, the notion of local motion does not make a lot of sense when considering the visual system performance since the purpose of the visual system is not to estimate a dense flow field. Thus defining global readouts is necessary in order to compare output from models with observable quantities measured in neuroscience experiments. Moreover behavioural and perceptual experiments provide numerous types of data such as perceived motion direction and speed or smooth pursuit eye movements.

In this paper we set the basis for a new benchmark methodology which is based on human visual performance. Section 2 describes the main difficulties to design such an evaluation methodology. Section 3 proposes several stimuli and associated readouts selected for motion estimation evaluation. Section 4 presents baseline results. The scoring procedure is illustrated on an example and we show our baseline results which includes both biologically inspired artificial vision models and computer vision models. Section 5 concludes and mentions possible extensions of this work.

Stimuli, scoring procedure and baseline results are available online at:
<http://www-sop.inria.fr/neuromathcomp/psymotionbench>

2 Comparison Difficulties

2.1 Stimulus Parametrisation

Comparing models performance to biological data requires the definition of an homogeneous stimuli set. For example the stimuli are characterised by their physical size, the distance to the observer, and their visual field size. In a benchmark stimuli set we need to ensure a constant mapping between the physical and the

numerical dimensions, otherwise incoherences will occur in the simulated results. Other physical quantities such as duration and luminance also require precise mappings.

2.2 Discretisation

Stimuli discretisation implies the necessity to define properly a scale factor for converting real-world values characteristics into computer parameters. The main problem with discretisation procedures is the *aliasing problem*. One has to make sure that frequency of the input does not cross the Nyquist frequency.

The spatio-temporal discretisation maps the time and the visual field as a succession of discrete images uniformly sample. Such a representation is geometrically different from the log-polar retinotopic disposition of the visual cortex. Moreover the precision of the input has to be sufficient to avoid aliasing problems with the stimuli sizes used by the experimentalists. The finer the precision the larger the data and a good compromise between those two quantities has to be chosen.

Luminance is usually encoded by an eight bits values at each pixel. Such a coarse quantisation is a severe restriction since contrast has a profound impact of the temporal dynamics of most visual percepts and is responsible for many dynamical non-linearities.

2.3 Inhomogeneity of the Motion Representations

All motion models do not have the same motion representation. Their output can be described by global velocity likelihoods [25], velocity distributions at every position [15, 5], filter responses [1], time-correlated spike trains [13], or 2D flow fields [3]. A typical biological model of motion integration might include V1 layers with filter-like responses, MT layers corresponding to local pattern translational motions and MST layers giving indications of global rotation or expansion motions. Because of the variety of motion representations it is necessary to define common observable quantities which are comparable to experimental measurements. These common observable quantities are called *readouts* in both simulations and experiments presented herein.

2.4 Lack of Ground Truth

In computer vision the ground truth is the true velocity field, which is easily defined for synthetic videos, and which can also be estimated for real scene videos. For example in [2], the authors proposed videos of real scenes with the true velocity field. Algorithms can be evaluated based on local comparisons of the 2D flow fields against the estimated flow resulting from different algorithms.

In psychophysical studies the notion of ground truth is less obvious and it is impossible to define it in a strict sense. For example, one has to handle the great variability between subjects or between trials for a single subject. The concision

of data reported in the literature, often a mean and a standard deviation, does not allow the extraction of the statistical laws underlying the data. Moreover many stimuli are bi-stable or multi-stable, and an additional difficulty in defining a ground truth.

Among the set of experimental stimuli studied in neuroscience some provided results at different levels. For instance, the coherence level necessary for perceiving global motion in a random dot patterns has been measured in human subjects but also in single neurons in areas MT, MST, and LIP. In the 2D motion integration example being considered here, a consistent set of global direction estimates have been collected at these different levels as well as for human perception, and monkey and human smooth pursuit [22]. When available these datasets collected for different responses with a single set of motion stimuli should be used to benchmark models.

Given the diversity of the neuroscience experiments, capturing the main properties and results of motion estimation appears to be a complex task. For this reason we restrict our study to a set of fundamental questions described in the following section.

3 Database Design

In this paper we focus on two fundamental aspects of motion integration. Namely, we want to evaluate models performance with stimuli showing the respective influence between 1D versus 2D cues, and the dynamics of motion integration. We chose four stimuli fitting into two classes: line-drawings objects and gratings (see Figure 1). For the purpose of our evaluation we selected stimuli for which smooth pursuit eye movements and motion perception data were available. This section presents the stimuli in more details.

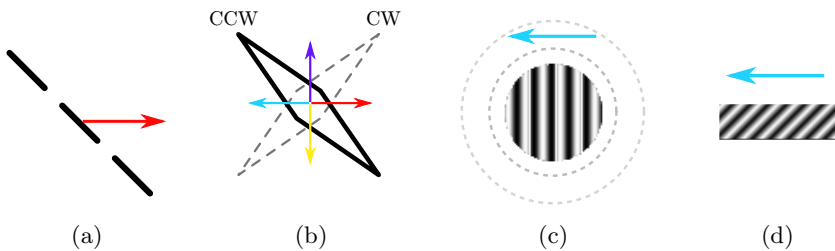


Fig. 1. Database design. The proposed stimuli fit into two classes: line drawings and gratings. (a) Translating bar. (b) Translating diamond. (c) Grating size. (d) Barber pole.

3.1 Line-Drawing Objects

Translating Bars. In [20, 8] the authors consider tilted translating bars. Pursuing a translating bar which true motion direction is not normal to its orientation leads to initial deviation in the *smooth pursuit* eye movement direction. For example, the peak directional error for a tilted bar of 20° length, 0.5° width and 92 cd/m^2 luminance, moving with $16^\circ/\text{s}$ is about 30° [8].

To obtain a model evaluation procedure, the slope of the directional errors could be analysed with respect to bar length, number of bar tiles. Indeed, as the bar length is increased it becomes more complicated to recover its true direction. Likewise it easier to pursue one long bar, if it is tiled into several sub bars [20, 8].

Translating Diamonds. In [24] the authors consider diamond stimuli translating either vertically or horizontally. Due to the local orientations of the diamonds edges with respect to the translating direction, these stimuli mimic type II plaids. Indeed the vector average of the edge motions is biased 44° away from the object's direction. The stimuli thus provide an interesting example to study the influence of 1D and 2D cues on motion integration.

Changing the configuration of the stimulus, by using clockwise (CW) or counter-clockwise (CCW) stimuli, or by varying the direction of the translation, does not influence the ability to pursuit the translating diamonds. In all the cases, the initial pursuit direction as well as the fastest perceptual estimates are biased towards the vector average of the edge motions. It is only after a few hundred milliseconds of exponential direction error decay that the eyes correctly track the object or that human subjects report the correct direction of motion.

3.2 Gratings

Gratings Sizes. In [4] the authors use a drifting grating viewed through a circular aperture. The orientation of the grating is constant and orthogonal to its drifting direction, but the diameter of the circular aperture varies among the stimuli. The authors quantify the change in eye direction during several time windows with respect to the diameter of the aperture. Their goal is to provide a quantitative measure of the spatial summation area, i.e. the smallest diameter leading to the strongest change in eye position. Such spatial summation functions can be seen as a global readout of the motion integration performed in area MT. It is however also possible to look at the perceptual effects of such stimuli: varying sizes of grating patches affect motion detection as well as motion after effect. Many psychophysical studies have been conducted on the perceptual consequences of the centre-surround interactions in early visual areas (see [32] for a review) and it becomes possible to compare these results for the properties of neuronal receptive fields in area V1, MT, or MST in macaque monkeys.

Barber Pole. In the classical barber pole illusion, a translating grating is viewed through a rectangular aperture, leading to two orthogonal sets of 2D

cues [38]. The larger set of 2D cues originates from the longest side of the rectangular aperture, while the smaller set of 2D cues originates from the shortest side. According to psychophysical experiments, as well as neurobiological data, the final perceived motion direction is the same as the orientation of the elongated side of the aperture, after an initial direction orthogonal to the grating orientation [23]. The perceived motion direction thus corresponds to the 2D cues with the greater number of occurrences.

Again, similar observations are available at psychophysical [12, 19] and neuronal [27] levels. It is thus possible to compare model output with a global readout such as time-dependant ocular pursuit but also to compare the dynamics of single model neurons with that of V1 and MT neurons.

4 Results

4.1 Readouts Definition

Common output is necessary in order to compare models together and abstract all implementation dependent issues coming from the large variety of motion models. For example, if eye-movement like output can be defined, then models can be compared together in term of dynamics. As illustrated by this example, our goal here is to propose output formats corresponding to classical readouts as defined in psychophysics. Our goal is to describe for each readout, what they are supposed to measure, and how they are measured in psychophysics.

Since the notion of local measurement has no clear interpretation in term of neural architecture or activity, the readouts defined herein correspond to global motion estimations. For example, it is known that the preferred motion of neurons in both V1 and MT changes depending on the stimulus [28], or that the perceived motion and the neural activity can differ [17]. Considering not only the local estimates but also more distant features makes an important difference with classical computer vision methodology for optical flow where only precision of local estimates matters.

Our goal is to define qualitatively which common outputs are needed from models (i.e. readout inspired from real neuroscience experiments). However no general formula for readouts can be given for three main reasons. The first reason is that the cortical mechanisms leading to a readout from neural activity are usually not clearly established and *a fortiori* it is hard to model them rigorously. The second reason is that the variety of motion representations in models makes it impossible to write a general formula that would be valid for any kind of representation. The last reason is that readouts defined here are sometimes inherent to models. Indeed some models already provide an eye-movement output [25] whereas others considers neural activity in cortical areas [6].

In this article two kinds of evaluation are considered: The *static* evaluation considers only the result at convergence whereas the *dynamic* evaluation focuses on the dynamics of motion integration.

For the *dynamic evaluation*, the readout is expressed as a time independent value, such as a perceived motion direction. Given a stimulus, some experiments

require subjects to give their perception concerning the motion they perceive. In general, this perceived motion readout ignores time evolution. From a modelling point of view, the perceived motion readout can be a global velocity corresponding to the steady state. For the 2D motion integration tasks, we can assume that perceived motion corresponds to the final output from eye movement readout.

For the *static evaluation*, the readout is expressed as a time dependant value, such as smooth pursuit eye movements. Voluntary eye movement to track motion are directly related to our interpretation of the scene in term of motion content. Primates use two types of voluntary eye movement to track objects of interest: smooth pursuit and saccades. Pursuit eye movements are driven by visual motion and rely on both low-level and high-level motion processing. Pursuit initiation is critically dependent upon visual motion processing in cortical areas MT and MST. It presents the interest of being a simple motor responses that requires an accurate estimate of the global direction and speed of a single object, despite its properties such as shape or colour. It is therefore a good probe of object motion processing and in particular it reflects many of the dynamical properties of low level motion computation. From a modelling point of view, smooth pursuit eye movement is a single time-dependent vector, and we only consider the eye direction since speed is generally ignored in experiments.

4.2 Scoring Procedure

The full scoring procedure for each class of stimuli is available online. For each stimulus, instructions are detailed (see for example Figure 2 for translating diamonds stimuli). In order to show the general idea, let us explain what is the scoring procedure for static evaluation of the translating diamonds stimuli. Our reference paper for this case will be [24] as it presents ocular following measurements that we can use in our evaluation.

For a given an approach, our evaluation procedure starts from the estimated global motion direction at every frame and for each of the stimuli in this class, i.e. for translating diamonds translating in one out of the four possible directions (right, up, left, down) and oriented either clockwise (CW) or counter clockwise (CCW). Let us denote by $e_{\mathcal{S}}(t)$ the estimated global direction dynamics for a stimulus \mathcal{S} , with $t \in [0, 450 \text{ ms}]$ and $\mathcal{S} \in \{\text{right, up, left, down}\} \times \{\text{CW, CCW}\}$. From this global estimated direction, the instantaneous direction error $\varepsilon_{\mathcal{S}}(t)$ is defined by

$$\varepsilon_{\mathcal{S}}(t) = e_{\mathcal{S}}(t) - \hat{e}_{\mathcal{S}}(t),$$

where $\hat{e}_{\mathcal{S}}(t)$ is the true object motion. Some results are shown in Figure 3 (a) for the biologically inspired artificial vision model proposed in [35]. Here the estimated global motion direction was obtained from the MT layer activity.

Our goal is to compare this estimated direction error $\varepsilon_{\mathcal{S}}(t)$ to the direction error observed with human subjects (see Figure 3 (b) from [24]). We followed the same procedure as the one defined in [24]. Estimated direction error is fitted with the function.

Translating diamonds: static

Masson, G.S. and Stone, L.S. J Neurophysiol, 2002.

1. For each stimulus, compute global direction

a. The static (final) global motion direction is computed for each stimulus



$$d_s \in [0, 2\pi[\quad \text{with } s \in \{\text{up, down, left, right,}\} \times \{\text{CW, CCW}\}$$

2. For each stimulus, we compute a direction error

- a. We know the true object motion direction e_s for each stimulus.
- b. We can compute the difference between the two

$$e_s = \hat{d}_s - d_s$$

3. We score the direction errors

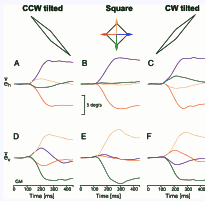
- a. Assuming a normal law with data from the literature, where \hat{e}_s is the average direction error, and σ_s its deviation:

$$k_s = G_{\sigma_s}(e_s - \hat{e}_s)$$

Translating diamonds: dynamic

Masson, G.S. and Stone, L.S. J Neurophysiol, 2002.

1. For each stimulus, extract eye direction error



a. The eye direction dynamics is denoted for each stimulus s by:

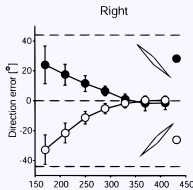
$$e_s(t) \in [0, 2\pi[$$

$$\text{with } t \in [0, 450], s \in \{\text{up, down, left, right,}\} \times \{\text{CW, CCW}\}$$

b. From the veridical object motion, \hat{e}_s , we compute a direction error e_d :

$$e_s(t) = \hat{e}_s(t) - e_s(t)$$

2. For each of the motion direction, we fit the error



a. For a given motion direction, we average absolute errors for the clockwise and counter clockwise orientations.

$$h_d(t) = 0.5 \text{ abs}(e_{d,cw}(t) - e_{d,ccw}(t))$$

b. We fit the result with the following function

$$f_{A,B,\tau}(t) = A \exp\left(-\frac{t}{\tau}\right) + B$$

3. We score the fitted parameters

Assuming a normal law with data from the literature.

Fig. 2. Example of slides describing the scoring procedure (for the translating diamonds stimuli) which are available on the benchmark website

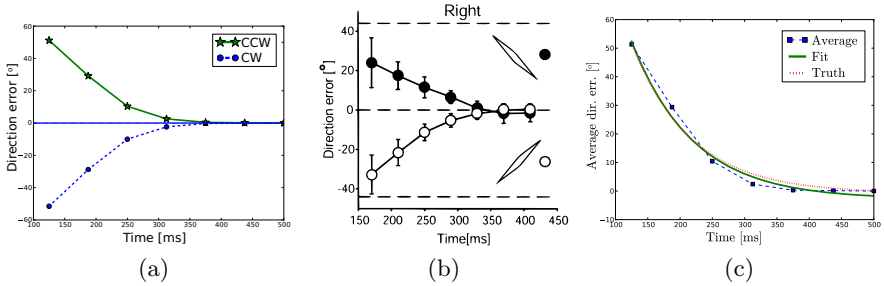


Fig. 3. Scoring procedure for translating diamonds (rightward-motion) (a) Estimated direction error of the biologically inspired artificial vision model proposed in [35]. (b) Oculo-motor dynamics recorded from human subjects. Reproduced from [24]. (c) Comparison between the estimated direction error for all stimuli from the class (Average), the exponential fit on this estimated direction error (Fit) and the fit on the experimental data (Truth).

$$f_{\alpha,\beta,\tau}(t) = \alpha \exp(-t/\tau) + \beta,$$

where α , β , and τ are the fitting parameters to be adjusted. These parameters can then be compared to what is obtained with human observers. In [24] the authors estimated the average values and standard deviations for these parameters (denoted respectively by $\mu(\hat{\eta})$ and $\sigma(\hat{\eta})$ for a parameter $\eta \in \{\alpha, \beta, \tau\}$). Assuming a Gaussian distribution for parameters coming from human subjects and given a stimulus \mathcal{S} , we defined a score $s \in [0, 1]$ for each parameter by:

$$s_{\eta} = \exp(-(\eta - \mu(\hat{\eta}))^2 / \sigma(\hat{\eta})^2), \quad \text{with } \eta \in \{\alpha, \beta, \tau\}.$$

Finally, a global score can be obtained by averaging scores over all stimuli and parameters.

4.3 Baseline Results

We applied our evaluation methodology to both biologically inspired artificial vision models [6, 35] and computer vision models [16, 21, 34, 11] by running either the original implementation from the authors or the code that was available in the OpenCV library [10]. A single set of parameters were experimentally tuned in order to achieve the overall best score across all experiments. As defined in section 4.1, we discuss below the results obtained in the static and dynamic evaluations. For each scenario, results are presented into tables by scores between zero (low performance) and one (high performance). Algorithms are ranked according to their average score across all experiments. Complete details for each stimuli and evaluation procedure can be found on the associated website.

Static Evaluation. Results are presented in Table 1. As a general comment, it is interesting to remark that models performance somewhat follows research

Table 1. Static evaluation results. For each approach and each experiment a score between 0 (worse) and 1 (best) is given depending on the final motion direction error. BMOCV denotes the block matching algorithm found in the OpenCV library.








Approach	Avg.				
TMK-10 [35]	1.00	1.00	1.00	1.00	1.00
SRDB-10 [34]	0.86	1.00	1.00	0.65	0.78
BM-10 [11]	0.74	1.00	1.00	0.00	0.98
BN-07 [6]	0.68	1.00	1.00	0.36	0.38
LK-81 [21]	0.45	0.81	0.00	0.99	0.00
HS-81 [16]	0.39	0.52	0.00	1.00	0.03
BMOCV [10]	0.19	0.00	0.32	0.44	0.00

Table 2. Dynamic evaluation results. For each approach and each experiment a score between 0 (worse) and 1 (best) is given depending on the fitting procedure described in the main text.

Approach	Avg.			
TMK-10 [35]	0.68	1.00	0.96	0.08
LK-81 [21]	0.37	0.75	0.36	0.00
SRDB-10 [34]	0.37	0.75	0.36	0.00
BM-10 [11]	0.35	0.50	0.36	0.18
BN-07 [6]	0.32	0.50	0.39	0.07
BMOVC [10]	0.31	0.50	0.36	0.05
HS-81 [16]	0.26	0.75	0.03	0.00

evolution. For example the seminal approaches for optical flow proposed by Horn and Schunck [16] or Lucas and Kanade [21] show quite a poor performance on most stimuli. The fact that these approaches are differential and not multi-scale largely explains this performance. Being differential, the optical flow is estimated based on the brightness consistency assumption, which is a local indication. Thus when there is a majority of 1D cues, with hardly no texture, the input to differential algorithms is not very informative and leads to an aperture problem that is hard to solve numerically.

Considering multi-scale approaches is today one classical method to solve the aperture problem more efficiently. This solution is now used by most current models, such as the recent models by [34, 11], which are now among the best computer vision models (see the latest results online from [2]). Interestingly, those models also perform well for most of our experiments.

Biologically inspired artificial vision models [6, 35] show high performance. In particular the model proposed in [35] obtains the maximum score. The major strength of this model is that its design is naturally multi-scale as it is inspired from the multi-layer architecture of the brain cortical areas (v_1 and MT) with proper connectivity patterns. This is one important result of this evaluation methodology because it allows to show that taking biology into account can lead to extra performance.

Dynamic Evaluation. Results are presented in Table 2. Studying the dynamical properties of motion integration is quite a new topic and very few biological data (psychophysical, oculo-motor, or neural dynamics) were available when most models were proposed. For this reason, most models were often static, i.e. they were interested in estimating the final percept or optical flow, ignoring how the solution evolves in time to the final percept.

Studying the dynamics has mainly been considered in [35]. This model shows the best performance on translating bars and diamonds, considering the recent neuronal, psychophysical, and behavioural findings [26, 27, 24, 37]. For all other models, since there is no true dynamics, scores are not very informative. Yet, dynamics remain an open issue in the modelling community. None of models tested here performed correctly on the gratings stimulus, suggesting that the underlying mechanisms remain to be found.

5 Conclusion

In this article we set the basis for a new evaluation methodology for motion models which is based on human performance. This work generalises in a rigorous way the evaluation procedures done for most motion models proposed in computational neuroscience. By carefully defining a unified database and proper scoring procedures, it is now possible to perform non-biased comparisons between models, since stimuli are not optimised for a given approach. Our database is freely available on the web together with scoring instructions and results. We provided baseline results for both biologically inspired artificial vision models and state-of-the-art computer vision models. Our results also contribute to show what is the interest to consider biologically inspired models in the computer vision community. By considering stimuli from the psychophysics community, one can further challenge motion models, in addition to existing optical benchmarks.

The proposed evaluation methodology can of course be extended. In this study several properties affecting the motion integration mechanisms were ignored. For instance, disparity used in binocular experiments is missing, and thus it could be probably possible to evaluate other kinds of models where motion and depth are combined [7, 39, 30, 33]. Another property ignored herein is the contrast. In a wide range of psychophysical and neurobiological stimuli, contrast has a considerable effect on motion integration. For instance the receptive field size in areas v_1 MT changes with contrast [31, 29]. Contrast also influences behavioural results [24, 37, 9]. Those kinds of stimuli variations should also be considered in

a more comprehensive data set. One could also think about other stimuli. For instance, it is well known that most of the motion stimuli are multi-stable. In the case of drifting plaids one can perceive either two gratings with different velocities, or one single plaid motion [18]. Incorporating this multi-stability in models is still only at the sketch level in models [14, 36, 35], and mostly ignored in motion benchmarks.

Acknowledgements. The authors particularly thank Pascal Mamassian for enriching thoughts related to this work. This research work has received funding from the Région PACA, the CNRS, the European Community (FACETS, IST-FET, Sixth Framework, N°025213, and SEARISE, Seventh Framework, N°215866), and the Agence Nationale de la Recherche (ANR, NATSTATS).

References

- [1] Adelson, E.H., Bergen, J.R.: Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A* 2(2), 284–299 (1985)
- [2] Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. In: *International Conference on Computer Vision, ICCV 2007* (2007)
- [3] Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. *IJCV* 12(1), 43–77 (1994)
- [4] Barthelemy, F.V., Vanzetta, I., Masson, G.S.: Behavioral receptive field for ocular following in humans: dynamics of spatial summation and center-surround interactions. *Journal of Neurophysiology* 95(6), 3712 (2006)
- [5] Bayerl, P., Neumann, H.: Disambiguating visual motion through contextual feedback modulation. *Neural Computation* 16(10), 2041–2066 (2004)
- [6] Bayerl, P., Neumann, H.: Disambiguating visual motion by form-motion interaction—A computational model. *International Journal of Computer Vision* 72(1), 27–45 (2007)
- [7] Beck, C., Neumann, H.: Interactions of motion and form in visual cortex – a neural model. *Journal of Physiology - Paris* 104, 61–70 (2010)
- [8] Biber, U., Ilg, U.J.: Initiation of smooth-pursuit eye movements by real and illusory contours. *Vision Research* 48(8), 1002–1013 (2008)
- [9] Born, R.T., Pack, C.C., Ponce, C., Yi, S.: Temporal evolution of 2-dimensional direction signals used to guide eye movements. *Journal of Neurophysiology* 95, 284–300 (2006)
- [10] Bradski, G.: *The OpenCV Library*. Dr. Dobb’s Journal of Software Tools (2000)
- [11] Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2010)
- [12] Castet, E., Charton, V., Dufour, A.: The extrinsic/intrinsic classification of two-dimensional motion signals with barber-pole stimuli. *Vision Research* 39(5), 915–932 (1999)
- [13] Cesmeli, E., Lindsey, D.T., Wang, D.L.: An oscillatory correlation model of human motion perception. In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000*, vol. 4 (2000)
- [14] Giese, M.: *Dynamic Neural Field Theory for Motion Perception*. Springer (1998)

- [15] Grossberg, S., Mingolla, E., Pack, C.: A neural model of motion processing and visual navigation by cortical area MST. *Cerebral Cortex* 9(8), 878–895 (1999)
- [16] Horn, B.K.P., Schunck, B.G.: Determining optical flow. *Artificial Intelligence* 17, 185–203 (1981)
- [17] Huang, X., Albright, T.D., Stoner, G.R.: Stimulus dependency and mechanisms of surround modulation in cortical area MT. *Journal of Neuroscience* 28(51), 13889 (2008)
- [18] Hupé, J.M., Rubin, N.: The dynamics of bi-stable alternation in ambiguous motion displays: a fresh look at plaids. *Vision Research* 43(5), 531–548 (2003)
- [19] Kooi, T.L.: Local direction of edge motion causes and abolishes the barberpole illusion. *Vision Research* 33(16), 2347–2351 (1993)
- [20] Lorenceau, J., Shiffrar, M., Wells, N., Castet, E.: Different motion sensitive units are involved in recovering the direction of moving lines. *Vision Research* 33(9), 1207–1217 (1993)
- [21] Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *Proceedings of Imaging Understanding Workshop*, pp. 121–130 (1981)
- [22] Masson, G.S., Ilg, U.J. (eds.): *Dynamics of Visual Motion Processing Neuronal, Behavioral, and Computational Approaches*, 1st edn. Springer (2010)
- [23] Masson, G.S., Rybarczyk, Y., Castet, E., Mestre, D.R.: Temporal dynamics of motion integration for the initiation of tracking eye movements at ultra-short latencies. *Visual Neuroscience* 17(05), 753–767 (2000)
- [24] Masson, G.S., Stone, L.S.: From following edges to pursuing objects. *Journal of Neurophysiology* 88(5), 2869 (2002)
- [25] Montagnini, A., Mamassian, P., Perrinet, L., Castet, E., Masson, G.S.: Bayesian modeling of dynamic motion integration. *Journal of Physiology-Paris* 101(1-3), 64–77 (2007)
- [26] Pack, C.C., Born, R.T.: Temporal dynamics of a neural solution to the aperture problem in visual area MT of macaque brain. *Nature* 409, 1040–1042 (2001)
- [27] Pack, C.C., Gartland, A.J., Born, R.T.: Integration of contour and terminator signals in visual area MT of alert macaque. *The Journal of Neuroscience* 24(13), 3268–3280 (2004)
- [28] Pack, C.C., Hunter, J.N., Born, R.T.: Contrast dependence of suppressive influences in cortical area MT of alert macaque. *Journal of Neurophysiology* 93(3), 1809 (2005)
- [29] Pack, C.C., Hunter, J.N., Born, R.T.: Contrast dependence of suppressive influences in cortical area MT of alert macaque. *Journal of Neurophysiology* 93(3), 1809–1815 (2005)
- [30] Salgado, A., Sánchez, J.: Temporal Constraints in Large Optical Flow Estimation. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) *EUROCAST 2007*. LNCS, vol. 4739, pp. 709–716. Springer, Heidelberg (2007)
- [31] Sceniak, M.P., Ringach, D.L., Hawken, M.J., Shapley, R.: Contrast's effect on spatial summation by macaque V1 neurons. *Nature Neuroscience* 2(8), 733–739 (1999)
- [32] Seriès, P., Georges, S., Lorenceau, J., Frégnac, Y.: A network view of the structure of center/surround modulations of V1 receptive field properties in visual and cortical spaces. *Neurocomputing* 38, 881–888 (2001)
- [33] Strecha, C., Van Gool, L.: Motion - Stereo Integration for Depth Estimation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002, Part II*. LNCS, vol. 2351, pp. 170–185. Springer, Heidelberg (2002)

- [34] Sun, D., Roth, S., Darmstadt, T.U., Black, M.J.: Secrets of Optical Flow Estimation and Their Principles. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition (2010)
- [35] Tlapale, É., Masson, G.S., Kornprobst, P.: Modelling the dynamics of motion integration with a new luminance-gated diffusion mechanism. *Vision Research* 50(17), 1676–1692 (2010)
- [36] Veltz, R., Faugeras, O.: Local/global analysis of the stationary solutions of some neural field equations. *SIAM J. Applied Dynamical Systems* 9(3), 954–998 (2010)
- [37] Wallace, J.M., Stone, L.S., Masson, G.S.: Object motion computation for the initiation of smooth pursuit eye movements in humans. *Journal of Neurophysiology* 93(4), 2279–2293 (2005)
- [38] Wallach, H.: Über visuell wahrgenommene Bewegungsrichtung. *Psychological Research* 20(1), 325–380 (1935)
- [39] Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 optical flow. In: Proceedings of the British Machine Vision Conference, BMVC (2009)

An Algorithm for Automatically Discovering Dynamical Rules of Adaptive Network Evolution from Empirical Data

Hiroki Sayama

Collective Dynamics of Complex Systems Research Group
Departments of Bioengineering & Systems Science and Industrial Engineering
Binghamton University, State University of New York
P.O. Box 6000, Binghamton, NY 13902-6000, USA
sayama@binghamton.edu

Abstract. An algorithm is proposed for automatic discovery of a set of dynamical rules that best captures both state transition and topological transformation in the empirical data showing time evolution of adaptive networks. Graph rewriting systems are used as the basic model framework to represent state transition and topological transformation simultaneously. Network evolution is formulated in two phases: extraction and replacement of subnetworks. For each phase, multiple methods of rule discovery are proposed and will be explored. This paper reports the basic architecture of the algorithm, as well as its implementation and evaluation plan.

Keywords: Adaptive networks, automatic rule discovery, graph rewriting systems, generative network automata, algorithm.

1 Introduction

Modeling and predicting state-topology coevolution in adaptive networks is now becoming well recognized as one of the most significant challenges in complex network research [1-3]. To provide a novel framework for modeling adaptive network dynamics, I proposed to use graph rewriting systems [4,5] as a means of uniform representation of state-topology coevolution. This framework, called Generative Network Automata (GNA), is among the first to systematically integrate graph rewritings in the representation and computation of complex network dynamics that involve both state transition and topological transformation. However, it has remained an open question how one could derive a rule set of a GNA-based model from empirical data of network evolution.

Here I propose an algorithm that automatically discovers a set of dynamical rules that best captures state transition and topological transformation expressed in the empirical data. Network evolution is formulated using the GNA framework and the subnetwork extraction and replacement phases are analyzed separately. Multiple methods are proposed and will be tested for each phase. This paper reports the basic architecture of the algorithm, as well as its implementation and evaluation plan.

2 Generative Network Automata

The theoretical framework used in this paper is Generative Network Automata (GNA) [4,5]. Its working definition is described below.

Configuration: In the GNA framework, a network consists of dynamical nodes and directed links between them. Undirected links can also be represented by a pair of directed links symmetrically placed between nodes. Each node takes one of the (finitely or infinitely many) possible states defined by a node state set S . The links describe referential relationships between the nodes, specifying how the nodes affect each other in state transition and topological transformation. Each link may also take one of the possible states in a link state set S' (not considered within the scope of this paper). A configuration of a GNA at time t is a combination of states and topologies of the network. Formally, it is defined as $G_t = \langle V_t, C_t, L_t \rangle$, where:

- V_t : A finite set of nodes of the network at time t .
- $C_t : V_t \rightarrow S$: Node states at time t .
- $L_t : V_t \rightarrow \{V_t \times S'\}^*$: Links and their states at time t . This maps each node to a list of destinations of outgoing links and the states of those links.

Dynamics: States and topologies of a GNA are updated through repetitive GNA rewriting events, each of which consists of the following three steps:

1. Extraction of part of the GNA (subGNA) that will be subject to change.
2. Production of a new subGNA that will replace the subGNA selected above.
3. Embedding of the new subGNA into the rest of the whole GNA.

The GNA evolution model can be formally defined by the following triplet $\langle E, R, I \rangle$:

- E : An extraction mechanism that determines which part of the GNA is selected for the updating. It is defined as a function that takes the whole GNA configuration and returns a specific subGNA in it to be replaced.
- R : A replacement mechanism that produces a new subGNA from the subGNA selected by E and also specifies the correspondence of nodes between the old and new subGNAs. It is defined as a function that takes a subGNA configuration and returns a pair of a new subGNA configuration and a mapping between nodes in the old subGNA and nodes in the new subGNA.
- I : An initial configuration of the GNA.

The above $\langle E, R, I \rangle$ triplet is sufficient to uniquely define a specific GNA evolution model in this framework. Figure 1 illustrates how these mechanisms work together in a rewriting event.

The function of the extraction and replacement mechanisms (E and R) may be defined as either deterministic or stochastic, as opposed to typical deterministic graph grammatical systems [6]. A stochastic representation of GNA dynamics will be particularly useful when applied to the modeling of real-world complex network data,

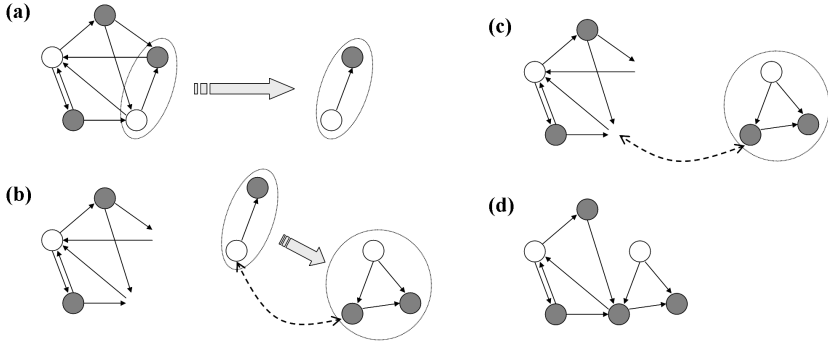


Fig. 1. GNA rewriting process (from [4,5]). (a) The extraction mechanism E selects part of the GNA. (b) The replacement mechanism R produces a new subGNA as a replacement of the old subGNA and also specifies the correspondence of nodes between old and new subGNAs (dashed line). This process may involve both state transition of nodes and transformation of topologies. The “bridge” links that used to exist between the old subGNA and the rest of the GNA remain unconnected and open. (c) The new subGNA produced by R is embedded into the rest of the GNA according to the node correspondence also specified by R . In this particular example, the top gray node in the old subGNA has no corresponding node in the new subGNA, so the bridge links that were connected to that node will be removed. (d) The updated configuration after this rewriting event.

in which a considerable amount of random fluctuations and observation errors are inevitable.

Also, the GNA framework is unique in that the mechanism of subGNA extraction is explicitly described in the formalism as an algorithm E , not implicitly assumed outside the replacement rules like what other graph rewriting systems typically adopt (e.g., [7]). This algorithmic specification makes global rewriting events possible (as well as local rewriting ones) and allows more flexibility in representing diverse network evolution and less computational complexity in implementing their simulations.

3 Proposed Algorithm

In this section, I describe the proposed algorithm for automatic discovery of rewriting rules from network evolution data. Within the scope of this paper, I will simplify the problem by requiring the data to satisfy the following:

1. A given data set is a series of configurations of labeled directed networks in which labels (states) and topologies coevolve over discrete time steps (Fig. 2 (a)).
2. The data set contains information about the correspondence of nodes between every pair of two successive time points (Fig. 2 (a)).
3. States are discrete, finite, and assigned only to nodes, not to links.

4. Changes that take place between successive time points are reasonably small so that they can be identified as one small network rewriting event per each time step.
5. The extraction mechanism E and the replacement mechanism R are memoryless, i.e., they produce outputs solely based on inputs given to them.

Here I note that the GNA framework has a significant advantage for the algorithm design. It formulates the network evolution using two separate phases, i.e., the extraction of subGNA (performed by E) and its replacement (performed by R). Therefore, the estimation and construction of models of E and R can be conducted independently and concurrently using separate training data sets, which will make the algorithm simple and tractable.

A general procedure of the proposed algorithm is explained below (Fig. 2).

- (1) *Preprocess the original network evolution data using data-dependent heuristics, if necessary, so that they meet all the aforementioned requirements.*
- (2) *Detect the difference between each pair of configurations at two successive time points (G_t, G_{t+1}) and represent it as a rewriting event $s_t \equiv r_t$ (Fig. 2 (b)), where s_t is a subGNA to be replaced, r_t is another subGNA that replaces s_t , and “ \equiv ” denotes correspondence from nodes in s_t to nodes in r_t .*

The difference between two configurations ($G_t = \langle V_t, C_t, L_t \rangle$, $G_{t+1} = \langle V_{t+1}, C_{t+1}, L_{t+1} \rangle$) will be detected in the following way:

- i. Let A be a set of nodes in G_t which disappeared in G_{t+1} ($A = \{x \mid x \in V_t \wedge x \notin V_{t+1}\}$).
- ii. Let B be a set of nodes in G_{t+1} which did not exist in G_t ($B = \{x \mid x \in V_{t+1} \wedge x \notin V_t\}$).
- iii. Add to A and B all the nodes whose states or neighbors changed between G_t and G_{t+1}
 $(D = \{x \mid x \in V_t \wedge x \in V_{t+1} \wedge (C_t(x) \neq C_{t+1}(x) \vee L_t(x) \neq L_{t+1}(x))\})$,
 $A = A \cup D$, $B = B \cup D$.

At this point, A and B contain the nodes that experienced some changes (enclosed by solid lines in Fig. 2 (b)).

- iv. Add to A and B all the nodes which have a link to any of the nodes in A ($D' = \{x \mid x \in V_t \wedge x \in V_{t+1} \wedge L_t(x) \cap A \neq \{\emptyset\}\}$, $A = A \cup D'$, $B = B \cup D'$).

The above step includes in A and B additional nodes that may have influenced the rewriting event (enclosed by dashed lines in Fig. 2 (b)).

- v. Let s_t and r_t be subgraphs of G_t and G_{t+1} induced by nodes in A and B , respectively.

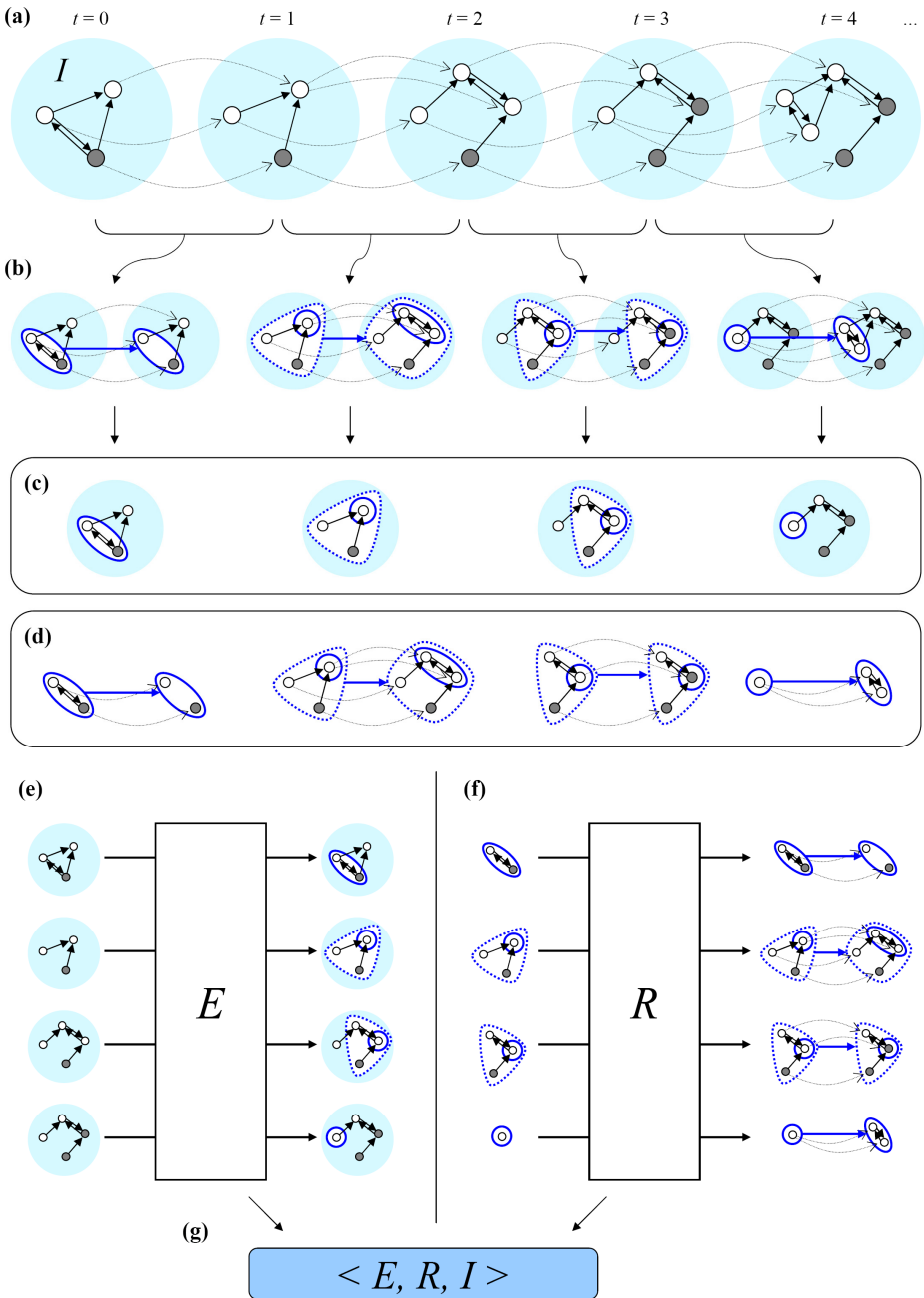


Fig. 2. Overview of the proposed algorithm. (a) Original network evolution data starting with the initial configuration I . (b) Detection of rewriting events at every time step. (c) Training data for the extraction mechanism E . (d) Training data for the replacement mechanism R . (e, f) Construction of models of E and R based on the training data. (g) Final GNA model.

Then the detected rewriting event is represented as $s_i \Rightarrow r_i$, where “ \Rightarrow ” is the set of all the node correspondences between s_i and r_i present in the original data.

- (3) *Construct a model of the extraction mechanism E by using $\{ (G_i, s_i) \}$ as training data, where G_i is the input given to E and s_i the output that E should produce (Fig. 2 (c), (e)).*

This step is the most challenging part in this algorithm development effort. The task to be achieved in this step is to identify an unknown mechanism that chooses a subset of a given set of nodes. Exact identification of an unknown computational mechanism is theoretically not possible in general. Therefore, I will test several heuristic approaches as candidates of practical solutions, including:

i. *Statistical analysis of node properties*

I will first test a simple statistical approach, where correlations will be measured, for the whole training data, between several node properties and the probability of a node to be selected for rewriting. Node properties to be used will include state, degree (absolute or relative) and local clustering coefficient (absolute or relative), among others. If a clear correlation is found, it will be used as the mechanism of E .

ii. *Statistical selection from multiple mechanisms*

In the second approach, I will assume several predefined candidate mechanisms (e.g., random selection, preferential selection based on node degrees, etc.) and calculate the probability for each extraction result given in the training data to occur with each candidate mechanism. If a mechanism includes parameters, they will be optimized to attain the maximal probability. This calculation will be conducted and aggregated for the whole training data to evaluate how likely the given training data could result from each of the candidate mechanisms. Then the mechanism with highest likelihood will be returned as the estimated mechanism of E .

iii. *Parameter estimation for a preprogrammed mechanism*

In some situations the actual extraction mechanism E may be known to the researcher. If this is the case, the mechanism can be preprogrammed in detail, with several parameters left unspecified to give room for fitting to the training data. Then I will use the same probability calculation method as in the second approach to optimize the parameters.

iv. *Evolutionary search with Genetic Programming*

I will also test an evolutionary approach, where possible mechanisms of E will be represented by short pseudo codes and be evolved using Genetic Programming techniques.

- (4) *Construct a model of the replacement mechanism R by using $\{ (s_i, s_i \Rightarrow r_i) \}$ as training data, where s_i is the input given to R and $s_i \Rightarrow r_i$ the output R should produce (Fig. 2 (d), (f)).*

In this step, the task can be achieved in a much simpler manner than in step (3), though technically it still remains identification of an unknown mechanism. This is because a single rewriting event typically involves just a few nodes so the number of possible inputs given to the replacement mechanism R is virtually finite in contrast to the number of possible inputs to E that is virtually infinite. Therefore I will use straightforward pattern matching methods to construct a model of R from the data.

Specifically, the algorithm will construct R as a simple procedure that searches for a rewriting event in the training data whose left hand side matches the given input. If there is only one such event found, the event itself will be the output of R . If multiple events are found, the output will be determined either deterministically (e.g., event with greatest frequency) or stochastically (e.g., random selection with weights set proportional to event frequencies). Or, if no event is found, either identity (“input \Rightarrow input”; no change) will be returned or seek similar events will be sought using partial graph matching schemes.

- (5) *Construct a complete GNA model by combining the results of the above steps (3) and (4) together with the initial configuration I (Fig. 2 (g)).*

4 Summary and Future Work

In this paper, I proposed an algorithm for automatic rule discovery of adaptive network evolution from empirical data, and described its outline as well as multiple candidate methods for some of its components. The implementation of the algorithm is currently ongoing in Python, partly based on the existing NetworkX module [8]. The completed implementation will be made freely available to researchers and other professionals under an open-source license.

The implemented algorithm will be evaluated firstly via application to abstract data generated by artificial GNA models used in my preliminary study [4,5]. The algorithm will be applied to several sample simulation runs selected from those data to check if they could correctly recover the actual network rewriting rules used to produce the data. In this testing, the correct answers are already known, so it will be possible to evaluate the success/failure ratio of each algorithm implementation and revise it to improve the accuracy of its outputs. Several additional complexities will also be introduced into the abstract network models, including more than two states on nodes, more subGNA rewriting options, and simultaneous application of multiple rewriting events, to test and revise the algorithm under model variations. After this initial evaluation is completed, I will also plan to test the algorithm by applying it to real-world network evolution data (such as [9,10]).

As the research progresses, the entire architecture or the details of each step may be revised as needed. At later stages of the project, expansions of the algorithm to broader classes of complex adaptive network dynamics will also be considered. Specifically, I plan to investigate how to incorporate continuous states, how to incorporate links with states, and how to handle the possibility for E or R to have its own internal memory.

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grant No. 1027752.

References

1. Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74, 47–97 (2002)
2. Gross, T., Blasius, B.: Adaptive coevolutionary networks: a review. *J. R. Soc. Interface* 5, 259–271 (2008)
3. Gross, T., Sayama, H. (eds.): *Adaptive Networks: Theory, Models and Applications*. Springer (2009)
4. Sayama, H.: Generative network automata: A generalized framework for modeling complex dynamical systems with autonomously varying topologies. In: *Proceedings of The First IEEE Symposium on Artificial Life*, pp. 214–221. IEEE (2007)
5. Sayama, H., Laramée, C.: Generative network automata: A generalized framework for modeling adaptive network dynamics using graph rewritings. In: Gross, T., Sayama, H. (eds.) *Adaptive Networks: Theory, Models and Applications*, pp. 311–332. Springer (2009)
6. Rozenberg, G. (ed.): *Handbook of Graph Grammars and Computing by Graph Transformation, Foundations*, vol. 1. World Scientific (1997)
7. Kurth, W., Kniemeyer, O., Buck-Sorlin, G.H.: Relational Growth Grammars – A Graph Rewriting Approach to Dynamical Systems with a Dynamical Structure. In: Banâtre, J.-P., Fradet, P., Giavitto, J.-L., Michel, O. (eds.) *UPP 2004. LNCS*, vol. 3566, pp. 56–72. Springer, Heidelberg (2005)
8. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using NetworkX. In: Varoquaux, G., Vaught, T., Millman, J. (eds.) *Proceedings of the 7th Python in Science Conference (SciPy 2008)*, pp. 11–15 (2008), NetworkX, <http://networkx.lanl.gov>
9. Fowler, J.H., Jeon, S.: The authority of Supreme Court precedent. *Social Networks* 30, 16–30 (2008)
10. Fowler, J.H., Jeon, S.: Supreme Court Citation Network Data website, <http://jhfwolwer.ucsd.edu/judicial.htm>

An Evaluation of the Invariance Properties of a Biologically-Inspired System for Unconstrained Face Recognition

Nicolas Pinto¹ and David Cox²

¹ Massachusetts Institute of Technology, Cambridge, MA 02139, USA
pinto@mit.edu

² Rowland Institute at Harvard, Cambridge, MA 02142, USA
cox@rowland.harvard.edu

Abstract. A key challenge in building face recognition systems — biologically-inspired or otherwise — is evaluating performance. While much of face recognition research has traditionally used posed photographs for evaluation, recent efforts have emerged to build more naturalistic, unconstrained test sets by collecting large numbers of face images from the internet (e.g. the “Labeled Faces in the Wild” (LFW) test set [1]). While such efforts represent a large step forward in the direction of realism, the nature of posed photographs from the internet arguably represents an incomplete sampling of the range of variation in view, lighting, etc. found in the real world. Here, we evaluate a family of large-scale biologically-inspired vision algorithms that has previously proven to perform well on a variety of object and face recognition test sets [2], and show that members of this family perform at a level of performance that is comparable with current state-of-the-art approaches on the LFW challenge. As a counterpoint to internet-photo based approaches, we use synthetic (rendered) face images where the amount of view variation is controllable and known by design. We show that while there is gross agreement between the LFW benchmark and synthetic benchmarks, the synthetic benchmarks reveal a substantially greater degree of tolerance to view variation than is apparent from the LFW benchmark in models containing deeper hierarchies. Furthermore, such an approach yields important insights into which axes of variation are most challenging. These results suggest that parametric synthetic benchmarks can play an important role in guiding the progress of biologically-inspired vision systems.

Keywords: biologically-inspired, computer vision, face recognition, performance evaluation.

1 Introduction

The face recognition abilities of biological visual systems are currently unrivaled by artificial systems, particularly in unconstrained environments. A natural strategy that follows from this observation is to seek direct inspiration from biology, building artificial visual systems that attempt to capture aspects of the

computational architecture of the brain, in the hope of eventually mimicking its abilities. Such efforts to model visual computations done by the brain have a long history, at least dating back to Fukushima’s Neocognitron (1980; [3]). More recent experiments with biologically-inspired models have shown them to be highly competitive in a variety of different face and object recognition contexts [4, 5, 6, 7, 8].

Recently, interest in unconstrained face recognition has grown, driven largely by the creation of the *Labeled Faces in the Wild (LFW)* face recognition test set, which has provided a standardized benchmark against which to measure progress. While much work has been done on face recognition in relatively constrained environments (e.g. posed photographs, under controlled lighting conditions [9, 10, 11, 12, 13, 14]), until recently, relatively few available image sets have tackled face recognition in less controlled circumstances. More recently, thanks in large part to the rise of the internet, it has become possible to assemble large collections of face images “in the wild” in the sense that they come from a wide variety of sources and were not posed for the purpose of research.

As in other computer vision domains, biologically-inspired models have achieved highly-competitive performance on the LFW challenge since its inception [7, 15]. More recently, Pinto et al. [2] described a large-scale feature search approach in which thousands of candidate biologically-inspired feature sets are rapidly “screened” to find model architectures that are well suited to a given problem domain. Here, we apply this method to the LFW challenge, and find that it achieves high levels of performance, on par with state-of-the-art methods, even without using any particularly sophisticated machine-learning backend.

However, while these models achieve excellent performance on the LFW challenge set, this set provides little direct insight into why one model performs better than another, and the extent to which the LFW set — which is primarily composed of posed photographs of celebrities — is reflective of the “real” problem of unconstrained face recognition is not entirely clear. In particular, it is not clear that this set contains an accurate sampling of the range of view variation found in the real world [7, 15] since most images are frontal views, and some of the examples of a given individual are taken on the same day, at the same event (e.g. multiple photos of Halle Berry taken from the academy awards ceremony). Thus, while the LFW challenge is clearly useful, and an improvement over more controlled sets, it does not provide an obvious path to the full evaluation of a vision model, nor is it clear how performance on the LFW sets will transfer to other real-world scenarios.

As an complement to the LFW set, we here draw upon carefully-crafted synthetic image sets. While synthetic images have fallen out of favor in the computer vision community in recent years, advances in 3D rendering software have increasingly narrowed the gap between real and synthetic imagery, and rendered images offer several critical advantages over collected photographs. In particular, rendered images allow for complete knowledge and control over the view, position, scale, lighting, presence of other objects etc. in a scene. As a result, synthetic test sets that span whatever range of variation the experimenter

desires can be easily generated, and tasks of parametrically variable difficulty can be constructed. Importantly, such data sets also allow one to specifically test the performance of a model as a function of variation in view, lighting, etc [6]. The ability of a model to tolerate such variation – referred to as “invariance” in the parlance of neuroscience — is a critical property of natural vision systems, and is a key stumbling block in the creation of artificial systems.

2 Methods

2.1 Biologically-Inspired Visual Representations

In the experiments presented below, we studied a family of biologically-inspired visual representations designed to model various stages of visual cortex in the brain.

We used two basic sub-classes of models: 1) *V1-like*, a simple one-layer model with fixed parameters, designed to mimic cortical visual area V1 [6], and 2) multi-layer “High-Throughput” (HT) models, generated by way of a large scale screening approach [2].

Both models classes are characterized by a cascade of linear and nonlinear processing steps (see Figure 1), with *V1-like* having just one layer (and with filters kernels constrained to be Gabor wavelets), and the *HT* models having either two or three layers (referred to hereafter as *HT-L2* and *HT-L3*, respectively).

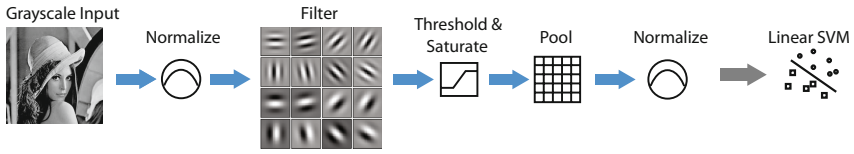
Our *V1-like* implementation was taken without modification from [6, 7]. Similarly, the *HT-L2* and *HT-L3* models were generated according the high-throughput screening approach described in [2] except with randomly generated filters instead of filters trained using an unsupervised learning approach. The details of the *HT-L2* and *HT-L3* models are described in greater detail below.

2.2 High-Throughput-Derived Multilayer Visual Representations: *HT-L2* and *HT-L3*

In this study, we considered the best two- and three-layer models generated from a high-throughput screening model selection procedure. An important feature of the generation of these representations, according to the scheme set forth in [2], is the use of a massively parallel, high-throughput search over the parameter space of possible instances of a large class of biologically-inspired models. Details of this model class and the high-throughput screening (model selection) procedure have been described before [2] but are summarized below for convenience.

Model Architecture: Candidate models were composed of a hierarchy of two (*HT-L2*) or three layers (*HT-L3*), with each layer including a cascade of linear and nonlinear operations that produce successively elaborated nonlinear feature-map representations of the original image. A diagram detailing the flow of operations is shown in Figure 1, and, for the purposes of notation, the cascade of operations is represented as follows:

V1-like



Multi-layer

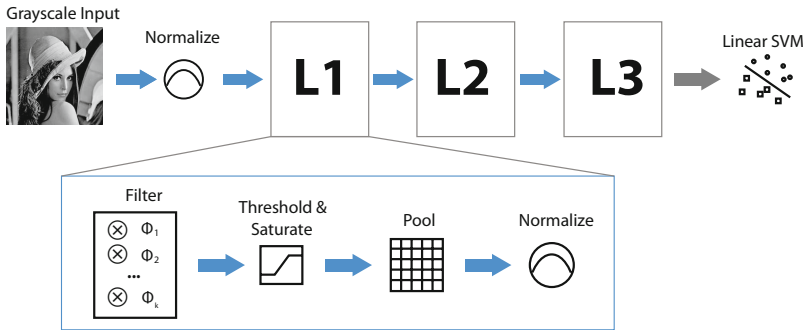


Fig. 1. A schematic diagram of the system architecture of the family of models considered. Each model consists of one to three feedforward filtering layers, with the filters in each layer being applied across the previous layer.

$Layer^0 :$

$$\text{Input} \xrightarrow{\text{Grayscale}} \text{Normalize} \xrightarrow{N^0}$$

$Layer^1 :$

$$N^0 \xrightarrow{\text{Filter}} F^1 \xrightarrow{\text{Activate}} A^1 \xrightarrow{\text{Pool}} P^1 \xrightarrow{\text{Normalize}} N^1$$

and generally, for all $\ell \geq 1$:

$Layer^\ell :$

$$N^{\ell-1} \xrightarrow{\text{Filter}} F^\ell \xrightarrow{\text{Activate}} A^\ell \xrightarrow{\text{Pool}} P^\ell \xrightarrow{\text{Normalize}} N^\ell$$

Details of these steps along with the range of parameter values included in the random search space are described next.

Input and Pre-processing. The input of the *HT-L2* and *HT-L3* models were 100x100 and 200x200 pixel images, respectively. In the pre-processing stage, referred to as $Layer^0$, this input was converted to grayscale and locally normalized:

$$N^0 = \text{Normalize}(\text{Grayscale}(\text{Input})) \tag{1}$$

where the **Normalize** operation is described in detail below. Because this normalization is the final operation of each layer, in the following sections, we refer to $N^{\ell-1}$ as the input of each $Layer^{\ell>0}$ and N^ℓ as the output.

Linear Filtering. The input $N^{\ell-1}$ of each subsequent layer (i.e. $Layer^\ell, \ell \in \{1, 2, 3\}$) was first linearly filtered using a bank of k^ℓ filters to produce a stack of k^ℓ feature maps, denoted F^ℓ . In a biologically-inspired context, this operation is analogous to the weighted integration of synaptic inputs, where each filter in the filterbank applied at a particular image location represents a different cell.

Definitions: The filtering operation for $Layer^\ell$ is denoted:

$$\mathbf{F}^\ell = \mathbf{Filter}(N^{\ell-1}, \Phi^\ell) \tag{2}$$

and produces a stack, F^ℓ , of k^ℓ feature maps, with each map, F_i^ℓ , given by:

$$F_i^\ell = N^{\ell-1} \otimes \Phi_i^\ell \quad \forall i \in \{1, 2, \dots, k^\ell\} \tag{3}$$

where \otimes denotes a correlation of the output of the previous layer, $N^{\ell-1}$ with the filter Φ_i^ℓ (e.g. sliding along the first and second dimensions of $N^{\ell-1}$). Because each successive layer after $Layer^0$ is based on a stack of feature maps, $N^{\ell-1}$ is itself a stack of 2-dimensional feature maps. Thus, the filters contained within Φ^ℓ are, in turn, 3-dimensional, with their third dimension matching the number of filters (and therefore, the number of feature maps) from the previous layer (i.e. $k^{\ell-1}$).

Parameters:

- The filter shapes $f_s^\ell \times f_s^\ell \times f_d^\ell$ were chosen randomly with $f_s^\ell \in \{3, 5, 7, 9\}$ and $f_d^\ell = k^{\ell-1}$.
- Depending on the layer ℓ considered, the number of filters k^ℓ was chosen randomly from the following sets:
 - In $Layer^1$, $k^1 \in \{16, 32, 64\}$
 - In $Layer^2$, $k^2 \in \{16, 32, 64, 128\}$
 - In $Layer^3$, $k^3 \in \{16, 32, 64, 128, 256\}$

All filter kernels were fixed to random values drawn from a uniform distribution.

Activation Function. Filter outputs were subjected to threshold and saturation activation function, wherein output values were clipped to be within a parametrically defined range. This operation is analogous to the spontaneous activity thresholds and firing saturation levels observed in biological neurons.

Definitions: We define the activation function:

$$\mathbf{A}^\ell = \mathbf{Activate}(\mathbf{F}^\ell) \tag{4}$$

that clips the outputs of the filtering step, such that:

$$\mathbf{Activate}(\mathbf{x}) = \begin{cases} \gamma_{max}^\ell & \text{if } x > \gamma_{max}^\ell \\ \gamma_{min}^\ell & \text{if } x < \gamma_{min}^\ell \\ x & \text{otherwise} \end{cases} \tag{5}$$

Where the two parameters γ_{min}^ℓ and γ_{max}^ℓ control the threshold and saturation, respectively. Note that if both minimum and maximum threshold values are $-\infty$ and $+\infty$, the activation is linear (no output is clipped).

Parameters:

- γ_{min}^ℓ was randomly chosen to be $-\infty$ or 0
- γ_{max}^ℓ was randomly chosen to be 1 or $+\infty$

Pooling. The activations of each filter within some neighboring region were then pooled together and the resulting outputs were spatially downsampled.

Definitions: We define the pooling function:

$$\mathbf{P}^\ell = \mathbf{Pool}(\mathbf{A}^\ell) \tag{6}$$

such that:

$$\mathbf{P}_i^\ell = \mathbf{Downsample}_\alpha \left(\sqrt[p^\ell]{(A_i^\ell)^{p^\ell} \odot \mathbf{1}_{a^\ell \times a^\ell}} \right) \tag{7}$$

Where \odot is the 2-dimensional correlation function with $\mathbf{1}_{a^\ell \times a^\ell}$ being an $a^\ell \times a^\ell$ matrix of ones (a^ℓ can be seen as the size of the pooling “neighborhood”). The variable p^ℓ controls the exponents in the pooling function.

Parameters:

- The stride parameter α was fixed to 2, resulting in a downsampling factor of 4.
- The size of the neighborhood a^ℓ was randomly chosen from $\{3, 5, 7, 9\}$.
- The exponent p^ℓ was randomly chosen from $\{1, 2, 10\}$.

Note that for $p^\ell = 1$, this is equivalent to blurring with a $a^\ell \times a^\ell$ boxcar filter. When $p^\ell = 2$ or $p^\ell = 10$ the output is the L^{p^ℓ} -norm □.

Normalization. As a final stage of processing within each layer, the output of the Pooling step was normalized by the activity of their neighbors within some radius (across space and across feature maps). Specifically, each response was divided by the magnitude of the vector of neighboring values if above a given threshold. This operation draws biological inspiration from the competitive interactions observed in natural neuronal systems (e.g. contrast gain control mechanisms in cortical area V1, and elsewhere [16, 17])

Definitions: We define the normalization function:

$$\mathbf{N}^\ell = \mathbf{Normalize}(\mathbf{P}^\ell) \tag{8}$$

such that:

$$N^\ell = \begin{cases} \rho^\ell \cdot C^\ell & \text{if } \rho^\ell \cdot \left\| C^\ell \otimes \mathbf{1}_{b^\ell \times b^\ell \times k^\ell} \right\|_2 < \tau^\ell \\ \frac{C^\ell}{\left\| C^\ell \otimes \mathbf{1}_{b^\ell \times b^\ell \times k^\ell} \right\|_2} & \text{otherwise} \end{cases} \tag{9}$$

¹ The L^{10} -norm produces outputs similar to a *max* operation (i.e. *softmax*).

with

$$C^\ell = P^\ell - \delta^\ell \cdot \frac{P^\ell \otimes \mathbf{1}_{b^\ell \times b^\ell \times k^\ell}}{b^\ell \cdot b^\ell \cdot k^\ell} \quad (10)$$

Where $\delta^\ell \in \{0, 1\}$, \otimes is a 3-dimensional correlation over the “valid” domain (i.e. sliding over the first two dimensions only), and $\mathbf{1}_{b^\ell \times b^\ell \times k^\ell}$ is a $b^\ell \times b^\ell \times k^\ell$ array full of ones. b^ℓ can be seen as the normalization “neighborhood” and δ^ℓ controls if this neighborhood is centered (i.e. subtracting the mean of the vector of neighboring values) before divisive normalization. ρ^ℓ is a “magnitude gain” parameter and τ^ℓ is a threshold parameter below which no divisive normalization occurs.

Parameters:

- The size b^ℓ of the neighborhood region was randomly chosen from $\{3, 5, 7, 9\}$.
- The δ^ℓ parameter was chosen from $\{0, 1\}$.
- The vector of neighboring values could also be stretched by gain values $\rho^\ell \in \{10^{-1}, 10^0, 10^1\}$. Note that when $\rho^\ell = 10^0 = 1$, no gain is applied.
- The threshold value τ^ℓ was randomly chosen from $\{10^{-1}, 10^0, 10^1\}$.

2.3 Final Model Output Dimensionality

The output dimensionality of each candidate model was determined by the number of filters in the final layer, and the x-y “footprint” of the layer (which, in turn, depends on the subsampling at each previous layer). In the model space explored here, the possible output dimensionality ranged from 256 to 73,984.

2.4 Screening (Model Selection)

A total of 5,915 *HT-L2* and 6,917 *HT-L3* models were screened on the *LFW* View 1 “aligned” set [18]. We selected the best model from each “pool” for further analysis on the *LFW* View 2 set (Restricted Protocol). Note that *LFW* View 1 and View 2 do not contain the same individuals and are thus mutually exclusive sets. View 1 was designed as a model selection set while View 2 is used as an independent validation set for the purpose of comparing different methods.

Examples of the screening procedure for *HT-L2* and *HT-L3* models on the *LFW* View 1 task screening task are shown in Figure 2. Performance of randomly generated *HT-L3* models ranged from chance performance (50%) to better than 80% correct; the best five models were drawn from this set and are denoted *HT-L3-1st*, *HT-L3-2nd*, and so on. An analogous procedure was undertaken to generate five two-layer models, denoted *HT-L2-1st*, *HT-L2-2nd*, etc. For the purposes of the present paper, we only considered the best model from each group (i.e. *HT-L2-1st* and *HT-L3-1st*).

2.5 Synthetic Face Images

In order to assess model performance on an image set with a known amount of variation, we generated a set of 3D-rendered face images. 3D face meshes were

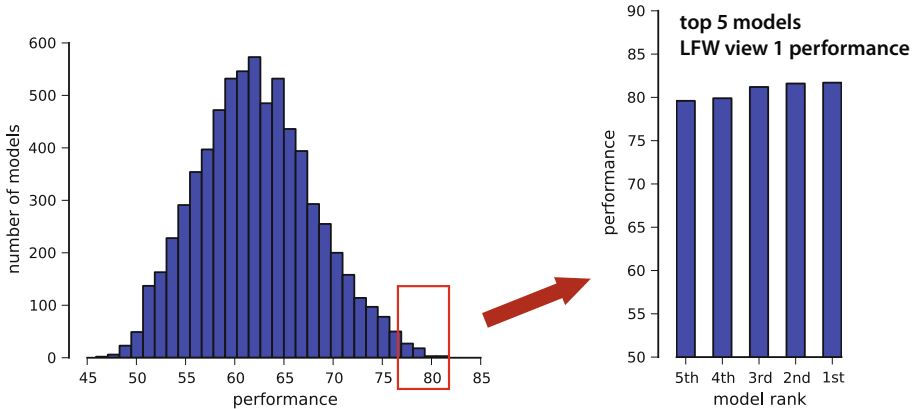


Fig. 2. The high-throughput screening process used to find good representations. Here, data is shown for the screening of *HT-L3* models. A distribution of the performance of approx. 7,000 randomly generated models is shown on the left, with the top five high-performing models replotted on the right. Following screening, the models were evaluated exclusively with sets that do not overlap with the screening set.

Table 1. Performance of the family of biologically-inspired models on the *LFW* challenge set (restricted view 2). For the *HT-L2* and *HT-L3* models, the cross-validated performance of the top 5 randomly-generated models is shown (e.g. 1st, 2nd, etc.). The performance of the simpler single layer *V1-like* model [7] is provided for comparison.

	5th	4th	3rd	2nd	1st
<i>V1-like</i>	77.0 \pm 0.5				
<i>HT-L2</i>	77.8 \pm 0.4	81.3 \pm 0.4	81.5 \pm 0.6	80.8 \pm 0.4	81.0 \pm 0.3
<i>HT-L3</i>	82.8 \pm 0.6	82.3 \pm 0.4	83.3 \pm 0.4	83.9 \pm 0.3	84.1 \pm 0.3

randomly generated using the FaceGen [19] software package and were rendered using the free POV-Ray ray-tracer [20]. For each rendered image, a model rotation (azimuth and elevation), position (x and y), and scale were drawn from a uniform distribution and the models were rendered with a common light source (Figure 3(a)). For the experiments presented here, rotation, size, and position were combined into a single composite “variation level” wherein the variation in the pixel-level euclidean norm was equalized for each kind of variation (e.g. one “unit” of rotation variation produced an equivalent pixel-level change as one “unit” of position variation). Examples of several variation “levels” are shown in Figure 3(a).

The rendered face/head was next composited onto one of four kinds of backgrounds: no background, a white noise background, a phase-scrambled natural background (approximately equivalent to 1/f noise), and a randomly chosen natural background, chosen from a large pool of outdoor background images (Figure 3(b)). Care was taken to ensure that the same background image was never used in more than one final image.

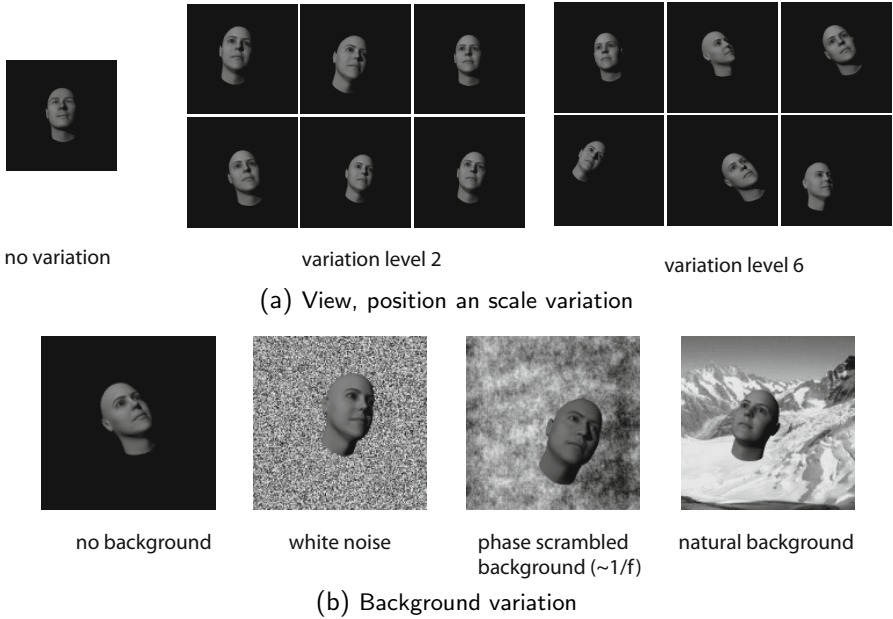


Fig. 3. Synthetic face stimuli

2.6 Classification and Performance Evaluation

To evaluate the performance of a given model with a given stimulus set, we trained a multi-class support vector machine (SVM) classifier [21] using a one-vs-all configuration [22] for each target class. Training and test data were strictly segregated, and performance was evaluated using five 250 train / 50 test random folds of the data. Error bars in all plots show the standard deviation of performance across these five folds.

3 Results

3.1 LFW Performance

Performance on the LFW data set for these models is presented in Table 1. Performance ranged as high as 84.1% percent correct for the best HT-L3 model, achieving performance within a few percent of state-of-the-art methods [23, 24]. While more sophisticated kernel blending techniques have previously been used to achieve better performance on the LFW challenge set by leveraging multiple feature representations (e.g. [15]), we here restrict ourselves here to unblended model performance for the sake of clarity. Further, for simplicity, we also here only consider the best-performing model from each group (i.e. HT-L2-1st and HT-L3-1st).

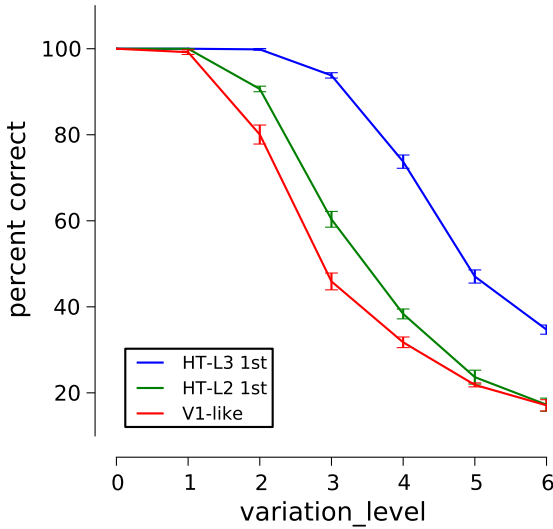


Fig. 4. Model performance on synthetic faces as a function of level of variation

3.2 Performance as a Function of Variation Level

The synthetic face evaluation sets used here provide us with the ability to parametrically control the level of rotation, position and scale variation that our models are required to tolerate. Figure 4 shows the performance the best models from each model class (V1-like, HT-L2, HT-L3) as a function of (composite) variation level for an eight-way face classification task.

3.3 Effect of Number of Faces to Be Discriminated

To further explore the behavior of our models with a controlled stimulus, we examined model performance as a function of the number of faces to be discriminated. In particular, we considered cases with two, four, six, and eight faces. Performance, grouped by model is shown in Figure 5 and is shown grouped by variation level in Figure 6. Predictably, absolute performance level is depressed as a larger number of faces is considered, as is the chance performance level (dotted line). Interestingly, the rate at which performance falls off varies between models as a function of both number of faces to be discriminated, and as a function of variation level. The stability of the performance of the largest/deepest model — HT-L3-1st — is most pronounced when large number of faces and large amounts of variation are considered. Differences between models are far less pronounced with smaller numbers of faces and lesser degrees of variation.

3.4 Effect of Background

To explore the role of background variation, we evaluated model performance with four different background conditions: no background, white-noise back-

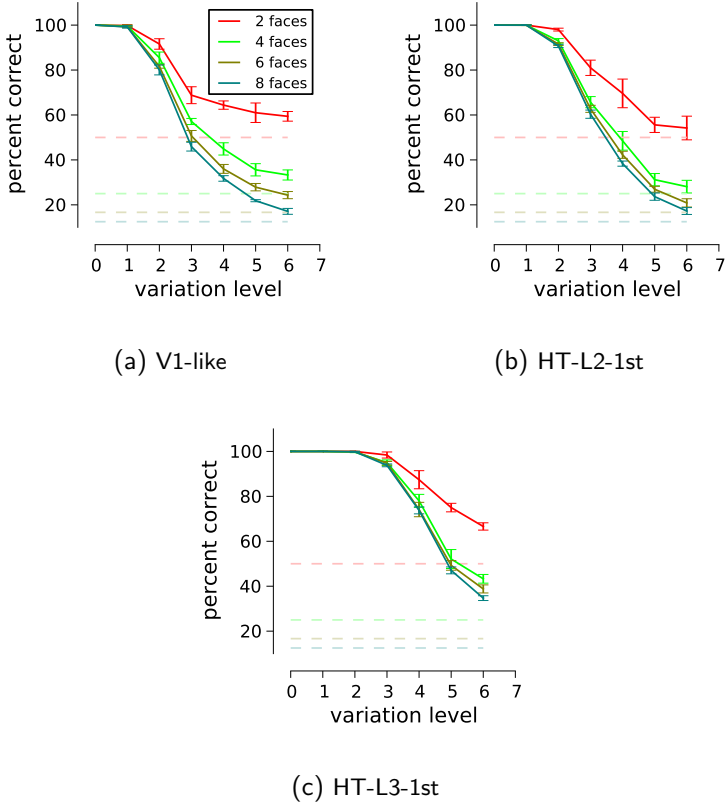


Fig. 5. Effect of number of synthetic faces to be discriminated, sorted by model

ground, phase-scrambled natural backgrounds (i.e. approx. $1/f$ noise), and natural backgrounds. Performance as a function of background and variation level is shown Figure 7. Choice of background was found to have a profound effect on model performance. In the absence of a background, the performance for most models remained high, even at relatively high levels of variation in view, position, and scale (e.g. greater than 90% performance at variation level 4 for the HT-L3-1st and V1-like models). However, the inclusion of any background resulted in a precipitous drop-off in performance for all models, except for the HT-L3-1st model, whose performance degraded gradually. In general, progressively more realistic backgrounds proved increasingly difficult for all models.

4 Discussion

While it is standard practice to test computer vision algorithms with standardized “natural” image test sets such as the LFW set, the performance obtained on such a set provides a relatively narrow window onto behavior of a given system.

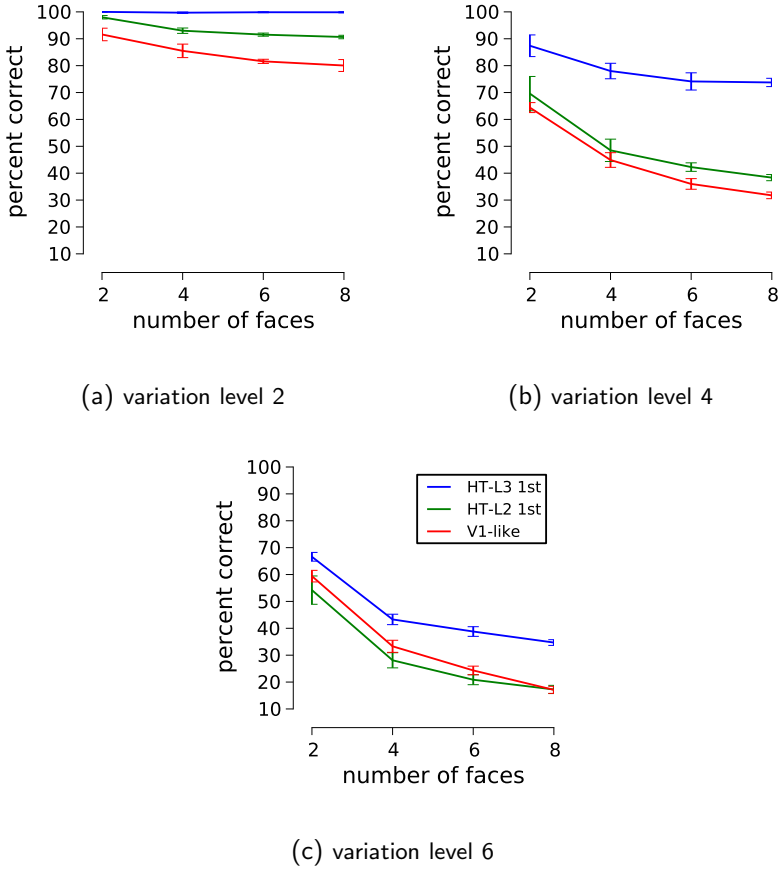


Fig. 6. Effect of number of synthetic faces to be discriminated, sorted by variation level. Note that the performance was 100% in all cases for the zero variation condition (data not shown).

Here, we used synthetic test images, rendered with known amounts of variation, to provide a much richer multidimensional assessment of the invariance properties of a class of models that have achieved high levels of performance on the LFW set.

While the ordinal performance of the one-, two- and three-layer models considered here is roughly the same as is observed for the LFW set (i.e. V1-like < HT-L2 < HT-L3), tests with synthetic sets reveal that the model with the deepest hierarchy (HT-L3) is substantially better able to tolerate variation in view, position, scale and background as compared to the other models considered here. This dramatic difference was not at all apparent from the LFW performance, where the best HT-L3 model performed only a few percent higher than its nearest rivals. While there is no hard evidence one way or another, we speculate that the relatively compressed range of performance between the various models on

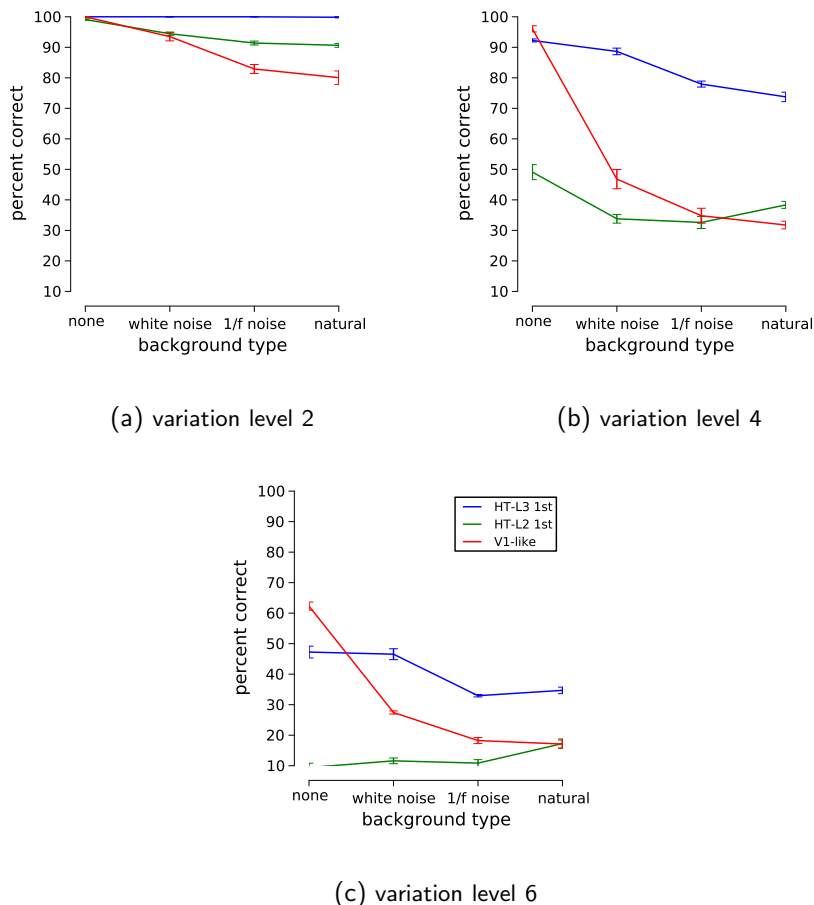


Fig. 7. Effect of background type on performance with synthetic faces. Note that the performance was 100% in all cases for the zero variation condition (data not shown).

the LFW set is reflective of the relatively limited range of view variation found in that set. Indeed, when we examine a relatively low level of variation with our synthetic faces, we see a similarly compressed range of performance variation across the models.

More broadly, our results suggest that the level of variation present in a set, both in terms of view and in terms of background can have a large effect on the “dynamic range” within which one has the ability to distinguish between models. Indeed, without any background, and at low levels of variation, the differences between models can become vanishing small, and in some cases can even reverse. These results underscore the importance of building sets, be they synthetic or natural, that contain more realistic ranges of variation.

References

1. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled Faces in the Wild. TR UMass (2007)
2. Pinto, N., Doukhan, D., DiCarlo, J.J., Cox, D.D.: A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS Comput. Biol.* (2009)
3. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybernetics* (1980)
4. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. In: *PAMI* (2007)
5. Mutch, J., Lowe, D.G.: Object class recognition and localization using sparse features with limited receptive fields. In: *IJCV* (2008)
6. Pinto, N., Cox, D.D., DiCarlo, J.J.: Why is Real-World Visual Object Recognition Hard. *PLoS Comput. Biol.* (2008)
7. Pinto, N., DiCarlo, J.J., Cox, D.D.: Establishing Good Benchmarks and Baselines for Face Recognition. In: *ECCV* (2008)
8. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: *ICCV* (2009)
9. ORL Face Set (2010), <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html> (accessed July 15, 2010)
10. Yale Face Set (2010), <http://cvc.yale.edu> (accessed July 15, 2010)
11. CVL Face Set (2010), <http://www.lrv.fri.uni-lj.si/facedb.html> (accessed July 15, 2010)
12. AR Face Set (2010), <http://www2.ece.ohio-state.edu/~aleix/ardatabase.html> (accessed July 15, 2010)
13. Phillips, P., Moon, H., Rizvi, S., Rauss, P.: The FERET evaluation methodology for face-recognition algorithms. In: *PAMI* (2000)
14. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-PIE. *Image and Vision Computing* (2009)
15. Pinto, N., DiCarlo, J.J., Cox, D.D.: How far can you get with a modern face recognition test set using only simple features? In: *CVPR* (2009)
16. Geisler, W.S., Albrecht, D.G.: Cortical neurons: isolation of contrast gain control. *Vision Research* (1992)
17. Rolls, E.T., Deco, G.: *Computational neuroscience of vision*. Oxford University Press (2002)
18. Taigman, Y., Wolf, L., Hassner, T., Tel-Aviv, I.: Multiple one-shots for utilizing class label information. In: *BMVC* (2009)
19. FaceGen (2010), singularinversions.com (accessed July 15, 2010)
20. POV-Ray Raytracer (2010), www.povray.org (accessed July 15, 2010)
21. Schölkopf, B., Smola, A.: *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. The MIT Press (2002)
22. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. In: *JMLR* (2004)
23. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and Simile Classifiers for Face Verification. In: *ICCV* (2009)
24. Wolf, L., Hassner, T., Taigman, Y.: Similarity Scores Based on Background Samples. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) *ACCV 2009. LNCS*, vol. 5995, pp. 88–97. Springer, Heidelberg (2010)

Epidemic Spread in Adaptive Social Networks with Community Structure

Leah B. Shaw¹ and Ilker Tunc²

¹ College of William and Mary, Williamsburg VA 23187, USA
lbshaw@wm.edu

<http://lbshaw.people.wm.edu/>

² College of William and Mary, Williamsburg VA 23187, USA

Abstract. When an epidemic spreads in a population, individuals may adaptively change the structure of their social contact network in response. We study the spread of epidemics in an adaptive network with community structure. Community structure is a characteristic of social networks that has been neglected in previous adaptive network epidemic models. We model the effect of heterogeneous communities on infection levels. We also show how an epidemic can alter the community structure.

Keywords: adaptive networks, community structure, epidemics.

1 Introduction

We consider an epidemic spreading in a social network in which nodes represent individuals and links their interactions. Node dynamics are assumed to be SIS (susceptible-infected-susceptible). People are assumed to change their social behaviors during an epidemic as in [1]. Avoidance of the disease is implemented by rewiring susceptible nodes away from infected nodes to other susceptible nodes. Previous models for epidemic spread in adaptive networks [1,2,3] have not included community structure. However, it is often thought that most social networks have community structure, in which groups of nodes have high connectivity within the group and relatively few connections to other groups [4].

2 Model and Results

Our model is that of [1] with the following alterations. We use two parameters d and f to generate an initial network with two communities A and B. The parameter d is the probability that when generating a link the link begins with a node in community A. f is the probability that the link connects communities A and B. We fix $d > 0.5$ so that we obtain two heterogeneous communities having different average degrees. By changing f , we change the strength of the community structure. We generally consider strong community structure (few cross connections). When rewiring links, we preserve the community structure by using two rewiring parameters (dependent on f and d) for the probability α

a susceptible node in community A rewires to another susceptible in the same community and likewise with β for B.

We performed Monte Carlo simulations for a system of $N = 10^4$ nodes and $K = 10^5$ links. Further, we derived mean field ordinary differential equations for both node and link dynamics using a moment closure approximation for three point terms as in [12,13].

We find that the extent of community structure affects the infection levels. When the communities are heterogeneous, for example if A has high average degree and B has low average degree, the infection level in the two communities can be either similar or drastically different depending on the extent of community structure and the infection rate p . If the communities are weakly connected (e.g., $f \sim 10^{-3}$), there can exist distinct threshold infection rates p_A, p_B in the two communities. For $p < p_A$, a stable endemic state does not exist. For $p_A < p < p_B$, the infection is endemic in A but appears only stochastically in B. Finally, for $p > p_B$, the infection is endemic in both communities. If on the other hand the two communities are more strongly connected (e.g., $f \sim 0.1$), the infection threshold and infection levels are similar in both communities.

We have also found changes in the community structure as a result of the epidemic spread. Our rewiring rules are chosen such that if SI links occurred randomly throughout the system, the original community structure would be preserved. However, if community A is above the epidemic threshold and community B is below, there are disproportionately more SI links in the A community and fewer in the B community. As a result, there is a net flux of links away from A and toward B until the system reaches a steady state with fewer AA links and more BB links than would be expected from the initial network structure dictated by the parameters d and f . Thus the epidemic spread alters the community structure and has a homogenizing effect.

We have studied both epidemic thresholds and alteration of community structure via both Monte Carlo simulations of the full system and our lower dimensional mean field model. We find both approaches to be in good agreement in their predictions of steady state infection levels and average numbers of links within and between the communities.

References

1. Gross, T., D’Lima, C.J.D., Blasius, B.: Epidemic Dynamics on an Adaptive Network. *Phys. Rev. Lett.* 96, 208701 (2006)
2. Shaw, L.B., Schwartz, I.B.: Fluctuating Epidemics on Adaptive Networks. *Phys. Rev. E* 77, 066101 (2008)
3. Shaw, L.B., Schwartz, I.B.: Enhanced Vaccine Control of Epidemics in Adaptive Networks. *Phys. Rev. E* 81, 046120 (2010)
4. Newman, M.E.J.: The Structure and Function of Complex Networks. *SIAM Review* 45, 167–256 (2003)

A Dynamical Mechanism for the Evolution and Breakdown of Cooperation in the Snowdrift Game in Adaptive Networks

Gerd Zschaler¹, Arne Traulsen², and Thilo Gross¹

¹ Max-Planck-Institut für Physik komplexer Systeme,
Nöthnitzer Str. 38, 01187 Dresden, Germany
zschaler@pks.mpg.de

² Max-Planck-Institut für Evolutionsbiologie,
August-Thienemann-Str. 2, 24306 Plön, Germany

Abstract. We present a novel dynamical mechanism promoting cooperation in the snowdrift game on an adaptive network. In infinite systems, asymptotic full cooperation can be achieved, while sudden breakdowns of highly cooperative states are observed in finite systems.

We consider the evolutionary dynamics of the snowdrift game on an adaptive network, where the coupling between strategy evolution and topological evolution provides a mechanism promoting cooperation *dynamically* and leading to asymptotic full cooperation in the limit of infinite system size.

In our model, the nodes of an undirected network represent agents and the links represent interactions among them. Each agent is assigned a strategy, either unconditional cooperation or unconditional defection. The agents' interactions are described in terms of a pairwise cooperative game, namely the snowdrift game. In this game, an agent receives an abstract payoff that is determined by the pairing of its own and the opponent's strategy. Each agent obtains a total payoff from all the interactions with its neighbors. While mutual cooperation in each interaction is the social optimum, unilateral defection yields a higher payoff and undermines the evolution of cooperation. For this reason, additional mechanisms are typically required to allow cooperators to thrive in evolutionary games [1–3].

The evolution of the system is driven by two processes: the agents can imitate the strategies of successful neighbors or cut the link to a less successful neighbor and reconnect it to a random agent. In contrast to previous work that mainly focused on local update rules [4–6], the agents use non-local information about the general performance of the strategies to assess the success of a strategy in our model. The agents thus update their strategies by imitating what they perceive as the more successful one within their “information horizon”, i.e., within a neighborhood of a fixed radius. In the same way, they rewire their links depending on this non-local information. Both processes are associated with rate constants, which define their relative time scales, and selection intensities, which control how strongly they are influenced by the performance of the agents' strategies. Thus, the strategy dynamics feed back on the topological evolution of the system, rendering it an adaptive network [7].

The adaptive nature of the model leads to a novel dynamical mechanism promoting cooperation. In full simulations of the network, we find oscillations in the number of cooperating players when the link rewiring rate is greater than a critical value. With increasing rewiring rate, or information horizon, the amplitude and period of these oscillations also increase. In the limit of infinite system size, the time-averaged payoff of the cooperators equals that of the defectors in this regime. Nevertheless, for the case of infinite information horizon, a state of full cooperation is approached asymptotically if the rate of topological change exceeds a finite threshold. Using a moment expansion approach and pair approximation [8, 9], we derive a low-dimensional analytical model relating this phenomenon to the bifurcation structure of the underlying dynamical system. Asymptotic full cooperation is thus achieved by the formation of a homoclinic loop in a global bifurcation, which is triggered by sufficiently fast and strongly selective rewiring. Furthermore we show that in finite populations, this mechanism can lead to periods of almost full cooperation interrupted by recurrent collapses to episodes of predominant defection.

A sufficiently large information horizon is necessary to observe the homoclinic transition in our model. We note, however, that this assumption is not necessary for the oscillations to appear, even though it is conceivable that global information can be accessible in social systems through, e.g., general beliefs, rumors, or the mass media. If the agents' information access is restricted to small finite information horizons, the present model still exhibits oscillations. Furthermore, a transition similar to the homoclinic bifurcation was found numerically in a different model [10], in which local update rules are assumed. In our model, asymptotic full cooperation is achieved dynamically by the interplay between topological and strategical evolution, rather than by assembling characteristic "cooperator-friendly" topologies. In finite populations, fluctuations destabilize highly cooperative states, leading to spontaneous collapses of the latter. The homoclinic mechanism in combination with noise may thus be a relevant ingredient in the systemic failure of cooperation that is found in real-world systems [11].

References

- [1] Nowak, M.A., Sigmund, K.: *Science* 303, 793–799 (2004)
- [2] Doebeli, M., Hauert, C.: *Ecol. Lett.* 8, 748–766 (2005)
- [3] Macy, M.W., Flache, A.: *Proc. Natl. Acad. Sci. U. S. A.* 99, 7229–7236 (2002)
- [4] Zimmermann, M.G., Eguíluz, V.M., San Miguel, M., Spadaro, A.: *Adv. Complex Syst.* 3, 283 (2000)
- [5] Santos, F.C., Pacheco, J.M., Lenaerts, T.: *Proc. Natl. Acad. Sci. U. S. A.* 103, 3490–3494 (2006)
- [6] Pacheco, J.M., Traulsen, A., Nowak, M.A.: *Phys. Rev. Lett.* 97, 258103 (2006)
- [7] Gross, T., Blasius, B.: *J. R. Soc. Interface* 5, 259–271 (2008)
- [8] Gross, T., D'Lima, C.J.D., Blasius, B.: *Phys. Rev. Lett.* 96, 208701 (2006)
- [9] Keeling, M.J., Eames, K.T.D.: *J. R. Soc. Interface* 2, 295 (2005)
- [10] Szolnoki, A., Perc, M.: *New J. Phys.* 11, 093033 (2009)
- [11] Turchin, P.: *Historical Dynamics: Why States Rise and Fall*. Princeton Studies in Complexity. Princeton University Press (2003)

Evolutionary Dynamics of Cooperation under the Distributed Prisoner's Dilemma

Flávio L. Pinheiro^{1,2}, Francisco C. Santos^{3,1}, and Jorge M. Pacheco^{4,1}

¹ ATP-group, CMAF, Complexo Interdisciplinar, Lisboa, Portugal

² Centro de Física da Universidade do Minho, Braga, Portugal

³ DEI & INESC-ID, Instituto Superior Técnico, TU Lisbon, Lisboa, Portugal

⁴ Dep. de Matemática e Aplicações, Universidade do Minho, Braga, Portugal

Abstract. Humans contribute to a broad range of cooperative endeavors. In many of them, the amount or effort contributed often depends on the social context of each individual. Recent evidence has shown how modern societies are grounded in complex and heterogeneous networks of exchange and cooperation, in which some individuals play radically different roles and/or interact more than others. We show that such social heterogeneity drastically affects the behavioral dynamics and promotes cooperative behavior, whenever the social dilemma perceived by each individual is contingent on her/his social context. The multiplicity of roles and contributions induced by realistic population structures is shown to transform an initial defection dominance dilemma into a coordination challenge or even a cooperator dominance game. While locally defection may seem inescapable, globally there is an emergent new dilemma in which cooperation often prevails, illustrating how collective cooperative action may emerge from myopic individual selfishness.

Keywords: Cooperation, Complex Networks, Self-Organization, Evolutionary Game Theory.

1 Introduction

Quite often we are confronted with situations in which the act of giving is more important than the amount given. Take for instance a charity event. Some celebrities are usually invited to participate. Their appearance is given maximal audience, and shown contributing a large amount of money. With their media coverage, which is impressive to many, promoters hope to induce a large number of (much smaller) contributions from anonymous (non-celebrities, the overwhelming majority) charity participants, who feel compelled to contribute given the fact that their role model (the celebrity) contributed. Clearly the majority imitates the act of giving and not the amount given.

Many other examples from real life could be provided along similar lines, from trivia, to fads, to stock markets, to Humanitarian causes up to the salvation of planet Earth [1, 2]. Many of these situations provide examples of public goods games (PGG) [3] which are often hard to dissociate from reputation building, social norms and moral principles [4, 5, 6, 7, 8].

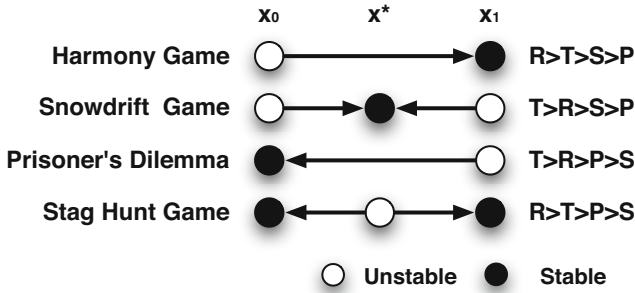


Fig. 1. Fixed points of the Replicator Equation for the four main dilemmas defined by the ranking between the elements of the 2-person game payoff matrix: Reward for mutual cooperation (R), Temptation to defect (T), Sucker’s Payoff (S) and Punishment for mutual defection (P)

When two individuals meet each other in a one shot game where each one can choose between two possible actions, we are in the presence of a situation well known from game theory as a 2 person game involving two strategies. The possible outcomes of such interaction can be summarized in a 2x2 payoff matrix. If the two strategies are to Cooperate and to Defect we typically face a dilemma of cooperation and it is common to write the possible outcomes in the payoff matrix as:

$$\begin{matrix} & C & D \\ C & \begin{pmatrix} R & S \\ T & P \end{pmatrix} & \end{matrix} \tag{1}$$

where R (Reward) and P (Punishment) are the payoffs for mutual cooperation and mutual defection respectively, whereas S (Sucker’s) and T (Temptation) are the payoffs associated with cooperation and defection when the players use different strategies. Depending on the relative value of these four payoffs, four different dilemmas can be defined as illustrated in Figure 1. These dilemmas cover a wide range of situations one encounters in the real world, from social to microbial interactions. In the Evolutionary Game Theory (EGT) framework, we study the dynamics of populations which are conventionally assumed as infinite and well-mixed. The population dynamics is described by the well known replicator equation which has two trivial fixed points ($x = 0, x = 1$) and may, eventually, have one additional interior fixed point (x^*), see Figure 1.

The simplest PGG involves two individuals. Both have the opportunity to contribute a cost c to a common pool. A Cooperator (C) is one who contributes; otherwise she is a Defector (D). The total amount contributed is multiplied by an enhancement factor F and equally shared among the two participants. Hence, player i ($i = 1, 2$) using strategy s_i ($s_i = 1$ if C, 0 if D) gets a payoff $P_i = Fc(s_1 + s_2)/2 - cs_i$ from this game, leading to the following payoff matrix:

$$\begin{array}{cc}
 & C & D \\
 \begin{array}{c} C \\ D \end{array} & \begin{pmatrix} (F-1)c & Fc/2 - c \\ Fc/2 & 0 \end{pmatrix} &
 \end{array} \quad (2)$$

For $F \leq 1$ Ds dominate unconditionally. For $F = 2$ no strategy is favored in well mixed populations (neutral drift); yet, for $F > 2$, it is better to play C despite the fact that, in a mixed pair, a D collects a higher payoff than a C. For $1 < F < 2$ the game falls into the famous symmetric one-shot two-person prisoner's dilemma [9], on which many central results have been obtained over the years, in particular in the context of evolutionary game theory [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31].

In the social dilemma with the payoff matrix given by equation (2), every C pays a fixed cost c per game, providing the corresponding same benefit to the partner. However, if what matters is the act of giving and not the amount given, then there is no reason to assume that everybody contributes the same cost c to each game. Depending on the amount associated with each individual contribution, the overall result of the evolutionary dynamics may change [32, 33]. The two person game introduced above provides not only the ideal ground to introduce such a diversity of contributions, but also an intuitive coupling between game dynamics and social structure: The first (second) individual contributes a cost c_1 (c_2) if playing C and nothing otherwise. Hence, player i ($i = 1, 2$) now gets the following payoff from this game:

$$P_i = F(c_1 s_1 + c_2 s_2)/2 - c_i s_i \quad (3)$$

reflecting the symmetry breaking induced by possibly different contributions from different cooperating individuals. This poses a natural question: Who will acquire an evolutionary edge under these conditions?

Often the amount that each individual contributes is correlated with the social context she is actually embedded in [21, 34]. Modern communities are grounded in complex social networks of investment and cooperation, in which some individuals play radically different roles and interact more and more often than others. Empirical studies have demonstrated that social networks share both small-world properties and heterogeneous degree distribution [35, 36, 37]. In such heterogeneous communities, where different individuals may be embedded in very different social environments, it is hard to imagine that every C will always provide the same amount in every game interaction, hence reducing the problem to the standard two-person prisoner's dilemma studied so far. In the context of N-person games played in prototypical social networks, it has been found that the diversity of contributions greatly favors cooperation. However, and similar to the relation between two-body and many-body interactions in the Physical Sciences, N-person public goods games have an intrinsic complexity which cannot be anticipated from two-person games. As such it is not clear in which way heterogeneous networks, which naturally induce the symmetry breaking alluded to before, enhance or inhibit the evolution of cooperation.

2 Model and Methods

We will consider finite populations structured by means of complex networks, where each node represents an individual, with links representing interactions between them.

Our study will be focused on three topologies: Scale-Free (SF), generated using a direct implementation of the Barabási-Albert (BA) model based on growth and preferential attachment [38]; Exponential (EXP), generated replacing the preferential attachment by uniform attachment in the BA model. Both these networks are heterogeneous, as in general different nodes will exhibit different degree. Finally, we shall also consider Regular networks (REG), which are homogeneous in the sense that all nodes exhibit the same degree. It is known that social networks fall somewhere between the limits of SF and REG networks [39], and hence the choice of studying an intermediate class represented by the EXP networks.

We will study two regimes in the framework of the Prisoner's Dilemma: the conventional one (CPD) where each C contributes the same cost c to each game she plays, and the distributed one (DPD) where Cs now equally distribute a given cost c among all games they play.

In each time-step, every individual engages in a 2-person PGG with each of his neighbors. The accumulated payoff from all interactions is associated with reproductive fitness (f_i) or social success, which determines the behavior in the next generation. We adopt the so-called pairwise comparison rule [40, 41, 42] for the social learning dynamics: Each individual (x) copies the behavior of a randomly chosen neighbor (y) with a probability given by the Fermi distribution from statistical physics:

$$p = (1 + e^{-\beta(f_y - f_x)})^{-1} \quad (4)$$

in which β , known as the intensity of selection, plays the same role as the inverse of temperature in Physics. By this definition the probability increases with the fitness difference. For $\beta \ll 1$ we fall into the weak selection regime, as selection provides a small perturbation to random drift; opposite to this scenario, imitation dynamics is obtained for high values of β . Throughout this work we adopt the strong selection regime ($\beta = 10.0$) as this enhances the influence of the individual fitness (and hence the role played by the social network) in the selection process. In all our simulations we will keep $c = 1$.

3 Results

Figure 2 shows the final fraction of cooperators, corresponding to the average, over 10^3 runs and networks, of the value obtained for this fraction, in each run, after 10^5 generations. This is done for each point (F and degree) and for the three different population structures.

Figure 2A shows the outcome of evolving the conventional 2-person PD ($1 < F < 2$), in which case each player contributes a fixed amount c to each

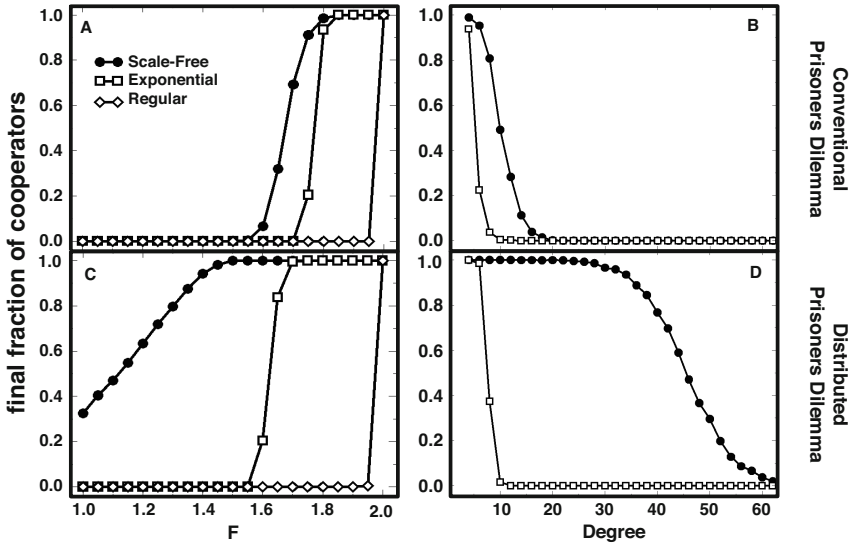


Fig. 2. Final fraction of cooperators as a function of i) the enhancement factor F (panels A and C) and ii) the degree for the EXP and SF networks with $F = 1.8$ (panels B and D). Panel A: Under CPD Cooperation is able to dominate on SF networks (filled circles), unlike what happens on REG structures (empty squares). On exponential networks, intermediate levels of cooperation emerge, as a result of the heterogeneity of such topologies (empty diamonds). Panel C: Under DPD the advantage of Cs is dramatically enhanced when the same cost is evenly shared among each neighbor. As expected, abandoning the well-mixed regime leads to a break-up of neutrality for $F = 2$. Panels B and D : Cooperation is able to dominate on sparse networks. Yet only under DPD, combined with high levels of heterogeneity attained on Scale-free networks, one observes the maintenance of cooperative behavior in highly connected populations. The results were obtained for networks of 10^3 nodes and variable average degree ($z = 4$ in panels A and C) starting with 50% of Cs randomly distributed in the population.

game she participates. The existence of a minority of highly connected individuals in SF networks (line and filled circles) allows the population to preserve high cooperative standards, while on homogeneous networks (line and empty diamonds), Ds dominate for the entire range of parameters, as a result of the pairwise comparison rule adopted [43]. Heterogeneous networks thus pave the way for the emergence of cooperation. Highly connected individuals (i.e. hubs) work as catalysers of cooperative behavior, as their large number of interactions allows them to accumulate a high fitness. This, in turn, leads them to act as role models for a large number of social ties. To the extent that hubs are Cs, they influence the vast majority of the population to follow their behavior. Clearly, this feature has a stronger impact on SF networks than on EXP networks, the difference between these two types of networks stemming from the presence or absence, respectively, of the preferential attachment mechanism.

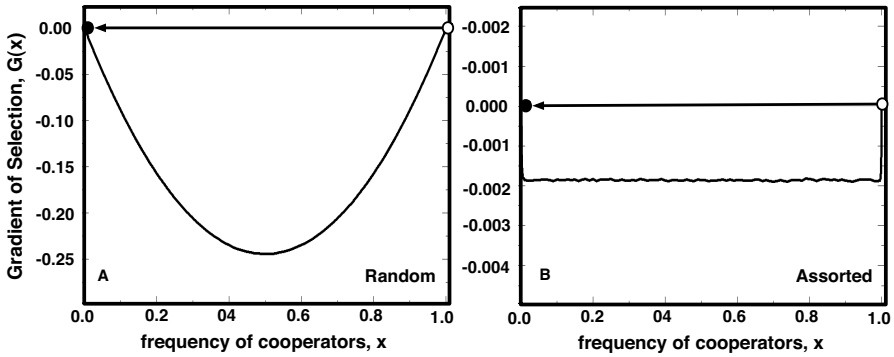


Fig. 3. The Gradients of Selection of Regular Networks for $F = 1.25$ for both random and assorted distributions of strategies. For both distributions the $G(x)$ is always negative, meaning, that the features of in the initial prisoner's dilemma remain intact at a population-wide level.

The results in Figure 2A and 2B are based on a CPD assumption while Figure 2C and 2D considers the DPD dilemma. While on homogeneous networks the fate of cooperation is the same as before - it amounts to rescaling the intensity of selection - heterogeneity in the amount contributed by each individual to each game creates a remarkable boost in the final number of Cs for the entire range of F , which increases with increasing heterogeneity of the underlying network. Comparison with the results of Figure 2A shows that under DPD preferential attachment plays a prominent role, since it constitutes the network wiring mechanism distinguishing EXP networks from SF networks. Changing from CPD to DPD induces moderate boosts in the equilibrium fraction of Cs on EXP networks, but a spectacular boost of cooperation on SF networks: Hubs become extremely influential under the DPD.

The previous analysis shows that heterogeneous networks boost cooperation (Figure 2). A close-up analysis to how the structures that make up such heterogeneous networks, the hubs, interact with each other gives us a glimpse of the way in which they favor cooperation, as a result of the symmetry breaking game dynamics [44]; the same analysis carried out on homogeneous networks is useless as symmetry is kept intact on such networks.

In order to probe deeper into the mechanism(s) underlying the prevalence of cooperators in the DPD, we start by defining the finite population analog $G(x)$ of the gradient of selection under the replicator dynamics ($x(1-x)(f_c - f_d)$): $G(x) = T^+(x) - T^-(x)$, where $T^+(x)$ ($T^-(x)$) is the average frequency of transitions increasing (decreasing) the number of Cs for each random configuration with xN Cs, valid for any population size and structure. This is a mean-field variable. Consequently, doing so we overlook the microscopic details of the competition and self-organization of Cs and Ds, but we gain an overview of the game dynamics in a mean-field perspective. G becomes positive whenever cooperation is favored by evolution and negative otherwise. Whenever $G = 0$, selection

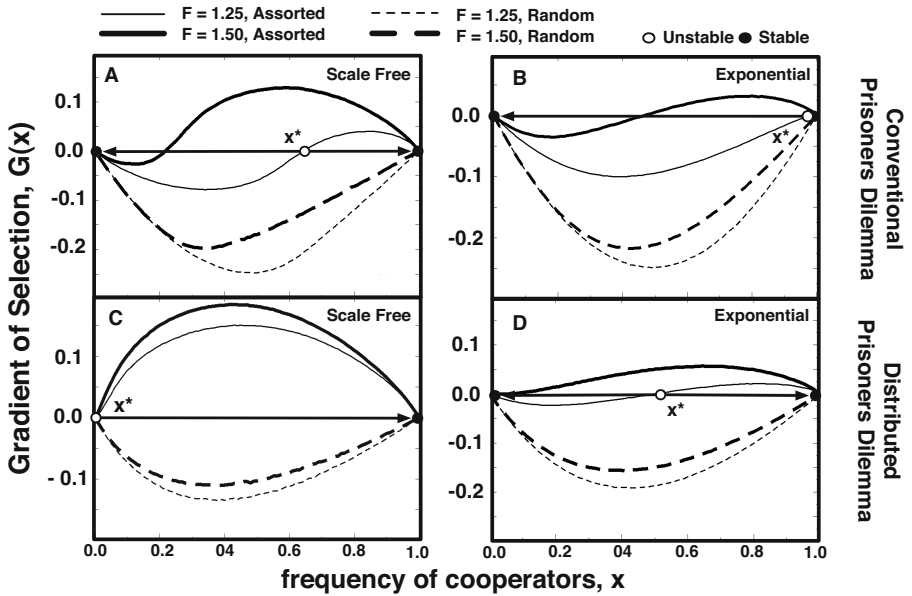


Fig. 4. Gradients of Selection $G(x)$ for $F = 1.25$ (thin lines) and $F = 1.50$ (thick lines) for random (dashed lines) and assorted (solid lines) distributions of strategies for both types of heterogeneous networks (EXP and SF). Under the CPD paradigm (panels A and B) and with the appropriate value of F , heterogeneous networks lead to the appearance of an unstable fixed point x^* (open circles) characteristic of a coordination game. Under DPD (panels C and D), the change in the effective game is even more marked and in the case of SF networks the game transformation occurs between a $G(x)$ always negative (prisoner's dilemma) to a scenario where it is positive for most values of x leading to a scenario characteristic of a Harmony game, where cooperators dominate unconditionally. In both panels the networks employed had 10^3 nodes and an average degree $z = 4$, and $\beta = 10.0$.

becomes neutral and evolution proceeds mainly by random drift. Naturally, G will depend implicitly on the population structure, on the fraction x of Cs and also on how these Cs are spread in the network.

At start each individual in the population is assigned a strategy (C or D) randomly, with equal probability, such that correlations between individuals with the same strategy are not favored. When the population evolves such correlations are expected to build up, as Cs breed Cs and Ds breed Ds [25]. Hence one expects that a possible outcome of evolution is the assortation of strategies where each C(D) has, at least, one C(D) in his neighborhood. This distribution of Cs and Ds replicates the observations of all numerical simulations we have done. For that reason we will always compare the results of random strategy configurations against $G(x)$ of assorted strategy configurations (Figures 3 and 4).

In Figures 3 and 4 we plot the $G(x)$ as a function of x for both strategy distributions, for two different values of F and both prisoner's dilemma paradigms (CPD and DPD).

On homogeneous networks (Figure 3) Ds are always advantageous. On heterogeneous networks the initial prisoner's dilemma (dashed lines) is effectively transformed into a different game (full lines). Figure 4A and 4B indicates that, in the case of CPD, introducing diversity in roles and positions in the social network effectively leads to a coordination game, characterized (in an infinite, well-mixed population) by a critical fraction x^* above which Cs are always advantageous ($G < 0$ for $x < x^*$ and $G > 0$ for $x > x^*$, Figure 1). This result provides a powerful qualitative rationale for many results obtained previously on heterogeneous networks under strong selection [16, 17, 21] in which degree heterogeneity is shown to induce cooperative behavior, inasmuch as the initial fraction of Cs is sufficient to overcome the coordination threshold. Moreover, Figure 4C shows that changing the contributive scheme from CPD to DPD in SF population structures acts to change a prisoner's dilemma effectively into a Harmony game where Cs become advantageous irrespectively of the fraction of Cs ($x^* \approx 0$).

4 Conclusion

The present study puts in evidence the impact of breaking the symmetry of cooperative contributions to the same two-person game. On strongly heterogeneous networks, the results of Figure 2 provide an impressive account of the impact of this diversity of contributions. Our results suggest that whenever the act of cooperation is associated to the act of contributing, and not to the amount contributed, cooperation blooms insofar as the structure of the social web is heterogeneous, leading individuals to play diverse roles. The multiplicity of roles and contributions induced by the social structure effectively transforms a local cooperative dilemma into a global coordination game [45] our results provides additional evidence that the assortment of strategies arising from the intricate nature of collective dynamics of cooperation in a complex network leads to a change in the effective game played by the population and for that reason, while locally cooperation can be understood as a prisoner's dilemma, globally it is effectively described by a coordination dilemma [45].

Acknowledgements. The authors acknowledge financial support from FCT-Portugal. The authors would also like to thank Marta D. Santos and Sven Van Segbroeck for useful discussions.

References

1. Milinski, M., Semmann, D., Krambeck, H.J., Marotzke, J.: Stabilizing the earth's climate is not a losing game: Supporting evidence from public goods experiments. Proc. Natl. Acad. Sci. U S A 103, 3994–3998 (2006)

2. Dreber, A., Nowak, M.: Gambling for global goods. *Proc. Natl. Acad. Sci. U S A* 105, 2261–2262 (2008)
3. Hardin, G.: The tragedy of the commons. *Science* 162, 1243–1248 (1968)
4. Nowak, M.A., Sigmund, K.: Evolution of indirect reciprocity by image scoring. *Nature* 393, 573–577 (1998)
5. Ohtsuki, H., Iwasa, Y.: How should we define goodness? reputation dynamics in indirect reciprocity. *J. Theor. Biol.* 231, 107–120 (2004)
6. Nowak, M.A., Sigmund, K.: Evolution of indirect reciprocity. *Nature* 437, 1291–1298 (2005)
7. Pacheco, J.M., Santos, F.C., Chalub, F.A.: Stern-judging: A simple, successful norm which promotes cooperation under indirect reciprocity. *PLoS Comput. Biol.* 2, e178 (2006)
8. Hauert, C., Traulsen, A., Brandt, H., Nowak, M.A., Sigmund, K.: Via freedom to coercion: The emergence of costly punishment. *Science* 316, 1905–1907 (2007)
9. Rapoport, A., Chammah, A.M.: *Prisoner's Dilemma: A Study in Conflict and Cooperation*. University of Michigan Press (1965)
10. Hofbauer, J., Sigmund, K.: *Evolutionary Games and Population Dynamics*. Cambridge University Press (1998)
11. Nowak, M.A.: *Evolutionary dynamics: exploring the equations of life*. Belknap/Harvard University Press (2006)
12. Nowak, M.A., May, R.M.: Evolutionary games and spatial chaos. *Nature* 359, 826–829 (1992)
13. Darwin, C.: *The Descent of Man and Selection in Relation to Sex*. Murray (1871)
14. Traulsen, A., Nowak, M.: Evolution of cooperation by multilevel selection. *Proc. Natl. Acad. Sci. U S A* 103, 10952–10955 (2006)
15. Nowak, M.A., Sasaki, A., Taylor, C., Fudenberg, D.: Emergence of cooperation and evolutionary stability in finite populations. *Nature* 428, 646–650 (2004)
16. Santos, F.C., Pacheco, J.M.: Scale-free networks provide a unifying framework for the emergence of cooperation. *Phys. Rev. Lett.* 95(098104) (2005)
17. Santos, F.C., Pacheco, J.M., Lenaerts, T.: Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proc. Natl. Acad. Sci. U S A* 103, 3490–3494 (2006)
18. Santos, F.C., Pacheco, J.M.: A new route to the evolution of cooperation. *J. Evolution Biol.* 19, 726–733 (2006)
19. Santos, F.C., Rodrigues, J.F., Pacheco, J.M.: Graph topology plays a determinant role in the evolution of cooperation. *Proc. Biol. Sci.* 273(51–55) (2006)
20. Poncela, J., Gómes-Gardeñes, J., Floría, L.M., Moreno, Y.: Robustness of cooperation in the evolutionary prisoner's dilemma on complex networks. *New J. Phys.* 9(187) (2007)
21. Santos, F.C., Santos, M.D., Pacheco, J.: Social diversity promotes the emergence of cooperation in public goods games. *Nature* 454, 213–216 (2008)
22. Szabó, G., Fáth, G.: Evolutionary games on graphs. *Phys. Rep.* 446, 97–216 (2007)
23. Ebel, H., Bornholdt, S.: Coevolutionary games on networks. *Phys. Rev. E* 66(056118) (2002)
24. Zimmermann, M.G., Eguiluz, V.M., San Miguel, M.: Coevolution of dynamical states and interactions in dynamic networks. *Phys. Rev. E* 69(065102) (2004)
25. Santos, F.C., Pacheco, J.M., Lenaerts, T.: Cooperation prevails when individuals adjust their social ties. *PLoS Comput. Biol.* 2, e140 (2006)
26. Pacheco, J.M., Traulsen, A., Nowak, M.A.: Coevolution of strategy and structure in complex networks with dynamical linking. *Phys. Rev. Lett.* 97, 258103 (2006)

27. Hanaki, N., Peterhansl, A., Dodds, P.S., Watts, D.J.: Cooperation in evolving social networks. *Management Sci.* 53, 1036–1050 (2007)
28. Tanimoto, J.: Dilemma solving by the coevolution of networks and strategy in a 2x2 game. *Phys. Rev. E* 76, 021126 (2007)
29. Poncela, J., Gómez-Gardeñes, J., Floría, L.M., Sánchez, A., Moreno, Y.: Complex cooperative networks from evolutionary preferential attachment. *PLoS One* 3, e2449 (2008)
30. Skyrms, B., Pemantle, R.: A dynamic model of social network formation. *Proc. Natl. Acad. Sci. U S A* 97, 9340–9346 (2000)
31. Pacheco, J.M., Traulsen, A., Ohtsuki, H., Nowak, M.A.: Repeated games and direct reciprocity under active linking. *J. Theor. Biol.* 250, 723–731 (2008)
32. Masuda, N.: Participation costs dismiss the advantage of heterogeneous networks in evolution of cooperation. *Proc. R. Soc. B* 274(1620), 1815–1821 (2007)
33. Tanimoto, J., Yamauchi, A.: Does “game participation cost” affect the advantage of heterogeneous networks for evolving cooperation? *Physica A* 389(11), 2284–2289 (2010)
34. Perc, M., Szolnoki, A.: Social diversity and promotion of cooperation in the spatial prisoner’s dilemma game. *Phys. Rev. E* 77(011904) (2008)
35. Albert, R., Barabási, A.: *Statistical mechanics of complex networks*. *Rev. Mod. Phys.* 74, 47–98 (2002)
36. Dorogotsev, S.N., Mendes, J.F.F.: *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press (2003)
37. Onnela, J.P., Saramaki, J., Hyvonen, J., Szabó, G., Lazer, D.: Structure and tie strengths in mobile communication networks. *Proc. Natl. Acad. Sci. U S A* 104, 7332–7336 (2007)
38. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* 286, 509–512 (1999)
39. Amaral, L.A., Scala, A., Barthelemy, M., Stanley, H.: Classes of small-world networks. *Proc. Natl. Acad. Sci. U S A* 97, 11149–11152 (2000)
40. Szabó, G., Töke, C.: Evolutionary prisoner’s dilemma game on a square lattice. *Phys. Rev. E* 58(69–73) (1998)
41. Hauert, C., Szabó, G.: Game theory and physics. *Am. J. Phys.* 73, 405–414 (2005)
42. Traulsen, A., Nowak, M.A., Pacheco, J.: Stochastic dynamics of invasion and fixation. *Phys. Rev. E* 74(011909) (2006)
43. Ohtsuki, H., Nowak, M.A.: The replicator equation on graphs. *J. Theor. Biol.* 243(86–97) (2006)
44. Pacheco, J.M., Pinheiro, F.L., Santos, F.C.: Population structure induces a symmetry breaking favoring the emergence of cooperation. *PLoS Comput. Biol.* 5(12) (2009)
45. Skyrms, B.: *The Stag hunt and the evolution of social structure*. Cambridge University Press (2004)

A Pheromone Based Mobile Agent Migration Strategy for Servicing Networked Robots

W. Wilfred Godfrey and Shivashankar B. Nair

Indian Institute of Technology Guwahati, Assam, India
{w.godfrey, sbnair}@iitg.ernet.in

Abstract. In this paper we describe how mobile agents carrying a resource as a payload can prove to be useful in searching networked robots that require their services. While the agents migrate within the network in a conscientious manner, robots requiring their services diffuse pheromones to attract and guide them through the shortest path. The bidirectional and parallel search on part of the robot and the agent culminates in a faster convergence. The paper also compares the results derived by using this method with those obtained using two other algorithms. The results and discussions clearly indicate that this pheromone based algorithm is better suited for both static and dynamic networked robotics scenarios.

1 Introduction

Of late there has been considerable research in finding applications for mobile agents [1, 2]. Even though such agents have useful properties and look intuitively advantageous, researchers have not been able to tap their potential to the fullest. While they have been used in scenarios for searching and routing [3], their more specific features like payload carrying ability as also cloning have hardly been exploited. Mobile agents have been used in applications with robots and robotic networks. Researchers have already used robots in conjunction with mobile agents for a variety of functions [4, 5]. Mobile agents, in each of these cases, have been used either as an alternative to the traditional communication paradigms or for collecting data and forwarding to a centralized server. Many other features of these agents such as inter-agent communication, goal oriented, adaptive and proactive behaviors still remain to be effectively utilized.

A model for a mobile agent based multi-robot network that uses immune system metaphors has been proposed by Godfrey and Nair [6]. The mobile agents assist in gathering and providing information to various robots forming a network. The agents used in their architecture migrate from one robot to another and provide a means for information retrieval and exchange, thereby providing for a mechanism for inter-robot information sharing. Information in this context could mean the knowledge of how a task is to be performed. Though the concept of information sharing is achieved in this architecture, the round robin strategy for mobile agent migration to discover a robot requiring a service seems grossly inefficient. Several network related strategies have

reportedly been implemented for agent migration [7, 8, 9]. Minar et al [10] describe two algorithms for mapping a network of hosts using mobile agents – viz. the conscientious and super conscientious approaches. These agents gather topological information of an underlying static network in a distributed fashion.

The work reported herein portrays a strategy to attract a specific mobile agent, in the network, carrying a resource as its payload, to a robot requesting for the same. Robotic nodes diffuse pheromones [11, 12, 13] into the network as and when they need a service while the mobile agents moving conscientiously within the network track these pheromones to eventually reach and service them. We have compared our method with those proposed by Gaber et al [14, 15, 16] and found that using pheromones seem to be a far better alternative in both static and dynamic multi-robot networks.

2 Resource Discovery Methods

A survey of resource discovery methods can be found in [17]. In this section we describe two existing methods which make use of the mobility of agents to discover resources and later compare them with the Pheromone-Conscientious proposed by us. Our test-bed comprises a network of robots, each capable of hosting mobile agents. The agents carry, as payload, code for various tasks to be executed by the robot. A robot is provided with a set of tasks for which it does not have the associated code, initially. It thus sends a request for the same into the network. The mobile agents, on receiving the request, migrate to this robotic node and provide the code to complete the service.

2.1 Conscientious Approach

A mobile agent using the conscientious approach [10] aims at uniformly visiting all the nodes comprising the network. The agent maintains a queue of “visited nodes”. As it migrates, it tries to avoid nodes within this queue and in the process migrates to other nodes in its sojourn within the network. In doing so it maintains uniformity in its visits to the nodes and services those that have requested for a service which it carries as payload.

2.2 Gaber-Bakhouya’s Random Walk and Cloning Based Approach (G-B Algorithm)

Gaber-Bakhouya’s approach [14, 15] uses a mobile agent for locating resources available at distant nodes in a peer-peer network. Their algorithm works as follows- The node which requires to locate a service creates a mobile agent termed as the request agent. A service could constitute a set of resources {R1, R2, R3, R4} each of which is available at separate nodes in the network. The request agent is capable of cloning at each hop. Based on the number of neighbors of the current node, this agent chooses one path to a node and sends a clone each along the others. Every agent and

clone has a Time to Live (TTL) parameter, which is decremented at each hop. If the TTL becomes zero before the service discovery is completed, the agent/clone contacts the requestor node to find whether any other agent has discovered the path to all these resources. If the requestor node responds negatively, the agent extends its TTL and proceeds in its search; else it kills itself. Once an agent discovers the path to all these resources, it sends back the path information to the requestor thereby completing its task. An adaptive immune approach [16] has reportedly been used to contain the population size of the clones. A reinforcement learning algorithm facilitates the strengthening of paths towards this set of resources.

3 Pheromone-Conscientious (P-C) Mobile Agent Migration Strategy for Multi-robot Systems

The work reported in this paper augments the multi-robot mobile agent architecture proposed in [6] with a combined pheromone cum conscientious strategy. The mobile agents opt for a conscientious approach while the robots requiring a service use pheromone diffusion to attract the concerned agents to satiate a service request. The Multi-Mobile Agent based Multi-Robot networked system has the following features:

- i) Absence of a centralized server for hosting services.
- ii) Mobile agents carrying payload (resources) keep migrating across different robotic nodes conscientiously.
- iii) Robots need not have the resources (programs required for task execution) when they are deployed initially, thus facilitating a novice user to add and use robots on the fly.
- iv) The robotic network could be static or dynamic.

A robot is issued a set of tasks by a human operator. A set of tasks for which the robot does not have relevant code to effect the execution could be looked upon as a service $S = \{C_{T1} C_{T2}, C_{T3} \dots C_{Tn}\}$ where C_{Ti} is the code for the task T_i . A robot that needs a service is termed as a Robot Requesting Service (RRS). The mobile agents carry one or more of the C_{Ti} s as their payload. The work described in this paper intends to reduce the time required by the relevant mobile agents to reach and satiate the RRS with the requested C_{Ti} s, thereby providing for a quicker and on-site service. Once the programs for all the requested tasks are received at the RRS, they are ordered in the desired sequence, compiled and executed by the robot. This facilitates a novice user to hook on a robot to such a network and allow it to discover the essential programs. The main thrust of this paper is to describe a proactive RRS discovery process, on part of the mobile agents, using pheromones. The mobile agents normally use the conscientious method for migration within the network. They consciously avoid those robotic nodes recently visited and migrate towards those not yet visited by maintaining a “recently visited node list”. An RRS attracts the relevant agents by diffusing pheromones onto its immediate neighbors which in turn diffuse them further. Relevant agents that populate the network eventually hit the pheromones trail and on doing so, migrate thereon, along the shortest path to the RRS. The mobile

agent and the RRS thus perform a proactive and concurrent bidirectional search facilitating a faster service within the network. While the mobile agent uses the conscientious approach till it senses a pheromone trail, the RRS spreads out a pheromone network around it to attract and guide the agent towards it.

3.1 Contents of the Pheromone

The virtual pheromone used herein has embedded within it, the following:

1) Task Identifier: Each task has an identifier and a pheromone is diffused specifically for that task so as to attract only those mobile agents that carry the relevant programs as their payload.

2) Concentration: Concentration of a pheromone trail has direct relevance to the proximity of the RRS. A higher concentration indicates that the RRS is close by. A mobile agent always chooses pheromone trails of higher concentration and hence reaches the RRS through the shortest available path.

3) Lifetime: Pheromones need to die out after the concerned mobile agent reaches and services an RRS, lest the agent be trapped at the RRS again for a redundant service. A life time for pheromones ensures that they die out after the concerned mobile agent reaches the RRS. However it may happen that they die out before the agent reaches the RRS. Therefore the RRS is programmed to periodically re-diffuse pheromones till the service request is completed.

4) Next Robot Identifier: Once a mobile agent selects a pheromone having the highest concentration at a node, it checks the Next Robot identifier of this pheromone to find the next robot node that will lead it to the RRS and then migrates towards it.

3.2 Diffusion of Pheromones

Diffusion of pheromones is carried out by a robotic node for every pheromone it receives from another by laying them across all its neighbors except the one from which it was received. Since the pheromone concentration gradient needs to be formed, the concentration of the pheromone as also its life time, are reduced proportionately. This maximum concentration (C_{max}) of a pheromone trail is laid between the RRS and its immediate one-hop neighbors. Successive diffusions have a concentration gradient given by equation 1.

$$\Delta C = 100/d \quad (1)$$

where d is the spanning length. The percolated pheromone concentration P_{pc} is given by equation 2.

$$\begin{aligned} P_{pc}(n) &= C_{max} && \text{for } n=1 \\ &= P_{pc}(n-1) - \Delta C && \text{for } n > 1 \end{aligned} \quad (2)$$

where n is the hop count from the RRS. Similar to the concentration gradient, a gradient of the pheromone life time (ΔL) is also used to ensure that they disappear in

accordance. Farther the pheromone from the RRS, the lesser is its life time. The lifetime gradient is proportional to the concentration gradient and is given by equation 3.

$$\begin{aligned}\Delta L &= L_{\max} (\Delta C/100), \\ &= L_{\max}/d\end{aligned}\quad (3)$$

where L_{\max} is the maximum value of life time. While laying the pheromones, their lifetimes are reduced over the hops using equation (4).

$$\begin{aligned}P_{lt}(n) &= L_{\max} && \text{for } n=1 \\ &= P_{lt}(n-1) - \Delta L && \text{for } n>1\end{aligned}\quad (4)$$

where n is the hop count from the RRS. If the tasks issued need to be procured and executed quickly, the RRS will correspondingly change the values of P_{lt} , ΔC and d to allow a higher and longer diffusion into the network. This will increase the chances of the relevant agents hitting these pheromones trails. The maximum life time needs to be fixed based on the application scenario. For instance a larger network would, on an average, mean that the mobile agent takes more time to find the pheromone trail. Under such conditions we prescribe a larger life time.

4 Experimental Results and Analysis

We present a comparison of results obtained by using the three methods (viz. Conscientious, G-B and P-C) for scenarios with single and multiple RRSs populating a static robotic network. We also present a short discussion on how the three algorithms would perform when used in a dynamic network.

4.1 With One RRS and Three Agents

Simulations for several scenarios were carried out with 25 robotic nodes. Table 1 shows the results of four such scenarios depicted in Figure 1 (a)-(d) using a single RRS requesting a service $S = \{C_{T1}, C_{T2}, C_{T3}\}$. In case of the Conscientious and PC methods, the initial locations of three mobile agents, carrying three different resources, were changed for every run of the simulation as shown in the Figure 1. Likewise, the static resources were shifted accordingly for the G-B method. The number of hop counts taken by each of the agents to reach the RRS with their resources to service it is shown in Table 1. The hops taken, agents spawned/cloned and the messages/pheromones diffused together give us an idea of the amount of network resources expended in effecting a service. All these consume energy and/or time. The P-C method, carried out for 25 robotic nodes had a pheromone spanning depth of 2. As is evident, the P-C method outperforms its Conscientious counterpart due to the parallel and bidirectional search used. The use of pheromones greatly reduces the wayward movement of the mobile agents carrying resources. The P-C

method takes in all 12 hops (for all three agents) and 12 pheromone diffusions in the first network scenario. In the G-B method, wherein the RRS sends agents and their clones to discover the static resources, the first agent discovers all the resources in 10 hops. The clones generated by this agent move in parallel and take an extra 10 hops. These agents may further clone and hop or pass messages all of which amount to a value greater than 15. Though these hops occur in parallel and can be attributed to a faster search, they consume both network bandwidth and energy of the nodes. The G-B algorithm also prescribes message passing between the agent/clone and the RRS every time the TTL becomes 0. The same is true for updating the path, and sending the path back to the RRS. These overheads have not been calculated here but if added makes the P-C algorithm perform far better as the mobile agents actually reach the services to the RRS. The path calculated by the learning mechanism used in the G-B algorithm is also restricted to a certain set of resources comprising a service. This path may prove to be suboptimal when proper subsets of the resources are taken into consideration.

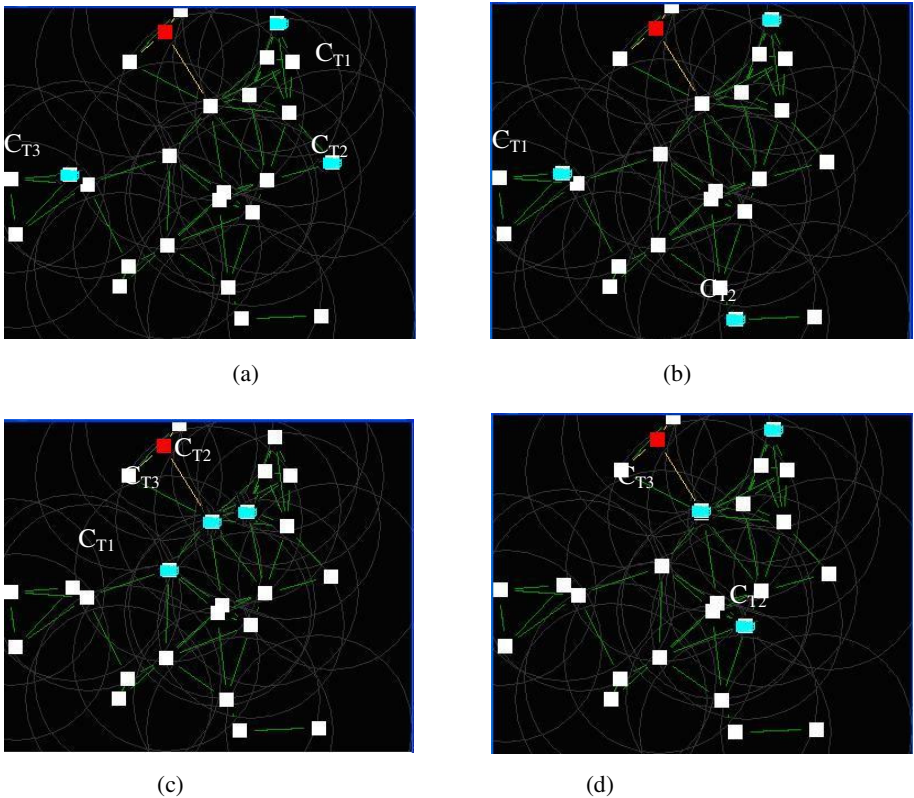


Fig. 1. Snapshots of simulations using 25 nodes. Scenarios (a)-(d) depict the placement of the agents/resources (in Blue) and that of the Requesting node (in Red).

Table 1. Table depicting the number of hops taken by three agents in the Conscientious and P-C methods to bring the resources to the RRS together with the pheromones diffused in the latter. The number of agents, number of hops consumed to locate the resource locations in the G-B method are also shown.

Scenarios	Conscientious Method			P-C Method			G-B Method			
	No. of hops			No. of hops			No. of pheromones diffused by the RRS	No. of hops taken by the Agent that first locates all the resources	No. of clones generated by the Agent that first located all the resources	No. of other Clones and Messages
	1 st Agent	2 nd Agent	3 rd Agent	1 st Agent	2 nd Agent	3 rd Agent				
(a)	26	10	45	5	3	4	12	10	10	Greater than 15
(b)	17	37	33	8	7	4	12	14	10	Greater than 15
(c)	29	6	15	5	5	6	12	4	2	Greater than 15
(d)	31	5	8	4	4	10	12	7	6	Greater than 15

4.2 Multiple RRS

If the above experiment were to be repeated with the multiple RRSs requesting for the same resources (viz. C_{T1} , C_{T2} , C_{T3}) at the same time, the G-B method would generate a large number of clones as also messages. It could result in an increase in energy consumed per node and also bandwidth. Though the G-B algorithm describes an immune approach to restrict clonal expansion, it has not taken into account the amount of processing involved per node and the subsequent delays caused in the search. As the number of RRSs increase clones too increase thereby increasing the effective time required for a migration. On the other hand, the P-C method is conservative in its use of both energy and network bandwidth and strives to bring the resources to the agent in times comparable to that of the G-B algorithm. A modified P-C algorithm for closely knit RRSs that request the same resource described in [18], allows for decreasing the service period.

4.3 Dynamic Multi-robot Network

Unlike the G-B algorithm that caters only to static networks, the P-C algorithm can be used in conjunction with dynamic networks with a moderate degradation in performance. In a dynamic network the links to neighboring robot nodes could make or break. Under these conditions, if the G-B algorithm were used, the agents would never be able to trace their path backwards. Even a single link failure can cause an agent to be lost. In the P-C algorithm, the agents could be made to come closer to the RRS by re-diffusing pheromones with low life times at a higher frequency and greater spanning depth. Since the pheromones have a short life time, those between separated

nodes will die out faster. The higher re-diffusion frequency will allow fresh and new pheromones to be laid in the current state of the network.

5 Conclusions

In this paper, we have discussed a pheromone based migration strategy for servicing networked robots. A comparison of this method with the Conscientious and the G-B algorithms is also presented. Observations indicate the viability of using the P-C algorithm which is conservative in its consumption of both network bandwidth and energy. Further unlike the G-B algorithm, it can perform even when the network is dynamic. The G-B algorithm, on the contrary may prove to be beneficial in terms of time in static networks provided we neglect bandwidth problems and energy consumed per node. We thus conclude that the P-C algorithm described herein seems to perform better than the G-B algorithm when used in conjunction with multi-robot networks.

References

1. Outtagarts, A.: Mobile Agent Based Applications: A Survey. *International Journal of Computer Science and Network Security* 9(11), 331–339 (2009)
2. Giovanni, V.: Mobile Agents: Ten Reasons for Failure. In: 5th International Conference on Mobile Data Management, Berkeley, California, pp. 298–299 (2004)
3. Chen, W., Zhang, Y.: A Multi-constrained Routing Algorithm based on Mobile Agent for MANET Networks, pp. 16–19. IEEE Computer Society, IEEE Press, New York (2009)
4. Kambayashi, Y., Takimoto, M., Kodama, Y.: Controlling biped walking robots using Genetic algorithms in mobile agent environment. In: 3rd IEEE International Conference on Computational Cybernetics, pp. 29–34 (2005)
5. Cragg, L., Hu, H.: A Multi-agent system for distributed control of networked mobile robots. *Measurement and Control* 38(10), 314–319 (2005)
6. Godfrey, W.W., Nair, S.B.: An Immune System Based Multi-robot Mobile Agent Network. In: Bentley, P.J., Lee, D., Jung, S. (eds.) ICARIS 2008. LNCS, vol. 5132, pp. 424–433. Springer, Heidelberg (2008)
7. Zhang, J., Ma, J., Yang, J., Cheng, L.: An Improved Migration Strategy of Mobile Agent. In: 4th International Conference on Fuzzy Systems and Knowledge Discovery, vol. 1, pp. 577–581 (2007)
8. Hayfa, Z., Hammadi, S.: A Migration strategy of mobile agents for the transport network applications. *Journal of Mathematical Computation and Simulation* 76(5-6), 345–362 (2008)
9. Kwon, H.C., Kim, H.H., Lee, J.T., Yoo, K.J.: A Migration strategy of mobile agent. In: 8th International Conference on Parallel and Distributed Systems, pp. 706–712 (2001)
10. Minar, N., Kramer, K.H., Maes, P.: Cooperating Mobile Agents for Mapping Networks. In: 1st Hungarian National Conference on Agent Based Computing, pp. 34–41 (1999)
11. Kitamura, N., Yoshiyuki, N., Koji, F.: Development of a desktop swarm robot system based on pheromone communication. *Artificial Life Robotics* 14(3), 329–331 (2009)
12. Parunak, H., Dyke, V., Brueckner, S., Sauter, J.A.: Digital Pheromones for Coordination of Unmanned Vehicles. In: 1st International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 246–263 (2002)

13. David, P., Daily, M., Estowski, R., Howard, M., Lee, C.: Pheromone Robotics. In: *Autonomous Robots*, vol. 11(3), pp. 319–324. Kluwer Academic Publishers (2001)
14. Gaber, J., Bakhouya, M.: Mobile Agent-Based Approach for Resource Discovery in Peer-to-Peer Networks. In: Joseph, S., Despotovic, Z., Moro, G., Bergamaschi, S. (eds.) *AP2PC 2006*. LNCS (LNAI), vol. 4461, pp. 63–73. Springer, Heidelberg (2008)
15. Bakhouya, M., Gaber, J.: A Query Routing Approach Based on Users' Satisfaction for Resource Discovery in Service-Oriented Networks. *World Wide Web* 13(1-2), 61–73 (2010)
16. Bakhouya, M., Gaber, J.: Adaptive Approach for the Regulation of a Mobile Agent Population in a Distributed Network. In: *The Fifth International Symposium on Parallel and Distributed Computing*, pp. 360–366 (2006)
17. Meshkova, E., Riihijärvi, J., Petrova, M., Mähönen, P.: A Survey on Resource Discovery mechanisms, peer-to-peer and service discovery frameworks. *Computer Networks* 52(11), 2097–2128 (2008)
18. Godfrey, W.W., Nair, S.B.: Mobile Agent Cloning for Servicing Networked Robots. In: Desai, N., Liu, A., Winikoff, M. (eds.) *PRIMA 2010*. LNCS, vol. 7057, pp. 336–339. Springer, Heidelberg (2012)

Knowledge Sharing in Social Network Using Game Theory

Ping Zhu¹, Guiyi Wei¹, Athanasios V. Vasilakos², and Hung-Yu Wei³

¹ College of Computer & Information Engineering,
Zhejiang Gongshang University, Hangzhou, Zhejiang, P.R. China
{jackyzhu, weigy}@mail.zjgsu.edu.cn,

² Department of Computer and Telecommunications Engineering,
University of Western Macedonia, Kozani, Greece
vasilako@ath.forthnet.gr

³ Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
hywei@cc.ee.ntu.edu.tw

Abstract. Stimulating is an important mechanism in Virtual Community (VC) during the Knowledge Sharing (KS) process. In this paper, we combine the power of game theory and stimulating mechanism together to optimize the KS process in Social Network (SN). We first model the basic stimulating mechanism as a static game of complete information, under which the stimulating threshold for Nash Equilibrium (NE) is derived. Next, we modify the static model by introducing the KREPS-MILGROM-ROBERTS-WILSON (KMRW) reputation model, where the dynamic case is studied and the Perfect Bayesian Equilibrium is proved. We then propose a novel *rational stimulating* mechanism by combining the finitely repeated game with basic stimulating mechanism together. Theoretical analyzing indicates that, by introducing incomplete information, the *rational stimulating* achieves a lower cost; through stimulating, the Perfect Bayesian Equilibrium's condition is satisfied and the KS rate will approach 100% as long as the KS process is repeated enough. Finally, we extend our *rational stimulating* mechanism to the multi-person model.

Keywords: Social Network, Virtual Community, Knowledge Sharing, Stimulating, Game Theory.

1 Introduction

Social networks are built upon the idea that there exists a determinable structure to how people know each other, whether directly or indirectly [1]. In such networks, people are connected and cooperate through one or more specific types of interdependency through common social relationships [2-4].

As one of the kernel technology in SN, Knowledge Sharing concerns about how to turn individual knowledge into organizational knowledge [5,6]. Problems arouse during KS process involves how to increase the KS rate, how to avoid hitchhike and how to make most efficient utilization of knowledge [7-10].

The game theory [11,12] is a powerful tool to model the interaction among SN members and to analyze the optimal cooperation strategies. The static game of complete information based KS model can only be used to analyze member's one time KS behavior. In dynamic game of complete information, the finitely repeated game can not form collaboration behavior among members. Although we can achieve the Subgame Perfect Nash Equilibrium in infinitely repeated game, nevertheless, in real world SN, the KS process can not repeat endlessly. While in incomplete information case of KMRW model [13], the Perfect Bayesian Equilibrium can be achieved [14], however, the condition that equilibrium must satisfy is not easy to obtain and control [12].

While SN without coordination can not accomplish KS simply, the stimulating mechanism in VC [15] provides a feasible trick. A meticulously designed stimulating mechanism can greatly arouse member's enthusiasm as well as increase the KS rate. If we transfer different form of stimulating into numerical value defined as *stimulating cost*, then the key problem in stimulating mechanism is that how does the SN coordinator optimally set this value under the premise of guaranteeing a high KS rate among all the members in SN. Although a large *stimulating cost* can motivate the KS process, however, the cost to SN coordinator is non-neglectful; on the other hand, a small *stimulating cost* may not promote the members enough to join KS.

In this paper, we propose a novel *rational stimulating* mechanism by combining the finitely repeated game with basic stimulating mechanism together, who will optimize each other during the KS process. Through *rational stimulating*, Game helps Stimulating to reduce its cost; Stimulating guarantees the existing of Perfect Bayesian Equilibrium in return of Game's help.

The rest of paper is organized as follows: Section 2 describes the problem in KS process and gives out a basic solution using basic stimulating mechanism, where the *stimulating cost* is left as a problem to handle. In Section 3.1 and 3.2, we analyze the finitely repeated game of incomplete information and leave the Perfect Bayesian Equilibrium condition as another problem to solve. In Section 3.3, we propose and explore our *rational stimulating* mechanism. In Section 4, we extend the *rational stimulating* mechanism to multi-person case. Finally in Section 5, we conclude our paper.

2 Basic Stimulating Mechanism

2.1 Problem Description

In this section it is assumed that there are only two members in SN, named m_1 and m_2 respectively. They simultaneously choose actions and each member's payoff function is common knowledge between themselves. Further, we suppose that m_1 and m_2 are both rational and will take their dominant strategy as their best response to each other. The benefit of KS can be quantized, and so is the cost. We can obtain the following static game of complete information: the KS benefit for m_1 and m_2 are both of b ; the KS cost is defined to be c , where $b, c > 0$ holds. As Table 1 shows, the KS process can be represented in the accompanying

bi-matrix (We use the row of bi-matrix to indicate m_1 's action, and m_2 is the column).

Table 1. Knowledge Sharing Based on Static Game of Complete Information

	sharing	not sharing
sharing	$(b - c, b - c)$	$(-c, b)$
not sharing	$(b, -c)$	$(0, 0)$

If both the two members choose *sharing* strategy, then the payoff for each one is $b - c$; if one of them (suppose m_i) chooses *not sharing*, then $m_j, j \neq i$ will get a negative payoff of $-c$ and m_i will get a b payoff; if neither of them chooses *sharing*, both will get nothing, represented as zero in the bi-matrix. Under assumption of rationality, both of m_1 and m_2 's dominant strategy are *not sharing*, and the corresponding equilibrium is thus (*not sharing, not sharing*). While the SN coordinator expects a scenery of (*sharing, sharing*), the rationality brings the KS process into Prisoners' Dilemma.

2.2 Basic Stimulating Mechanism in Knowledge Sharing

To help SN members walk out of the Prisoners' Dilemma and arrive into the (*sharing, sharing*) equilibrium, the SN coordinator can take stimulating mechanism. Define SN coordinator's *stimulating cost* to be s . The payoff function by introducing stimulating mechanism can be represented by Table 2.

Table 2. Knowledge Sharing Based on Static Game of Complete Information with Basic Stimulating Mechanism

	sharing	not sharing
sharing	$(b + s - c, b + s - c)$	$(s - c, b)$
not sharing	$(b, s - c)$	$(0, 0)$

In basic stimulating mechanism, member m_i achieves an additional award s as long as he takes the *sharing* strategy. Under assumption of rationality, we can easily get the equilibrium of this improved KS process in the following two cases:

$$NE = \begin{cases} (N, N), & \text{if } s < c \\ (S, S), & \text{if } s > c \end{cases} \tag{1}$$

While in equations (1), S indicates *sharing*, N represents *not sharing*. The SN members can achieve the (*sharing, sharing*) equilibrium in condition that the *stimulating cost* provided by SN coordinator is larger than KS cost.

So much for this, we have derived the *stimulating cost* for SN coordinator. Providing $s > c$, the rational SN members can achieve their NE. However, as

described in Section 1, a large *stimulating cost* results in a non-neglectful cost to SN coordinator. So the new problem arising here is that whether the threshold of s , which is c till now, can be further decreased under the premise of this decreased *stimulating cost* could still guarantee SN members' (*sharing, sharing*) equilibrium. In the following sections, we will find out an improved threshold by introducing SN member's uncertainty.

3 Rational Stimulating Mechanism

The static game of complete information discussed in Section 2 can only be used to analyze member's one time KS behavior. Although we can extend it to the dynamic model and analyze the multi-stage KS process, however, the finitely repeated game can not form collaboration behavior among members. Further, if the game is repeated infinitely, the Subgame Perfect Nash Equilibrium can be achieved. Nevertheless, in real world SN, the KS process will not repeat endlessly. Fortunately, the KMRW reputation model [13] provides a Perfect Bayesian Equilibrium solution within some finite stages under incomplete information condition. In this section, we will explore the KMRW reputation model in our KS process with two SN members. Extension to multi-person KMRW model is discussed in Section 4.

3.1 Two Stage Knowledge Sharing

Suppose m_1, m_2 are a little of complex than in the previous section by introducing incomplete asymmetric information. To be concrete, we assume that m_1 has private information about his strategy with probability p of playing the *following* strategy and probability $1 - p$ of *playing rationally*. Moreover, m_2 does not know which type m_1 actually belongs to, the only thing he knows is m_1 's probability distribution $(p, 1 - p)$. The *following* strategy provides that m_1 will first choose *sharing* then mimic m_2 's previous strategy; *playing rationally* means that the player will act according to its dominant strategy in the current stage. In an actual SN, the *following* member (referred as non-rational later in paper) can be explained as an *action echo*, who always follows other members' opinion in purpose of raising his status in SN.

The timing of m_1 and m_2 is described as follows:

- The SN coordinator knows the type for m_1 , with probability p of non-rational and probability $1 - p$ of rational.
- m_1 and m_2 choose *sharing* or *not sharing* in the first stage. The non-rational m_1 will choose *sharing* according to the *following* strategy; m_2 will choose his strategy rationally.
- On observing the result of the first stage, m_1 and m_2 choose *sharing* or *not sharing* in the second stage. The non-rational m_1 mimics m_2 's first step strategy; rational m_1 and m_2 play rationally.
- The payoff of the two stage KS is the sum of each stage's payoff.

Denote the rational and non-rational m_1 as m_1^r and m_1^n respectively. Similar to the finitely repeated game of complete information, *not sharing* is the dominant strategy for both m_1^r and m_2 . So in the second stage of KS process, m_1^r and m_2 will play *not sharing*. Because m_2 will surely choose *not sharing* in the second stage, m_1^r is not necessary to hide his type in the first stage, so he will also choose *not sharing* in the first stage. Considering of m_1^n 's *following* strategy, the equilibrium path of two stage knowledge sharing can be represented by Table 3, where $X \in \{S, N\}$ according to the *following* strategy.

Table 3. Two Stage Knowledge Sharing with Incomplete Asymmetric Information

	$t = 1$	$t = 2$
m_1^n	S	X
m_1^r	N	N
m_2	X	N

If m_2 choose *sharing* in the first stage, then the average payoff (without stimulating) of m_2 in the two stage KS process is:

$$p \times (b - c) + (1 - p) \times (-c) + p \times b \tag{2}$$

If m_2 choose *not sharing* in the first stage, then the average payoff of m_2 in the two stage KS process is:

$$p \times b \tag{3}$$

From equations (2) and (3), we can solve the condition of m_2 sharing his knowledge in the first stage:

$$\begin{aligned}
 & p \times (b - c) + (1 - p) \times (-c) + p \times b > p \times b \\
 \Rightarrow & p > \frac{c}{b}
 \end{aligned} \tag{4}$$

Inequality (4) indicates that according to m_2 's priori knowledge of m_1 's type, to promote knowledge sharing, the SN coordinator should guarantee $p > \frac{c}{b}$. However, given m_1 , p is fixed; given m_2 , b and c are fixed. It seems that the SN coordinator has nothing to do with adjusting inequality (4). We will solve this problem in Section 3.3 where the *rational stimulating* mechanism is discussed. Here, we continue our analysis in finitely repeated game of incomplete information in three stage case.

3.2 Three Stage Knowledge Sharing and the General T Stage Case

Suppose inequality (4) is satisfied, we will derive the sufficient condition for the equilibrium path of three stage KS as Table 4 shown. Under this equilibrium, m_1^r 's average payoff is $(b - c) + b$; m_2 's average payoff is $b - c + p \times (b - c) + (1 - p) \times (-c) + p \times b$. We will next prove that m_1 and m_2 has no incentive to derive the equilibrium path described in Table 4.

Table 4. Three Stage Knowledge Sharing with Incomplete Asymmetric Information

	$t = 1$	$t = 2$	$t = 3$
m_1^n	S	S	S
m_1^r	S	N	N
m_2	S	S	N

If m_1^r chooses *not sharing* in the first stage, m_2 will know m_1 's type is m_1^r so as to choose *not sharing* in the following stages. To cope with m_2 , m_1^r will also choose *not sharing*. The resulting KS process is shown in Table 5. According to Table 5, m_1^r 's average payoff is b which is lower than the equilibrium path payoff $2b - c$ (assuming $b > c$ holds), so m_1^r will choose *sharing* in the first stage.

Table 5. Three Stage Knowledge Sharing: m_1^r 's Deviation

	$t = 1$	$t = 2$	$t = 3$
m_1^n	S	S	N
m_1^r	N	N	N
m_2	S	N	N

If m_2 chooses *not sharing* in the first stage, according to *following* strategy, m_1^n will mimic him and choose *not sharing* in the second stage. Considering that m_1^n will surely choose *not sharing* in the third stage, m_1^r is not necessary to hide his type in the second stage, so m_1^r will take its dominant strategy *not sharing* in stage two. The resulting KS process is shown in Table 6.

Table 6. Three Stage Knowledge Sharing: m_2 's Deviation

	$t = 1$	$t = 2$	$t = 3$
m_1^n	S	N	X
m_1^r	S	N	N
m_2	N	X	N

If m_2 choose *not sharing* in the second stage, the average payoff is b . So we can get the condition of m_2 having no incentive to deviate from equilibrium path:

$$\begin{aligned}
 & b - c + p \times (b - c) + (1 - p) \times (-c) + p \times b > b \\
 \Rightarrow & p > \frac{c}{b}
 \end{aligned}
 \tag{5}$$

which is the same to inequality (4).

If m_2 choose *sharing* in the second stage, the average payoff will be $b - c + p \times b$. Once again, we can get the condition of m_2 having no incentive to deviate from equilibrium path:

$$\begin{aligned}
 & b - c + p \times (b - c) + (1 - p) \times (-c) + p \times b \\
 & > b - c + p \times b \\
 \Rightarrow & p > \frac{c}{b}
 \end{aligned} \tag{6}$$

which is the same to inequality (4) and (5).

Combine inequalities (4), (5) and (6), we conclude that under condition of $p > \frac{c}{b}$, there exists a Perfect Bayesian Equilibrium in the three stage KS process, which improves the KS rate in SN. To be general, we can have the following theorem:

Theorem 1 (T Stage Knowledge Sharing). *Given two SN members m_1 and m_2 , who satisfy $p > \frac{c}{b}$. There exists a Perfect Bayesian Equilibrium in the T stage knowledge sharing process, under condition that m_1^r and m_2 both take the sharing strategy in the previous T-2 stages and the last two stages is taken as Table 3 shown.*

Theorem 1 indicates that according to m_2 's priori knowledge $(p, 1 - p)$, to promote knowledge sharing, the SN coordinator should guarantee $p > \frac{c}{b}$. However, as referred in section 3.1, given m_1 and m_2 , p, b and c are all fixed. The condition, $p > \frac{c}{b}$, Theorem 1 relying on, is not naturally satisfied. In the next subsection, we propose a novel *rational stimulating* mechanism, which contributes to both satisfying Perfect Bayesian Equilibrium condition and reducing the *stimulating cost* in SN's members KS process.

3.3 Rational Stimulating in Knowledge Sharing

Theorem 2 (Rational Stimulating with Two Members). *By introducing incomplete information in the basic stimulating mechanism, the optimal stimulating value s_{opt} satisfies:*

$$s_{opt} > \max\{c - pb, c - \frac{2}{3}b, c - 2(1 - p)b\} \tag{7}$$

Moreover, as KS process's repeating times T increases, the KS rate $\eta(T)$ also increases, which will approach 100% in the limit case as equation (8) shown:

$$\lim_{T \rightarrow \infty} \eta(T) = 1 \tag{8}$$

Proof. If m_1, m_2 play the game according to Theorem 1, the KS rate $\eta(T)$ is computed as:

$$\eta(T) = \frac{(T - 2)(b - c) + p(b - c) + (1 - p)(-c) + pb}{T(b - c)} \tag{9}$$

According to *L'Hospital Rule*, equation (8) can be easily achieved from equation (9). By taking the first derivative of equation (9) with respect to T, we have:

$$\eta'(T) = \frac{2b(1 - p) - c}{T^2(b - c)} \tag{10}$$

Combining the Perfect Bayesian Equilibrium conditions (inequalities (4.5,6)) and the first derivative of $\eta(T)$ (equation (10)), we get the condition under which there exists Perfect Bayesian Equilibrium and the KS rate $\eta(T)$ increases as T increases:

$$\begin{cases} \frac{c}{b} < p < 1 - \frac{1}{2} \frac{c}{b} \\ c < b \\ \frac{c}{b} < 1 - \frac{1}{2} \frac{c}{b} \end{cases} \tag{11}$$

Notice that in inequality (11), given SN members m_1 and m_2 , p, b, c are constants. We have already assumed that the KS cost is lower than benefit, so condition $c < b$ is naturally satisfied. The problem mentioned in the previous section still remains. That is, give m_1 and m_2 , inequalities $\frac{c}{b} < p < 1 - \frac{1}{2} \frac{c}{b}$ and $\frac{c}{b} < 1 - \frac{1}{2} \frac{c}{b}$ can not be naturally satisfied.

Rewrite Table 2 as Table 7. Denote $c' = c - s$, then Table 1 and Table 7 jointly mean that the KS cost c under our *rational stimulating* mechanism can be variable, which is adjusted through *stimulating cost* provided by SN coordinator. Substitute c with c' in inequality (11), we finally get inequality (18).

Table 7. Another Form for Knowledge Sharing with Basic Stimulating Mechanism

	sharing	not sharing
sharing	$(b - (c - s), b - (c - s))$	$(-(c - s), b)$
not sharing	$(b, -(c - s))$	$(0, 0)$

The significance of Theorem 2 can be explained as follows: given SN members m_1 and m_2 , although p, b, c are fixed, we can still achieve the finitely repeated KS process’s Perfect Bayesian Equilibrium by coordinator optimally setting the *stimulating cost* as s_{opt} . On the other side, by modeling the KS process as an incomplete information dynamic game, compared to the basic stimulating mechanism, the SN coordinator can reduce its *stimulating cost* in management of SN activities. By adopting the *rational stimulating* mechanism, the SN coordinator uses a lower cost to achieve an prosperous scenery of knowledge sharing among different SN members.

4 The Multi-person Knowledge Sharing

In the original work of KMRW reputation model [13], only two players with incomplete asymmetric information were discussed. However, in real world SN, there are always more than two members sharing their knowledge. In this section, we extend both of KMRW reputation model and *rational stimulating* mechanism to the multi-person environment.

Suppose there are M members in SN, named m_1, m_2, \dots, m_M respectively. They simultaneously choose actions and each member’s payoff function is common knowledge between themselves. Each of these M players is free to choose

between *sharing* and *not sharing*. While the *sharing* strategy is taken by member m_i , a cost of c is accompanied with m_i and a potential benefit of b is ready for some players who will acquire m_i 's knowledge. We further assume that every member in SN is seeking knowledge all the time, which means when m_i takes the *sharing* strategy, other $M - 1$ players ($m_j, j \neq i$) always obtain a benefit of b . While the *not sharing* strategy is taken by member m_i , no additional cost is needed. The payoff of m_i relies on the number of his neighbors who take *sharing* strategy.

4.1 M-Member Knowledge Sharing

Theorem 3 (Nash Equilibrium with M Members). *The Nash Equilibrium with M members in the KS process is a natural extension of two member Prisoner's Dilemma, which can be described as:*

$$NE^M = (N, N, \dots, N) \tag{12}$$

Proof. Consider one possible strategy combination:

$$\mathbb{P} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_M)$$

Where $\mathbf{p}_i \in \{N, S\}, i = 1, 2, \dots, M$ represents for m_i 's strategy. m_i 's payoff can be calculated as:

$$u_i = k_i \times b - c_i \tag{13}$$

Where $k_i \leq M - 1$ denotes the number of m_i 's neighbors who take the *sharing* strategy and m_i 's cost c_i is defined as:

$$c_i = \begin{cases} c & \text{if } m_i \text{ shares knowledge,} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Notice that, the first part of equation (13) has nothing to do with m_i 's strategy. For a given k_i , m_i 's best response is surely *not sharing*. ■

4.2 M-Member Knowledge Sharing with Basic Stimulating

To help SN members walk out of the Prisoners' Dilemma and arrive into the (S, S, \dots, S) equilibrium, the SN coordinator can again take stimulating mechanism. As an improvement to Theorem 3, the basic stimulating mechanism in multi-person KS process can be described as

Theorem 4 (Basic Stimulating with M Members). *By introducing the stimulating mechanism, the Nash Equilibrium with M Members can be migrated to*

$$NE_{bs}^M = (S, S, \dots, S) \tag{15}$$

Proof. In basic stimulating mechanism, m'_i 's payoff can be calculated as:

$$u_i = k_i \times b - c_i + s_i \tag{16}$$

Where $k_i \leq M - 1$ denotes the number of m'_i 's neighbors who take the *sharing* strategy and m'_i 's cost c_i is defined according to equation (14), the *stimulating cost* s_i is defined as:

$$s_i = \begin{cases} s & \text{if } c_i = c, \\ 0 & \text{otherwise} \end{cases}$$

Given k_i , m'_i 's best response is to be *sharing* under condition of $s > c$. ■

4.3 Multi-person Extension of KMRW and Rational Stimulating

In our model, all $m_i, i = 1, 2, \dots, M$ are assumed to have private information about his strategy with probability p of being m_i^n and probability $1 - p$ of being m_i^r . For any $j, j \neq i$, m_j dose not know which type m_i actually belongs to, the only thing he knows is m_i 's probability distribution $(p, 1 - p)$. For a given m_i , it's assumed to have connection with all the other $M - 1$ members in the SN, and an M -member game with incomplete asymmetric information is played. We have the following theorem:

Theorem 5 (T Stage Multi-person Knowledge Sharing). *Given M SN members m_1, m_2, \dots, m_M , who satisfy $p > \frac{c}{b}$. There exists a Perfect Bayesian Equilibrium in the T stage knowledge sharing process, under condition that $m_j^r(j \neq i)$ and m_i both take the sharing strategy in the previous $T - 2$ stages and the last two stages are taken as Table 8 shown, where $\bar{i} = \{1, 2, \dots, i - 1, i, \dots, M\}$.*

Proof (Induction on T). Given that for each $\tau = 2, 3, \dots, T - 1$, Theorem 5 holds. Then for a $\tau=T$ stage game,

- ① $m_j^r, j \neq i$ has no incentive to deviate from the equilibrium path in T stage game. If m_j^r chooses *not sharing* in stage $\tau < T - 1$, m_i will know m_j 's type is m_j^r and will choose *not sharing* in the following $T - \tau$ stages. The payoff from τ to T in equilibrium path and deviation path are $(T - 2 - \tau + 1) \times (b - c) + b$ and b respectively. So $m_j^r, j \neq i$ has no incentive to deviate from the equilibrium path.
- ② m_i has no incentive to deviate from the equilibrium path in T stage game. According to equilibrium path, the payoff of m_i from stage $\tau < T$ to T to can be calculated as:

$$(M - 1) \times \{2(b - c) + [(T - 2) - (\tau + 2) + 1](b - c) + p(b - c) + (1 - p)(-c) + pb\} \tag{17}$$

If m_i chooses *not sharing* in stage τ . All $m_j^r, j \neq i$ will mimic this strategy and choose *not sharing* in stage $\tau + 1$. m_j^r will also choose *not sharing* strategy for two reasons:

- *not sharing* dominates *sharing* in stage $\tau + 1$;
- *not sharing* hide m'_j 's type and will achieve a payoff of at least zero from stage $\tau + 2$ to T; while *sharing* will expose himself to m_i who will choose *not sharing* in the rest stages and get a payoff of exactly zero.

Suppose this process for m_i continues until stage $\tau + \varphi, \varphi \geq 0$ (m_i chooses *not sharing*, all m_j^n and m_j^r also choose *not sharing*). In stage $\tau + \varphi + 1$, m_i adopts the *sharing* strategy. The continuation game from stage $\tau + \varphi + 2$ to T thus constitute a $T - (\tau + \varphi + 2) + 1$ stage repeated game. According to our hypothesis, this game can be played according to the equilibrium path. We still need to discuss four different cases according to the $\tau + \varphi$ value:

- If $\tau + \varphi = T$. m'_i 's payoff is $(M - 1) \times b$ in stage τ and zero in all the other stages. Notice that $p > \frac{c}{b}$, we have $pb > c$. Thus m'_i 's payoff is less than equation (17);
- If $\tau + \varphi = T - 1$. m'_i 's payoff is $(M - 1) \times b$ in stage τ , $(M - 1) \times (-c)$ in stage T and zero in all the rest stages, which is less than equation (17);
- If $\tau + \varphi = T - 2$. m'_i 's payoff is $(M - 1) \times b$ in stage τ , $(M - 1) \times (-c)$ in stage T - 1, $(M - 1) \times pb$ in stage T and zero in all the rest stages, which is less than equation (17);
- If $\tau + \varphi < T - 2$. m'_i 's payoff is $(M - 1) \times b$ in stage τ , $(M - 1) \times (-c)$ in stage $\tau + \varphi + 1$, $(M - 1)[(T - 2) - (\tau + \varphi + 2) + 1](b - c) + p(b - c) + (1 - p)(-c) + pb]$ from stage $\tau + \varphi + 2$ to T and zero in all the rest stages, which is also less than equation (17). ■

Table 8. The Last Two Stages of T Stage Multi-Person Knowledge Sharing

	$t = 1$	$t = 2$
$m_{i_1}^n$	S	X
$m_{i_1}^r$	N	N
...
$m_{i_{M-1}}^n$	S	X
$m_{i_{M-1}}^r$	N	N
m_i	X	N

According to Theorem 5, we can get the optimal *stimulating cost* under M -member environment:

Theorem 6 (Rational Stimulating with Multi-person). *By introducing incomplete information in the basic stimulating mechanism, the optimal stimulating cost s_{opt}^M satisfies:*

$$s_{opt}^M > \max\{c - pb, c - \frac{2}{3}b, c - 2(1 - p)b\} \tag{18}$$

The proof of Theorem 6 is similar with Theorem 2.

5 Conclusion

Knowledge Sharing is one of the kernel technology in SN. During the KS process, an efficient stimulating mechanism can greatly arouse member's enthusiasm as well as the KS rate. Although traditional stimulating mechanism can motivate the KS process, however, the cost to SN coordinator is non-neglectful. In this paper, a novel *rational stimulating* mechanism is proposed. By combining the power of game theory and basic stimulating mechanism together during the KS process, we successfully reduce the *stimulating cost*. We also solve the Perfect Bayesian Equilibrium condition problem and the KS rate is proved to approach 100% as long as the KS process is repeated enough. We also extend both of KMRW reputation model and *rational stimulating* mechanism to the multi-person environment.

References

1. Churchill, E.F., Halverson, C.A.: Social Networks and Social Networking. IEEE Intelligent Systems 9(5), 14–19 (2005)
2. Carrol, J.M., et al.: Notification and awareness: synchronizing task-oriented collaborative activity. International Journal of Human-Computer Studies 58(5), 605–632 (2003)
3. Diamadis, E.T., Polyzos, G.C.: Efficient cooperative searching on the Web: system design and evaluation. International Journal of Human-Computer Studies 61(5), 699–724 (2004)
4. Fischer, G., et al.: Beyond binary choices: Integrating individual and social creativity. International Journal of Human-Computer Studies 63(4-5), 482–512 (2005)
5. Li, X., Montazemi, A.R., Yuan, Y.: Agent-based buddy-finding methodology for knowledge sharing. Information and Management 43(3), 283–296 (2006)
6. Nonaka, I., Takeuchi, H.: The Knowledge-creating Company: How Japanese Companies create the Dynamics of Innovation. Oxford University Press (1985)
7. Murphy, C., et al.: GenSpace: Exploring social networking metaphors for knowledge sharing and scientific collaborative work. In: 23rd IEEE/ACM International Conference on Automated Software Engineering, pp. 34–41 (2008)
8. Hsu, M.S., et al.: Mutual choices of knowledge sharing in social networks. In: IEEE International Engineering Management Conference, vol. 2, pp. 471–1475 (2004)
9. Yang, J., Chen, Y.: A social network-based system for supporting interactive collaboration in knowledge sharing over peer-to-peer network. International Journal of Human-Computer Studies 66(1), 36–50 (2008)
10. Chow, W., Chan, L.: Social network, social trust and shared goals in organizational knowledge sharing. Information & Management 45(7), 458–465 (2008)
11. Fudenberg, D., Tirole, D.: Game Theory. M.I.T. Press (1991)
12. Gibbons, R.: A Primer in Game Theory. Prentice-Hall (1992)
13. Kreps, D., Milgrom, P., Roberts, J., Wilson, R.: Rational Cooperation in the Finitely Repeated Prisoners' Dilemma. Journal of Economic Theory 27(2), 245–252 (1982)
14. Lin, W.S., Zhao, H.V., Liu, K.: Cooperation Stimulation Strategies for Peer-to-Peer Wireless Live Video-Sharing Social Networks. IEEE Transactions on Image Processing 19(7), 1768–1784 (2010)
15. Rheingold, H.: The virtual community: homesteading on the electronic frontier. The MIT Press (2000)

Control of Snake Type Biomimetic Structure

Mircea Ivanescu¹, Nicu Bizdoaca¹, Hani Hamdan²,
Mario Eltabach³, and Mihaela Florescu¹

¹ University of Craiova,
A.I.Cuza, 13, Romania

² SUPELEC, Dept. of Signal Processing and Electronic Systems,
3 rue Joliot-Curie, 91192 Gif-sur-Yvette cedex, France,

³ VALORTIM-CETIM, Noise and vibration department,

52 avenue Félix-Louat, 60300 Senlis, France

{ivanescu,nicu,florescu}@robotics.ucv.ro,
Hani.Hamdan@supelec.fr, Mario.Eltabach@cetim.fr

Abstract. Robotic cooperative tasks impose, in many cases, a grasping action. Grasping by coiling it is one of the most versatile action. The present article propose a frequency stability criterion based on the Kahman – Yakubovich – Popov Lemma for the hyper-redundant arms with continuum element that performs the grasping function by coiling. Dynamics of the biomimetical robot during non-contact and contact operations, for the position control, is studied. An extension of the Popov criterion is developed. The P control algorithms based on SMA snake-type robot actuators are introduced. Numerical simulations and experimental results of the snake type robot motion toward an imposed target are presented.

Keywords: biomimetics, control, robot,snake-type robot.

1 Introduction

Snake type robot and continuum robot represents “state of art” for robotics specialist. The medical robots which allow minimal invasive medical techniques (surgery in the throat, inside the heart, in the stomach) is one of the strongest and actual application of this structure. Ingenious snake type robots consist of multiple miniaturized stages that are connected in series, or flexible links that function as both link and joint. Developing any of these robots poses a common set of problems: design optimization, choice of sensing, kinematic modeling, procedure planning and real-time control. This is the reason that remarkable studies are reported regarding:

- kinematics - Gravagne [2] analyzing the kinematic model of “hyper-redundant” robots, set up term of continuum” robots, Chirikjian and Burdick introduce “backbone curve” of hyper-redundant robot, that captures the robot’s macroscopic geometric features),

- control - Mochiyama investigated the problem of controlling the shape of an HDOF rigid-link robot spatial curves [7], in [8], the “state of art” of continuum robots, their areas of application and some control issues are presented)
- technological implementation - [9, 10] deal with several technological solutions for actuators used in hyper-redundant structures and with conventional control systems.

Control problem of snake-type biomimetic robotic structure presented in the current paper has as target robot grasping function by coiling.

2 Proposed Technological Implementation

In this paper a 2D model for snake-type robot is discussed. The assumption is correct because even the serpentine technological robots operate in 3D space, the grasping function of these arms is, generally, a planar function.

The technological proposed implementation is presented in Fig.1. It consists of layered structures that ensure the flexibility, driving and position measuring. The high flexibility is obtained by an elastic backbone rod. The driving layer is made up of two antagonistic SMA actuators, A and B, each of them having a number of SMA fibers that are connected to the ends of the beam and determine its bending by current control. These SMA fibers are well suited for grasping force control due to their high strength to weight ratio.

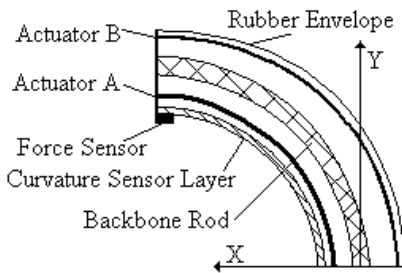


Fig. 1. The segment layer structure

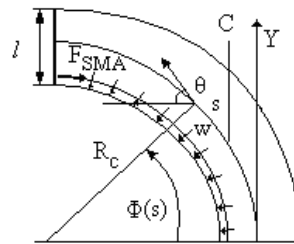


Fig. 2. The body robot parameters

The measuring layer is represented by an electro-active polymer curvature sensor. This sensor is placed on the boundary of the beam and allows for its curvature measuring by the resistance measuring. The sensor system is completed by a number of force sensors placed at each terminal of the beam segment. A rubber envelope protects and isolates this layer structure from the operator environment.

The general form of the body snake-type robot consists of a number (N) of segments and the last m segments ($m < N$) represent the grasping terminals.

As a theoretical model, we shall consider the beam in Fig.2 with the length L and the thickness l . This beam has been deflected into a circular arc by a SMA fiber. The

beam is composed of concentric arcs. The neutral arc defines the curvature of the beam,

$$c_v = \frac{d\phi}{ds}, \quad \phi = \frac{s}{R_c} \tag{2.1}$$

where ϕ represents the angle of the current position, s is the arc length from the origin, and R_c is the radix of the arc.

We denote the equivalent force developed by the SMA actuators at the end of the beam ($s=L$) by χ , the force density and the distributed force along the beam exercised by the SMA fibers on the beam surface by w and F , respectively, and τ is the equivalent moment of the beam.

From [11], we have the following relations

$$w = \frac{dF}{ds}, \quad w = \chi \cdot c_v, \quad \tau = \chi \cdot \frac{l}{2} \tag{2.2}$$

$$dF = \chi \cdot d\phi, \quad dF = -\chi \cdot d\theta \tag{2.3}$$

3 Robot Dynamic Model for Grasping by Coiling

The grasping function control is represented by the force control between the body robot and load. Consider that the robot body has achieved the desired position defined by the surface (object).

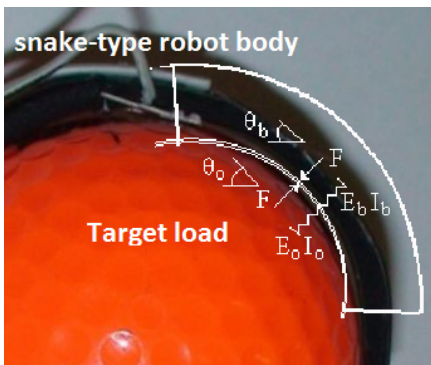


Fig. 3. The grasping model

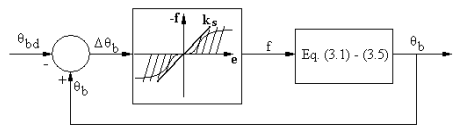


Fig. 4. The grasping control closed loop system

In Fig.3, an object with elastic and damping parameters $E_o I_o$, b_o and c_o , respectively, is grasped by coiling. Using the same procedure as developed in [16], the dynamic model of the two bodies in contact, body and load, is represented by the following partial differential equations,

$$\dot{\tilde{e}} = \tilde{A} \frac{\partial^2 \tilde{e}}{\partial s^2} + \tilde{B} \tilde{e} + \tilde{c} f \tag{3.1}$$

$$\tilde{e}(0, s) = 0; \tilde{e}(t, 0) = \frac{\partial \tilde{e}(t, 0)}{\partial s} = 0 \tag{3.2}$$

$$E_b I_b \cdot \frac{\partial e_b(t, L)}{\partial s} = \tau^* \tag{3.3}$$

$$e_b(t, L) = e_\theta(t, L); \frac{\partial e_o(t, L)}{\partial s} = \frac{\partial e_b(t, L)}{\partial s} \tag{3.4}$$

$$\frac{\partial \dot{\tilde{e}}(t, L)}{\partial s} = -\alpha_1 \tilde{e}(t, L) - \alpha_2 \dot{\tilde{e}}(t, L) \tag{3.5}$$

Where $\tilde{e}(t, s) = [e_b(t, s), e_o(t, s)]^T$, with $e_b(t, s) = \theta_b(s) - \theta_{bd}(s)$; $e_o(t, s) = \theta_o(s) - \theta_{od}(s)$, f is the force error, τ^* is the control variation, and the indices b and o specify the parameters of the beam and object, respectively.

$$\tilde{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{I_{\rho b}}(E_b I_b + E_o I_o) & 0 & -\frac{E_o I_o}{I_{\rho b}} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{E_o I_o}{I_{\rho o}} & 0 & \frac{E_o I_o}{I_{\rho o}} & 0 \end{bmatrix}; \tilde{B} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{b_b}{I_{\rho b}} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{b_b}{I_{\rho o}} \end{bmatrix}; \tilde{c} = \begin{bmatrix} 0 \\ \frac{c_b}{I_{\rho b}} \\ 0 \\ -\frac{c_o}{I_{\rho o}} \end{bmatrix} \tag{3.6}$$

4 Control Algorithm

The grasping force control is the second problem of the grasping control. A force sensor network is used to account for the contact between the robot body and the load. We notice, from (2.2), that the force density w is constant along the robot segment and, in a steady state, w can be approximated to f . A force sensor with the position $s = s^* \in [0, L]$ is used to measure the contact force. The contact force – displacement relation of the sensor is assumed to lie in the positive sector (Fig.4).

$$\Delta F = -\psi(\Delta\theta), \psi(\Delta\theta) \cdot \Delta\theta \geq 0 \tag{4.1}$$

$$\psi(0) = 0 \text{ for } \Delta\theta = 0 \tag{4.2}$$

The nonlinearity $\psi(\Delta\theta)$ is single-valued, time invariant and constraint to a sector bounded by slope k_s which is assumed to meet

$$0 \leq \frac{\psi(\Delta\theta)}{\Delta\theta} \leq k_s < \infty \tag{4.3}$$

which is the case for most physically realistic elastic contacts.

In terms of the sensor characteristics, the convergence to zero of the error e_b is equivalent to the convergence to zero of the contact force error f

$$\lim_{t \rightarrow \infty} e_b(t, s^*) = 0 \Rightarrow \lim_{t \rightarrow \infty} f(t, s^*) = 0 \tag{4.4}$$

The sensor nonlinearity (4.1) – (4.3) and the dynamic model of the grasping contact described by (3.1) – (3.5) suggest the closed – loop system of Fig.4.

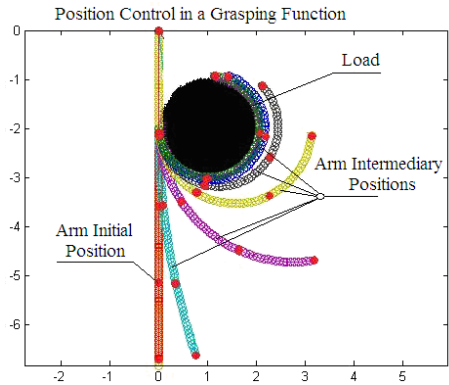
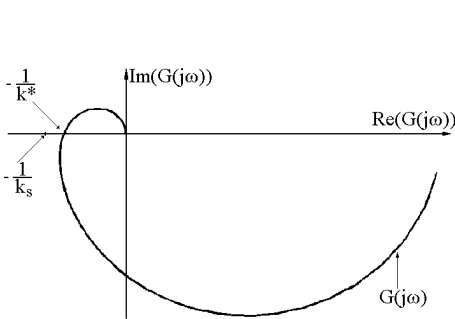


Fig. 5. The plot of $G(j\omega)$ for the grasping control **Fig. 6.** The simulation of the grasping operation control

Theorem. The closed – loop system (Fig.4) is absolutely stable if:

- (1) $(-\tilde{A} + \tilde{B})$ is a Hurwitzian matrix;
- (2) the pair $(-\tilde{A} + \tilde{B}, \tilde{c})$ is completely controllable;
- (3) there is a positive definite and symmetrical matrix P such that $(\tilde{A}^T P + P \tilde{A})$ is positive definite;

$$(4) \frac{1}{k_s} + \text{Re} \left[\tilde{n}^T \left(j\omega I - (-\tilde{A} + \tilde{B}) \right)^{-1} \tilde{c} \right] \geq 0 \tag{4.5}$$

- (5) the moment of the robot body verifies the relation

$$\tau^* = k_p e_b(t, L), \quad k_p > E_b I_b \tag{4.6}$$

Equations (4.1), (4.3) and (3.1) – (3.5) describe the closed loop system (Fig.4), consisting of a partial derivative equation linear system and a nonlinear element represented by the function $\psi(\cdot)$ belonging to the sector $[0, k_s]$. In this case, the condition (4) represents the Popov criterion for this class of systems. According to it, the system will be absolutely stable if the plot of $\tilde{G}(j\omega)$

$$\tilde{G}(j\omega) = \tilde{n}^T \left[j\omega I - (-\tilde{A} + \tilde{B}) \right]^{-1} \tilde{c} \quad (4.7)$$

crosses the negative real axis at a point that lies to the right of the critical point defined by $-\frac{1}{k_s}$ (Fig.5). For a pair “robot body segment – load” specified, the plot of $\tilde{G}(j\omega)$ has a well-defined characteristic and the intersection with the real axis determines the limit value of k. Let k^* be the corresponding value of the crossing point. The absolute stability is guaranteed if the sensor parameters meet the condition

$$k_s \leq k^* \quad (4.8)$$

5 Numerical Simulation and Experimental Results

A hyper-redundant manipulator with 4 segments is considered, Fig.6. The parameters of the robot body were selected run as follows: bending stiffness $E_b I_b = 1$, linear mass density $\rho_b = 0.5 \text{ kg/m}$, rotational inertial density $I_{\rho b} = 0.001 \text{ kg} \cdot \text{m}^2$ and damping ratio 0.35. These constants are realistic for long thin backbone structures. The grasping function is exercised by the last three segments of the robot body, the length of each segment is $L = 1$ (Geometrical parameters are scaled.). The load is a cylinder with the radix $R = 1$, bending stiffness $E_o I_o = 0.2$, rotational inertial density and damping ratio 0.22.

The plot of $G(j\omega)$, Fig.7, crosses the negative real axis at -0.14 , which imposes the limit of tension at $T^* = 7.15 \text{ N}$; the plot of $\tilde{G}(j\omega)$ crosses the negative real axis at -0.74 , which corresponds to the critical value of the force sector at $k^* = 1.3$.

In Fig .8 force phase portrait for a P – control algorithm (4.5) with $k_p = 24$ is illustrated. Please note the convergence to zero of the force error, but, also, the transient response of the system determined by the P – control law and a low damping factor of the system.

In order to verify the suitability of the control algorithm, a planar continuum terminal robot structure consisting by a layer structure has been employed for testing (Fig.9). The robot body consists of two $(25 \times 6 \times 4 \text{ mm})$ continuum segments with an

elastic backbone rod. The two antagonistic SMA actuators ensure the actuation system. Each actuator consists of G fibers in parallel. A polymer thick film layer which exhibits a decrease in resistance with an increase of the film curvature is used. Also, a Force Sensing Resistor is used at the end of each segment.

A Quancer based platform is used for control and signal acquisition. The load is represented by a sphere ball with $R_c = 0.02m$. A P-control with $k_p = 2.17$, $T_D = 5s$, $T_P = 7s$ is implemented. The contact force in the grasping operation is represented in Fig.10.

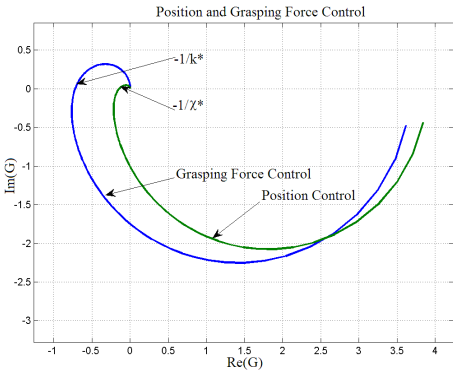


Fig. 7. The plot of $G(j\omega)$ for position and force control

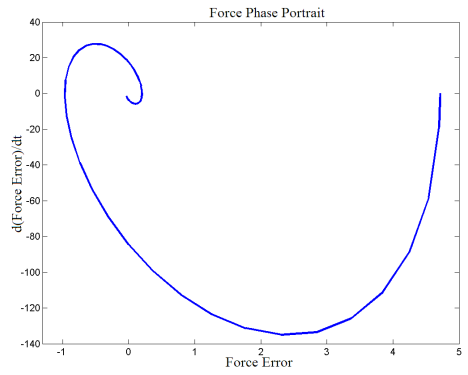


Fig. 8. The force phase portrait

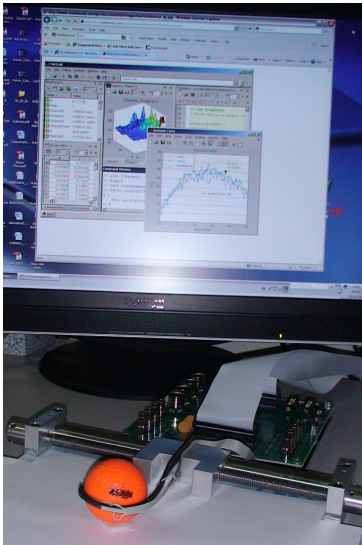


Fig. 9. Experimental platform

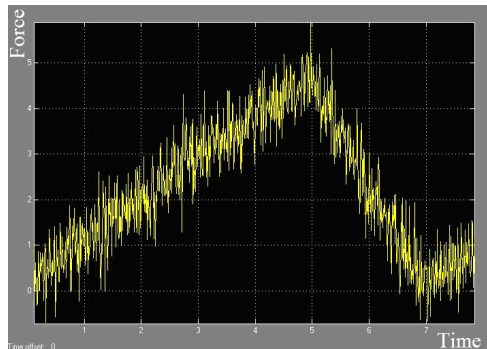


Fig. 10. The contact force diagram

6 Conclusions

In this paper, we have presented a control algorithm for biomimetic snake-type robot with continuum elements that performs the coil function for grasping. First, the dynamic model of continuum robot for the position control during non-contact operations with environment is studied. The P control algorithms are proposed. Then, the grasping control problem for the robot in contact with a load is analyzed. The control algorithms based on SMA actuators are introduced. Proposed approach is validated through simulations and experiments. Our future work is to use for actuation polymer based artificial muscle, in order to improve the robot flexibility and to develop an improved version of biomimetic robotics structure

Acknowledgement. This work was supported by CNCSIS –UEFISCSU, project number PNII – IDEI code 289/2008, “Reverse Engineering in Cognitive Recognition And Control Of Biomimetics ”, project director prof. Nicu Bizdoaca, Ph.D., University of Craiova.

References

1. Hemami, A.: Design of light weight flexible robot arm. In: Robots 8 Conference Proceedings, Detroit, USA, pp. 1623–1640 (1984)
2. Gravagne, I.A., Walker, I.D.: On the kinematics of remotely - actuated continuum robots. In: Proc. 2000 IEEE Int. Conf. on Robotics and Automation, San Francisco, pp. 2544–2550 (2000)
3. Gravagne, I.A., Walker, I.D.: Kinematic Transformations for Remotely-Actuated Planar Continuum Robots. In: Proc. 2000 IEEE Int. Conf. on Rob. and Aut., San Francisco, pp. 19–26 (2000)
4. Gravagne, I.A., Rahn, C.D., Walker, I.D.: Good Vibrations: A Vibration Damping Setpoint Controller for Continuum Robots. In: Proc. 2001 IEEE Int. Conf. on Robotics and Automation, Seoul, Korea, pp. 3877–3884 (2001)
5. Gravagne, I.A., Walker, I.D.: Uniform Regulation of a Multi-Section Continuum Manipulator. In: Proc. IEEE Int. Conf. on Rob. and Aut., Washington, A1-15, pp. 1519–1524 (2002)
6. Chirikjian, G.S., Burdick, J.W.: An obstacle avoidance algorithm for hyper-redundant manipulators. In: Proc. IEEE Int. Conf. on Robotics and Automation, Cincinnati, Ohio, pp. 625–631 (1990)
7. Mochiyama, H., Kobayashi, H.: The shape Jacobian of a manipulator with hyper degrees of freedom. In: Proc. 1999 IEEE Int. Conf. on Robotics and Automation, Detroit, pp. 2837–2842 (1999)
8. Robinson, G., Davies, J.B.C.: Continuum robots – a state of the art. In: Proc. 1999 IEEE Int. Conf. on Rob and Aut., Detroit, Michigan, pp. 2849–2854 (1999)
9. Ivanescu, M., Stoian, V.: A variable structure controller for a tentacle manipulator. In: Proc. IEEE Int. Conf. on Robotics and Aut., Nagoya, pp. 3155–3160 (1995)
10. Ivanescu, M., Florescu, M.C., Popescu, N., Popescu, D.: Position and Force Control of the Grasping Function for a Hyperredundant Arm. In: Proc. of IEEE Int. Conf. on Rob. and Aut., Pasadena, California, pp. 2599–2604 (2008)

11. Camarillo, D., Milne, C.: Mechanics Modeling of Tendon – Driven Continuum Manipulators. *IEEE Trans. on Robotics* 24(6), 1262–1273 (2008)
12. Wongratanaphisan, T., Cole, M.: Robust Impedance Control of a Flexible Structure Mounted Manipulator Performing Contact Tasks. *IEEE Trans. on Robotics* 25(2), 445–451 (2009)
13. Grant, D., Hayward, V.: Constrained Force Control of SMA Actuators. In: *Proc. ICRA 2000*, San Francisco, pp. 1314–1320 (2000)
14. Mihlin, S.G.: *Variationnie Metodi b Matematicheskvi Fizike*. Nauka, Moscva (1970)
15. Slotine, J.J., Li, W.: *Applied Nonlinear Control*. Prentice-Hall International Editions (1991)
16. Ivanescu, M., Bizdoaca, N., Florescu, M.C., Popescu, N., Popescu, D.: Frequency Criteria for the Grasping Control of a Hyper-Redundant Robot. In: *Proc. of IEEE Int. Conf. on Rob. and Aut. (ICRA 2010)*, Anchorage, Alaska, pp. 1542–1549 (2010)

A Filter for the Cooperative Kinase Network of Budding Yeast *Saccharomyces cerevisiae*

Jian-Qin Liu^{1,*} and Tadashi Nakano²

¹Kobe Advanced ICT Research Center,
National Institute of Information and Communications Technology,
588-2 Iwaoka, Iwaoka-cho, Nishi-ku, Kobe, Hyogo, 651-2492, Japan
liu@nict.go.jp

²Frontier Research Base for Global Young Researchers,
Graduate School of Engineering, Osaka University,
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan
tnakano@wakate.frc.eng.osaka-u.ac.jp

Abstract. In the cooperative kinase network, a kinase interacts with other kinases for sustaining cellular signaling processes that greatly influence the major functions of cells. Here a key question is how the interacting kinases form a filtering network to estimate the original signal in the presence of stochastic fluctuation caused by the interactions. In this short paper, a filter is designed to estimate the concentration of the molecular signal Ste20 of the MAPK (mitogen-activated protein kinase) cascade in budding yeast based on the Ste20-Ste11-Ste7 pathway, in which kinases interact with each other. The filter is tested in simulations and the result shows that the estimated signal can be used to recognize the original signal. It is concluded that the Ste20-Ste11-Ste7 pathway can be regulated to analyze cell cycle processes through the interactions among kinases in the MAPK cascade.

Keywords: Signaling pathway, kinase, MAPK cascade, bioinformatics.

1 Introduction

Motivated by a recent report on a kinase and phosphatase interaction network which shows that cross-talks among kinases play an important role in cellular signaling [1,2], we have investigated filtering mechanisms of the MAPK (mitogen-activated protein kinase) cascade in budding yeast *Saccharomyces cerevisiae* – the Ste20-Ste11-Ste7-Kss1/Fus3 pathway [3] where cross-talks act as the convolution factor of a layered filter. In this pathway, signaling is initiated from the top of the hierarchy of signaling processes involving Ste20 and streamed toward the bottom of the hierarchy where Kss1 or Fus3 is involved. Here MAPKKKK (mitogen-activated protein kinase kinase kinase kinase) is Ste20; MAPKKK (mitogen-activated protein kinase kinase kinase) is Ste11; MAPKK (mitogen-activated protein kinase kinase) is Ste7; MAPK (mitogen-activated protein kinase) has two types – Kss1 and Fus3 that have different effectors. This pathway determines the function of the cell cycle process of budding

* Corresponding author.

yeast where the kinases of the pathway interact with each other as well as other signaling molecules in the kinase interaction network. In the MAPK cascade, Ste20 also interacts with Cdc28, Cbk1 and Swe1; and Ste11 interacts with Pbs2 in addition to Ste7, Kss1, and Fus3.

In order to investigate the ability of the pathway to restore the original signal in the presence of stochastic effects, we have designed a filter based on the Ste20-Ste11-Ste7 pathway where Ste11 interacts with Pbs2, Ste11 with Kss1, and Ste11 with Fus3. The filter is designed based on four order derivatives to perform nonlinear filtering for random signals. Simulation experiments were conducted to understand the ability of the pathway to estimate the original signal level of Ste20 where the interactions of Ste20 with Cdc28, Cbk1, and Swe1 are formalized as a stochastic process.

2 Results

A simulation result is given in Figure 1 where the original signal level of Ste20 varies in the range of [2.57, 3.44] (nM) and the estimated result for the filter is stationary with the variation within [2.82,3.11] (nM). Owing to the stationary characteristic of the estimated result, the mean (expectation) of the original signal level is inferred as 3 nM with the error limitation ± 0.2 nM. When the periodic and random influence on Ste20 is considered, the filter can also restore the original signal with the error limitation of ± 0.2 nM. It is inferred that the filtering mechanism of budding yeast may help us to explain how the specificity of the MAPK cascade is sustained in the dynamic processes of cell division.

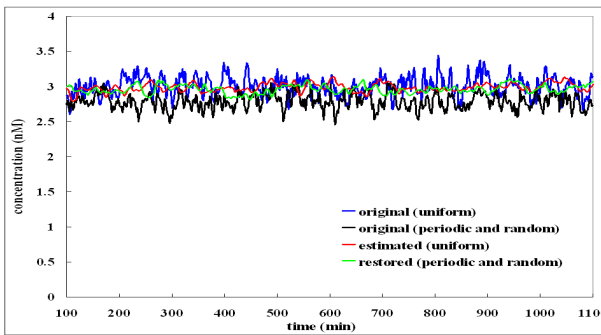


Fig. 1. Concentration-versus-time curve of the original signal (blue) under the uniform distribution; the original signal under the periodic and random influence (black); the estimated result of the filter (red) in the uniform case; the restored result of the filter (green) in the periodic random case

References

1. Breitkreutz, A., Choi, H., Sharom, J.R., et al.: A Global Protein Kinase and Phosphatase Interaction Network in Yeast. *Science* 328, 1043–1046 (2010)
2. Levy, E.D., Landry, C.R., Michnick, S.W.: Signaling Through Cooperation. *Science* 328, 983–984 (2010)
3. Elion, E.A., Qi, M., Chen, W.: Signaling Specificity in Yeast. *Science* 307, 687–688 (2005)

Application of Genetic Algorithm to Maximise Clean Energy Usage for Data Centres

Raymond Carroll, Sasitharan Balasubramaniam, Dmitri Botvich,
and William Donnelly

TSSG, Waterford Institute of Technology, Ireland
{rcarroll, sasib, dbotvich, wdonnelly}@tssg.org

Abstract. The communications industry is currently witnessing a continued increase in energy consumption, and this trend is predicted to increase even more in the coming years. This is largely driven by the popularity of the Internet, which continues to attract growing numbers of users who now rely on the Internet as part of their daily lives. A major factor behind this attraction is the multitude of services available on the Internet, ranging from web based services (e.g. facebook) to heavy power consuming services such as multimedia (e.g. youtube, IPTV). Therefore the data centres housing these services are seeing their energy consumption increase proportionally, now leading researchers to actively search for solutions to improve the energy efficiency of data centres. In this paper we propose a green data centre solution that makes data centres and services prioritise the usage of clean, renewable energy sources. The solution allows data centres to share information regarding renewable energy and cooling, in order to exploit variance between different countries energy and temperature profiles by moving services between data centres. We employ a genetic-algorithm to find the optimal placement of services on the data centres.

Keywords: Green Data Centres, Energy Efficiency, Genetic Algorithm.

1 Introduction

In recent years there has been a growing focus on the impact of the internet, and more specifically data-centres, on the environment, in terms of their increasing energy usage. Figure 1(a) shows that from 2000-2006 the energy usage for data centres in the US [1] more than doubled. It also depicts the predicted trends up to 2010, extrapolated based on both the historical data and also based on recent trends towards energy efficiency, where both show huge increases in energy usage.

While these trends do consider the impact of the move towards more energy efficient practices, they do not consider the impact that new technologies and computing models may have. For instance the growth in the usage of ‘smart’ phones in recent years has been exceptional. These phones are in essence resource limited computing platforms, where often times much of the processing is done in back-end service/application residing on the data-centre. Also, the recent move towards cloud

computing holds huge potential for increasing data centre usage. Cloud computing proposes to move all the majority of the application processing and data storage into the data centre, with ‘thin’ client devices running simple interfaces. These emerging trends suggest that data-centres could grow beyond what has been predicted, and continue this rate of growth into the future. At the same time, many countries are now actively pursuing more renewable sources of energy, through their own capital infrastructure projects or through grid feed-in tariff incentive schemes. This is illustrated in Figure 1(b), where we show the recent capacity increases in wind and solar energy within the EU states.

Based on these developments, our work attempts to address the problem of data-centre energy usage by allowing data-centre operators to determine a service placement strategy with the best renewable and cooling energy profile. This in turn reduces the overall carbon footprint of the data centre operator. To do this we employ a genetic algorithm (GA) based service placement approach, where the GA determines the most optimal service/data-centres pairings to maximize data-centre usage of renewable energy sources and minimise cooling energy.

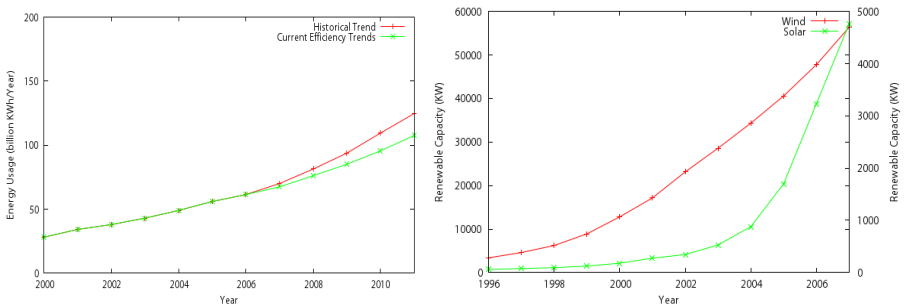


Fig. 1. (a) Data Centre Energy Trends (b) Wind and Solar Capacity increases of EU countries

2 Related Work

There is now an array of work being carried out in the area of energy efficiency for data centres. Many of these approaches focus on moving or scheduling workload in some way to achieve greater energy efficiency. Many of these have looked at consolidating workloads on a minimum number of servers in order to allow certain servers to be switched off or sleep to save power [2][3][4][5]. In [6] biological mechanisms are used to determine more efficient servers in a data centre where load is subsequently moved. Other works investigate how to make the data centres more efficient by reducing the load on the cooling systems through better workload placement and scheduling within and between data centres [7][8][9][10][11]. A similar approach to ours is taken in [12], where traffic load is moved between data centres based on electricity costs rather than renewable energy.

3 Sustainable Energy Prioritization Solution

In this section we introduce our proposed solution, outlining the service placement process and underlying genetic algorithm. Our goal is to place services on data centres in such a way as to maximise renewable energy usage and minimise cooling energy. The renewable energy consumption of a data-centre is measured through a value we call the renewable energy ratio (RER – Section 4), and this is used as a key metric in calculating the data centres which use more renewable energy. In terms of cooling, the ambient temperature of the country can have a significant effect on the efficiency of the cooling system. This efficiency is measured through a value termed the Coefficient of Performance (COP – Section 3.2) and so this value also plays a key role in calculating the data centres with the most energy efficient cooling. By using these values we can determine the best data-centres to place services on which will result in more renewable energy usage and lower cooling energy usage.

3.1 GA-Based Service Placement

The service placement process takes place in effectively three stages. Initially all data centres must co-ordinate and share information regarding their renewable and cooling energy usage levels, service usage details and data centre configuration information. This information forms the basis for our genetic algorithm to determine the fittest service configuration. In our solution the genetic algorithm is run periodically by a specific, pre-selected data centre (referred to as the GA-DC). However, for the purposes of redundancy each data centre is capable of running the algorithm and so the energy data is shared among all data centres.

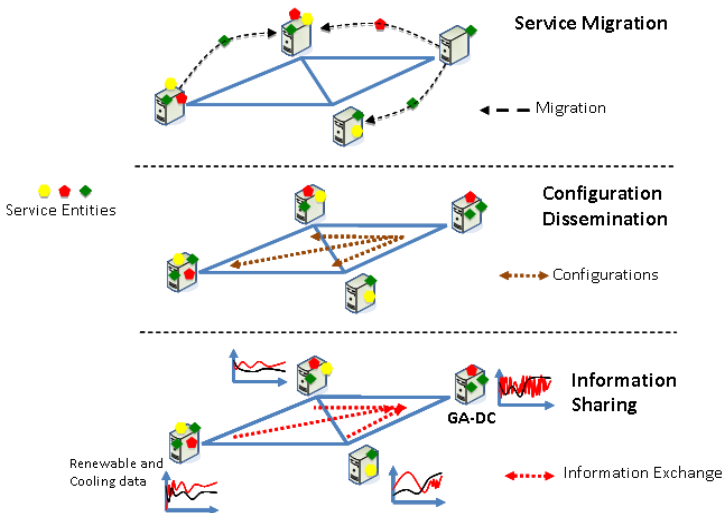


Fig. 2. Service Placement Process

As mentioned earlier the genetic algorithm will determine the most optimal service/data centre configuration, based on maximising the usage of renewable energy and minimising the consumption of energy in cooling the data centre. The genetic algorithm is described in detail in the following sections.

Once the optimal configuration has been found, this configuration is disseminated to all data centres. Each data centre then examines this configuration and implements its recommendations. To do this the data centre determines which of its currently hosted services have been selected to move and to what data centre. Once ascertained the data centres then migrates the selected services to their newly designated data centre. Each data centre is notified of the change in order to update their information registries and so that requests for the service at the originating data centre will be forwarded to its new location.

3.2 Genetic Algorithm

We begin the discussion of our genetic algorithm by looking at the fitness function we will use to determine the best solution. The fitness function is composed of two parts, the renewable energy consumed and the cooling energy consumed. These are conflicting optimization goals, since the data centres with the best renewable energy ratio are not necessarily the data centres with the best environmental conditions for efficient cooling. Below (1) we present our approach for maximising the renewable energy consumed and minimising the cooling energy used. Let the set of services be $S_i = \{s_1, s_2, \dots, s_i, \dots, s_N\}$, where N is the total number of services; the set of Data Centres be $DC_j = \{DC_1, DC_2, \dots, DC_j, \dots, DC_M\}$, where M is the total number of Data Centres. Let RER_j be the Renewable Energy Ratio of data centre j , CE_j be the Cooling Energy of Data Centre j , sl_{ij} be the service load of service i on data centre j , and DCC_j is the capacity of data centre j . In (1) below we present our fitness function which attempts to maximise the renewable energy consumed and minimise the cooling energy used.

$$\max \sum_{j=0}^{j < K} \sum_{i=0}^{i < N} ((1 - \alpha)(sl_i \cdot RER_{DC_j}) + a \frac{1}{CE_j}) \tag{1}$$

Subject to

$$\sum_{i=0} sl_{ij} < DCC_j \tag{2}$$

To determine the renewable energy consumed we use the renewable energy ratio (RER_j) of the data centre in question. The RER is the ratio of renewable energy production to total energy production in the data centres host country (see Section 4). This ratio gives us the best indication possible of what proportion of energy from renewable sources the data centre is consuming. However the quantity of renewable energy consumed is a factor of the load on the data centre also. As such we need to calculate the load exerted on the data centres by the service (sl_i).

The cooling energy (CE) of a data centre is calculated based on heat load (HL) to be removed from the data centre subject to the efficiency of the cooling system (COP)

in removing this heat (see (3)). The heat load is directly related to the energy being consumed by the computing equipment, which is then converted to heat. As such we calculate the heat load by determining the power being used in the data centre. This is shown in (4), where P_{max} is the maximum power a single server consumes at peak load, P_{idle} is the power a single server consumes while idle (load = 0), sl_i is the load exerted on the server by a single service i and ns is the number of servers in the data centre. Since the idle power is consumed irrespective of server workload, the workload only impacts the power consumed above the idle power ($P_{max} - P_{idle}$).

$$CE = \frac{HL}{COP} \tag{3}$$

$$HL = \sum_{i=1}^{i < N} (sl_i) \cdot (ns \cdot (P_{max} - P_{idle})) + ns \cdot P_{idle} \tag{4}$$

Before we discuss the COP in more detail, there are constraints on the GA which we must mention. The utilisation of the services assigned to a specific data centre cannot exceed the capacity of that data centre (2). In addition, a service must be assigned to only one data centre (especially important in crossover and mutation).

Coefficient of Performance (COP)

Critical to the calculation of the cooling energy fitness value is the Coefficient of Performance (COP) of each data centre. The COP value indicates the efficiency of the cooling system in removing the heat load from the data centre (5). A high COP means the thermodynamic process is more efficient.

$$COP = \frac{HeatLoad}{CoolingEnergy} \tag{5}$$

$$COP = \frac{1}{\frac{T_H}{T_C} - 1} \quad \begin{matrix} T_H = \text{Outside Temperature} \\ T_C = \text{Inside Temperature} \end{matrix} \tag{6}$$

Under the principles of thermodynamics [13], the efficiency of a typical heat pump is highly dependant on both the inside (target) temperature and the environmental (outside) temperature to which the removed heat is rejected. Therefore the greater the outside temperature is (for a set inside temperature) the more inefficient the system.

In most cases, once the outside temperature drops below the indoor temperature air conditioning is typically not required. However in the case of data-centres, the primary heat load is not coming from heat transfer from the environment but rather the computing equipment, so cooling is still required. In line with best practices of data centre cooling, we design each data-centre with a free cooling system in addition to conventional cooling. Free cooling allows data-centres to utilize the outdoor environmental conditions to part, or even fully, cool the data centres when conditions

allow. Typically this is when the outside temperature is below the indoor temperature, thus free cooling is also highly dependant on the weather conditions.

As a result we employ a COP model based on the assumption that both free-cooling and electric/mechanical-cooling are employed. Once the outdoor temperature is above the required cooling temperature, COP values based on standard electrical cooling are employed. However, when the outdoor temperature drops below the required indoor temperature we move to free cooling and adapt the COP model inline with the changeover. We do not subscribe to a specific free-cooling system, instead we generalize based on the assumption that free-cooling provides a significant improvement in efficiency of the cooling system.

For our non-free cooling COP model we adopt COP values from the ORNL [14] heat pump simulator. For free cooling we simply adapt the values of the ORNL COP such that when the outside temperature drops below the inside temperature, we adjust the COP relative to the original COP value (e.g. +40%). This aims to represent that, once the outside temperature is cooler than inside, the free-cooling system is in operation. However we do not assume that free-cooling COP is uniform, as the energy required by a free-cooling system can vary depending on the extent by which the outside temperature is cooler than inside. For instance air-pumps may need to pump less air to cool the server room the cooler the outside temperature gets. So, using this model we can determine the COP based on the known outside and inside temperature.

The next step in implementing our GA solution is to encode our problem into a chromosome representation. In essence each chromosome is required to represent a configuration of the entire set of services placed across the nine selected data centres. In our representation each gene represents a single service placed on a single data centre. Specifically, each gene contains two parts, the service in question and an ordered list of binary values indicating the data centre on which the service is placed. In this way we must first calculate the fitness of each individual gene, by examining the workload details of the service, and the details of the data centre on which it is placed. Once each gene's fitness has been calculated we can then sum these values to determine the overall chromosome fitness.

3.3 Genetic Algorithm Operation

The algorithm begins by randomly creating a population of chromosomes subject to the constraints outlined before (2). Once the initial population has been generated, we calculate the fitness of each chromosome. Then, based on the elitism approach, we select the two fittest chromosomes to be carried forward to the next generation. In order to populate the remainder of the new generation, we select two parent chromosomes and perform crossover in order to generate new offspring chromosomes. We employ roulette wheel selection to choose the parents and then perform single point crossover to create the child chromosome(s). Again care must be taken when performing crossover, that the resulting chromosome(s) do not cause any data-centre to exceed its maximum capacity. In our algorithm mutation is carried out by simply changing one of the binary values representing the data centre on which a service resides. In other words this results in the service being placed on a different

data-centre than before, and again care must be taken to not break the constraints outlined.

3.4 GA Evaluation

In this section we perform some initial evaluations of the genetic algorithm itself, to ensure its correct operation. We vary the main GA parameters (population size, generations, crossover rate) as seen in the results presented in Figure 3.

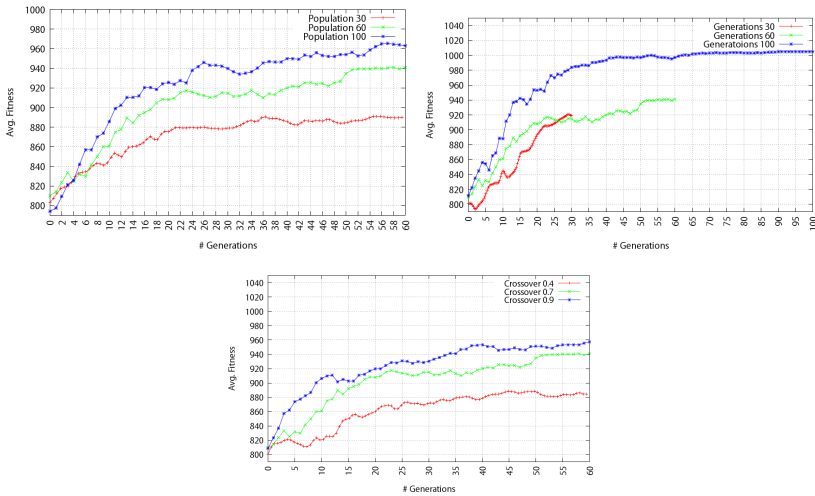


Fig. 3. (a) - Varying Generation Size (b) Varying Population Size (c) Varying Crossover Rate

We start by varying the number of generations for which the GA runs, as seen in Figure 3. As expected the greater the number of generations the better the overall result. The smaller generation size does obtain a reasonable fitness value quickly, but the 60 and 100 generation simulations are able to obtain higher values over time. Increasing the population size reduces the effects of randomness and gives a more diverse starting population. As expected, this leads to a stronger average population fitness, increasing in line with the population size increase. Finally, again as we would predict, higher crossover rates lead to more diversity in the populations and hence allow fitter, more optimal solutions to be found. At a low crossover rate we can see that the algorithm struggles to improve the population fitness since it is more difficult to breed new solutions from parents with higher fitness values.

4 Case Study

In order to properly present our solution we confine it to a fixed case study comprising a specific set of data centres and services, which will also be used later in the evaluation of our solution. We also specify the real temperature and energy data

we have obtained, which are used in the calculation of the fitness function. Nine data centres were selected from major European cities, including Dublin, London, Lisbon, Madrid, Milan, Athens, Amsterdam, Berlin and Copenhagen. These were chosen in an attempt to give a significant variation in both climatic conditions and sources of renewable energy used.

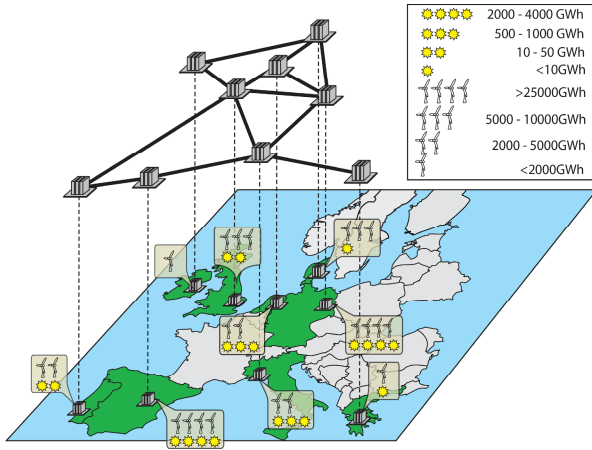


Fig. 4. Case Study Configuration

The data centres are connected in a network topology as depicted above, in line with the European Optical Network. The size of each data centre is determined by the number of servers it contains, which is relative to the population of the host country. Core to the approach taken in this paper is the use of real energy and weather data for the countries where the data centres reside. In line with this we have carried out a detailed search for data relating to the renewable energy production of each data centre host country, as well as temperature variations for a period covering January 2007 to December 2009. The energy production values, as described subsequently, are taken from [15].

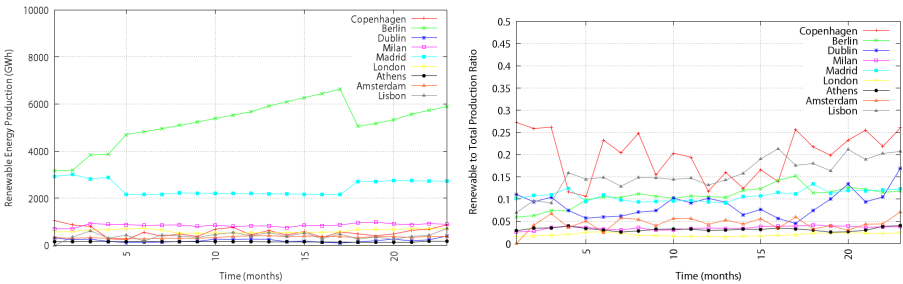


Fig. 5. (a) Renewable Energy Production, (b) Renewable Energy/Total Production Ratio

In Figure 5(a) we show the total renewable production for each of the selected data centres countries over the aforementioned period. Instinctively the larger countries

will have greater volumes of renewable energy production (e.g. Germany and Spain), with the country's policy for sustainable growth also affecting capacity values. Other factors, such as weather conditions and capacity increases account for variations in the values from month to month. In Figure 5(b) we present the renewable energy values as a fraction of the overall energy production of the data-centres home country. This gives a clearer representation of the countries with the most energy production, and hence those that are more desirable candidates for services to migrate to. In terms of the temperature variations of each country/data centre, we used data from the European Climate Assessment & Dataset (ECA&D) [16] project, recording real temperature data from across Europe. Also, in order to calculate the cost impact on data centres we also used the real energy unit price as reported by the European Commission (Eurostat [17]).

5 Simulation and Results

In the following section we perform a case study simulation of the potential renewable energy gains possible for a small sized data centre operator, based on the genetic algorithm and scenario outlined in the previous sections. The operator runs nine data centres distributed as seen in Figure 4. Within each of these data centres there are a varying number of servers (8-200) and services (16-400), proportionate to the population of the country. In terms of the server specifications, we stipulate a standardized server across all data centres with a maximum power draw (P_{max}) of 400w and an idle power draw (P_{idle}) of 150w. To represent the workload exerted on the server, each service is randomly assigned a value that denotes how much of the servers processing capability it is using. This value effectively represents each services utilisation at a given time. In this work we keep the request rate uniform (i.e. we do not alter the service workload values) in order to allow clear comparisons in the evaluation of our solution. The request does vary between data centres however, proportionate to the population of the host country.

The simulation runs over 23 simulated months where the evaluation of data centre/service configurations by the GA takes place each month. In our simulations we compare our proposed approach using the genetic algorithm to the scenario where services remain statically on their allocated data centre. In the static case services are allocated relative to the size of the data centre and remain there throughout the course of the simulation. In Table 1 we show the parameters used for our simulations.

Table 1. Simulation Parameters

<i>Parameter</i>	<i>Value</i>
Population Size	100
Mutation Rate	.7
Crossover Rate	.1
# Generations	60
Free-Cooling Efficiency	+40%
α	0.5

In Figure 6(a) we present the overall quantity of renewable energy used when employing both the genetic algorithm approach and the static approach. As you can see the GA based solution out performs the static solution. In this case the GA utilises, on average, 15.9% more renewable energy than static services which accounts for approximately 1566MWh of electricity. The overall energy usage (of IT equipment) remains constant for both solutions, indicating that the GA did not increase the renewable quantity simply by increasing the total energy utilisation.

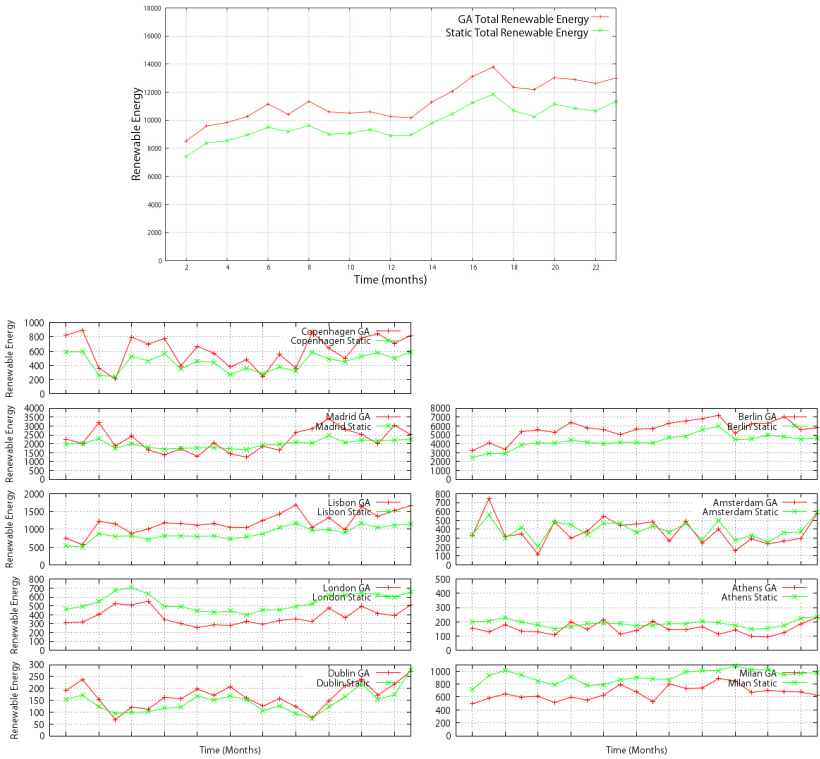


Fig. 6. (a) Total Renewable Energy (b) Renewable Energy per Data Centre

In Figure 6(b) we break this renewable energy usage down according to the individual data-centre usage. As expected, the change in the level of renewable energy used varies from data-centre to data-centre, depending on it’s renewable ratio. Many data centres (e.g. Lisbon, Copenhagen, Berlin) increase their renewable usage while others (e.g. London, Milan, Athens) perform worse, using more fossil-fuel based energy. Increases in renewable energy are as a result of more favourable conditions (i.e. higher renewable ratios) in that country and vice versa. We can see a strong correlation here with the utilisation as indicated in Figure 8, as data centres that increase utilisation also increase renewable enregy utilisation, while those with lower utilisation decrease renewable usage. Correlation can also be seen with the renewable

ratios in Figure 5(b), as those with generally higher renewable ratios gain renewable while those with lower ratios again perform worse.

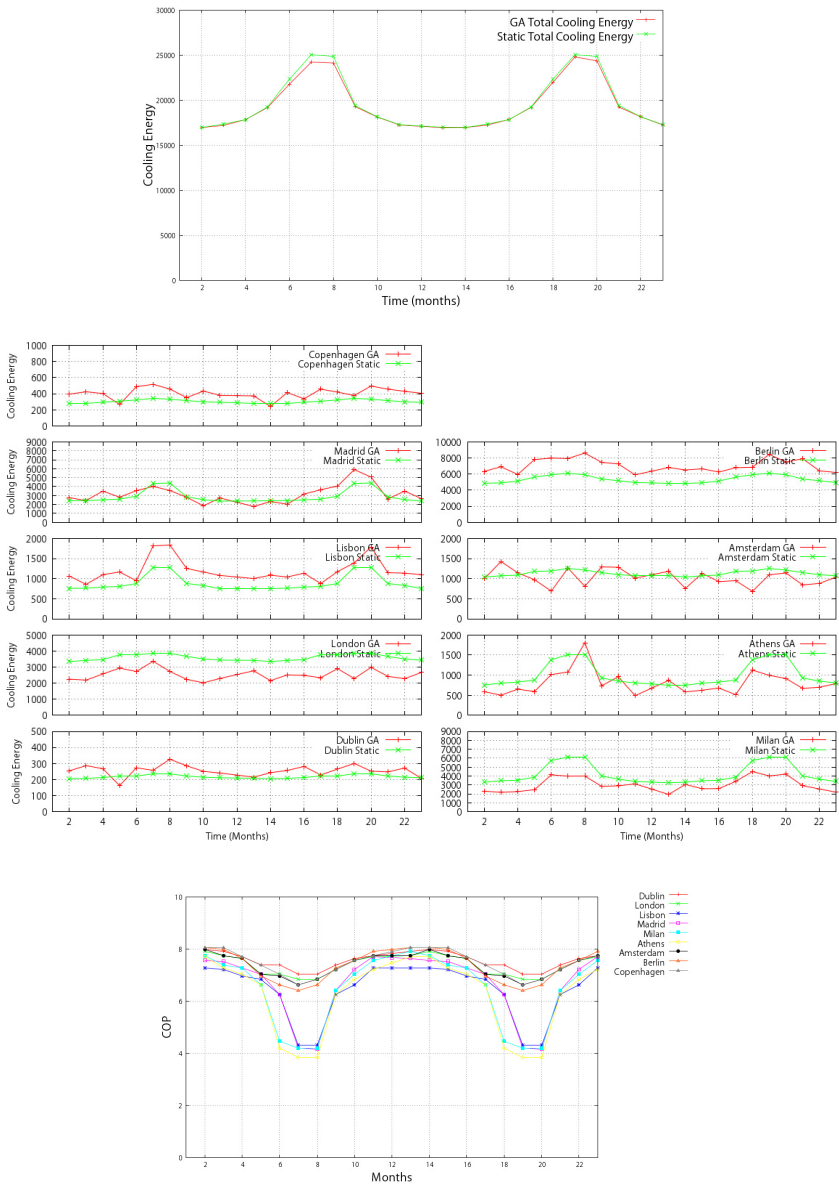


Fig. 7. (a) Total Cooling Energy (b) Cooling Energy per DC (c) COP per DC

In Figure 7(a) we present the total cooling energy used by both the genetic algorithm and static simulations. In the fitness function we aim to minimize the

cooling energy expended, where cooling energy is directly proportionate to the COP of the data centre and the load exerted on the data centre. Savings in cooling energy are made by moving load to countries with lower temperatures and hence better COP values. If there is no significant variance in the temperatures of the countries then there is little opportunity to make significant savings.



Fig. 8. Utilisation per DC

In Figure 7(c) we can see that, during the winter periods the COP values for each data centre are very close and so the savings observed are very small. During the summer periods the COP values begin to diverge and so the GA can find service placements that can provide energy savings. In general however the savings observed for cooling energy are small (at best 3% for months 6-8), and this is due to the geographic proximity of the data-centres. In Figure 7(b) cooling energy values for each individual data centre are shown. Again there is a strong correlation here with the utilisation values in Figure 8, where DCs with higher utilisation will see higher cooling energy. The cooling energy values may appear somewhat counter-intuitive at first, given that the data-centres with the best COPs generally show increased cooling energy values for the GA (e.g. Dublin, Copenhagen, Berlin). However, given that the data centres with the best COP values are targeted for service placement, this will lead to increased utilisation and hence increased cooling energy usage. Since these have the most efficient cooling conditions, the cooling cost is lower than on those data centres with higher COPs for the same load. In other words, by removing load from lower-efficiency data centres and placing it on more efficient data centres we reduce

the cooling energy consumed. There are some exceptions though, such as London and Amsterdam, who generally have good COP values but their cooling values do not necessarily reflect this. However we must consider the effects of the renewable aspect of the fitness function. Dublin, Copenhagen and Berlin also have good or very good renewable energy ratios, which make them more attractive for placement (i.e. higher fitness) while London and Amsterdam have poor or very poor renewable values. This offsets the effect of a positive COP value.

Figure 8 presents the utilisation experienced by each data centre over the course of the simulation. The utilisation values presented here are relative to the overall capacity of the entire data-centre group (i.e. all 9 data centres). As expected we can see that many of the data centres in the GA approach decrease capacity compared to the static while others increase. The data centres that consistently increase (Copenhagen, Lisbon, Dublin, Berlin) can be seen to correspond to those data centres that perform well in terms of renewable energy and also cooling energy. It must be noted that utilisation is also influenced by the capacity of the data centres. For instance Copenhagen is generally the best performer in terms of renewable energy and one of the top performers for cooling, yet Berlin's utilisation increase is significantly larger. This is simply because Berlin is a considerably larger data centre and can handle a much larger utilisation increase. In terms of reduced utilisation, we can see that London and Milan show significant reductions with Athens, Amsterdam and Madrid show varying levels of reduction. For London and Milan, both have very poor renewable energy ratios (specifically London).

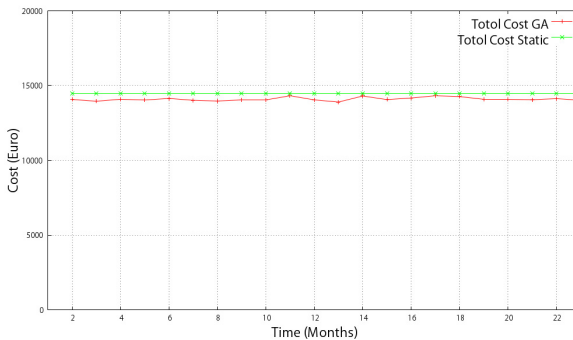


Fig. 9. Total Cost

Finally in Figure 9 we show the cost values incurred in both simulations. Since cost is not part of the fitness function this is presented only to evaluate the cost impact of our solution. Here it can be seen that the cost of our solution is lower than the cost of the static approach. However the cost decrease is very small (approx. 2.5% average), but the aim here is not to considerably reduce the cost, merely to ensure that our proposed approach does not come at a financial burden to the data centre operator. This is important in promoting the proposed solution to data centres operators, as increased costs will negatively impact the renewable energy benefits of the system.

6 Discussion and Future Work

Here we will briefly discuss some issues and notions that we feel warrant highlighting. Firstly, we abstract the energy grid of each country to be a ‘black box’, where we assume each energy source is fed into the grid and its source becomes indiscernible. In other words, the energy produced from renewable sources is not partitioned or reserved for specific usage, but is available in the grid for common usage in direct proportion to the rate at which it was produced. In electrical grids, detailed forecasting determines the required electrical capacity required at any given time. Consumption beyond this capacity requires additional generators to be brought online (typically using dispatchable, fossil fuel based energy sources such as natural gas) to cater for this additional consumption. In this regard, it may be suggested that causing a data centre to consume more energy in fact only utilises more non-renewable energy. However, when looking at the quantity of additional consumption our system places on the data centre, it equates to approximately 2×10^{-5} percent of the country’s electricity production. This minute change we believe would be covered by the grids forecasting model. If the system were to be adopted in a large scale this may present a more significant issue, however, given the disparity in data-centre locations it might still not be directly discernable. That is to say that not all data-centre operators will have data-centres in the same countries, so the load increase will not always affect the same countries. However, in future work we aim to factor more detailed information from the grid, so that processing load only moves to data centres when its energy consumption drops below the forecasted consumption.

Another issue that we discuss here is the additional energy and latency costs that may be incurred by moving quantities of services between data centres. Moving services could cause higher loads on networking equipment along the migration path, hence increasing their energy consumption. Also, moving services further from the source of requests could potentially increase delays times and hence reduce end user Quality of Experience (QoE). In future work we plan to expand our simulations to evaluate these effects on the underlying network infrastructure. However, we feel that these effects could be limited by placing a distance limit on migrating services or indeed integrating a distance metric directly into the move metric objective function itself. In this way we could ensure that we always attempt to minimise the effects of migration on energy and QoS.

7 Conclusion

Due to the increasing popularity of the Internet, the communication systems of the future are predicted to consume large quantities of energy. In particular, the data centres that house various types of Internet services are poised to be the most significant consumer of energy. While improving energy efficiency is one objective of modern society, another key objective is to move towards green, renewable energy sources to reduce our carbon footprint. In this paper we have proposed a green data centre solution that uses a GA-based service placement approach based on targeting

countries with the highest production of renewable energy and the best conditions for cooling. To validate our proposed solution, we carried out some demonstrative simulations by gathering data regarding the renewable energy production and temperature profiles of each country, and implementing a genetic algorithm that aims to maximise the quantity of renewable energy consumed and minimise cooling energy expended. From our simulations we have demonstrated that by employing this technique it is feasible to make significant improvements in the proportion of renewable energy utilised in data centre operation, hence reducing the quantity of fossil fuels burned and ultimately carbon emissions. We also demonstrated that cooling energy can be reduced in circumstances where there is significant variance in the country's temperature profiles. At the same time, we showed that this improved renewable energy utilisation did not come at an increased monetary cost for the operator.

References

1. ENERGY STAR. Report to Congress on Server and Data Center Energy Efficiency Public Law, 109-431. Environmental Protection Agency, Washington, D.C. (2007), http://www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf
2. Bradley, D.J., Harper, R.E., Hunter, S.W.: Workload-based Power Management for Parallel Computer Systems. *IBM Journal of Research and Development* 47, 703–718 (2003)
3. Das, R., Kephart, J.O., Lefurgy, C., Tesauro, G., Levine, D.W., Chan, H.: Autonomic multi-agent management of power and performance in data centers. In: *The Seventh International Conference of Autonomic Agents and Multiagent Systems* (May 2008)
4. Meisner, D., Gold, B.T., Wenisch, T.F.: PowerNap: Eliminating Server Idle power. In: *ACM SPLOS* (2009)
5. Rusu, C., et al.: Energy-Efficient Real-Time Heterogeneous Server Clusters. In: *Proceedings of RTAS* (April 2006)
6. Barbagallo, D., Di Nitto, E., Dubois, D.J., Mirandola, R.: A Bio-Inspired Algorithm for Energy Optimization in a Self-Organizing Data Center. In: Weyns, D., Malek, S., de Lemos, R., Andersson, J. (eds.) *SOAR 2009*. LNCS, vol. 6090, pp. 127–151. Springer, Heidelberg (2010)
7. Moore, J., Chase, J., Ranganathan, P., Sharma, R.: Making scheduling "cool": temperature-aware workload placement in data centers. In: *Proceedings of the USENIX Annual Technical Conference 2005 on USENIX Annual Technical Conference*, Anaheim, CA, April 10-15, p. 5 (2005)
8. Patel, C.D., Sharma, R.K., Bash, C.E., Graupner, S.: Energy Aware Grid: Global Workload Placement based on Energy Efficiency. In: *IMECE 2003-41443*, 2003 International Mechanical Engineering Congress and Exposition, Washington, DC (2003)
9. Bash, C., Forman, G.: Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center. In: *Proceedings of the USENIX Annual Technical Conference*, pp. 363–368 (2007)
10. Wang, L., et al.: Towards Thermal Aware Workload Scheduling in a Data Center. In: *Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms and Networks (IS-PAN 2009)*, Kao-Hsiung, Taiwan, December 14-16 (2009)

11. Sharma, R.K., Bash, C.L., Patel, C.D., Friedrich, R.J., Chase, J.S.: Balance of Power: Dynamic Thermal Management for Internet Data Centers. *IEEE Internet Computing* 9(1), 42–49 (2005)
12. Qureshi, A., Weber, R., et al.: Cutting the Electric Bill for Internet-Scale Systems. In: *SIGCOMM 2009*, Barcelona, Spain, August 17–21 (2009)
13. Moran, M.J., Shapiro, H.N.: *Fundamentals of Engineering Thermodynamics*. Wiley, New York (1995)
14. Oakridge National Laboratory (ORNL) Heat Pump Model, <http://www.ornl.gov/~wlj/hpdm/MarkVII.shtml>
15. International Energy Agency, <http://www.iea.org>
16. European Climate Assessment & Dataset (ECA&D), <http://eca.knmi.nl/>
17. European Commission: Eurostat, <http://epp.eurostat.ec.europa.eu/>

Safer Driving with Gaze Guidance

Laura Pomarjanschil^{1,2}, Michael Dorr^{1,3}, Christoph Rasche⁴,
and Erhardt Barth¹

¹ Institute for Neuro- and Bioinformatics, University of Lübeck,
Ratzeburger Allee 160, 23538 Lübeck, Germany
{laura,barth,dorr}@inb.uni-luebeck.de

² Graduate School for Computing in Medicine and Life Sciences, University of Lübeck

³ Schepens Eye Research Institute, Harvard Medical School

⁴ Image Processing and Analysis Laboratory, University “Politehnica” Bucharest
rasche15@gmail.com

Abstract. Our study explores eye movements and driving behaviour in safety-critical situations. We collected eye movements from subjects instructed to drive predetermined routes in a driving simulator. While driving, the subjects performed various cognitive tasks designed to divert their attention away from the road. The subjects were divided in two groups, a control and a “gaze-contingent cue” (GCC) group. For the latter, potentially dangerous simulator events, such as a pedestrian suddenly crossing the street, were highlighted with temporally transient GCCs, which were triggered if the subject did not look at the event. Preliminary results, with 11 GCC subjects and 9 controls, show a significantly diminished accident rate for the group that drove with gaze guidance.

Keywords: eye movements, driving, attention, gaze guidance, driver assistance.

1 Introduction

Driving is a visually guided behaviour and driving performance often suffers from limited attentional resources.

The analysis of eye movements can reveal where a subject is deploying attention and, in consequence, which aspects of a scene are being processed. Eye movements are influenced by the structure and the saliency of the viewed scene as well as by the task performed by the observer (Yarbus [1967], Hayhoe and Ballard [2005], Henderson [2003], Torralba et al. [2006]). Everyday tasks involve using complex eye movement patterns (Land and Hayhoe [2001], Rayner et al. [1998], Land and Lee [1994]). Nevertheless, eye movements can be reliably predicted by low-level features of the visual input (Vig et al. [2009]).

When driving a car, peripheral cues intertwine with foveal scanning to provide the driver with enough information (Mourant and Rockwell [1970], Land and Lee [1994]). The increase in scene complexity or the presence of distracting stimuli have immediate consequences both on the gaze pattern and

on the driving performance (Crundall et al. [1998], Chapman and Underwood [1998]).

Along with excessive speed and reckless behaviour, driver inattention and misperception errors are listed as major contributing factors for fatal traffic accidents (Mosedale et al. [2006], NHTSA [2009]).

One way to help inattentive drivers is to build enhanced vision systems that can direct their eye movements towards regions of increased risk. By using gaze-contingent interactive displays to display the visual information with increased salience in desired regions, a gaze guidance effect can be obtained (Barth et al. [2006]).

In the following we will show that gaze guidance techniques applied in a driving simulator improve the eye movement pattern of the drivers and considerably reduce accident rates.

2 Methods

Our experiments took place in a PC-based driving simulator with integrated eye tracking and with the ability to display gaze-contingent events.

The simulation environment reproduced a virtual city populated with cars and pedestrians involved in realistic traffic scenarios. Subjects drove predetermined routes inside the simulated city, while performing additional cognitive tasks. To guide the driver along the route, transparent directional arrows were overlaid on the road at intersections.

For the experiment, we designed three distinct routes, each stretching on average over a distance of 900 m. Amidst benign traffic scenarios, each route had approximately four critical sections. They consisted of pedestrians crossing or coming close to the street from angles of low visibility, creating the risk of a crash with the egocar (the subject-controlled car).



Fig. 1. Simulator scene. Because the driver is looking away, the pedestrian beginning to cross the street is highlighted with a gaze-contingent cue

The cognitive tasks were designed to act as a distractor. In the first task, the observers were instructed to count the number of floors on all buildings along the route, and to remember the approximate location of the one with the highest count. In the second task, the observers had to search for an item (e.g. a copy shop) on the route, and to report how many occurrences of it they observed, and where they were located. In the last trials they were told to drive freely, but they were verbally distracted by conversation with the experimenter.

An experiment consisted of nine trials resulting from the combination of each route with each task. Each subject drove each route three times, every time with a different cognitive load. All the experiments began with a short training route, in which the drivers were allowed to drive freely in a remote part of the city. Only when the observer was able to drive safely on the simulated road, the actual trials would begin. On average, an experiment lasted approximately thirty minutes.

For one subject group, the potentially critical events were highlighted with gaze-contingent cues (GCC). The chosen cue material was a simple, opaque, red overlay shaped like four rays converging on the pedestrian (see Fig. 1). Several shapes and colours were tested in order to make the cue as unobtrusive as possible, while leaving it salient enough to capture the driver's eye movements. The cue was only triggered if the observer did not look at the critical event, and was turned off immediately when the subject looked at it. The control group was not exposed to any GCCs.

We recorded data from 20 volunteers with normal or corrected to normal vision (7 female and 13 male, with ages between 20 - 45 years). All had a driving licence and at least one year driving experience. Their computer gaming experience was variable. 11 subjects were part of the gaze guidance aided group, with the remaining 9 being controls.

The setup consisted of two computer workstations, one running and displaying the simulation, the other acting as a server controlling the events and the eye tracking device. The server and the simulator were connected through a direct ethernet interface.

The participants were seated 70 cm away from the 22" display. They controlled the simulator with an off-the-shelf pedals and steering wheel system. Eye movements were recorded using a SensoMotoric Instruments RED250 remote eye tracker running at a sampling frequency of 250 Hz. The eye tracker device was connected via USB to the server. The display had a spatial resolution of 1680x1050 pixels and was viewed under an angle of about 38x24 degrees. Before each trial, a 9 point calibration was performed.

From the over 280 minutes of gaze data recordings, more than 54,000 saccades were extracted using a velocity-based algorithm (Böhme et al. [2006]). The simulator also recorded driving parameters such as speed, pedal position and steering wheel inclination at a frequency of 60 Hz.

3 Results

3.1 Influence of Cognitive Load

We first analyzed how the subjects' gaze was distributed across the screen. The three heat maps (Fig. 2) show the statistical distribution of the eye movement data across the display for each condition, throughout the experiment. The distribution was computed for all subjects ($N = 20$).

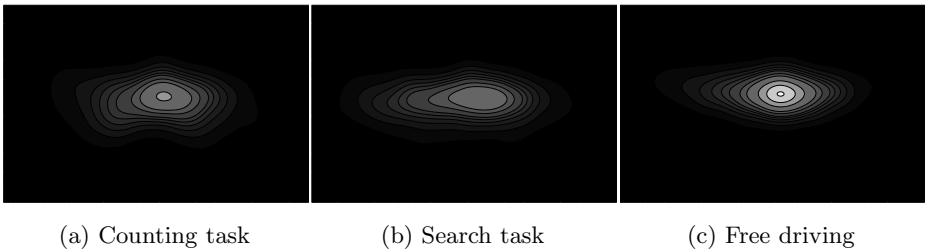


Fig. 2. Gaze distribution across the display. In the free driving task, the subjects fixated more in the centre of the screen; counting the number of floors led to a more vertically stretched gaze distribution.

Next, we examined the degree in which the saccade amplitudes varied with the cognitive load (Fig. 3). We plotted the empirical cumulative distribution function of the saccade amplitudes for every task. For each task, the saccade amplitudes follow significantly different distributions ($p < 0.001$, Kolmogorov-Smirnov test).

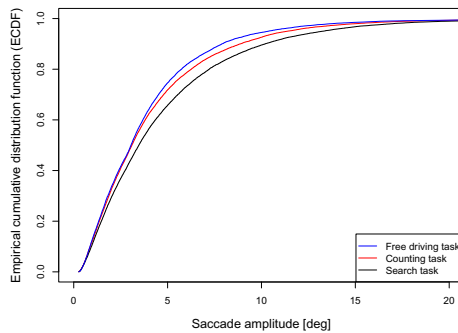


Fig. 3. Saccade amplitude distribution. In the free driving task (uppermost curve), the subjects performed more short saccades, while in the search task (bottom curve) there were more large, exploratory, saccades.

3.2 Efficiency of Gaze Guidance

For the analysis of the GCC influence on driving performance, no distinction was made between the cognitive tasks. The data were pooled for each observer group over all three conditions.

Reaction Times. We examined the reaction time between the triggering of a critical event and the first gaze hit on the pedestrian of interest (Fig. 4). The latency between the onset of a critical event and the first gaze hit tends to be smaller for the GCC subjects.

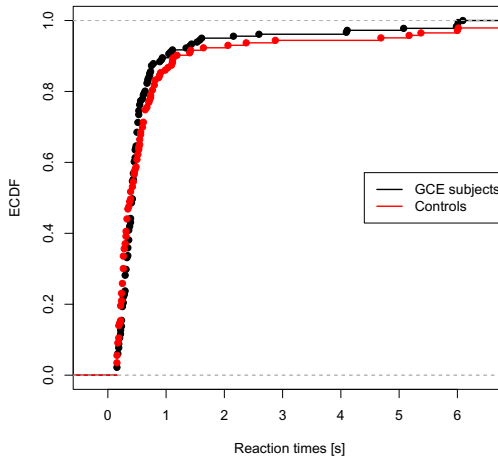


Fig. 4. ECDF of the reaction time to a critical scenario pedestrian (mean/s.d. 0.27/0.71 s for GCC subjects, 0.39/1.1 s controls)

Provoked Accidents. To evaluate the subjects' driving performance, we looked at the number of accidents caused in the experiment. An accident was considered a crash of the egocar against one of the critical scenario pedestrians.

Crashes were detected using a distance based-algorithm. An "encounter" was considered to be a crash if the distance between the centre of the egocar and the pedestrian was smaller than 1 m.

We found that the accident rate is substantially decreased for the gaze guidance subjects (0.021), being less than a third of that of the controls (0.067). This reduction is highly significant (95% confidence interval: [0.0393, 0.1004]).

4 Conclusions

We have performed experiments in a gaze-contingent driving simulator to study the impact of gaze guidance on driving performance. We found that subjects

exhibited different gaze behaviour under different cognitive tasks. When safety-critical events were highlighted with briefly flashed, gaze-contingent cues, drivers attended to these events more quickly. More importantly, such gaze guidance led to a significantly reduced number of accidents. Subjects reported that they were not distracted by the cues; a part of the cues went unnoticed.

Acknowledgement. Our research has received funding from the European Commission within the GazeCom project (IST-C-033816) of the FP6. All views herein are those of the authors alone; the European Commission is not liable for any use made of the information. Further support has been received from the Graduate School for Computing in Medicine and Life Sciences funded by Germany's Excellence Initiative [DFG GSC 235/1].

References

- Barth, E., Dorr, M., Böhme, M., Gegenfurtner, K.R., Martinetz, T.: Guiding Eye Movements for Better Communication and Augmented Vision. In: André, E., Dybkjær, L., Minker, W., Neumann, H., Weber, M. (eds.) PIT 2006. LNCS (LNAI), vol. 4021, pp. 1–8. Springer, Heidelberg (2006)
- Böhme, M., Dorr, M., Krause, C., Martinetz, T., Barth, E.: Eye movement predictions on natural videos. *Neurocomputing* 69(16–18), 1996–2004 (2006)
- Chapman, P.R., Underwood, G.: Visual search of driving situations: Danger and experience. *Perception* 27, 951–964 (1998)
- Crundall, D.E., Underwood, G., Chapman, P.R.: How much do novice drivers see? the effects of demand on visual search strategies in novice and experienced drivers. In: Underwood, G. (ed.) *Eye Guidance in Reading and Scene Perception*, pp. 395–417. Elsevier Science Ltd. (1998)
- Hayhoe, M., Ballard, D.: Eye movements in natural behavior. *Trends in Cognitive Sciences* 9(4), 188–194 (2005)
- Henderson, J.M.: Human gaze control during real-world scene perception. *Trends in Cognitive Sciences* 7(11), 498–504 (2003)
- Land, M.F., Hayhoe, M.: In what ways do eye movements contribute to everyday activities? *Vision Research* 41, 3559–3565 (2001)
- Land, M.F., Lee, D.N.: Where we look when we steer. *Nature* 369, 742–744 (1994)
- Mosedale, J., Purdy, A., Clarkson, E.: Contributory factors to road accidents. Technical report, Department for Transport, UK (2006)
- Mourant, R.R., Rockwell, T.H.: Mapping eye-movement patterns to the visual scene in driving: An exploratory study. *Human Factors* 12(1), 81–87 (1970)
- NHTSA. An examination of driver distractions as recorded in NHTSA database (2009)
- Rayner, K., Reichle, E.D., Pollatsek, A.: Eye movement control in reading: An overview and model. In: Underwood, G. (ed.) *Eye Guidance in Reading and Scene Perception*, pp. 243–68. Elsevier Science Ltd. (1998)
- Torralba, A., Oliva, A., Castelhano, M.S., Henderson, J.M.: Contextual guidance of eye movements and attention in real-world scenes: The role of global features on object search. *Psychological Review* 113(4), 766–786 (2006)
- Vig, E., Dorr, M., Barth, E.: Efficient visual coding and the predictability of eye movements on natural movies. *Spatial Vision* 22(5), 397–408 (2009)
- Yarbus, A.L.: *Eye Movements and Vision*. Plenum Press, New York (1967)

Infrastructure Optimization of Flight-Formation Inspired Self-organization for Address Configuration in Sensor Networks

Rui Teng, Bing Zhang, and Jian-Qin Liu

National Institute of Information and Communications Technology, Japan
{teng, zhang, liu}@nict.go.jp

Abstract. Self-organized sensor networks are expected to automatically configure sensor nodes into networks. Node addresses are essential for network communications, and the address size has a substantial impact on the amount of energy consumption during the long-term report of small sensing data. This paper introduces a Scalable and Dynamic Infrastructure based Configuration scheme (SDIC), which addresses the problem of how to organize a large number of sensor nodes to configure small addresses with scalability and energy efficiency. The special features that make our approach unique are the exploitation of an optimized autoconfiguration infrastructure that is inspired by the flight formation of migration birds. SDIC enables sequential address assignment in a deterministic manner without address conflicts. SDIC utilizes mechanisms of the optimized server term control to achieve the scalability of infrastructure and configuration operation. The evaluation results of SDIC show that it achieves small-size address with few conflicts and with energy efficiency.

Keywords: Self-organization, bird flight formation, address autoconfiguration, network infrastructure.

1 Introduction

Two features distinguish the sensor networks with the conventional Internet. One is the self-organization of sensor nodes. The other is energy efficient operations in the network. Small sensor nodes are deployed in a region of interest to observe physical phenomena such as temperature, humidity, or solar radiation. Battery-powered sensor nodes are expected to self-organize into networks and provide long-term operations of reporting data and keeping network connectivity [1],[2].

The IP addresses have a high energy cost for sensor nodes. It is possible for a sensor node to adopt a local unique address as its network-layer address, because delivery and process of sensing data in a sensor network are generally carried out inside a task area, such as a monitoring region [3]. In other words, each node in a distributed system can have a node address that is unique with respect to the connectivity of the network. A network-wide local unique address has a potentially

short address size. For instance, a typical sensor network of a few hundred nodes could use as few as 10 bits if the addresses were locally unique. Although sensor attributes, such as temperature, humidity and location, can be used to partially identify the sensor destinations in an information query, attribute addresses can not distinguish individual sensor node and the sources of sensing data.

To configure addresses for a large number of nodes, it is difficult to manually allocate the addresses to a number of nodes in sensor networks, and a reasonable option is to use autoconfiguration of sensor addresses. The address autoconfiguration is a self-organization procedure. It selects, allocates, and assigns a unique network address to each unconfigured node. However, most of conventional autoconfiguration schemes deal with the IP address configuration, generally within a large address space [4],[5],[6],[7],[8],[10],[12],[13]. Although a few schemes of Non-IP address autoconfiguration have also been adopted to configure addresses, these configuration schemes either assume using a pre-defined large address space for small scale nodes, or configuring the addresses that are not unique in the network [9], [11]. To configure small-size addresses of sensors, it requires that addresses being assigned within a potential smallest address space for a number of sensor nodes. This leads to a high probability of address collisions.

This paper introduces a Scalable and Dynamic Infrastructure based Configuration scheme (SDIC), which self-organizes sensor nodes to efficiently configure addresses in a small address space. SDIC utilizes the model of the birds' flight formation, in which all birds shares the responsibility of being lead bird by rotation to keep the team consistency and have energy saving [15]. Like the role rotation among birds in flight formation, SDIC constructs a controllable network infrastructure by utilizing dynamic roles of address server and address client among sensor nodes. The address servers manage the use of address space in a deterministic manner and assign address in a sequential way. Each address server only maintains small amount of state information (such as the address sequence) during the configuration. SDIC utilizes the optimized control of the server term that manages the role rotation of sensors to avoid overuse of energy at a certain node. As a result, SDIC has a temporary centralized infrastructure, which, however, operates in distributed manner for the long-run.

The rest of the paper is organized as follows: Section 2 presents related work. Section 3 describes the flight formation inspired infrastructure self-organization in SDIC scheme. Section 4 describes the basic SDIC autconfiguration. Section 5 describes the optimized control of address-server's term. Section 5 presents the simulation results of the proposed SDIC protocols. Section 6 concludes.

2 Background and Related Work

Conventional approaches of address autoconfiguration can be categorized as infrastructure-based or infrastructureless-based according to the utilization of predefined network infrastructure. In traditional IP networks, the address configuration is infrastructure-based due to the use of address servers. Dynamic Host Configuration Protocol (DHCP) operates based on centralized server-client architecture, is an effective address assignment approach in the IP Internet [8].

In mobile ad hoc networks, address configurations are infrastructureless: without the dependence of predefined address server. Autoconfiguration of node addresses in ad hoc networks has been extensively studied in recent years. There are mainly two types of address autoconfiguration in ad hoc networks, stateful autoconfiguration and stateless autoconfiguration. Stateful address autoconfiguration adopts address allocation table to maintain uniqueness of addresses, such as MANETconf [4], prophet [13]. Stateless autoconfiguration utilizes the on-demand duplicate address detection, such as WDAD, IETF zero-configuration [5],[14]. Many of them are based on Duplicated Address Detection (DAD) approach, which operates based on full distributed self-organization architecture [6]. The existing address autoconfiguration approaches are generally utilized for IP address configuration (for the ease of installation) in a relative small network. These approaches have a general target of dealing with the invalid address configuration caused by the network merge and partitions [4],[5],[7]. But such a target is not the ultimate goal in sensor networks, the topologies of which are generally static and energy efficiency is of more importance than the mobility issue.

A few approaches of address autoconfiguration have been proposed in sensor networks. Random addressing lets each sensor node select an address randomly in a defined address space. Random addressing is a simple way for each node to get an address. It is an infrastructure-less approach and operates in a full distributed organized manner. But the uniqueness of the address can not be guaranteed, and the incidence of address collisions will be high when the address size is short [9]. Treecast addressing has also discussed node address issues in sensor networks [10]. It is an infrastructure-based approach, however, it utilizes a static centric organized architecture assuming there are special sink servers already deployed in sensor networks. Furthermore, Treecast addressing does not consider the energy efficiency aspect for node addressing in sensor network, such as configuring small-size addresses. A MAC addressing scheme has been proposed in [11], which achieves small MAC address by adopting address reuse in the networks. However, the proposed approach cannot be utilized for configuring unique network-layer addresses. [16] has proposed energy-efficient MAC and network-layer addressing schemes in sensor networks. However, it have not addressed and analyzed the efficient organization of sensor nodes and the infrastructure optimization for address configuration. Spatial-time based address configuration was proposed in [17], in which the location of nodes and time of configuration are used to represent the node address. Due to the large configuration space of location and time, there are few address conflicts. However, the spatial-time configuration has not considered the problem of large address size caused by the using location and time in the address.

3 Flight-Formation Inspired Network Infrastructure Self-organization

3.1 Network Model

A sensor node consists of sensing, computing, and communication capabilities. The node address of a sensor is configurable in that the address and its size can be set up

in an on-demand manner rather than being predefined before the node joins the network. Unlike the conventional Internet nodes, the sensor node generally has a power constraint.

A self-organized sensor network does not depend on the predefined network infrastructure such as network servers. In contrast, sensor nodes cooperatively perform the role of network infrastructure, automatically configuring sensor nodes into a network. A typical self-organized sensor network operates as follows. Sensor nodes are firstly deployed in a field with a certain density that enables nodes to be connected, and sufficiently cover the sensing area. Then, the nodes, which operate in an unattended mode, self-organize into a network. Node addresses are essential for nodes to cooperatively form the network, and communicate with each other. After the configuration, sensor nodes start to collect and report the information about their surroundings. Each sensor node connects to others by either one hop connection or multi-hop connections.

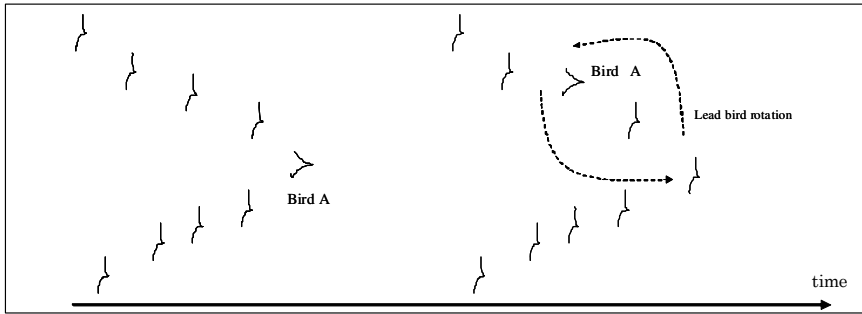
The task of address autoconfiguration is to assign each node a unique address in an automatic manner. To configure small-size unique addresses, it requires configuring addresses within potential smallest address space for a number of sensor nodes. However, it is difficult to utilize the individual operation of node configuration such as distributed random address selection and assignment, which is widely used in conventional autoconfiguration schemes. This is because there will be many collisions to configure a number of addresses in a small address space. The cooperation of self-organized sensor nodes is necessary to efficiently manage the address space.

3.2 Flight Formation Model

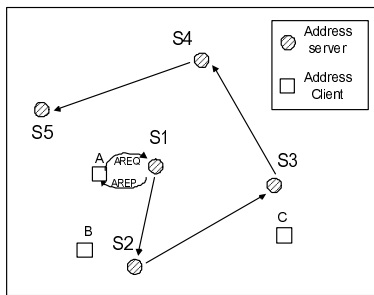
The self-organization for sensor nodes to cooperatively configure addresses is inspired by the formation flight model of birds. Many species of large birds, such as goose, duck, and other migration birds, fly together in formation. A group of such birds will typically fly in a large “V” type [15]. In the “V” type formation, generally there is one bird in the lead position at the vertex of “V”, and others trail behind in two lines, as shown in Fig.1 (a).

Researchers who have studied formation flight believe that birds fly in this way for two reasons. The first reason is that the shape of formation reduces the drag force that each bird experiences compared to it flying alone [18]. The second reason that may explain the flight is that the “V” formation allows the birds to communicate more easily with good visual contact of each other.

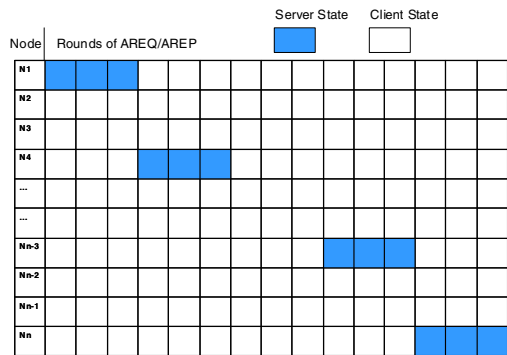
The formation flight keeps the consistency of speed and direction of birds as a group. Each bird takes the advantages of the power of the formation. But the lead bird in the vertex of “V” consumes more energy than others. When the lead bird gets tired, it will rotate back in the wing, and another bird takes the lead position, as shown in Fig.1 (a). This results in the dynamic formation of “V” to keep the energy balance among birds [15].



(a) Flight formation of birds



(b) Address configuration with server rotation



(c) Server term control

Fig. 1. Overview of SDIC address configuration. The bird flight formation model describes that all birds share the responsibility of being lead bird by rotation to keep the team consistency and to have energy balance. This brings insights into the self-organization of sensor networks for address autoconfiguration.

3.3 Insights from Flight Formation to SDIC

The bird flight formation model brings important insights into the self-organization of sensor networks for address autoconfiguration. First, flying birds as a group represent a well established team with shared goal (such as flight migration). With the flight formation, the team operates in a consistent manner with regard to speed and direction. Further, the formation result in the energy saving of birds during the flight. Second, to maintain the flight formation, each bird in the flight has its own roles and responsibilities for the entire group. The lead bird takes most of the responsibility during the flight, but other birds also become leaders when needed.

In SDIC configuration, a multihop server-client formation is self-constructed by sensor nodes. The server node manages the use of address space and assigns address to client sensors in a deterministic manner and keeps the consistency of address usage. This formation allows sensor nodes to configure address with little address conflicts, keeps the address assignment within the smallest necessary address space.

The sensors take advantage of the server-client formation with a high successful configuration rate, and save the configuration energy by using easy communication of unicast between server and clients.

In SDIC, like the flight formation, each sensor node has a role of becoming address client or address server. A node in the server state acts as an address server, and then takes the responsibility of address assignment, while a node in the address client state might issues the configuration request or relay packets for other sensors.

Like the lead bird in flight formation, the server node consumes more energy than other nodes. Thus it will rotate back to client after a certain term of server operation. In this paper, the optimized server term is studied by both of analysis and simulation. The server rotation avoids the overuse of energy at one node, balancing the energy consumption of the network. Further, server rotation provides the configuration scalability in the network.

4 Fundamental SDIC Address Autoconfiguration

SDIC configures addresses in a potential smallest address space by assigning address sequentially from low to high without the overuse of address space. The sequential assignment of node addresses requires a deterministic operation rather than an opportunistic operation so as to maintain the consistency of address configuration in self-organized sensor networks.

To realize these functions, SDIC scheme self-organizes sensor nodes into a *dynamic* server-client based network infrastructure. Each address server is automatically elected among sensor nodes, and has a limited server term, after which, another sensor node will be selected as a new server. As shown in Fig.1 (b), the address server performs the address assignment in its term to the address clients, which are nodes that have not had their address configured yet. In the configuration, an address client (e.g. node A) that asks for an address sends an *address-request (AREQ)* message to the server (e.g. S1). In response to the *AREQ* message, the address server selects an unused address in the address space and sends an *address-reply (AREP)* message, which includes the assigned address, to the address client.

At the beginning of configuration, when a node needs an address to be configured and does not have any information about the address server, the node searches for the address server by broadcasting a server-query message at a random time. If an address server already exists, the server should inform of its presence to the node that is searching for a server. If a searching node receives no reply within a certain period of time, it turns into a server initiation state, assigns itself as an address server, and then announces its status to other nodes by broadcasting a *server-announcement* message. Each sensor node that receives the server-announce message records the server address and also the source route to the address server for future connections. Thus, the route for an address client to deliver an *AREQ* message to an address server is obtained from the source route of a *server-announcement* message. And the route for the server to deliver an *AREP* message to an address client is obtained from the source route of the *AREQ* message from the address client.

SDIC utilizes *the server term control* to obtain the infrastructure scalability. The *server term control* computes and sets an optimized term for each address server. It controls the server rotation (as shown in Fig.1 (c)) among sensors to achieve both deterministic and distributed operation, leading to the server robustness by avoiding the overuse of server energy. A node in a server state perform the server role in a server term that is measured by a certain number of AREQ/AREP rounds in the configuration.

With the central control of address servers, node addresses can be assigned in a sequential manner to sensor nodes. This results in the smallest necessary address space and few address conflicts during the configuration procedure. That is, the address server assigns each address in a sequential increasing order, beginning at 0. A sequence of addresses is maintained at each server, and the address sequence is delivered to a new server from the server that finishes its server term.

5 Optimized Address-Server Term Control

Note SDIC sets each address server a server term. As shown in Fig.2, after an address server (such as S_n in the figure) has assigned addresses n times, it selects a new server from the clients that obtained its address in its server term. In this paper, the last client that is assigned with an address from current address server will be the candidate of new address server (e.g. S_{n+1} in the figure). After a new server is selected, the address sequence is delivered from the current server to the new server so as to keep the consistence of address configuration. A simple procedure of server rotation can just includes the new server assignment by delivering the address sequence message to the next server.

There should be a certain condition with regards to the server numbers, by adjusting which the energy consumption at the servers could be optimized. We calculate the server number that results in the minimum energy consumption at a server since the server generally consumes the largest part of energy.

At first we calculate the configuration cost in a centric address configuration operation without server rotation. We define following parameters.

N	The total number of nodes in the network
K	The number of address servers during the configuration
E_S	Energy consumption at an address server
E_{AC}	Energy consumption at an address client
E_{SI}	Energy consumption of address server initiation
E_{SC}	Energy consumption of the address server configuration
E_{SR}	Energy consumption of the address server rotations
E_{TX}	The energy consumption of transmitting a packet
E_{RX}	The energy consumption of receiving a packet

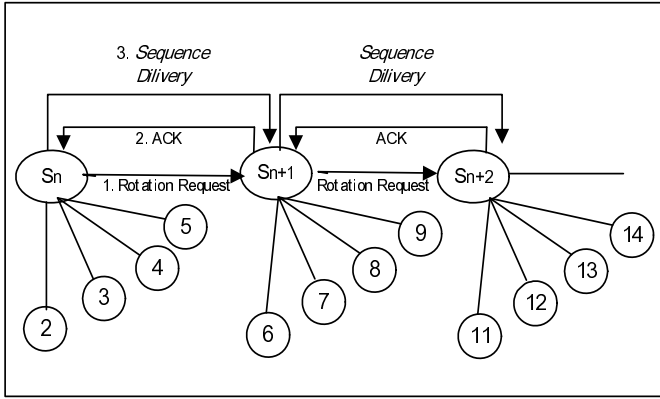


Fig. 2. Server term control. After an address server finishes its term, a new server will be selected to assign addresses to sensors.

The energy consumption at a server, $E_s[i]$, includes three parts: the energy consumption E_{SI} in the server initiation, E_{SC} in the address configuration, and E_{SR} in the server rotation. It can be computed as

$$\begin{aligned}
 E_s[1] &= E_{SI} + E_{SC} + E_{SR} \\
 &= E_{TX} \times 2 + (E_{RX} + E_{TX}) \times \left[\frac{N-1}{K} \right] \\
 &+ (E_{RX} + E_{TX}) \times (K-1)
 \end{aligned} \tag{1}$$

$$\begin{aligned}
 E_s[i] &= E_{SI} + E_{SC} + E_{SR} \\
 &= E_{TX} \times 1 + (E_{RX} + E_{TX}) \times \left[\frac{N-1}{K} \right] \\
 &+ ((E_{RX} + E_{TX}) \times (K-1) + E_{RX})
 \end{aligned} \tag{2}$$

where k is the number of servers during the configuration, each sever assigns (N/K) addresses. In this paper, we adopt the simple procedure of server rotation which just includes the new server assignment by delivering the address sequence message to the next server, and requires the new server to initiate an announcement broadcast. The server announcement lets other node the new server's ID and route so as to avoid much overhead of broadcasting for address clients to find server. The minimum energy consumption at a server can be expressed as follows.

$$\begin{aligned} \min(E_S [i]) &= \min(E_{SI} + E_{SC} + E_{SR}) && i=1; \quad (3) \\ &= \min(E_{TX} \times 2 + (E_{RX} + E_{TX}) \times [\frac{N-1}{K}] \\ &+ (E_{RX} + E_{TX}) \times (K-1)) \end{aligned}$$

$$\begin{aligned} \min(E_S [i]) &= \min(E_{SI} + E_{SC} + E_{SR}) && i>1; \quad (4) \\ &= \min(E_{TX} \times 1 + (E_{RX} + E_{TX}) \times [\frac{N-1}{K}] \\ &+ ((E_{RX} + E_{TX}) \times (K-1)) + E_{RX}) \end{aligned}$$

To derive the optimal server number K that yields the minimum $E_S[i]$, we calculate the derivative of $E_S[i]$ with respect to K as follows.

$$\begin{aligned} \frac{d}{dk} E_S [i] &= \frac{d}{dk} (E_{SI} + E_{SC} + E_{SR}) && (5) \\ &= (E_{RX} + E_{TX}) \times (-\frac{N-1}{K^2}) + (E_{RX} + E_{TX}) \end{aligned}$$

When $K = [\sqrt{N-1}]$, $\frac{d}{dk} E_S [i] = 0$ and the $E_S[i]$ has the smallest value. The corresponding server term is $[\frac{N}{K}]$. Thus, substituting $K = [\sqrt{N-1}]$ to equations (3) and (4) gives the minimum energy consumption at a server.

$$\begin{aligned} \min(E_S [i]) &= E_{TX} \times 2 + (E_{RX} + E_{TX}) \times [\sqrt{N-1}] + && i=1; \quad (6) \\ &(E_{RX} + E_{TX}) \times ([\sqrt{N-1}] - 1) \end{aligned}$$

$$\begin{aligned} \min(E_S [i]) &= E_{TX} \times 1 + (E_{RX} + E_{TX}) \times [\sqrt{N-1}] + && i>1; \quad (7) \\ &(E_{RX} + E_{TX}) \times ([\sqrt{N-1}] - 1) + E_{RX} \end{aligned}$$

As for the address clients, the corresponding energy cost is as follows.

$$\begin{aligned} \min(E_{AC}) &= 2 \times (E_{RX} + E_{TX}) + && (8) \\ &(E_{RX} + E_{TX}) \times ([\sqrt{N-1}] - 1) \end{aligned}$$

From the above analysis, we find that there is an optimized server number that gives rise to the lowest energy consumption at servers, leading to both small and balanced energy cost among sensor nodes.

6 Simulation Evaluation

6.1 Simulation Setup

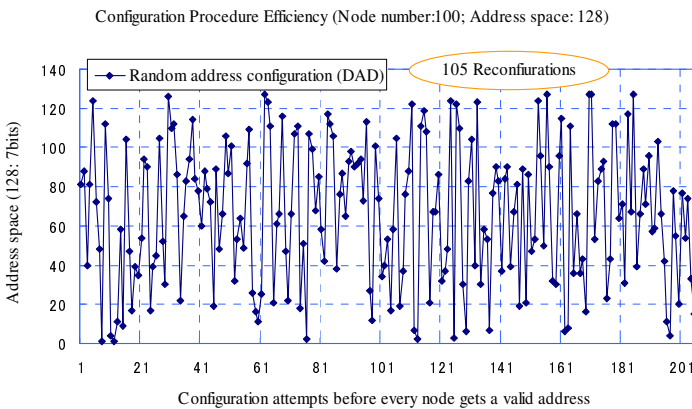
Unless otherwise stated, the simulation is setup as follows. The number of sensor nodes varied from 20 to 256. Each sensor is equipped with a radio module that has a transmission range of 10m. The sensor nodes are distributed over the rectangle fields: field of 100m×20m for 20 nodes, 100m×50m for 50 nodes, 100m×100m for 100

nodes, 100m×150m for 150 nodes, 100m×200m for 200 nodes, and 100m×250m for 250 nodes. Each node boots at a random time within 0-300seconds. Each node has a 32 bits unique temporary address, the difference of temporary address, temporary configured addresses, and configured addresses are identified by the packet type field in the packet header. As for SDIC used in the simulation, each server rotates its role to another node after it has assigned addresses 5 times. That is, a default number of addresses assignment in a round service of one address server is 5. The configuration message is assumed with the same packet size. The proposed SDIC scheme is evaluated and compared with a random address configuration (DAD) approach, because DAD is a widely used approach for address autoconfiguration in self-organized networks[4], [5], [6]. In DAD, each node configures address individually by randomly selecting a candidate address for configuration, and then detects its validity in the network. If the selected address is duplicated with other nodes, the node should configure again by selecting another candidate address.

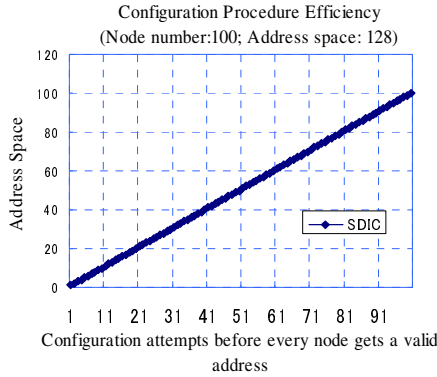
6.2 Numerical Results

A. Configuration Procedure Efficiency

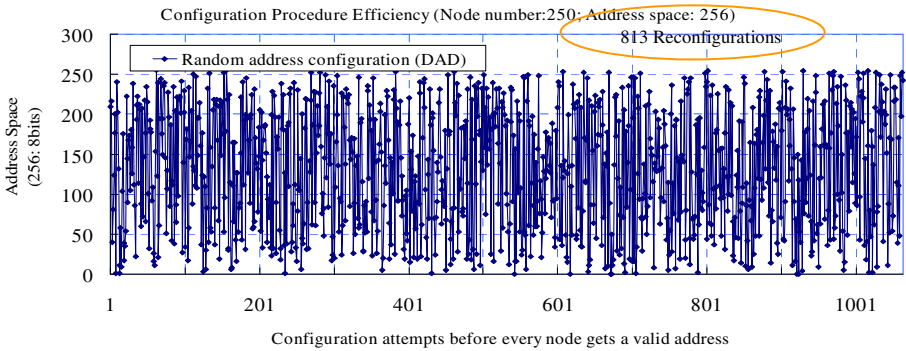
Fig.3. shows the configuration procedure efficiency that is measured by the total number of configuration attempts during the configuration. Here a configuration attempt refers to a round of address request and reply. As shown in the simulation results in Fig.3. (a) and (b), to configure addresses for 100 nodes in an address space of 128, random address configuration (DAD) requires 205% configuration attempts of that of SDIC. And to configure addresses for 250 nodes in an address space of 256, DAD requires about 425.2% configuration attempts of that of SDIC, as shown in Fig.3. (c) and (d). DAD requires much more attempts and has many reconfigurations to have each sensor node a valid address. This is because that the probability of address conflicts is high for configuring a number of addresses in a small address space by the random address selection. In contrast, SDIC has a high efficiency for each sensor node to successfully get a valid address in each configuration attempt. This is because that SDIC utilizes a temporarily centralized and deterministic operation of address servers, leading to few address conflicts.



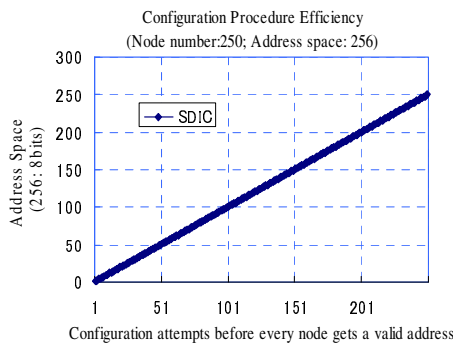
(a) Performance of DAD approach for configuring 100 nodes in address space of 128



(b) Performance of SDIC approach for configuring 100 nodes in address space of 128



(c) DAD approach for configuring 250 nodes in address space of 256



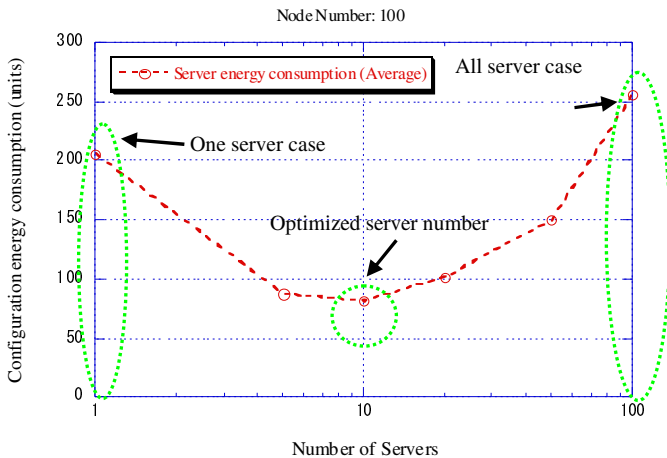
(d) SDIC approach for configuring 250 nodes in address space of 256

Fig. 3. Configuration procedure efficiency. To configure addresses for 100 nodes in an address space of 128, random address configuration (DAD) requires 205% configuration attempts of that of SDIC. And to configure addresses for 250 nodes in an address space of 256, DAD requires about 425.2% configuration attempts of that of SDIC.

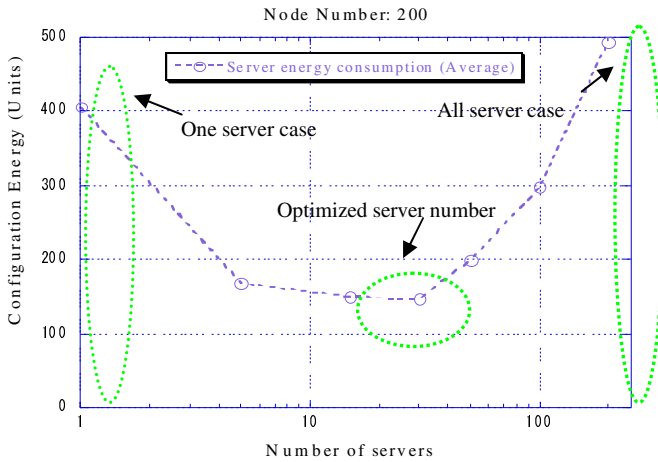
B. Configuration Scalability: The Impact of Server Term Control

Fig.4 shows the impact of server number during the configuration on the average energy consumption at the server nodes. The energy is measured by the energy unit that is the energy consumption in transmitting or receiving a configuration message. Since every server has the same server term, the number of servers reveals the server term of a server. More number of servers means small server term of an address server. The energy consumed at the server nodes is large in cases of either the server number is very small or the server number is very large. When there is only one server in the configuration, much configuration messages, such as AREQ and AREP, overhead will focus on the server, leading to large energy consumption. Further, there is a much unbalance of energy consumption between the address server and clients. On the other hand, when the entire nodes are selected to perform a server’s role the energy consumption is large at both server and client. This is because that many servers causes much server rotation operations, leading to the large energy consumption.

One the other hand, we can observe that there is a optimized number of servers, which achieves the smallest energy consumption at the server nodes. When the server number approaches to a certain number such as 10 in the network of 100 nodes, and 15 of a network with 200 nodes, the energy consumption of servers has the lowest level. This result is in accordance with the analysis of the optimized server number.



(a) The impact of the server number (100 nodes case)



(b) The impact of the server number (200 nodes case)

Fig. 4. Impact of the server term control. There exist optimized server numbers, which result in the smallest energy consumption at server nodes.

7 Conclusion

This paper introduced a SDIC approach that self-organizes the sensor nodes like the flight formation of birds to cooperatively configure energy efficient addresses. SDIC configures addresses for a large number of sensor nodes within a small address space by constructing a controllable and optimized network infrastructure. Like the role management among birds in flight formation, sensor nodes are self-organized with the roles of address servers and address clients to manage the use of address space. SDIC controls server term for the server's rotation among sensors to achieve server robustness and the energy efficiency of configuration. Consequently, the roles and duties of server/clients are dynamically assigned to sensor nodes and are performed in a distributed way.

SDIC configures local unique addresses sequentially, leading to small address size. The configuration is scalable with regard to address conflicts, energy consumption of servers, and information maintained in the servers. Simulation results show that SDIC avoids address conflicts and requires little reconfiguration compared with the conventional address autoconfiguration schemes, especially when there are many nodes in the network. With the efficient server term control, SDIC achieves low and balanced energy consumption among sensor nodes. The use of energy-efficient addresses in sensor network and the bird-flight formation inspired self-organization also highlight the future applications of sensor networks, in which sensor nodes can perform various sensing tasks by cooperatively sharing the roles of network managements with energy efficiency and scalability.

References

1. Estrin, D., Govindan, R.: Next Century Challenges: Scalable Coordination in Sensor Networks. In: ACM/IEEE Conf. MobiCom 1999, pp. 263–270 (1999)
2. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. *IEEE Transactions on Wireless Communications*, 660–670 (October 2002)
3. Mainwaring, A., Polastre, J., Robert, S., David, C., John, A.: Wireless Sensor Networks for Habitat Monitoring. In: ACM International Workshop on Wireless Sensor Networks and Applications, pp. 88–97 (2002)
4. Nesargi, S., Prakash, R.: MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network. In: INFOCOM 2002, pp. 1059–1068 (2002)
5. Vaidya, N.H.: Weak duplicate address detection in mobile ad hoc networks. In: Proc. of ACM MobiHoc, pp. 206–216 (2002)
6. Perkins, C.E., Malinen, J.T., Wakikawa, R., Belding-Royer, E.M., Sun, Y.: IP Address Autoconfiguration for Ad Hoc Networks. Internet Engineering Task Force, MANET Working Group (July 2000), draft-ietf-manet-autoconf-01.txt
7. Gunes, M., Reibel, J.: An IP Address Configuration Algorithms for Zeroconfig Mobile Ad Hoc Networks. In: International Workshop on Broadband Wireless Ad Hoc Networks and Services (2002)
8. Aboba, B., Cheshire, S.: Dynamic Host Configuration Protocol (DHCP) Domain Search Option. RFC 3397 (November 2002)
9. Elson, J., Estrin, D.: Random, Ephemeral Transaction Identifiers in Dynamic Sensor Networks. In: Int'l Conf. Distributed Computing Systems (ICDCS 2001), pp. 459–468 (2001)
10. PalChaudhuri, S., Du, S., Saha, A., Johnson, D.: Treecast: A Stateless Addressing and Routing Architecture for Sensor Networks. In: 4th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN 2004), pp. 221–228 (2004)
11. Schurgers, C., Kulkarni, G., Srivastava, M.B.: Distributed assignment of encoded mac addresses in sensor networks. In: Proc. of ACM Mobihoc, pp. 295–298 (2001)
12. Motegi, S., Yoshihara, K., Horiuchi, H.: Address Autoconfiguration for Event-Driven Sensor Network. IEICE Transaction on Communication, 950–957 (March 2005)
13. Zhou, H., Ni, L.M., Mutka, M.W.: Prophet address allocation for large scale MANETs. In: Proc. INFOCOM, pp. 1304–1311 (2003)
14. The IETF Zero Configuration Networking (zeroconf) Working Group, <http://www.zeroconf.org/zeroconf-charter.html>
15. Nahavandi, A.: Art and Science of Leadership. Prentice Hall, Upper Saddle River (2000)
16. Teng, R., Morikawa, H., Aoyama, T., Zhang, B.: Network-Layer and MAC-layer address autoconfiguration in self-organized sensor networks. In: 6th International Conference on ITS Telecommunications, pp. 1005–1010 (June 2006)
17. Yamazaki, K., Sezaki, K.: Spatio-temporal addressing scheme for mobile ad hoc networks. In: TENCON 2004, pp. 223–226 (2004)
18. Weimerskirch, H., Martin, J., Clerquin, Y., Alexandre, P., Jiraskova, S.: Energy saving in flight formation. *Nature* (October 18, 2001)

Colour Saliency on Video

Michael Dorr^{1,2}, Eleonora Vig², and Erhardt Barth²

¹ Schepens Eye Research Institute, Harvard Medical School,
MA 02114 Boston, USA

`michael.dorr@schepens.harvard.edu`

² Institute for Neuro- and Bioinformatics, University of Lübeck, Germany
`{vig,barth}@inb.uni-luebeck.de`

Abstract. Much research has been concerned with the notion of bottom-up saliency in visual scenes, i.e. the contribution of low-level image features such as brightness, colour, contrast, and motion to the deployment of attention. Because the human visual system is obviously highly optimized for the real world, it is reasonable to draw inspiration from human behaviour in the design of machine vision algorithms that determine regions of relevance. In previous work, we were able to show that a very simple and generic grayscale video representation, namely the geometric invariants of the structure tensor, predicts eye movements when viewing dynamic natural scenes better than complex, state-of-the-art models. Here, we moderately increase the complexity of our model and compute the invariants for colour videos, i.e. on the multispectral structure tensor and for different colour spaces. Results show that colour slightly improves predictive power.

Keywords: video saliency, eye movements, intrinsic dimension, multi-spectral structure tensor.

1 Introduction

The human visual system uses a sophisticated approach to efficiently cope with the vast amounts of data that enter the eye and which need to be processed in real time. Only information from a small central fraction of the visual field, the fovea, is processed at high spatial resolution; more peripheral information is processed only at a very coarse scale and is used mainly for action guidance. One particular problem that the human vision system seems to solve surprisingly well is then when and where to direct the fovea via eye movements to sample all relevant aspects of a visual scene.

Early work found that fixated image regions differed from non-fixated regions in their low-level features such as contrast [10] or higher-order statistics [7]. Nevertheless, it is still a matter of debate whether these altered image statistics at fixation are actually causal of eye movements [4], or whether it is high-level objects that draw attention [3].

For machine vision applications and systems, however, the distinction between a causal and a mere correlative contribution of saliency to eye movement guidance is rather philosophical. It is safe to assume that the human visual system

is highly optimized for the real world, and thus mimicking its performance will find the most informative regions in a scene. Consequently, many models have been developed for saliency on both static images and videos [6,5,8,13]. Typically, these models first extract a range of biologically-inspired low-level features, such as brightness, colour, contrast, orientation, and motion on multiple spatio-temporal scales, and then fuse this information into a single saliency map that assigns a single value of “interestingness” to each image location.

Contrary to these often complex models with a high number of parameters, in previous work we have successfully modelled eye movements using a simple and very generic video representation: the geometric invariants of the structure tensor that capture the amount of spatio-temporal intensity variation [11]. Based on these invariants, we can derive the intrinsic dimensionality of the video, that is the number of degrees of freedom that are used locally. For example, at a stationary edge, the signal changes in only one spatio-temporal direction (orthogonal to the edge), and thus edges constitute $i1D$ regions; transient corners, on the other hand, change in all directions and are therefore $i3D$. One important finding is that the predictive power increases with intrinsic dimensionality: in other words, corners are more informative than edges, and transient features are more informative than their stationary counterparts. A further, surprising finding is that prediction based on this generic video representation outperforms complex state-of-the-art models [12].

So far, the geometric invariants were only computed in grayscale on the luma channel. In the following, we shall compute the invariants on a multispectral structure tensor in order to investigate whether the incorporation of colour information can improve eye movement predictability in dynamic natural scenes.

2 Methods

2.1 The Multispectral Structure Tensor

To estimate the intrinsic dimension of a given video region Ω , we choose a linear subspace $E \subset \mathbb{R}^3$, of highest dimension, such that

$$\frac{\partial f}{\partial \mathbf{v}} = 0 \text{ for all } \mathbf{v} \in E,$$

with the intrinsic dimension of $f = 3 - \dim(E)$ [9]. E can be estimated as the subspace spanned by the set of unity vectors that minimize the energy functional

$$\varepsilon(\mathbf{v}) = \int_{\Omega} \left| \frac{\partial f}{\partial \mathbf{v}} \right|^2 d\Omega = \mathbf{v}^T J \mathbf{v},$$

where the *structure tensor* J [11] is given by

$$J = \int_{\Omega} \nabla f \otimes \nabla f d\Omega$$

with the tensor product \otimes . Alternatively, we can then write

$$J = \omega * \begin{pmatrix} f_x f_x & f_x f_y & f_x f_t \\ f_x f_y & f_y f_y & f_y f_t \\ f_x f_t & f_y f_t & f_t f_t \end{pmatrix}$$

with a spatio-temporal lowpass filter kernel ω and partial derivatives f_x , i.e. $f_x = \partial f / \partial x$. Therefore, E is the eigenspace associated with the smallest eigenvalue of J , and the intrinsic dimension of f corresponds to the rank of J . Instead of performing an eigenvalue analysis, the intrinsic dimension can also be obtained from the symmetric invariants of J :

$$\begin{aligned} H &= 1/3 \text{ trace}(J) &&= \lambda_1 + \lambda_2 + \lambda_3 && (iD \geq 1) \\ S &= |M_{11}| + |M_{22}| + |M_{33}| &&= \lambda_1 \lambda_2 + \lambda_2 \lambda_3 + \lambda_1 \lambda_3 && (iD \geq 2) \\ K &= |J| &&= \lambda_1 \lambda_2 \lambda_3 && (iD = 3). \end{aligned}$$

For a multispectral image sequence, we look for the subspace E of highest dimension such that, in Ω ,

$$\frac{\partial \mathbf{f}}{\partial \mathbf{v}} = 0 \text{ for all } \mathbf{v} \in E.$$

Note that \mathbf{f} is now a vector from \mathbb{R}^q (for an image sequence with q colour channels), so we choose an appropriate scalar product for $\mathbf{y} = (y_1, \dots, y_q)$ and $\mathbf{z} = (z_1, \dots, z_q)$ such that $\mathbf{y} \cdot \mathbf{z} = \sum_{k=1}^q a_k y_k z_k$, with positive weights a_k that can be used to assign higher importance to certain colour channels, and we arrive at the multispectral structure tensor

$$J = \int_{\Omega} \begin{bmatrix} \|\mathbf{f}_x\|^2 & \mathbf{f}_x \cdot \mathbf{f}_y & \mathbf{f}_x \cdot \mathbf{f}_t \\ \mathbf{f}_x \cdot \mathbf{f}_y & \|\mathbf{f}_y\|^2 & \mathbf{f}_y \cdot \mathbf{f}_t \\ \mathbf{f}_x \cdot \mathbf{f}_t & \mathbf{f}_y \cdot \mathbf{f}_t & \|\mathbf{f}_t\|^2 \end{bmatrix} d\Omega.$$

In our implementation, we chose 5-tap spatio-temporal binomials for ω and for smoothing the video sequence before taking the derivatives, and J was computed for a spatio-temporally downsampled version of the original video (factor four in space and time). Saliency was then determined as the average energy of the geometric invariants in an 8x8 pixel window around a location.

2.2 Colour Spaces

The colour space RGB is commonly used in computer graphics and stores images with red, green, and blue components. Video formats, however, often exploit the reduced colour resolution of the human visual system and thus our original videos had been recorded in the $Y'CbCr$ format with one luma and two chroma channels (of halved resolution). When using $Y'CbCr$ directly, the dynamic range of the luma channel is much larger than that of the chroma channels, and the contribution of colour to $J_{Y'CbCr}$ is small. We therefore computed the standard deviation of each channel in our set of videos (Y' : 124.8; C_b : 39.7; C_r : 44.9) and used their inverse for the weights a_y, a_{cb}, a_{cr} to obtain the colour space $Y'CbCr_{weighted}$. As a baseline, we computed the invariants in grayscale on the Y' channel only.

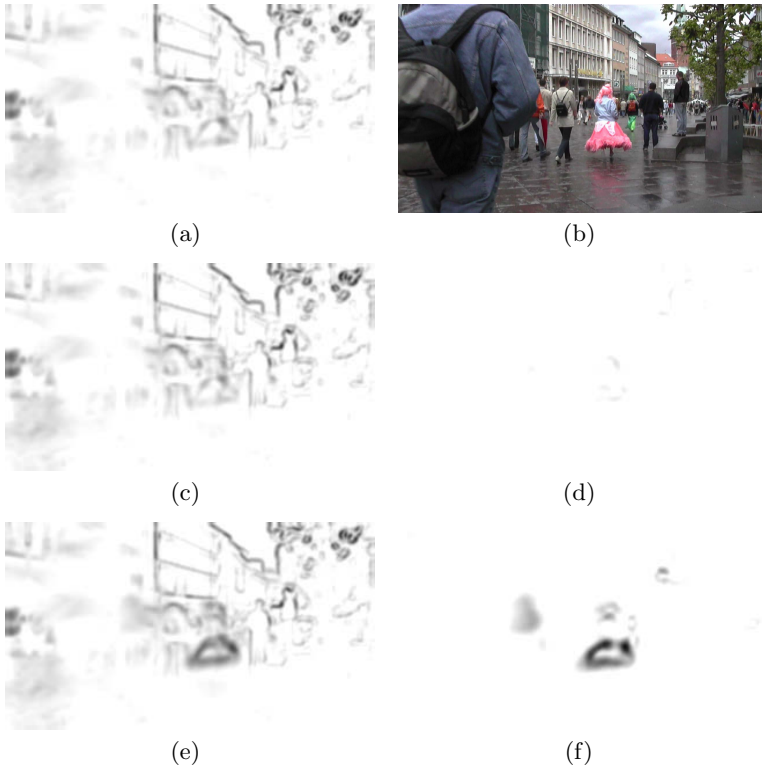


Fig. 1. Example frames for colour saliency. a) Invariant H computed on the luma channel only (inverted for legibility). b) Original video frame. c) H on J_{RGB} . d) Absolute difference between a) and c). Even though colour information is represented in RGB , the difference is small. e-f) H on $J_{Y'CbCr_weighted}$ and absolute difference to a).

2.3 Experimental Data

We used a large eye movement database of about 40000 saccades obtained from 54 subjects watching 18 high-resolution movies (1280 by 720 pixels, 29.97 fps, about 20 s duration each) described in detail elsewhere [2]. For a set of negative examples, we did not generate random data, but shuffled scanpaths on different movies in order to keep the spatio-temporal distribution of positive and negative samples over all movies constant. These samples were then classified with a Maximum Likelihood classifier based on one of the invariants.

3 Results

The ROC scores for the geometrical invariants H , S , and K on the multispectral structure tensor for different colour spaces are shown in Fig. 2. We can replicate the previous result that regions with higher intrinsic dimensionality are also

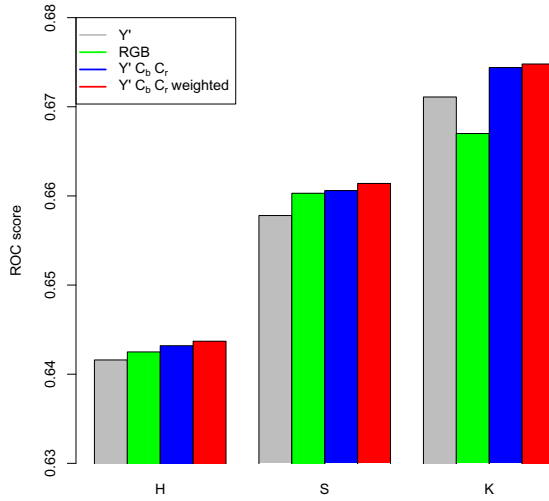


Fig. 2. ROC scores for eye movement predictability of the geometrical invariants of the multispectral structure tensor in different colour spaces. The higher the intrinsic dimension, the higher the predictability ($K > S > H$); saliency on colour video predicts eye movements better than on grayscale video (Y').

more predictive of eye movements ($K > S > H$). Furthermore, the inclusion of colour information improves predictive power, but only slightly. The differences between the different colour spaces are very small, except for the invariant K on RGB , which performs worse even than the grayscale K .

4 Conclusion

We have previously found that a simple, generic model of spatio-temporal intensity variation can predict eye movements on natural videos at least as well as complex state-of-the-art models. In the present manuscript, we have incorporated colour information into our model while maintaining its conceptual simplicity (but at increased computational cost). Results show that indeed colour improves predictive power, but only moderately so. Whether this is due to a ceiling effect or to a relatively small contribution of colour to eye guidance in dynamic natural scenes remains to be determined. Future work will also incorporate further colour spaces, such as HSV, which is particularly sensitive for skin colour, or the perceptually equidistant LAB space.

Acknowledgements. Our research has received funding from the European Commission within the GazeCom project (IST-C-033816) of the FP6, and was further supported by NIH grants EY018664 and EY019281. All views herein are

those of the authors alone; the European Commission is not liable for any use made of the information.

References

1. Bigün, J., Granlund, G.H., Wiklund, J.: Multidimensional orientation estimation with application to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(8), 775–790 (1991)
2. Dorr, M., Martinetz, T., Gegenfurtner, K., Barth, E.: Variability of eye movements when viewing dynamic natural scenes. *Journal of Vision* 10(10), 1–17 (2010)
3. Einhäuser, W., Spain, M., Perona, P.: Objects predict fixations better than early saliency. *Journal of Vision* 8(14), 11–26 (2008)
4. Elazary, L., Itti, L.: Interesting objects are visually salient. *Journal of Vision* 8(3), 1–15 (2008)
5. Itti, L., Baldi, P.: A principled approach to detecting surprising events in video. In: *Proc. IEEE Conf on Computer Vision and Pattern Recognition*, pp. 631–637 (2005)
6. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(11), 1254–1259 (1998)
7. Krieger, G., Rentschler, I., Hauske, G., Schill, K., Zetzsche, C.: Object and scene analysis by saccadic eye-movements: an investigation with higher-order statistics. *Spatial Vision* 13(2,3), 201–214 (2000)
8. Le Meur, O., Le Callet, P., Barba, D., Thoreau, D.: A coherent computational approach to model bottom-up visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(5), 802–817 (2006)
9. Mota, C., Stuke, I., Barth, E.: The Intrinsic Dimension of Multispectral Images. In: *MICCAI Workshop on Biophotonics Imaging for Diagnostics and Treatment*, pp. 93–100 (2006)
10. Reinagel, P., Zador, A.M.: Natural scene statistics at the centre of gaze. *Network: Computation in Neural Systems* 10, 341–350 (1999)
11. Vig, E., Dorr, M., Barth, E.: Efficient visual coding and the predictability of eye movements on natural movies. *Spatial Vision* 22(5), 397–408 (2009)
12. Vig, E., Dorr, M., Martinetz, T., Barth, E.: A Learned Saliency Predictor for Dynamic Natural Scenes. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) *ICANN 2010, Part III*. LNCS, vol. 6354, pp. 52–61. Springer, Heidelberg (2010)
13. Zhang, L., Tong, M.H., Cottrell, G.W.: SUNDAY: Saliency Using Natural Statistics for Dynamic Analysis of Scenes. In: *Proceedings of the 31st Annual Cognitive Science Conference*, Amsterdam, Netherlands (2009)

Bio-inspired Transputer Based-Fuzzy Mobile Robot

Ebrahim Mattar¹, Khalid Al-Mutib², Hani Hamdan³, and Mohamad Hamdan⁴

¹ University of Bahrain, P.O. Box 32038, Kingdom of Bahrain

² King Saud University, College of Computer and Information Sciences,
Department of Software Engineering, P.O. Box 51178, Kingdom of Saudi Arabia

³ SUPELEC, Department of Signal Processing and Electronic Systems,
Rue Joliot-Curie, 91192 Gif-sur-Yvette cedex, France

⁴ Université Libanaise, Faculté de Génie, Department of Electrical and Electronics,
Engineering, Beirut, Lebanon

ebrgallaf@eng.uob.bh, mutib@ksu.edu.sa, Hani.Hamdan@supelec.fr,
mhamdan@ul.edu.lb

Abstract. Mobile robots have been used widely in large number of intelligent based applications, where a random manoeuvring is an essential part of such systems, [1,2]. In this regards, experience has shown that mobile robots do require advanced parallel intelligent based control mechanisms. In this article, it is shown how a mobile robot can be controlled in movement via some rule based system. In this sense, the design, construction and testing of an autonomous mobile robot have already been defined in the literature. A Transputer Embedded Real-Time Controller has been widely used for industrial applications due to their advances. Here, a Fuzzy based Transputer System is used (on board the robot) to meet some intelligence requirements for the navigation and obstacle avoidance. A parallel fuzzy controller is implemented for a mobile robot guidance and obstacle avoidance. This is based on parallel processing of sonar data that do come from the mobile sensory system.

Keywords: Bio-Inspired Robots, Parallel Processing, Occam Processes, Mobile Robots Kinematics, Fuzzy Systems, Autonomous Robot.

1 Introduction

1.1 A Mobile Robotics System

The mobile robot indicated here in this article is known as the " Occam Mobile Robot". The robot has been modeled and controlled through parallel processing units. Occam parallel processes has been used to control the mobile robot is shown in Fig. (1). For the design of the robot, the following assumptions were made accordingly:

- {a} The mobile robot must be capable to follow an (x,y) path in any direction (θ) over the plane.
- {b} It must be robust in its mechanical design.

- {c} It must include sufficient sensors and actuators.
- {d} It has to be easily controlled and of ease to maintain.
- {e} It should be intelligent to make navigation decisions on its own.
- {f} It should have simple mechanical design.
- {h} All steering axes are perpendicular to the surface.

In addition, the following operational assumptions were assumed:

- {i} The mobile robot is to move on a planar surface.
- {ii} The translation friction at the point of contact between a wheel and the surface is large enough so that no translational slip occur.
- {iii} The rotational friction at the point of contact between a wheel and the surface is small enough so that rotational slip may occur.

Transputer Embedded Real-Time Controller has been widely used for industrial applications due to their advances. Fuzzy based Transputer system is used on board the robot to meet the intelligence requirement for the navigation and obstacle avoidance. Parallel fuzzy controller is implemented for the robot guidance and obstacle avoidance. This is based on processing of sonar data that do come from the mobile sensory system.

1.2 Proposed Control System

The system proposed here, is to allow a fuzzy controller to control the movement and the maneuver actions of a mobile robot. This is done via the use of a parallel processing of the sensory data. Sensory data are processed in parallel fashion, hence a fuzzy controller output is computed accordingly quickly. The proposed fuzzy controller is then used to control the movement of the mobile robot. The fuzzy controller has been trained and defined using the operators experience.

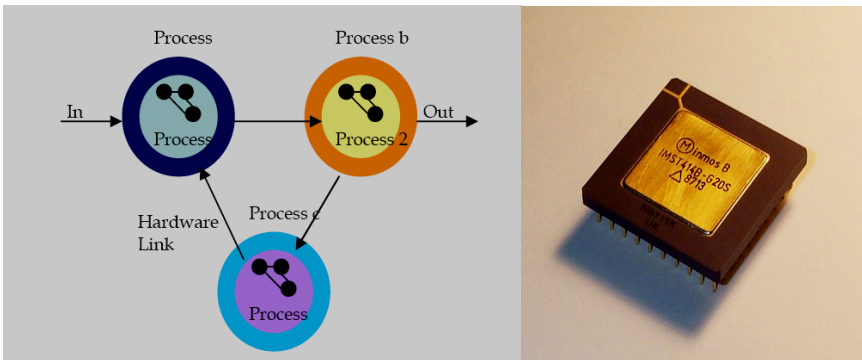


Fig. 1. The Occam process, and a T414 Transputer chip. The transputer architecture was designed as an Occam engine.

2 Mobile Robot Physical Construction

2.1 Robot Physical Body

We define the following: The Occam Mobile Robot can follow a path in any direction but first it must rotate around its axis to face that direction. It has two diametrically opposed drive wheels. Because of the simplicity of its mechanical design it has simple kinematics. The mobile robot weighs about two kilograms in a rectangular shape of width (30cm) and length of (40cm). It has two wheels of diameter (10 cm) located at the centre of the robot used for motion and steering. Each wheel actuates through a gear (10:1 ratio) by an independent DC motor. A shaft encoder is attached to each motor for wheel position sensing. Two free-wheeling castors are located one in the front and one at the back of the robot to balance the robot. The corners of the robot circumference are flat and with (45°) angle from the adjacent flat wall. These flat corners are used to mount the corner ultrasonic sensors in order to enable the robot to see in that direction. The sensors are mounted around the circumference of the robot body, and they are at 30cm high from floor. The sensors are slightly tilted upward to prevent ultrasonic echo reflection from the ground floor. The mobile robot is powered by four rechargeable (12V batteries), 2×12Ah and 2×6Ah. These batteries are configured to supply (+12V, -12V, +24V and -24V). A voltage converter is also used to provide (+5V).

2.2 Robot Kinematics Development

To provide a framework within which to develop the robot kinematic models, Muir and Neuman, 1986 [3], defined a wheeled mobile robot as: “ A robot capable of locomotion on a surface solely through the action of wheel assemblies mounted on the robot and in contact with the surface. A wheel assembly is a device which provides or allows relative motion between its mount and a surface on which it is intended to have a single point of rolling contact”, [2]. Kinematics is the study of the geometry of motion. All kinematics are derived by straightforward application of the axioms and corollaries of the transformation algebra, [3]. In the context of the wheeled mobile robots, the motion of the robot can be determined from the geometry of the constraint imposed by the motion of the wheels.

3 The Mobile Robot Fuzzy Controller

3.1 The System Controller

In the mobile robot fuzzy controller, the eight ultrasonic sensor range measurement data are the input to the controller and the output is the left and the right wheel speeds as shown in Fig. 2.

This fuzzy controller is integrated with the sensor fusion and the low level PID controller to form the complete controller for the mobile robot as shown in Fig. (3).

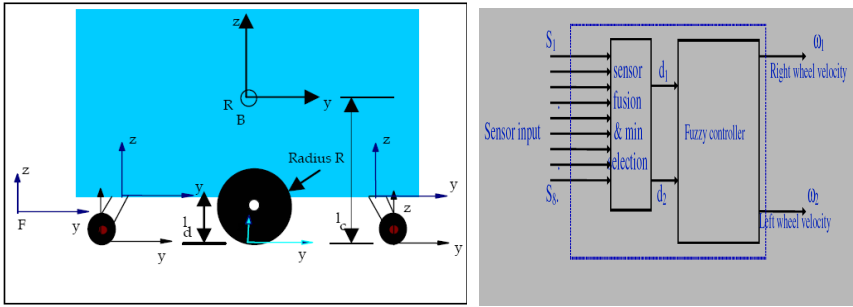


Fig. 2. Body coordinates assignments. Input-Output Mobil Robot Fuzzy Controller.

The overall controller starts with a user interface where the user can enter the desired control parameters and in this interface the statuses of the mobile robot will be displayed before starting the controller. The robot checking routine will perform self testing on the various parts of the hardware of the robot and report any problem to the user before starting. The sensor fusion is the integration of all of the eight ultrasonic sensors to form the input to the fuzzy controller. The robot sensors are grouped in to four regions as shown in Fig. 4.

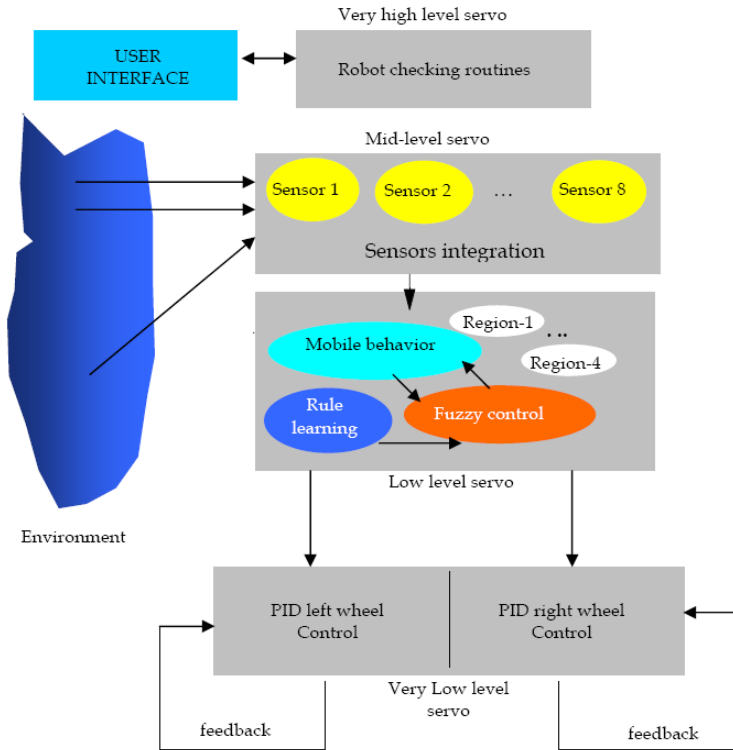


Fig. 3. Block diagram of the overall controller for the mobile robot

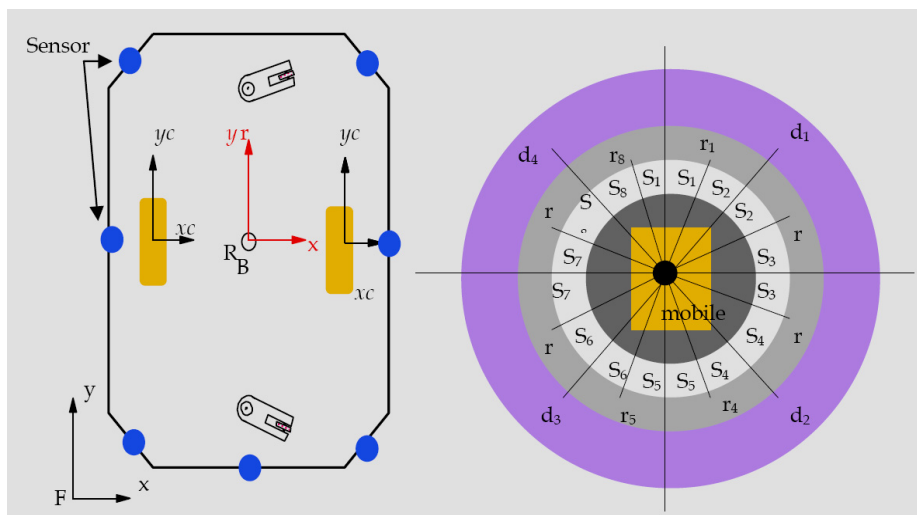


Fig. 4. Mobile sensing units: As grouped in four regions

Region #1 consist of sensor (1, 2 and 3). Region #2 consist of sensor 3, 4 and 5. Region #3 consist of sensor 5, 6 and 7. Region #4 consist of sensor #7, 8 and 1. In this arrangement some sensors are part of more than one region and this grouping is consistent with the fuzzy group classification where the borders between the sets (regions) are not crisp. In the process of selecting a region out of the four, the following steps are implemented:

- 1) Read all the sensors (S_1 to S_8), and pass to controller.
- 2) Arrange the sensor output value in ascending order.
- 3) Identifying the three sensors with the lowest value.
- 4) Check if two of the three are in one region.
- 5) If yes, that region will be selected.
- 6) If no, go to step one again.

The selection of the region will define the fuzzy rule that will be used in the fuzzy controller.

3.2 Fuzzy Control Information Processing

The processing of the information in the fuzzy process will first include the fuzzification of the input variables and then the defuzzification of the output variables as discussed in (4.4).

3.3 Fuzzification of Sensors Data (Variables d_{min1} and d_{min2})

Once the region is selected the information supplied by the sensor in that region will indicate the nearest obstacle to the robot and the next nearest obstacle (d_{min1} and d_{min2})

facing that region. The fuzzy control can be realized by the control of the robot wheels rotation to keep the robot within a pre-set distance from the obstacle facing that region of the robot. A fuzzy operator converts the crisp input data, say $\{d\}$, into the linguistic values (\hat{d}) determined as labels of fuzzy sets given by Equ (1):

$$\text{fuzzifier}(d_1, d_2) \Rightarrow \epsilon (d_1^{\wedge}, d_2^{\wedge}) \tag{1}$$

where *fuzzifier* denotes a Fuzzification operation. From now on, the sign (^) representing the fuzzy set will be omitted for simplicity. The input linguistic variables d_1 and d_2 are expressed by linguistic terms as illustrated by the membership function in Equ. (2) and Equ. (3) and in Fig. 5. respectively. The meanings of the linguistic terms are given in Table (1).

$$\{es\ vvs\ vs\ s\ m\ b\ vb\ vvb\ eb\} \subseteq m(d_1) \tag{2}$$

$$\{es\ vvs\ vs\ s\ m\ b\ vb\ vvb\ eb\ eeb\} \subseteq m(d_2) \tag{3}$$

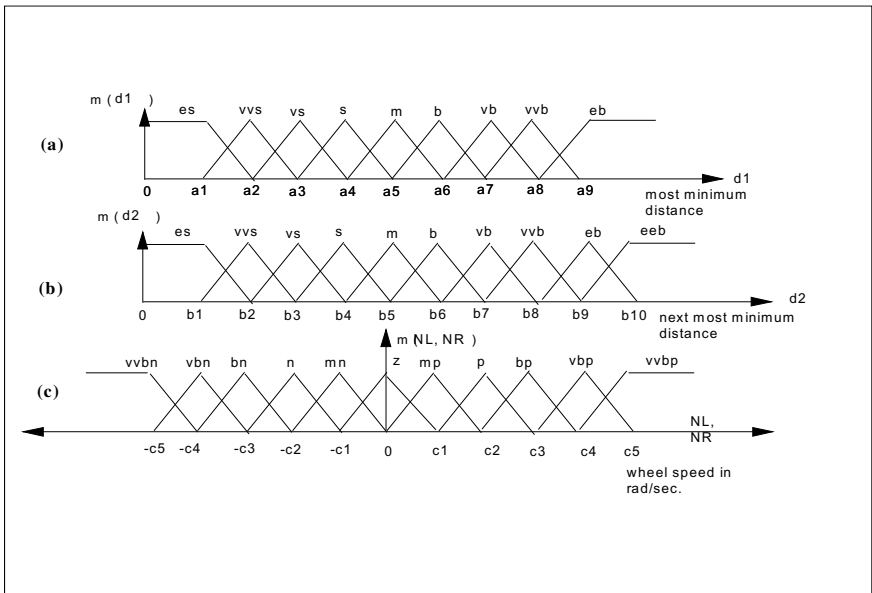


Fig. 5. Fuzzification of the input and output variables

The fuzzy output is the speed of the right and the left wheels in (rad/sec). They are expressed by the linguistic values with membership functions having a triangular shape functions as shown in Fig. 5. The halfway of this membership function is determined by an initial value of zero. The linguistic provisions for the output variables are given in Equ (4) and in Table (2) :

$$\{vvpn, vbn, bn, n, mn, z, mp, p, bp, vbp, vvpn\} \subseteq m(N_L) \tag{4}$$

Table 1. Membership for the input variables d_1 and d_2

Symbols	Linguistic variable
es	extremely small
vvs	very very small
vs	very small
s	small
m	medium
b	big
vb	very big
vvb	very very big
eeb	extremely big

Table 2. Membership for output variables (N_L and N_R)

Symbols	Linguistic variable
vvpn	very very big negative
vbn	very big negative
bn	big negative
n	negative
mn	medium negative
z	zero
mp	medium big
p	positive big
bp	big positive
vbp	very big positive
vvpb	very very big positive

In Equ. (4), the eleven states membership function is expressed in terms of degree of membership function (MF). Fuzzy subsets embody elements with degree of membership. On the other hand, fuzzy FM $\mu_z(\)$ of the fuzzy set{ }, assigns a real number between (0 to 1) to every element in the universe of discourse. With reference to the membership functions of the fuzzy sets defined in Fig. 5., these membership values indicate the degree to which the element (a number) belongs to the fuzzy set. A fuzzy membership function can have various shapes depending on the designer's preference and the need for it in a particular fuzzy controller. The triangular shape is the most common shape of the membership function in the literature and the easiest to implement, [7]. After isolating the control features of the fuzzy controller and decomposing each variable into fuzzy sets, the conceptual model is completed. This done by writing the production rules describing the action taken on each combination of the control variables. The mobile fuzzy controller is a two input (d_1, d_2) and one output which is the desired speed of the motor (refer to Fig. 5.). It is convenient to consider such control as an individual matrix of actions in an ($d_1 \times d_2$) array.

The fuzzy states of one control variable (d_1) form the horizontal axis and the fuzzy states of the other control variable (d_2) form the vertical axis. At the intersection of a row and column is the fuzzy state of the solution variable. For this system the representation assumes the shape of ($d_1 \times d_2 \times N_R$) cube. Systems of higher order are most difficult to conceptualize but, in principle, the same kind of correspondence between control and solution variables exists. This form of representation, very

common in the control engineering field, is called a Fuzzy Associated Memory or the (FAM). Table. 3. shows the FAM for the right wheel speed and in Table. 4. the FAM for the left wheel speed.

3.4 Construction of Rule Base Fuzzy System

The rule base for realizing each behaviour can be constructed based on the operator experience. For the fuzzy controller, the partial mapping is to be translated into linguistic rules. The fuzzy rule has an (IF-THEN) format as follows :

Table 3. The FAM for the right wheel speed (Region 1)

d ₁ \ d ₂	es	vvs	vs	s	m	b	vb	vvb	eb
es	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
vvs	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
vs	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
s	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
m	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
b	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
vb	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
vvb	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
eb	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
eeb	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb

The FAM for the right wheel speed (Region 1)

d ₁ \ d ₂	es	vvs	vs	s	m	b	vb	vvb	eb
es	z	vvpb	vvpb	vvpb	vvpb	vvpb	vvpb	vvpb	vvpb
vvs	vvpb	p	mp	vvpb	vvpb	vvpb	vvpb	vvpb	vvpb
vs	vvpb	bp	mp	bp	vbp	vvpb	vvpb	vvpb	vvpb
s	vvpb	vbp	bp	bp	p	vbp	vvpb	vvpb	vvpb
m	vvpb	vvpb	vbp	vvpb	vbp	bp	vbp	vvpb	vvpb
b	vvpb	vvpb	vvpb	vvpb	bp	vbp	vbp	vvpb	vvpb
vb	vvpb	vvpb	vvpb	vvpb	vbp	vbp	vbp	vvpb	vvpb
vvb	vvpb	vvpb	vvpb	vvpb	vbp	vvpb	vbp	vvpb	vvpb
eb	vvpb	vvpb	vvpb	vvpb	vvpb	vvpb	vvpb	vvpb	vvpb

Table 4. The FAM for the left wheel speed. The linguistic rules of the left wheel speed in region two.

d ₁ \ d ₂	es	vvs	vs	s	m	b	vb	vvb	eb
es	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
vvs	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
vs	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
s	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
m	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
b	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
vb	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
vvb	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb
eb	vbp	vbp	vbp	vbp	vvpb	vvpb	vvpb	vvpb	vvpb

The linguistic rules of the left wheel speed The FAM for the right wheel speed (Region 1)

Table 4. (continued)

$d_1 \backslash d_2$	es	vvs	vs	s	m	b	vb	vvb	eb
es	z	bp	bp	bp	bp	bp	bp	bp	bp
vvs	bp	z	mp	vb	vvp	vvp	vvp	vvp	vvp
vs	bp	bp	mp	bp	vb	vvp	vvp	vvp	vvp
s	bp	vb	bp	mp	p	vb	vvp	vvp	vvp
m	bp	vvp	vb	vvp	p	bp	vb	vvp	vvp
b	bp	vvp	vvp	vvp	bp	p	vb	vvp	vvp
vb	bp	vvp	vvp	vvp	vb	vb	bp	vb	vvp
vvp	bp	vvp	vvp	vvp	vvp	vvp	vb	vvp	vvp
eb	bp	vvp	vvp	vvp	vvp	vvp	vvp	vvp	vvp

IF (d_1 is *mb* AND d_2 is *vb*) then (N_L is *vvp* AND N_R is *vb*)

where $d_1, d_2, N_L,$ and N_R are the fuzzy variables and *mb, vb, vvp* and *vb* are the fuzzy subsets in the universe of discourses (X, Y and Z). The fuzzy rule base consists of several rules of the form given above. The experience of the operator play a big role in the shape and form of the rules and in the number of the rules, in the rule base fuzzy controller. In this fuzzy controller, the membership functions ($\mu_1, \mu_2, \mu_3, \mu_4$) with association with the (d_1) and (d_2) are computed as follows, [7]:

$$x = \text{Int} ((d_1+1)/10) \quad y = \text{Int} ((d_2+1)/10)$$

where (Int) is the integer part of the expression :

$$xx = (10x + 5 - d_1) \quad yy = (10y + 5 - d_2)$$

and complement of these fuzzy sets is given by:

$$(C_{xx} = 1 - xx) \quad (C_{yy} = 1 - yy)$$

$$\begin{aligned} \mu_1 &= \mu_i(d_1) \oplus \mu_i(d_2) \\ \mu_1 &= \min (xx, yy) \\ \mu_2 &= \min (xx, C_{xx}) \\ \mu_3 &= \min (C_{xx}, yy) \\ \mu_4 &= \min (C_{xx}, C_{yy}) \end{aligned} \tag{5}$$

In Equ. (5) is the fuzzy OR operator for the fuzzy set which select the minimum of the two variables given. The membership function in Equ. (5) is used in the defuzzification process, [12].

3.5 Defuzzification of Output Variables

In order to determine the crisp output (N_L, N_R) from the fuzzy control action, a defuzzification process is required , [8]. The centre of gravity defuzzification is a

commonly used method utilized in the control engineering applications. In the centre of gravity method we define the value (α) located under the centre of gravity of the area that is given by the function $\mu_{i=1,2,3,4}$. The crisp output is founded by, [8]:

$$N_L = \frac{\alpha_1\mu_1 + \alpha_2\mu_2 + \alpha_3\mu_3 + \alpha_4\mu_4}{\mu_1 + \mu_2 + \mu_3 + \mu_4} \tag{6}$$

where ($\alpha_1, \alpha_2, \alpha_3, \alpha_4$) are the corresponding values for the fuzzy rule for the N_L output. And for the second crisp output N_R the same Equ. (5) will apply using the corresponding ($\alpha_1, \alpha_2, \alpha_3, \alpha_4$) values. Each one of the four regions of the sensor grouping of Fig. 4. has two base rules, ones rule for the left wheel and another rule for the right wheel. The mobile robot fuzzy controller described above can be illustrated further by Fig. 6. This figure shows the interaction of the various parts of the fuzzy controller, the robot environment and the robot dynamics. In addition to the four regions rules, there is a fifth region which is the negative of the rule for region one. This fifth region is applied whenever the robot gets trapped in a situation where it cannot go left or right. This region will be selected by the fuzzy controller to reverse the robot from the entrapment. After the robot is out of the entrapment, one of the other four regions rules will be selected as applicable. The above fuzzy controller has been divided in two parts: the obstacle avoidance routine and then a

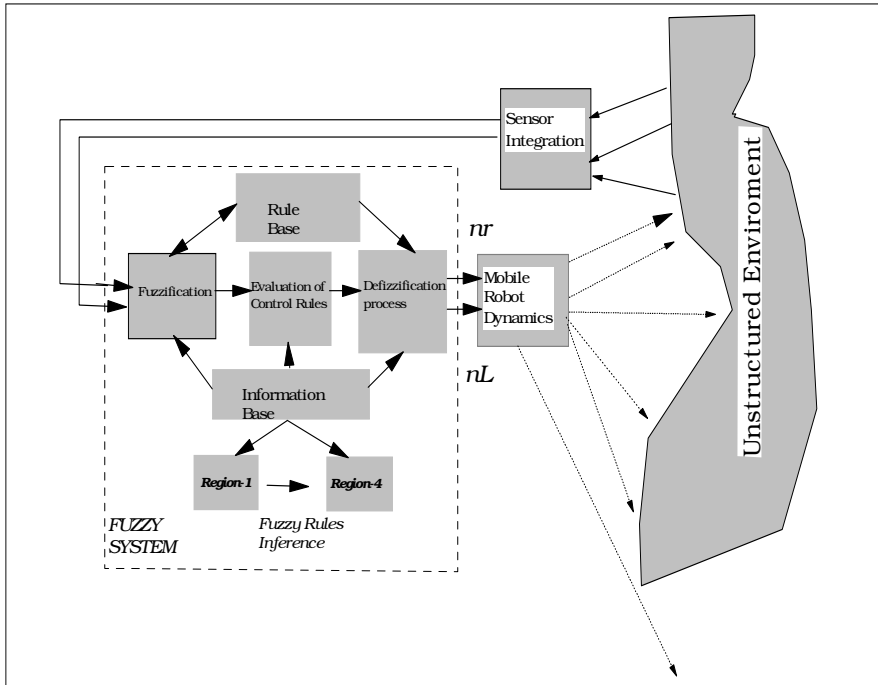


Fig. 6. Overall configuration of the fuzzy controller and its interaction with the robot dynamic and with the environments surrounding the robot speed in region one

goal seeking routine. The obstacle avoidance routine has been tested by simulation and by experiment which will be shown in the next section.

4 Simulation and Experimental Evaluation Fuzzy Controller

4.1 The Simulation Environment

The obstacle avoidance fuzzy controller has been simulated using the fuzzy rules and the kinematics of the mobile robot using MATLAB programming algorithms. The simulation x-y space has been made similar to the size of the operating area (6×6 meters) of the robot in the experimental setup. The simulation program starts by asking the initial (x,y) location in the (x,y) plane and the orientation angle with respect to the x-y world coordinate. The simulation shows that the robot moves around the plane geometry governed by the fuzzy-base rules without colliding with the surrounding boundaries. The simulation results are shown in Fig. 7. with the mobile robot as moving circles. The robot was tested, in real-time, using this fuzzy controller used in the simulation but was written in Occam language. Fig. 7. shows the experimental court (circles) with the erected walls. The robot in the experimental testing exhibited similar behavior to the one shown in this simulation. The experimental behavior of Occam Robot was recorded on a video tape.

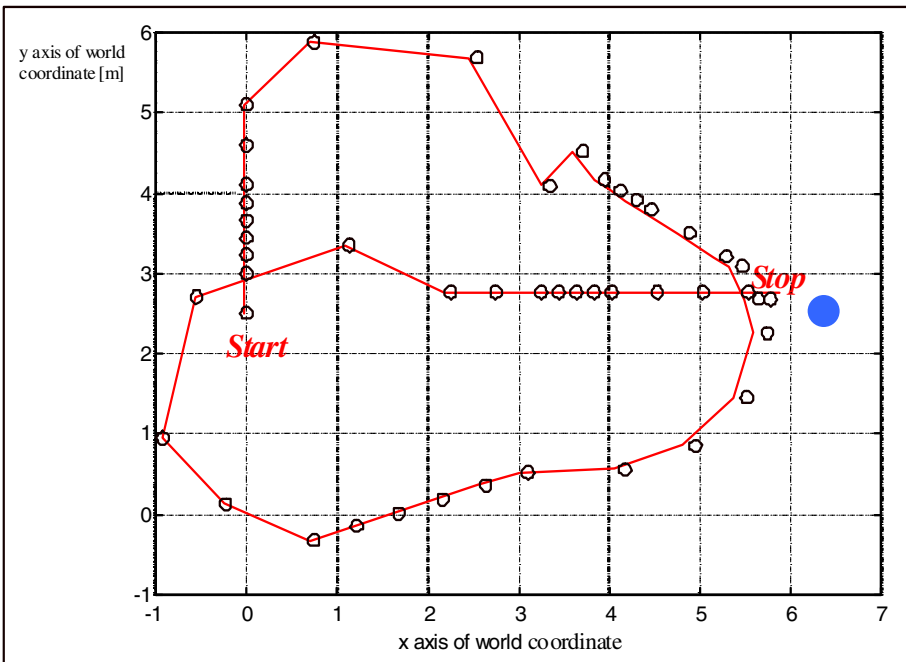


Fig. 7. The simulation of the robot movement using fuzzy controller with the robot moving in a circle. It started at 90° orientation at $x = 2.5$ and $y = 2.5$.

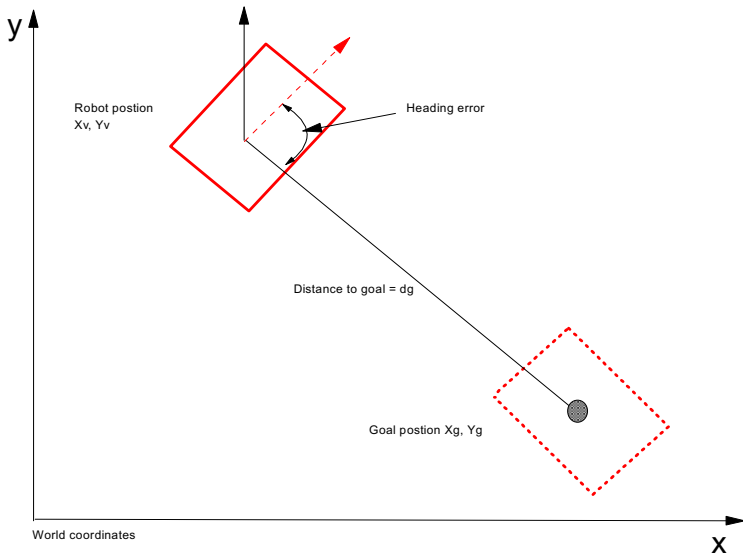


Fig. 8. The robot goal seeking and the global coordinate system

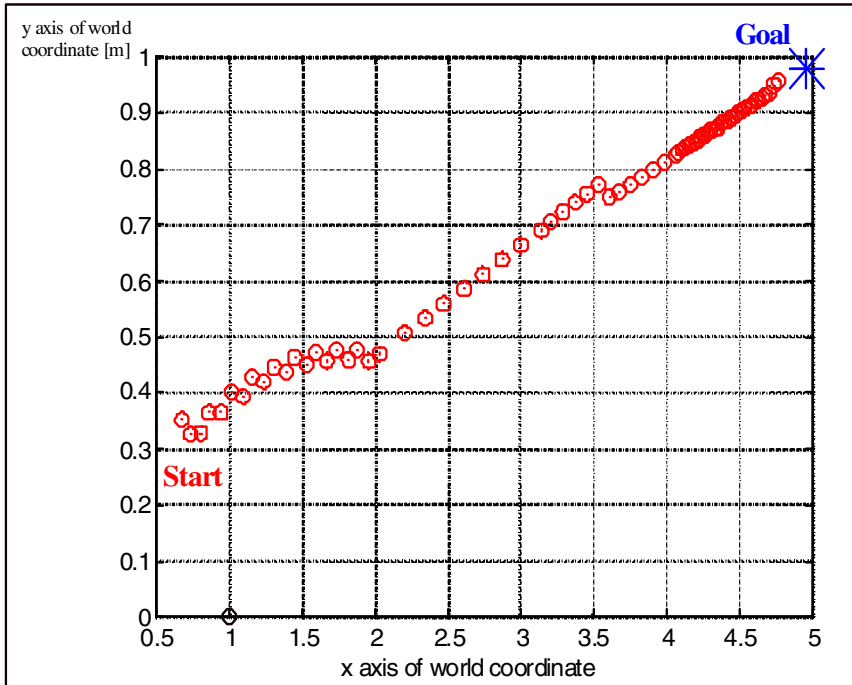


Fig. 9. The goal seeking simulation, the robot started at $x=0.75, y=0.35$ and 90° orientation. First the robot rotated until it faced the goal and then proceeded to the goal.

4.2 Robot Goal Seeking

The goal seeking simulation is shown in Fig. 9. Here the robot starts from known position and orientation in the world coordinate and stopped at the location of the desired goal position. The control algorithm enables the robot to start from a known position and stop at the specified goal position. If the robot encounters any obstacle in its way, the obstacle avoidance controller will take control of the mobile robot until the path of the robot is clear from any obstacle within the seeing range of the robot. After the path is clear of any obstacle the goal seeking controller will be active again.

4.3 Robot Collision Avoidance

The goal seeking algorithm is based on first rotating the robot until it does face the goal if it is not already facing it, than the algorithm try to reduce the distance between the robot and the goal to zero as shown in Fig. 9., where the distance to goal d_g , [8]:

$$d_g = \sqrt{(x_g - x_v)^2 + (y_g - y_v)^2}, \quad \text{and heading error } \theta_{HE} = (\theta_v + \theta_g)$$

5 Conclusions

In this paper a Transputer-based embedded controller has been successfully used to drive the mobile robot. Coordinate system frames are hence used for modeling purpose, as has been presented in details. The ability of the Transputer embedded controller (as a good example of parallel computation), to perform a demanding parallel fuzzy controller, has been demonstrated in a real-time experimental testing of the mobile robot in the obstacle avoidance behavior. Ultrasonic sensor fusion in the mobile robot navigation have been accomplished successfully. For such moving mobile robots, the parallel implementations of fuzzy controller and its advantages have been outlined. The obstacle avoidance fuzzy controller has been constructed and tested by simulation and experimentally. The goal seeking fuzzy controller has been also tested via some simulated robot movements and hence verified by some experimental tests.

References

1. Reister, D.B., Pin, F.G.: Time-Optimal Trajectories for Mobile Robots With Two Independently Driven Wheels. *The International Journal of Robotics Research* 13(1), 38–54 (1994)
2. Shgiley, J.E.: *Kinematic Analysis of Mechanisms*. McGraw-Hill, New York (1969)
3. Muir, P.F., Neuman, C.P.: Kinematics Modeling of Wheeled Mobile Robots. *Journal of Robotics Systems* 4(2) (1987)
4. Sarkar, N., Yun, X., Kumar, V.: Control of Mechanical Systems With Rolling Constraints: Application to Dynamic Control of Mobile Robots. *The International Journal of Robotics Research* 13(1), 55–69 (1994)

5. McKerrow, P.: Introduction to Robotics. Addison-Wesley Longman Publishing Co., Inc., Boston (1991) ISBN: 0201182408
6. Pedrycz, W.: Fuzzy Control and Fuzzy Systems, p. xiv + 225. Research Studies Press (1989) ISBN 0471 9231 17
7. Beom, H.R., Cho, H.S.: A Sensor-Based Navigation for a Mobile Robot Using Fuzzy Logic and Reinforcement Learning. IEEE Transaction on Systems, Man, and Cybernetics 25(3) (1995)
8. Baxter, J.W., Bumby, J.R.: Fuzzy control of a mobile robotic vehicle. In: Proceeding of Institute of Mechanical engineering, IMechE
9. Klir, G., Yuan, B.: Fuzzy Sets And Fuzzy Logic, Theory And Applications, 1st edn. Prentice Uranologist PTR (1995)
10. Meystel, A.: Autonomous Mobile Robots, Vehicles with Cognitive Control. World Scientific Publishing Co. Pte. Ltd. (1991)
11. Klafter, R.D., Chmielewski, T.A., Negin, M.: Robotic Engineering an Integrated Approach. Prentice-Hall (1989)
12. Mir, S.A., Zinger, D.S., Elbuluk, M.E.: Fuzzy Controller for Inverter Fed Induction Machines. IEEE Transaction on Industry Applications 30(1) (1994)
13. Benmounah, A.: Transputer Control of an AGV, Design, Construction, And Testing of a Mobile Platform. Ph.D. Thesis, University of Reading, UK (June 1991)
14. Maamri, M.: Control of Mobile Platform in a visually monitored environment. Ph.D. Thesis (May 1991)

Bio-inspired Visual Information Processing – The Neuromorphic Approach

Woo Joon Han and Il Song Han

Korea Advanced Institute of Science and Technology,
335 Gwahak-ro, Yuseong-gu, 305-701 Daejeon, Republic of Korea
{wjh208, i.s.han}@kaist.ac.kr

Abstract. This paper describes the bio-inspired visual information processing by neuromorphic system, mimicking the primitive behavior of visual cortex. The neuromorphic components are investigated for implementation of the visual signal selectivity of cortex, based on the CMOS conductance-based synaptic connections and neurons of Hodgkin-Huxley formalism. The proposed neuromorphic system exhibits the biologically plausible function mimicking the cat's visual cortex experimentation of Hubel and Wiesel. The detection of human head figure or pose demonstrates the feasibility of vision applications.

Keywords: neural networks, CMOS, vision, neuromorphic, visual cortex, simple cell, Hodgkin-Huxley formalism.

1 Introduction

There have been many works proposed recently for neuromorphic circuits and system, by mimicking both the functional and physiological characteristics of biological systems. We describe here the bio-inspired implementation of primary visual cortex, based on the neuron of Hodgkin-Huxley formalism and the visual cortex experimentation of Hubel and Wiesel. In this paper, the elements of neuromorphic implementation of visual cortex are presented with the orientation tuned map of synaptic weights and the spiking neuron, based on the electronically programmable MOSFET conductance.

The feasibility of neuromorphic VLSI visual cortex is investigated by simulated experimentation based on the CMOS 0.18 μ m technology, with demonstrated vision applications of detecting the fire, human head figure, or pose from image sequences.

2 Primary Visual Cortex Function and Bio-inspired Spiking Neuron Based on CMOS Circuit

The physiological studies about visual cortex from the investigation of cat's striate cortex by Hubel and Wiesel have confirmed the consensus of knowledge [1], though there are many models about visual cortex. The idea on the primary visual cortex of simple cell motivated various theories of object recognition from characters to complex

natural images [2]. For an idea of neural system implementation, the research about neurophysiology introduced the principles and demands of biologically plausible electronic implementation. In this paper, we propose the new way of implementing the neuromorphic VLSI for the primary visual cortex, inspired by the ideas on the primary visual cortex by Hubel and Wiesel's experimentation and the neurophysiological model of neuron by Hodgkin and Huxley [3]. The design motivation is from the well-known experimentation of simple cell in Fig 1. by Hubel and Wiesel.

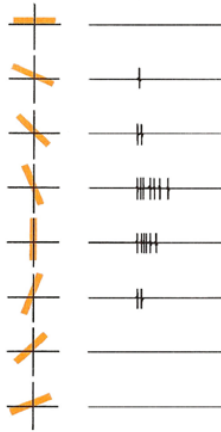


Fig. 1. Response of the cat's cortex to shining a rectangular slit of light, in various orientations [1]

The experimentation of spike burst with static line is aimed to mimic, while there is another experimentation of complex cell based on moving stimulus by Hubel and Wiesel.

The controlled conductance by CMOS transistors is used as an element of our neuromorphic system, which have been studied for biologically plausible analog-mixed neural networks VLSI [4, 5].

2.1 Neuromorphic Neuron Based on Voltage-Controlled CMOS Conductance

The Hodgkin-Huxley (H-H) formalism is a widely adopted idea of neuron's biophysical characterisation and dynamics. An electrical equivalent circuit model of Fig. 2 (a) is known as the empirical model of H-H formalism, which describes quantitatively the dynamics of the voltage-dependent conductance. Although most of particular neural networks tasks do not exhibit any major advantages based on H-H formalism, asynchronous spikes are considered as a principle element of high level or large scale neural computing system. The H-H formalism is widely of interest for its biophysical dynamics, though its complexity in computation is prohibitively high. Hence asynchronous dynamics of the H-H formalism is adopted as the idea of neuron model.

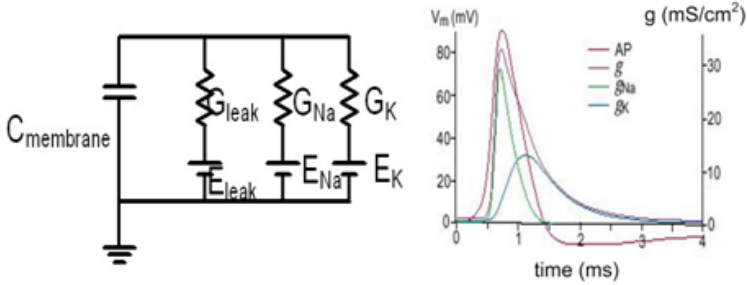


Fig. 2. (a) An electrical equivalent circuit of a neuron, Hodgkin-Huxley formalism (b) dynamics of asynchronous spike and refractory period vs. the membrane potential [3]

The overall dynamic behaviour of biological neuron is illustrated by the ion-based conductance controlled by membrane potential (or Action Potential), with the illustrated dynamics of conductance in Fig. 2(b) [4].

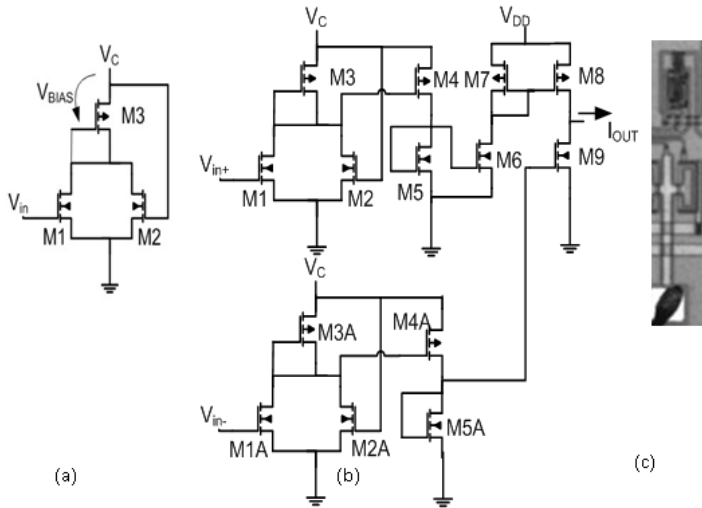


Fig. 3. (a) Voltage-controlled linear conductance by a pair of MOSFETs in the triode region, (b) the tunable linear transconductance circuit, (c) the chip photograph of CMOS transconductor

A circuit of Fig 3 (a) was proposed as a voltage-controlled linear conductance circuit by a PMOS transistor and a pair of identical NMOS transistors M1 and M2, while the conductance of MOS transistors is one of essential components in the analogue circuit design. The circuit of Fig. 3 has been investigated for various neural networks applications, from implementing synapses to neuron [4, 5].

The total drain current I_D of M1 and M2 can be expressed as

$$I_D = \alpha (V_{GS} - V_{BIAS} - 2V_{TH})(V_C - V_{BIAS}) . \tag{1}$$

where $V_{GS} = V_{inDC} \pm \Delta V_{in}$ is the gate-source voltage of transistor M1, $V_{DS} = V_C - V_{BIAS}$ is the drain-source voltage of transistors M1 and M2, and V_C is the tuning voltage of transconductance.

As the repeated two circuit in Fig. 3(b) operate in the same condition, the output current I_{OUT} is

$$I_{OUT} = Gm \Delta V_{in} \tag{2}$$

where $I_{OUT} = I_{D3} - I_{DA}$. The transconductance circuit of Fig. 3(b) can be used as a programmable conductance of neuron’s ion-channel or a synaptic connection with pulse/spike inputs.

2.2 Spiking Neuromorphic Circuit Mimicking the Primary Function of Visual Cortex

The tuning properties of orientation selectivity have been believed to play the key role for perception in visual cortex. As shown in Fig. 1, the tuning of specific neurons to the orientation of visual stimulus probably depends on the tuning features after passive or active learning for the earlier processing of natural image. The rule we assume is very simple as illustrated in Fig. 4, though some modifications are likely necessary for being more plausible to the natural system. Here, the tuned feature map (or connection) of 5×5 synaptic weights is based on the reference stimulus to match, with the minor adjustment depending on the output. The tuned feature map of vertical orientation is illustrated in Fig. 4(b), with the synaptic connections to 24 neurons (visual sensor, equivalent to a pixel) and itself. The synaptic weights of Fig. 4(b) are in the ration of (1: -0.6: 0.1 for black : grey : white). The six types of input stimulus (50×50 pixels) are experimented with the feature map (as synaptic connections) and spiking neurons based on H-H formalism.

The VLSI neuron is implemented by CMOS transconductance circuit of Fig. 3 in $0.18 \mu m$ CMOS technology. The spike burst output of Fig. 5 is observed by SPICE simulation, where the neuromorphic visual cortex mimic the biological spike burst of Fig. 1 from the experimentation work of Hubel and Wiesel.

The feasibility of bio-inspired neuromorphic system is demonstrated with its plausibility to primary simple cell function of visual cortex as exhibited in Fig. 5. The tuned feature characteristics of other orientations ($-45^\circ, 25^\circ$) are evaluated with the consistent outcomes as expected in the original experimentation of Fig. 1.

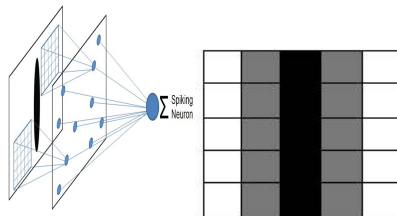


Fig. 4. The artificial primary visual cortex model with orientation selective synaptic weights to mimic the simple cell

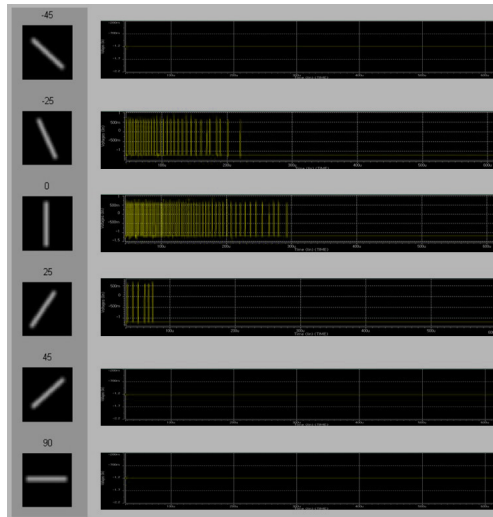


Fig. 5. The simulated spike burst of VLSI visual cortex to stimulus in various orientations

3 Bio-inspired Visual Information Processing and Application to Object Detection

The visual signal processing, particularly for vision application has been widely investigated, however, there is always a challenge such as the robustness to environmental change. The proposed bio-inspired processing is applied to example cases for evaluating its feasibility, as animal or human usually has the reasonable robustness. The previous research demonstrated the robustness to a certain application of object detection, i.e. the vehicle license plate detection. The license plate detection was investigated for the flexible detection based on the rectangle with the right angle, regardless of the aspect ratio or the whole size. It is based on the particular selective response to orientation at the right angle, i.e. presenting both components of horizontal and vertical orientation of the still image. It demonstrated the robust detection under some environmental interference such as the shiny reflection from the nearby area of license plate, in addition to the different sizes from the various distance [5]. In this paper, the application of multi-directional selectivity on the video information is investigated further for the detection of outdoor fire and human pose.

3.1 Human Pose Detection Based on the Edges of Multi-Direction

The visual perception processing is the one with the fast growing interests, with the emerging demands in applications of human detection or intelligent man-machine interface. In this paper, the neuromorphic vision system is evaluated for detecting

human or human pose based on the video image. There are two major principles employed – first, the head of human has the high density of orientation components regardless of front view or side view. The other principle is the head linked to torso, while the arm is also linked to torso. Example cases are illustrated in Fig. 6, where the multi-directional orientation components (based on Fig. 4) are used to prepare the orientation feature [6].



Fig. 6. Detection of passengers in different poses based on the multi-directional edges

The image of Fig. 7 illustrates the complex orientation features, which are caused by both the poor illumination condition and complicated environment. The orientation features of test image shows the complicated non-human objects, such as the glossy pattern of marble floor, reflection on the mirror like floor, wall-mounted art work and etc. In fact, the outside wall of the building is the transparent glass which causes the continuous and substantial changes in the illumination level and direction. Those features from the environmental cause is reduced by taking the difference of extracted orientation features of two video frames, as the background of two video frames has the similar features. The noise characteristics in the shadow area on the floor still remain in the reference features by two video frames. The application of neural net head detector generates the neuromorphic visual image, which has the high lightened area of human head. The equivalent of action potential locates the human head in the image, based on the winner neuron. The head area is successfully detected, while the head detection is successful throughout the entire video from appearing to disappearing. During the video experimentation of human head detection, the pose detection is evaluated with the movements of right arm. The pose detection is motivated for the alternative user interface of TV like haptic input to smart phone, which allows the unconstrained natural environment. The additional process is to introduce another neural network detector to characterize the arm, torso and background. The prior head detection can determine the region of interest as the arm is attached to the torso, as the head is also attached to the torso.

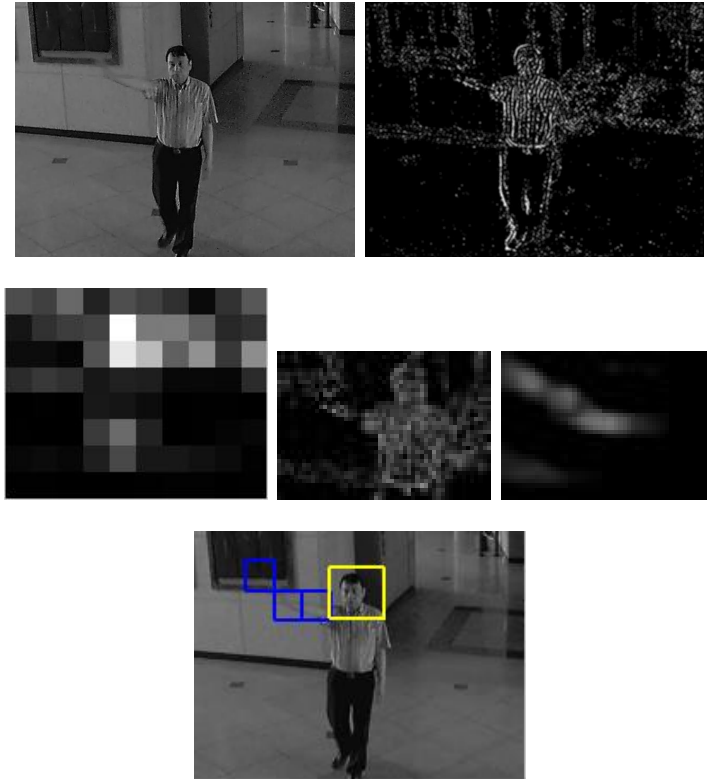


Fig. 7. Detection of human pose based on the neuromorphic vision: top-left to bottom-right, object video image, difference of detected edge components from two video frames, Action level of detection neuron, identified region of interest for detecting the pose of arm, neuromorphic visual image with the neural network arm pose detector, detected human head (yellow box) and arm pose (blue box)

The detected objects in Fig. 7 illustrate the successful feasibility of proposed pose detection, while the test on the entire video process shows the successful detection of arm when posed.

4 Conclusion

The bio-inspired neuromorphic vision is proposed as a feasible way of implementing the electronic hardware mimicking the primitive function of visual cortex, with application examples of object detection of various human pose. The example cases are successfully demonstrated by neuromorphic processing with the neural network detector or histogram detector. The neuromorphic circuit design is based on the linear controlled conductance of CMOS transistors, and is feasible for intelligent image sensors with the primitive visual cortex function by bio-plausible neurons of 10um x 10um in 0.18um CMOS technology.

The successful detection of outdoor fire or human pose is demonstrated with the simple bio-inspired principles, and the feasibility of neuromorphic vision is exhibited for various applications in the limited operation environment.

Acknowledgments. This research was supported by the grants from LG U+ and NRF.

References

1. Hubel, D.H., Wiesel, T.N.: Receptive Fields of Single Neurons in the Cat's Striate. *J. Physiol.* 148, 574–591 (1959)
2. Riesenhuber, M., Poggio, T.: Hierarchical Models of Object Recognition in Cortex. *Nature Neuroscience*, 1019–1025 (1999)
3. Hausser, M.: The Hodgkin-Huxley Theory of Action Potential. *Nature Neuroscience Supplement* 3, 1165 (2000)
4. Han, I.S.: Mixed-signal Neuron-synapse Implementation for Large Scale Neural Networks. *J. Neurocomputing*, 1860–1867 (2006)
5. Han, W.J., Han, I.S.: Bio-inspired Visual Signal Processing Based on Neuromorphic Circuit. In: 10th WSEAS International Conference on Instrumentation, Measurement, Circuit and Systems, pp. 131–136 (2010)
6. Han, W.J., et al.: Bio-inspired Neuromorphic Vision by CMOS Conductance for Intelligent Monitoring System of Passenger Emergency. In: Proc. ITS 2010, TP077-1 (2010)
7. Yilmaz, A., Javed, O., Shah, M.: Object Tracking: A Survey. *ACM Computing Surveys* 38, 1–45 (2006)
8. Yu, G., Slotine, J.: Visual Grouping by Neural Oscillator Networks. *IEEE Tr. Neural Networks* 20, 1871–1884 (2009)

Firing Pattern of Default Mode Brain Network with Spiking Neuron Model

Teruya Yamanishi¹, Jian-Qin Liu², and Haruhiko Nishimura³

¹ Fukui University of Technology, Fukui 910-8505, Japan
yamanisi@fukui-ut.ac.jp

² National Institute of Information and Communications Technology,
Kobe 651-2492, Japan

³ University of Hyogo, Chuo-Ku, Kobe 650-0047, Japan

Abstract. Recently, analyses of fMRI data have revealed functionally connected and interacting spontaneous active regions in the brain, which are referred as "Default Mode Brain Network". The fluctuations on BOLD signals of the default mode brain network have shown spatiotemporally correlated synchronization at a rate lower than 0.1 Hz in contrast to signals under concrete tasks like high frequency rhythms. Here we construct the default mode brain network by functionally connecting a neural network using functional correlation factors. For numerical simulations with Izhikevich's spiking neuron model, the condition on the slow synchronization of this network model is fixed, and the network dynamics is analyzed.

Keywords: spiking neuron model, default mode brain network, spontaneous activity, firing rate, synchronization.

1 Introduction

The functional magnetic resonance imaging (fMRI) has been yielding substantial results for the brain science. The visualization of active regions in the brain by fMRI is expected to reveal not only functions of the brain but also relations between the brain and the mind. For instance, how does one make a choice when two thoughts emerge? fMRI implied that both regions involved in abstract reasoning and those that process emotions get active in the brain under this situation. Greene and his group found using fMRI that if the dilemma is not so personal, the reasoning part of the brain is dominant when processing a difficult and personal moral dilemma [1].

Thus fMRI makes regions corresponding to any task light up, and also one more possible of fMRI on analyzing the brain is recently presented. It is observation for the slow fluctuations in the blood oxygen level dependent (BOLD) signal in the default or resting state of the brain for dozing situation [2–5]. The default or resting state means that someone does not focus on the external environment, namely be in the absence of an explicit task. The fluctuations have shown spatiotemporally correlated synchronization at a rate lower than 0.1 Hz.

These analyses of fMRI have revealed functionally connected and interacting spontaneous active regions in cerebral cortex, which is so-called "Default Mode Brain Network" and distinct from other systems within the brain. Being the activity in the default, the default mode brain network is a brain system much like the motor or visual systems and/or is associated with the ability of cognitive processing, development, aging, consciousness and psychiatric/neurological diseases. Then, some regions of the brain are in a system that has the spontaneous cognition and other memory-related functions from evidences in brain imaging of fMRI and anatomy [5]. The resting state, but these regions are active, is regarded to be related to the psychological function of the brain such as autobiographical memory. Integration of the neural activities on two states, namely the resting and the task-induced, tells us the entire story of the intrinsic organization of the dynamic networks of the brain that is formulated by anti-correlated signals [2]. Considering the fact that the brain is regarded as a dynamical system, the resting state of the brain is defined as the baseline. Therefore, the resting state is used as a condition of experiments compared with the task positive state, and the knowledge on the default mode brain network from multiple disciplines ranging from psychology to brain imaging technique is important for us to completely understand the brain's function at a systematical level.

2 Default Mode Brain Network and Its Modeling

The default mode brain network has been established as a new field highlighted in recent years though its history is short – just about a decade [5]. Of course, the related fundamentals in neuroscience leading to the advances of the default mode brain network research can be traced to 1970s [5]. In the early era of the fundamental research, the form of the image is rCBF (regional cerebral blood flow) that is obtained by using the nitrous oxide technique. The signals in this kind of images reflect the neural activity of the frontal area of the brain [5]. Later, other brain imaging techniques including PET (positron emission tomography), fMRI and other techniques are used, and the resolution of the obtained images becomes higher.

In informatics, the correlation network of the default mode brain network is verified by the evidences in neuroscience. The graphical representation of the default mode brain network consists of different nodes of the regions of the brain that are functionally connected and is formulated as an interaction of sub-systems. This kind of structure is expected to help us to explain how different parts of the brain perform a cooperated function. The intrinsic (spontaneous) activity of the resting state of the brain measured from BOLD signals demonstrates the brain's doing without any input/output. Table 1 is functional connecting core regions suggested as architectonics of the default mode brain network.

Currently, two aspects – the functions of the default mode brain network in neuroscience (e.g., neurophysiology) and pattern analyses of correlated signals (e.g., BOLD) [4] supported by related means of signal processing and data analysis (e.g., handling the noisy case [6]) – are the main streams of the research on

Table 1. Core regions of the default mode brain network. The corresponding brain areas generally imply Brodmann's areas from Refs. [2,5].

Regions	Abbreviation	Corresponding brain areas
Lateral temporal cortex	LTC	21
Medial prefrontal cortex	MPFC	24, 32ac, 10m/10r/10p, 9
Inferior parietal lobule	IPL	39, 40
Posterior cingulate cortex/restrosplenial cortex	PCC/Rsp	23/31, 26,29,30
Parahippocampal cortex	PHC	36
Hippocampal formation	HF	—

the default mode brain network. With the advanced of brain imaging techniques offering satisfactory resolution in space and time, the quantitative analyses of the default mode brain network in spatial and temporal dynamics become possible. For example, the stability of the default mode brain network for aging brain [7] is useful for modeling the temporal dynamics of one. Accordingly, it is a promising theme to explore the complete structure of the default mode brain network.

However, there is a problem in going forward it. It is how we can model a causality network from the correlation network based on theoretical models of nonlinear dynamics. So, we carry out simulation experiments on the default mode network by constructing a functionally connected system composed of eleven nodes with Izhikevich's spiking neuron model in order to solve this question. Basing on the spiking model proposed by Izhikevich [8,9] that explains eight patterns of neuron spikes with four variables and reproduces firing patterns exactly, we construct a functionally connected system composed of eleven nodes that corresponds to the regions of the brain, which is regarded as the default mode brain network here. Simulation experiments are carried out to investigate systematically the default mode brain network of the brain where firing patterns are mapped to neuron assemblies, and we attempt to shed new light on the root of dynamical systems of the brain.

3 Simulation Model and Method

Based on the functionally connected system of the brain in Refs. [2,5], we assume that the cortex region is composed of the neuron assembly formed by placing 160 excitatory neurons and 40 inhibitory neurons in one dimension, and construct a new neural network with the spiking neuron model for the default mode brain network. The ratio of the excitatory neuron's number to the inhibitory one in the region is fixed to follow that of anatomical insight for the human brain. In addition, the neural network takes into account 2 different intrinsic properties as follows:

– Hierarchic neuroanatomical connectivity structure

- Neuron assembly in each region

The network is taken to be randomly coupled with the strengths.

- Collective regions in clusters

The coupling strength between each region is given as the functional correlation one with the default mode brain network, which is named an *inter*-connection here. On the other hand, we call an *intra*-connection the coupling strength between each neuron.

The conceptual diagram of new neural network with this hierarchic structure is shown in Figure 1.

– Time delay in the transmission of information

The delay is simply considered as temporal transmission, and obtained by a common propagation velocity and 3D Euclidean distance between any 2 different regions. Then, we take the velocity as a parameter v , and estimate the distance from the typical locations of the regions in 3D space. Table 2 is the distance calculated using the human data.

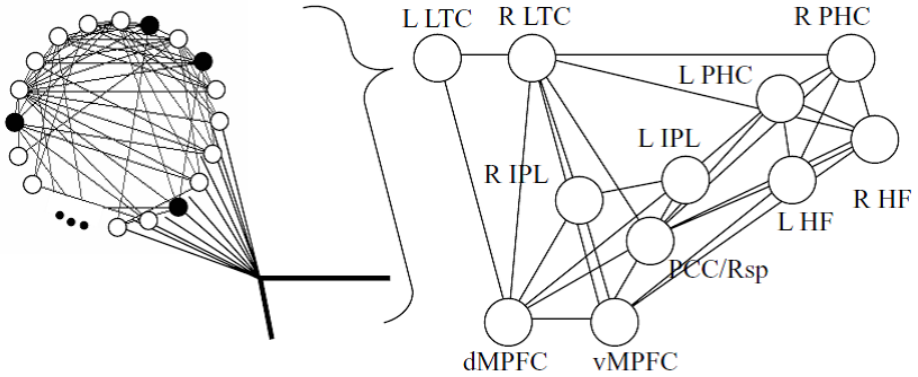


Fig. 1. A network model with the spiking neuron model on the default mode brain network is illustrated. Each region has 200 spiking neuron with intra-connections, and is functionally connected at correlation strengths from anatomical analyses.

Considering these intrinsic properties, the membrane potential of the i th neuron in the M -region with the synaptic current I_{Mi} is given by

$$\frac{dv_{Mi}(t)}{dt} = 0.04 v_{Mi}(t)^2 + 5 v_{Mi}(t) + 140 - u_{Mi}(t) + I_{Mi} + \sum_{Mj=\text{Firing}} s_{MiMj} + \alpha \sum_{N(N \neq M)} C_{MN} x_N(t - \tau_{MN}), \quad (1)$$

$$\frac{du_{Mi}(t)}{dt} = a_{Mi} (b_{Mi} v_{Mi}(t) - u_{Mi}(t)). \quad (2)$$

Table 2. 3D Euclidean distances between each region. The values are obtained by computing on the coordinate system of the Montreal Neurological Institute.

Nodes	L LTC	R LTC	dMPFC	vMPFC	L IPL	R IPL	PCC/Rsp	L PHC	R PHC	L HF	R HF
L LTC	—	120	106	78.1	72.5	126	67.3	36.3	89.6	39.0	82.5
R LTC		—	106	78.1	125.8	72.5	84.7	89.6	36.3	82.5	39.0
dMPFC			—	51.1	134	134	107	103	103	91.5	91.5
vMPFC				—	120	120	88.0	71.9	71.9	51.6	51.6
L IPL					—	88.0	41.4	57.9	91.0	82.2	103
R IPL						—	60.4	91.0	57.9	103	82.2
PCC/Rsp							—	36.4	50.6	55.9	64.0
L PHC								—	56.0	25.1	55.6
R PHC									—	55.6	25.1
L HF										—	44.0
R HF											—

s_{MiMj} , C_{MN} , and α in equation (II) are taken as the strength of intra-, inter-connection, and the parameter of the coupling strength for the inter-connection, respectively. Also, x_N is the firing rate of the region N , and τ_{MN} is the time delay from the region N to M [10]:

$$\tau_{MN} = \frac{L_{MN}}{v} , \tag{3}$$

where v and L_{MN} are the velocity as parameter and typical distance listed in Table 2, respectively. For simulation experiments, we use the result of hubs and subsystem within the default mode brain network mapped and estimated from functional connectivity analysis as C_{MN} in equation (II). As a result, our each region now is regarded as in the functional connectivity region in the brain. The values of C_{MN} are listed in Table 3. In addition, other parameters in equations (II)–(2) are listed in Table 4, and s_{MiMj} mentioned between neurons are randomly assigned values between 0 and 0.5 in the case of coupling with an excitatory neuron and between -1 and 0 for an inhibitory neuron.

Analyzing the fluctuations of the synchronization for neuron firing patterns calculated by equations (II)–(2), we define an order parameter for the region M [10]:

$$K_M = \frac{K'_M(t_f) - \langle K'_M(t_f) \rangle}{\langle K'_M(t_f) \rangle} , \tag{4}$$

where $\langle \rangle$ denotes the average over time, and

$$K'_M(t_f) = \langle |\Sigma_{i \in M} F_i(t) - \langle \Sigma_{i \in M} F_i(t) \rangle| \rangle , \tag{5}$$

with $F_i(t)$ representing the neuron firing i in the region M at the time t .

Table 3. Functional correlation strengths of each region within the default mode network [5,11]

Nodes	L LTC	R LTC	dMPFC	vMPFC	L IPL	R IPL	PCC/Rsp	L PHC	R PHC	L HF	R HF
L LTC	1.00	0.41	0.16	0.12	0.14	0.12	0.12	0.11	0.06	0.18	0.14
R LTC		1.00	0.16	0.18	0.07	0.20	0.19	0.08	0.10	0.15	0.17
dMPFC			1.00	0.47	0.22	0.31	0.34	-0.06	-0.10	-0.01	-0.04
vMPFC				1.00	0.27	0.31	0.52	0.11	0.06	0.20	0.16
L IPL					1.00	0.47	0.49	0.25	0.10	0.11	0.06
R IPL						1.00	0.42	0.12	0.05	0.09	0.07
PCC/Rsp							1.00	0.23	0.16	0.26	0.21
L PHC								1.00	0.57	0.31	0.28
R PHC									1.00	0.28	0.28
L HF										1.00	0.61
R HF											1.00

Table 4. Parameter values of the Izhikevich neuron model used in our simulations

Parameters	Excitatory neuron	Inhibitory neuron
Time scale a	0.02	0.02 ~ 0.10
Sensitivity b	0.20	0.20 ~ 0.25
Resting membrane potential c	-65 ~ -50	-65
Inactivity period d	2 ~ 8	2

4 Simulation Results and Progress of Research

Using the values of C_{MN} at Table 3, we examined the fluctuation in equation (4) and the power spectrum in the sliding time window of 500 ms shifted by steps of 500 ms for each region. For randomly connected neurons in each region, the result is shown in Figures 2 and 3 in case with the parameter of the coupling strength for the inter-connection $\alpha = 76$ in equation (1) and the velocity $v = 10$ mm/ms in equation (3). We find weakly synchronized firing patterns temporally for each region, and obtain the peak of the power spectrum for the fluctuation with the rate lower than 0.1 Hz. As the resting state is considered as a doze condition without strongly synchronized patterns like the alpha and/or gamma rhythms, one see that our model has correct results for their patterns. By changing the synaptic current I_{Mi} to large values, the firing patterns become to be strongly synchronized both spatially and temporally for each region, but the fluctuation with the rate lower than 0.1 Hz is relatively small.

Now, simulation experiments on the power spectrum of communities consisting of core regions indicated by the strength of the functional correlation are

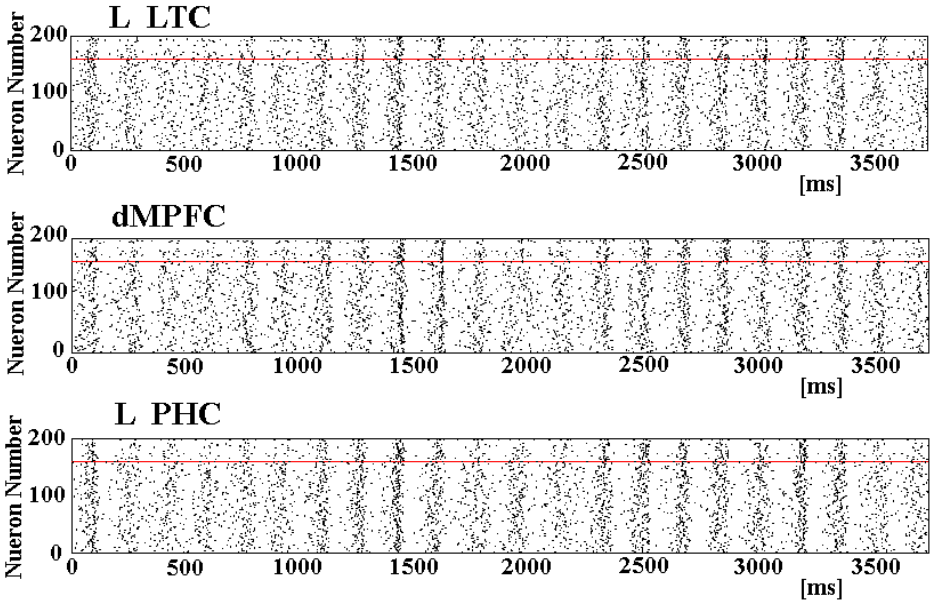


Fig. 2. Simulation results of the raster plots for L LTC, dMPFC and L PHC regions as typical cortices with $\alpha = 76$ and $v = 10$ mm/ms in equations (1)–(3). The firing excitatory (inhibitory) neurons are drawn in area below (above) the line at 160 in each graph.

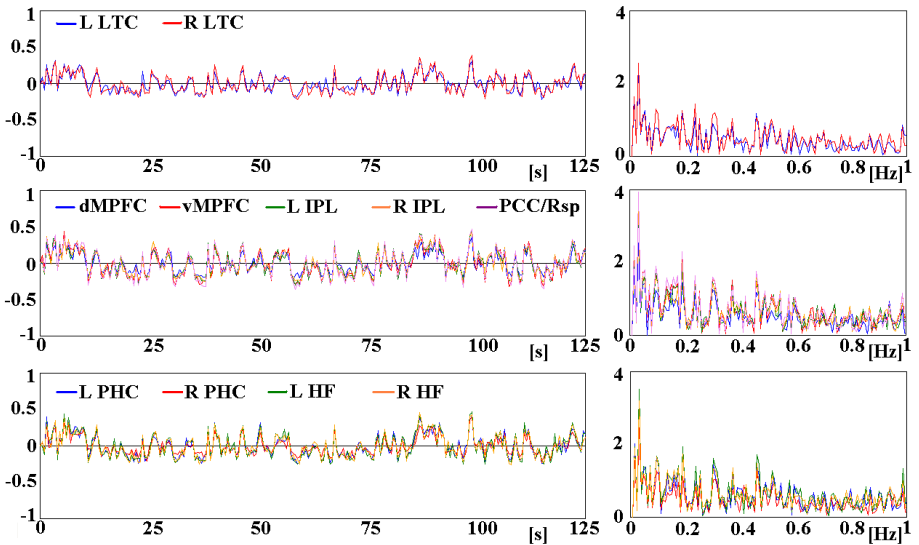


Fig. 3. (Left) Results of the synchronization order parameter K_M in equation (4). (Right) Power spectrum on each region with $\alpha = 76$ and $v = 10$ mm/ms.

simultaneously carried out, because the slow fluctuation of the default mode brain network is not observed at each core region, but at functional collective region constructed by core regions. Also, we are estimating the dependence on the amount of neuron to the power spectrum of the fluctuation.

References

1. Greene, J.D., et al.: The Neural Bases of Cognitive Conflict and Control in Moral Judgment. *Neuron* 44, 389–400 (2004)
2. Fox, M.D., et al.: The human brain is intrinsically organized into dynamic, anticorrelated functional networks. *PNAS* 102, 9673–9678 (2005)
3. Mason, M.F., et al.: Wondering Minds: The default Network and Stimulus-Independent Thought. *Science* 315, 393–395 (2007)
4. Fox, M.D., Raichle, M.E.: Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. *Nature Reviews of Neuroscience* 8, 700–711 (2007)
5. Buckner, R.L., Andrews-Hanna, J.R., Schacter, D.L.: The Brain’s Default Network: Anatomy, Function, and Relevance to Disease. *Ann. N. Y. Acad. Sci.* 1124, 1–38 (2008)
6. Gaab, N., Gabrieli, J.D.E., Glover, G.H.: Resting in peace or noise: Scanner background noise suppresses default-mode network. *Human Brain Mapping, Special Issue: Endogenous Brain Oscillations and Networks in Functional MRI* 29, 858–867 (2008)
7. Beason-Held, L.L., Kraut, M.A., Resnick, S.M.: Stability of Default-mode network activity in the aging brain. *Brain Imaging Behav.* 3, 123–131 (2009)
8. Izhikevich, E.M.: Simple Model of Spiking Neurons. *IEEE Trans. Neural Networks* 14, 1569–1572 (2003)
9. Izhikevich, E.M.: Which Model to Use for Cortical Spiking Neurons? *IEEE Trans. Neural Networks* 15, 1063–1070 (2004)
10. Deco, G., et al.: Key role of coupling, delay, and noise in resting brain fluctuations. *PNAS* 106, 10302–10307 (2009)
11. Vincent, J.L., et al.: Coherent spontaneous activity identifies a hippocampal-*parietal* memory network. *J. Neurophysiol.* 96, 3517–3531 (2006)

Program Equivalence Using Neural Networks

Tiago M. Nascimento^{1,2}, Charles B. Prado¹, Davidson R. Boccardo¹,
Luiz F.R.C. Carmo¹, and Raphael C.S. Machado¹

¹ INMETRO - National Institute of Metrology, Normalization and Industrial Quality
Rio de Janeiro - Brazil

² UFRJ - Federal University of Rio de Janeiro - Brazil

{tmnascimento, cbprado, drboccardo, lfrust, rcmachado}@inmetro.gov.br

Abstract. Program equivalence refers to the mapping between equivalent codes written in different languages – including high-level and low-level languages. In the present work, we propose a novel approach for correlating program codes of different languages using artificial neural networks and program characteristics derived from control flow graphs and call graphs. Our approach correlates the program codes of different languages by feeding the neural network with logical flow characteristics. Our evaluation using real code examples shows a typical correspondence rate between 62% and 100% with the very low rate of 4% false positives.

1 Introduction

Equivalence between two programs can be characterized by their executing behavior. The behavior equivalence can be established between distinct languages – including high-level and low-level languages. The *equivalence problem* refers to, given two programs, deciding if they present the same executing behavior or not. In the following, we present some scenarios of Software Engineering concerned with the equivalence problem.

1. **Platform migration.** Consider a developer who, for historic reasons, has a small part of a certain system written in a distinct programming language from the rest of the system. For maintainability reasons, this developer may track the homogeneity of the system by rewriting the routines in the predominant language of the system. One way to determine whether the “rewritten routines” were properly coded is to verify if they are equivalent to the “older routines”.
2. **Legacy software recovering.** Consider a developer who has a software system in production that, due to difficulties in its software configuration management, needs to “recover the baseline” of the software system in production, *i.e.*, for each software module that has a binary code in execution, the developer must identify among several versions of source code, which corresponds to that binary in execution. One possibility for this task could be recompiling the distinct source code modules, comparing the generated binaries with the ones in the production environment. This approach however, besides being impractical – given the huge amount of versions and

compilations that should be conducted, still has the possibility of failure in the case that the setting parameters are not exactly the same as those used in the compilation of the modules currently in the production.

3. **Introduction of non-intentioned behavior in the compilation process.** This scenario refers to the fact that the compilation process is a part of the software development and must be validated. In fact, the compilation process may introduce *bugs* and non-intentioned behavior in the software [1]. Besides, the software code and the compiler may be deliberately corrupted by the insertion of a malicious behavior (backdoor). Hence, it becomes interesting to have tools to directly compare the behavior described by source and binary code, independently of the compilation process.
4. **Software acquisition management.** Assume that a software manufacturer wishes to outsourcing the development of some libraries in a project. For so, the manufacturer gives some specifications to an independent developer, which returns the specified product developed. Naturally, such a product should be submitted to a battery of tests in order to characterize it according to the specifications. However, these tests normally are ineffective with hidden malicious behavior. Such a behavior is typically activated by the insertion of hidden undocumented commands — for instance, by using a counterintuitive sequence of keystrokes [2]. The detection of hidden behavior only can be done through of code analysis. In this case, the manufacturer may require not only the binary code but also the associated source code. Once the source code be analyzed, further step requires the mapping (equivalence) between the source and the binary code.

In the previous examples, we see that frequently the equivalence problem refers to the mapping of a binary code from a source code or of a binary code from another binary code, which we term “traceability”. A simple and direct way of performing such “traceability” is to reproduce exactly the development environment of the software developer and to compile the source code, verifying whether the generated binary code is as expected. For this approach we must assure that the compilation environment is the same. Such a hypothesis, however, may be impractical — as the case (2), in which the compilation settings may be lost due to flaws in the configuration management, or as the case (4), which would restrict the developer to a unique environment. Another drawback is related to the cost and the complexity required to keep several software development environments. Moreover, as demonstrated in the Thompson Turing Award Lecture [3]: unless the “language transformation” performed by the compiler can be completely characterized — which would require a binary analysis of the compiler code — it is not possible to guarantee that the compilation process, itself, does not introduce some kind of flaw or malicious behavior into the software (this is closely related to the example in case (3)). Another way of performing the software traceability is to audit the software development environment of the software developer. Such an approach presents disadvantages similar to those described

¹ These sequences of commands are sometimes called *easter eggs*.

in the previous paragraph, and it is likely to be ineffective when dealing with a malicious developer.

Summarizing, binary code verification is the only true way to detect hidden capabilities, as demonstrated by Thompson in his Turing Award Lecture [1]. Lest Thompson's paper be considered theoretical, his ideas have been put into practice by the malware W32.Induc.A [2]. On large and complex systems, however, binary verification can require long and laborious work to integrally track variable manipulations and to perform vulnerability analysis, so that a usual approach is to conduct such verification on the source code. Depending on the architectural complexity of the software, this software can be explicitly submitted to a white-box approach entailing a source code analysis. However, source code verification is not sufficient enough to give any guarantee about the behavior of the related binary code. To certify that the binary code being executed works properly, one must guarantee that such binary code was, in fact, generated from the approved source code through an honest compilation process.

Source code verification does not preclude the verification as to whether the binary code corresponds to its source code. Traceability of executable codes is the process for establishing the correspondence between source and object codes. That is, once the source code of a given software version is analyzed, evaluated and approved, it is necessary to verify whether a given binary code — which will be, in fact, in execution — corresponds to that source code.

In the present work we propose a different strategy to verify whether two programs of different languages (typically source code and binary machine code) describe the same software behavior. This paper presents a novel approach for performing program equivalence of program codes of different languages by using artificial neural network (ANN). More specifically, we collected properties of the control flow graphs and call graph, such as number of edges, number of nodes and number of functions, and we used an ANN to discover the degree of similarity of the program languages based on the collected properties.

The rest of the paper is structured as follows. Section 2 discusses the related works on program equivalence. Section 3 describes our proposed method by characterizing the properties extraction of program languages and by showing the application of an artificial neural network to discover the degree of similarity of the program languages. Section 4 presents the empirical evaluation of the method presented, followed by our concluding remarks.

2 Related Works

There are not many contributions related with program equivalence in the literature. However, there were works for verifying and analyzing of source and binary codes that may assist in achieving the proposed approach.

Quinlan *et al.* [3] proposed a framework for software defects verification (binary or source). However, it does not compare source and binary codes. Hassan *et al.* [4] observed that the architecture of some programs is intrinsically related with the their source and binary codes. They used two types of extractors:

a transfer control extractor of a code binary (LDX) and a label extractor of a C source code (CTAGX). After the extraction process, they conducted a comparison of the obtained results in order to infer the software architecture. Hatton [5] investigated the defect density as a relationship between a binary code and a source code. For so, he used the size (number of rows) of the source code and its defect per 1,000 lines to seek the relationship with the binary code. Neither of these works addresses the program equivalence problem.

Buttle [6] utilizes the program logical structure (control flow graph) of the binary code to match with the program logical structure of the source code. We also use program flow characteristics to match binary codes, however, in our approach other relationships that may coexist between source and binary codes in order to obtain a better matching were considered.

On a tangential direction there has been significant work in binary differing with the intent to review sequential versions of the same piece of software, to analyze malware variants of the same high-level language and to analyze security updates [7,8]. Most of these works use graph matching to compare the binaries. A good summary of these works may be found in [9], which also introduces anti-differing techniques with the intent to thwart algorithms based on graph matching. Research results from differing binaries [7,8,9], may thus be borrowed for the program equivalence problem for analogous constraints.

Some contributions in security use neural networks for cryptography approaches. A new digital image encryption algorithm using neural networks is presented in [10]. Such algorithm employs a hyper-chaotic cellular neural network using chaotic characteristics of dynamic systems.

Artificial intelligence based methods for software validation, verification and reliability can be found on the literature. The approaches in [11] and [12] propose the use neural networks for software reliability prediction. The former uses the prediction for software defects fix effort, while the second the prediction system is based on neural network ensembles. In [13], a neural network is used to predict a fault-prone module in a web application. In [14], a self-organizing system for reliability of modules is constructed.

The use of artificial neural networks was previously considered in [15], which described a method for verifying the correspondence between source and binary codes using artificial neural networks. The approach described in this paper improves upon that work by refining the extracted properties and by applying the method for program equivalence of distinct languages.

3 Proposed Approach

Our approach for program equivalence involves two steps. In the first step, we use software tools to extract characteristics of distinct program languages. Such characteristics could be simple ones, such as size, or more sophisticated ones, such as those derived from the control flow graphs or call graphs. In the second step, we use a nonlinear nondeterministic classifier to determine the correspondence of the program languages. In this section, we show how this approach was put in

practice with the use of four characteristics (size, number of procedures, number of nodes of control flow graph and number of edges of control flow graph) and an artificial neural network as classifier.

3.1 Extraction Process

In the compilation process there is a lot of lost information that should be taken into account when designing a program equivalence approach. The amount of available information of the compiled code is platform and compiler-specific. In the following, we mention some properties regardless of the source and binary codes, which may be explicitly or implicitly available, in order to give insights about the complexity of designing a program equivalence method.

Considering the fact that most embedded softwares are written in imperative language, properties such as variable names, variable types and procedure names may be lost during the compilation process since the compiler goal is to maximize the performance. This process normally decreases the legibility of the binary code, so representing low confidence for the mentioned properties to use in our program equivalence problem.

Data contents are not explicitly available in the source and binary codes. Nonetheless, these contents may be computed by data-flow analysis. The scope of the data-flow analysis for source and binary is faintly different. In the source code, the scope is at the variable level, meanwhile, in the binary code it is at memory and registers. This difference certainly increases the number of instructions contained in the binary in comparison to the respective number of the source code. Besides being positive for compiler data optimizations since it tracks fine-grained transitions, it requires more memory to analyze more code lines. Since the extraction of this property is complex, it is not suitable for our program equivalence problem.

The control sequentiality is certainly kept in the binary, albeit, its tree of execution is not clearly structured as such in the source code. A good summary of algorithms to structure the control sequentiality may be found in [16]. The control sequentiality describes the program logic of a certain code, and it can be characterized by call graphs and the individual function flow graphs. The call graphs show the caller-callee relationship. The individual function flow graphs represent the basic blocks and its flow of information based on conditional and unconditional branches.

The characteristics used in our program equivalence method are based on the program logic of the code (control sequentiality). These characteristics are number of nodes, edges and functions. The sizes in bytes of the codes were also used in our method. The sizes of the codes were obtained in a straightforward manner, however, the number of nodes and edges of the control flow graph for the source code were obtained by using a shell script that counted these properties. Such a control flow graph of the source code was built from the Gnu Compiler Collection with the parameter “fdump-tree-cfg”. For the extraction of the same characteristics from the control flow graph in the binary code, we used an idapython script over the IDA disassembler [17]. The number of the

functions for the source codes and binary codes were extracted from the call graphs, generated by the GNU cflow [18] tool for the source code and by the IDA disassembler for the binary code.

3.2 Artificial Neural Network

The fundamental point in program equivalence is to establish an association of the program languages by linking their intrinsic logical characteristics. For such, it is fundamental to find an efficient approach that combines the four different parameters extracted from logical program code (number of nodes, edges, functions, and the sizes of the codes (in bytes)).

At first analysis, some linear separation methods could be investigated to solve this problem. However, as will be shown in the further sections, this problem is both nonlinear and highly complex to solve using a linear method. For these reasons, we examine the use of artificial neural networks.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an “expert” in the category of information it has been given to analyze. Obviously, there are many kinds of Back-propagation networks applied to a large set of different problems, like: classification, recognition, prediction and others.

In this work, the main focus will be on neural networks applied to the classification problem. For that, we propose the use of Cascade-Forward Back-propagation Neural Networks, since they are widely used in this context [19,20,21]. In the next section, the details of neural network implementation will be presented.

Neural Network Implementation. To design a neural network, four characteristics (number of nodes, edges, functions, and the sizes of the codes (in bytes)) were fed into the input nodes of the one fully-connected cascade forward network using the Back-propagation training procedure [22]. From the empirical analysis, the best neural configuration was built with two neurons in the hidden layer. For the output layer, only one neuron was used (see Figure 1).

In order to simulate the neural network, the Matlab Neural Network Toolbox [23] was used. The selected activation function for the neurons was the hyperbolic tangent sigmoid. The target vector for the training phase was defined by establishing a target value of 1 when the input parameters represent equivalent codes (called class of true association), otherwise the target value was set to -1 (called class of false association).

4 Experimental Evaluation

We now present the results of an empirical evaluation for mapping equivalent codes written in distinct languages. Three evaluations were performed:

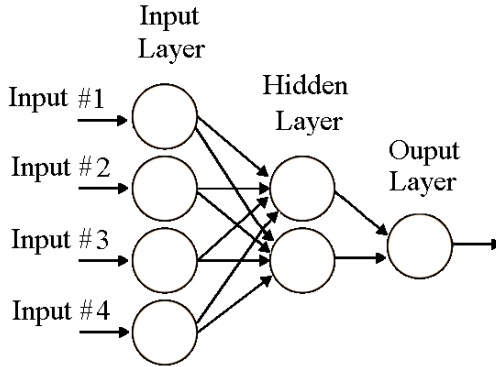


Fig. 1. Neural Network Topology

1) evaluation of malware modified binaries, 2) evaluation of binaries of different platforms and 3) evaluation of binaries generated from different compilers.

For all the evaluations, the percentage used for training phase was $\approx 70\%$ and the remaining samples were used to verify the capacity of generalization of the ANN. We studied the improvements of our method by comparing it against Support Vector Machine (SVM) — a technique typically applied to the construction of classifiers [24,25]. Our empirical evaluation shows that our method produces more precise results than SVM.

4.1 Evaluation of Malware Modified Binaries

Since most malwares are predominant for Windows environment, we performed the evaluation in this environment. Before starting to infect the system in order to extract the characteristics of the infected codes, we established some security policies to avoid malware dissemination. We setup an isolated machine for the malware infection and characteristics extraction. The extracted characteristics of the infected codes were used to build the set of false association.

For this evaluation, we collected 94 source codes, taken from [26,27] and compiled them using the gcc compiler of Windows platform. Figure 2 shows the training set of true association, *i.e.*, characteristics extracted from 70 pairs of correlated source and binary codes. The 24 remaining were utilized in the evaluation phase. The histograms show the distribution of the normalized variables: edges, nodes, functions and size, respectively, in the x-axis and their frequency in the y-axis. The control flow graph characteristics (edges and nodes) were fed into input #1 and input #2 of the ANN, respectively. The number of functions (extracted from the call graph) was fed into input #3, and the size into input #4. All the characteristics were correlated by subtracting the source code characteristic from the binary code characteristic except for the size characteristic, in which a division was applied. Before inputting such parameters into the ANN,

a normalization step was necessary since our ANN only accepts values in the interval $[-1,1]$. The normalization of the parameters was calculated by dividing all data of each parameter by the largest value of the same parameter.

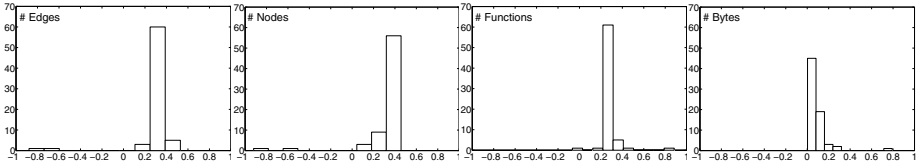


Fig. 2. Training set of the true association using correlated binary and source codes

The false association was created by infecting the 94 codes with four malwares (Virus.Cabanas.a, Virus.Win32.NGVCK.1003, Virus.Win32.Qudos.4250, Virus.Win32.Artelad.2173). Figure 3 shows the training set of false association using characteristics extracted from 280 infected samples (70 for each malware), in which the characteristics were correlated by subtracting the source code characteristic from the infected binary code except for the size characteristic, in which a division was applied. All the characteristics were normalized before inputting into the ANN. The 96 remaining infected samples were utilized in the evaluation phase.

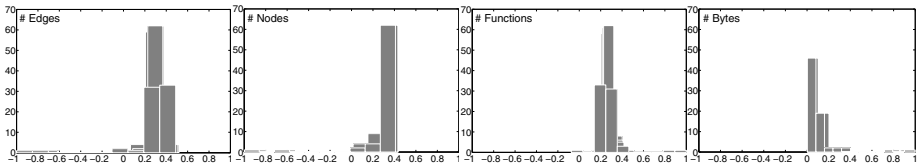


Fig. 3. Training set of the false association using malware modified binaries

Figures 4 and 5 show the sets utilized for the evaluation of the ANN. Table 1 shows the neural network results with respect to the number of hits and mistakes for both classes (true and false association). The data shows $\approx 4\%$ (4/96) false positives and $\approx 37\%$ (9/24) false negatives. The results of our approach are promising since the program equivalence is mainly concerned with a low rate of false positives.

Table 1. Neural network results of the evaluation of malware modified binaries

Class	True association	False association
True association	15	9
False association	4	92

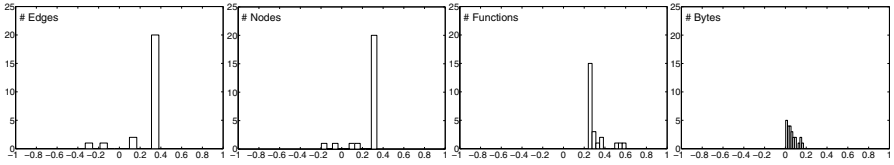


Fig. 4. Evaluation of the ANN using correlated binary and source codes

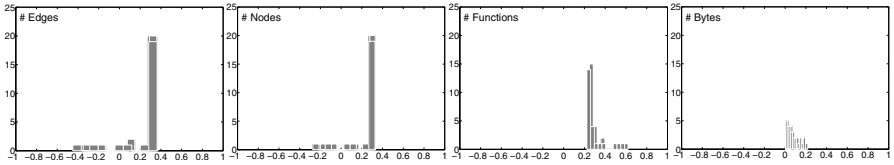


Fig. 5. Evaluation of the ANN using non-correlated binary and source codes

We compare the results of our proposed ANN with the results achieved using a Support Vector Machine (SVM). The classifier used was a two-norm and soft-margin SVM. The kernel function that maps the training data into kernel space was linear and the method to find the separating hyperplane was quadratic programming. However, the SVM did not converge to any separating hyperplane for the training sets exhibited in Figures 2 and 3. However, using only one malware (Virus.Cabanas.a) for the false association we found a division of the 4d-space — that is, a 3-d hyperplane that divides the 4-d space into two semi-spaces. In this experiment, the SVM correctly identified 19 among the 24 pairs of corresponding codes, with 7/23 ($\approx 30\%$) ratio of false positives. Observe that the SVM method achieved a reasonable accuracy regarding the identification of corresponding codes (exactly the same 19/24 as the ANN), but with a much higher number of false positives (7/23 against the 1/24 ratio of the ANN).

4.2 Evaluation of Binaries of Different Platforms

For this evaluation, we compiled 81 source codes using both gcc compiler on the Windows environment and gcc compiler on a Linux-like environment to build the true association. Figure 6 shows the training set of true association, *i.e.*, characteristics extracted from 60 pairs of correlated Windows binary and Linux binary. The 21 remaining were utilized in the evaluation phase. The normalization and extraction processes are analogous to the ones of the previous evaluation.

Figure 7 shows the training set for the false association, created naively by random values. Figures 8 and 9 show the sets utilized for the evaluation of the ANN. Table 2 shows the neural network results with respect to the number of hits and mistakes for both classes (true and false association). The data shows $\approx 9\%$ (2/21) false positives and zero false negatives. We used the same training sets exhibited in Figures 6 and 7 to feed the SVM. The SVM correctly identified

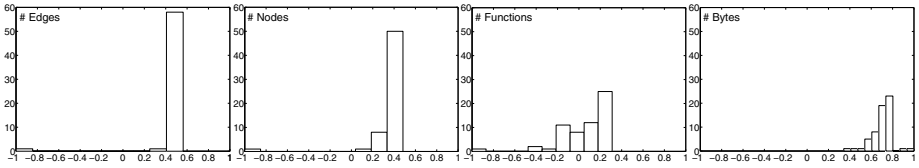


Fig. 6. Training set of true association using correlated binaries of different platforms

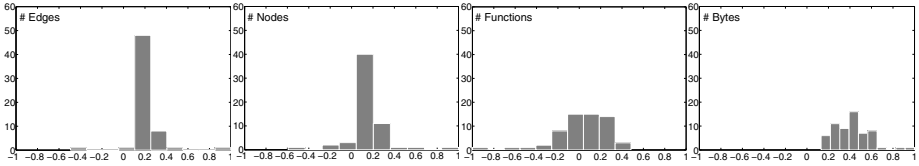


Fig. 7. Training set of the false association using non-correlated binaries of different platforms

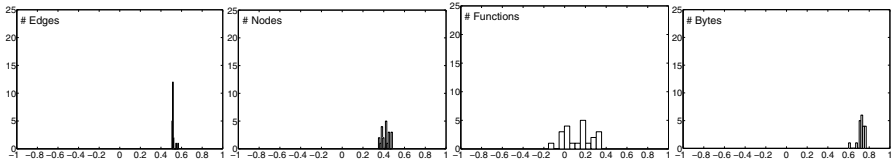


Fig. 8. Evaluation of the ANN using correlated binaries of different platforms

Table 2. Neural network results of the evaluation of binaries of different platforms

Class	True association	False association
True association	21	0
False association	2	19

all the 21 pairs of corresponding codes as the results of our ANN, with 3/21 ($\approx 14\%$) ratio of false positives against 2/21 ratio of our ANN approach.

4.3 Evaluation of Binaries Generated from Different Compilers

For this evaluation, we compiled 83 source codes using the gcc compiler and the Borland C++ compiler, both on the Windows environment. Figure 10 shows the training set of true association, *i.e.*, characteristics extracted from 63 pairs of correlated binary generated from the compilers above. The 20 remaining were utilized in the evaluation phase. The normalization and extraction processes are analogous to the ones of the previous evaluation.

Figure 11 shows the training set for the false association, created naively by random values. Figures 12 and 13 show the sets utilized for the evaluation of the ANN. Table 3 shows the neural network results with respect to the number of

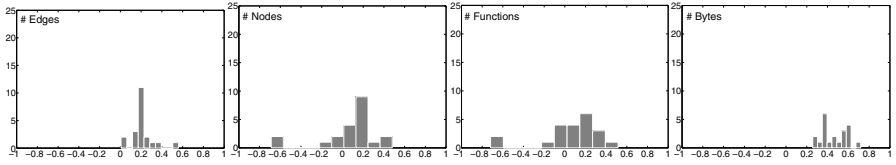


Fig. 9. Evaluation of the ANN using non-correlated binaries of different platforms

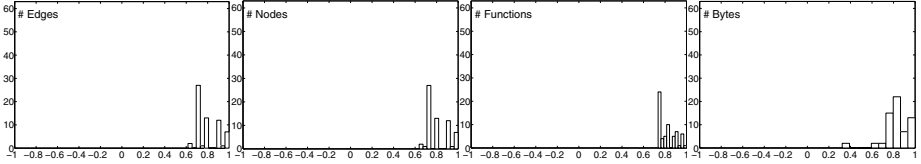


Fig. 10. Training set using correlated binaries generated by different compilers

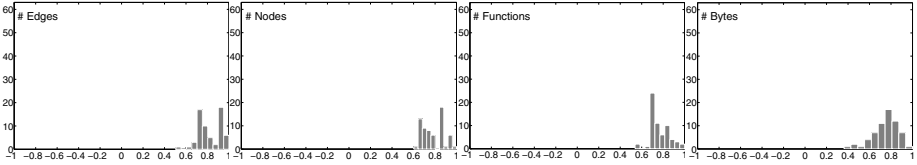


Fig. 11. Training set using different non-correlated compiled binaries

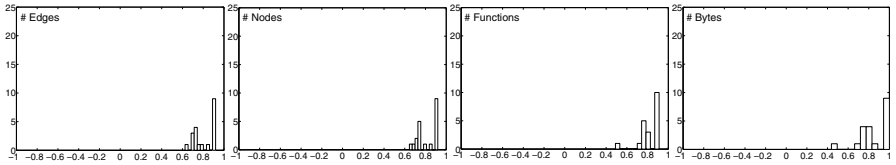


Fig. 12. Evaluation of the ANN using correlated binaries generated by different compilers

Table 3. Neural network results of the evaluation of binaries compiled with distinct compilers

Class	True association	False association
True association	19	1
False association	1	19

hits and mistakes for both classes (true and false association). The data shows 5% (1/20) false positives and 5% (1/20) false negatives. We used the training sets exhibited in Figures 10 and 11 to feed the SVM. The SVM correctly identified 18/20 pairs of corresponding codes against 19/20 of our ANN, with 10/20 (50%) ratio of false positives against 1/20 ratio of our ANN approach.

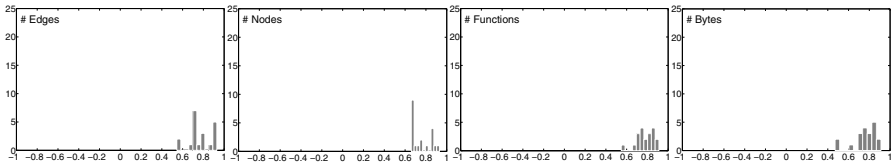


Fig. 13. Evaluation of the ANN using non-correlated binaries generated by different compilers

5 Conclusions

This paper deals with the issue of program equivalence of different languages — including high-level and low-level languages. The problem is fundamental for software validation: since the software evaluation is frequently based on source code analysis, it is important to guarantee that the compilation process did not introduce security flaws, backdoors or unwanted behaviors.

In the present work, we tackle the problem of program equivalence by extracting meaningful characteristics of program codes, obtaining such characteristics from program call graphs and control flow graphs. We use such characteristics to feed a nondeterministic classifier that decides whether a binary code corresponds to a given source code or whether a binary corresponds to another binary of a different platform or compiler. The originality of our approach lies on the extraction of characteristics of the call graphs and control flow graphs of the program codes, and on the use of an artificial neural network to decide the legitimacy of a binary code based on those characteristics. The performance of the proposed approach is well characterized through an experimental evaluation that, besides confirming a very low rate of false positives (considered as a basic requirement), also provides a reasonable amount of false negatives.

It could be argued that the proposed approach would not work in detecting simple binary code modifications that do not alter call graphs and control flow graphs, such as a change of a constant. We observe, however, that such modification would be immediately noted by the conventional functional tests performed on devices under verification. The kind of modification that our approach proposes to encounter is more subtle. For example, a malicious manufacturer could leave a backdoor that when activated transfers the execution control to a malicious behavior. Such a malicious modification would hardly be noticed by functional tests. However, it would not be possible for the manufacturer to include a backdoor without modifying the call graph and the control flow graph of the binary code. This makes the malicious modifications exactly the ones amenable to our approach based on call graph and control flow graph parameters.

An open question in the proposed approach, and the subject of ongoing research, concerns the necessity of also considering obfuscated codes during the training phase, to better understand its implications. For example, control-flow obfuscation alters the flow of control of the application by reordering statements, procedures, loops, obscuring flow of control using opaque predicates and

replacing transfer flow instructions. Using such obfuscation some properties used in our approach may change, so violating our results. Other possible avenues of research involve merging our current technique with data-flow strategies to circumvent attacks such as modification of variable contents. Finally, in future works we plan to investigate which other graph invariants — crossing number, cycle covering, chromatic number etc. — are meaningful to improve the correspondence of codes written in different languages.

References

1. Thompson, K.: Reflections on trusting trust. *Commun. ACM* 27(8), 761–763 (1984)
2. McDonald, J.: Delphi falls prey (2009), <http://www.symantec.com/connect/blogs/delphi-falls-prey> (last accessed October 2009)
3. Quinlan, D., Panas, T.: Source code and binary analysis of software defects. In: *CSIIRW 2009: Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research*, pp. 1–4. ACM, New York (2009)
4. Hassan, A.E., Jiang, Z.M., Holt, R.C.: Source versus object code extraction for recovering software architecture. In: *WCRE 2005: Proceedings of the 12th Working Conference on Reverse Engineering*, pp. 67–76. IEEE Computer Society, Washington, DC (1995)
5. Hatton, L.: Estimating source lines of code from object code. In: *Windows and Embedded Control Systems (2005)*, <http://www.leshatton.org/Documents/LOC2005.pdf>
6. Buttle, D.L.: Verification of Compiled Code. PhD thesis, University of York, UK (2001)
7. Wang, Z., Pierce, K., McFarling, S.: Bmat - a binary matching tool for stale profile propagation. *The Journal of Instruction-Level Parallelism* (2002)
8. Flake, H.: Structural comparison of executable objects. In: *Proc. of the Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*. IEEE Computer Society (2004)
9. Oh, J.: Fight against 1-day exploits: Diffing binaries vs anti-diffing binaries. In: *Blackhat Technical Security Conference (2009)*
10. Zhenga, J.: A digital image encryption algorithm based on hyper-chaotic cellular neural network. *Journal Fundamenta Informaticae* (2009)
11. Zeng, H., Rine, D.: A neural network approach for software defects fix effort estimation. In: *IASTED Conf. on Software Engineering and Applications*, pp. 513–517 (2004)
12. Zhenga, J.: Predicting software reliability with neural network ensembles. *Expert Systems with Applications* (36), 2116–2122 (2007)
13. Reddy, C.S., Raju, K.V.S.V.N., Kumari, V.V., Devi, G.L.: Fault-prone module prediction of a web application using artificial neural networks. In: *Proceeding (591) Software Engineering and Applications (2007)*
14. Lenic, M., Povalej, P., Kokol, P., Cardoso, A.I.: Using cellular automata to predict reliability of modules. In: *Proceeding (436) Software Engineering and Applications (2004)*
15. Boccoardo, D.R., Nascimento, T.M., Machado, R.C., Prado, C.B., Carmo, L.F.R.C.: Traceability of executable codes using neural networks. In: *Proceedings of the Information Security Conference (2010) (to appear)*

16. Moretti, E., Chanteperdrix, G., Osorio, A.: New algorithms for control-flow graph structuring. In: CSMR 2001: Proceedings of the Fifth European Conference on Software Maintenance and Reengineering, p. 184. IEEE Computer Society, Washington, DC (2001)
17. IdaPro: Ida pro - disassembler (2010), <http://www.hex-rays.com/idapro/> (last accessed January 2010)
18. Poznyakoff, S.: Gnu cflow (2010), <http://savannah.gnu.org/projects/cflow> (last accessed January 2010)
19. Ciocoiu, I.B.: Hybrid feedforward neural networks for solving classification problems. *Neural Processing Letters* 16(1), 81–91 (2002)
20. Asadi, R., Mustapha, N., Sulaiman, N.: New supervised multi layer feed forward neural network model to accelerate classification with high accuracy. *European Journal of Scientific Research* 33(1), 163–178 (2009)
21. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Prentice Hall (1998)
22. Hertz, J.A., Krogh, A.S., Palmer, R.G.: *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City (1991)
23. Moler, C.B.: *MATLAB — an interactive matrix laboratory*. Technical Report 369, University of New Mexico. Dept. of Computer Science (1980)
24. Men, H., Wu, Y., Gao, Y., Kou, Z., Xu, Z., Yang, S.: Application of support vector machine to heterotrophic bacteria colony recognition. In: CSSE (1), pp. 830–833 (2008)
25. Angulo, C., Ruiz, F., González, L., Ortega, J.A.: Multi-classification by using tri-class svm. *Neural Processing Letters* 23(1), 89–101 (2006)
26. Burkard, J.: C software (2010), <http://people.sc.fsu.edu/~burkardt/> (Last accessed January 2010)
27. Oliveira Cruz, A.J.: C software (2010), <http://equipe.nce.ufrj.br/adriano/c/exemplos.htm> (last accessed January 2010)

Representation of Spiking Neural P Systems with Anti-spikes through Petri Nets

Venkata Padmavati Metta¹, Kamala Krithivasan², and Deepak Garg³

¹ Bhilai Institute of Technology, Durg, India
vmetta@gmail.com

² Indian Institute of Technology, Chennai, India
kamala@iitm.ac.in

³ Thapar University, Patiala, India
deep108@yahoo.com

Abstract. Spiking Neural P(SN P) system with anti-spikes uses two types of objects called spikes and anti-spikes which can encode binary digits in a natural way. We propose a formal method based on Petri nets, which provides a natural and powerful framework to formalize SN P systems with anti-spikes. This enables the use of existing tools for Petri nets to study the computability and behavioural properties of SN P systems with anti-spikes.

1 Introduction

Spiking neural P systems (shortly called SN P systems) introduced in [1] as a variant of P systems, are biologically inspired parallel and distributed computing models inspired by the neurobiological behaviour of neurons sending electrical pulses of identical voltages called spikes to neighbouring neurons.

SN P system with anti spikes (shortly called SN PA system) introduced in [5], is a variant of an SN P system consisting of two types of objects, spikes(denoted as a) and anti-spikes(denoted as \bar{a}) participating in spiking and forgetting rules. We propose a relationship between SN P system with anti-spikes and Petri nets to complement the functional characterization of the behaviour of SN PA systems. The relationship between SN P systems and Petri nets is no means a new idea. Behavioural aspects of different variants of membrane systems were studied by translating them into equivalent Petri net models [4,3]. In [8], a formal translation has been given for basic class of SN P systems with delays into a class of Petri nets. In these Petri nets, places are used to represent neurons and spikes are encoded with tokens. Transitions are used to implement rules inside the neurons. Status places are associated with each place to maintain the state(open or closed) of the each neuron. The inhibitor and test arcs are used to test the status of the place and to only send the tokens to the open places.

To represent spikes and anti-spikes in SN PA systems, we use coloured Petri nets [2] in which tokens can of different types. We introduce coloured Petri nets with localities, with each transition is assigned a location based on the input place and locally sequential and globally maximal firing semantics to the

sequential firing in a neuron. The annihilation rule present in each neuron is encoded as a highest priority transition with exhaustive firing semantics. It is worth noting that as far as the rules are concerned, SN P systems are highly concurrent in nature and Petri nets play an important role in the modelling, analysis and verification of concurrent systems. As the procedure is direct, it involves less complexity in translation and the rich theoretical concepts and practical tools from well developed Petri nets can be used in the field of SN PA systems. We can also prove the correctness of the computation of SN PA system and also study the some behavioural properties like boundedness by means of Petri nets.

2 Spiking Neural P System with Anti-spikes

First we recall the definition of SN P system with anti-spikes (or SN PA system).

Definition 2.1 (*SN P system with anti-spikes*). Mathematically, we represent a spiking neural P system with anti-spikes of degree $m \geq 1$, in the form

$$\Pi = (O, \sigma_1, \sigma_2, \sigma_3, \dots, \sigma_m, syn, i_0), \text{ where}$$

1. $O = \{ a, \bar{a} \}$ is the alphabet. a is called *spike* and \bar{a} is called anti-spike.
2. $\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_m$ are neurons, of the form

$$\sigma_i = (n_i, R_i), \quad 1 \leq i \leq m, \text{ where}$$

- (a) n_i is the *multiset of spikes or anti-spikes* contained by the neuron.
- (b) R_i is a finite set of *rules* of the following two forms:
 - i. $E / b^r \rightarrow b'$ where E is a regular expression over a or \bar{a} , while $b, b' \in \{a, \bar{a}\}$, and $r \geq 1$.
 - ii. $b^r \rightarrow \lambda$, for some $r \geq 1$, with the restriction that $b^r \notin L(E)$ for any rule $E / b^r \rightarrow b'$ of type (1) from R_i ;
3. $syn \subseteq \{ 1, 2, 3, \dots, m \} \times \{ 1, 2, 3, \dots, m \}$ with $(i, i) \notin syn$ for $1 \leq i \leq m$ (*synapses* among neurons);
4. $i_0 \in \{ 1, 2, 3, \dots, m \}$ indicates the *output neuron*.

The rules of type $E / b^r \rightarrow b'$ are spiking rules, and they are possible only if the neuron contains n b 's such that $b^n \in L(E)$ and $n \geq r$. When neuron σ_i sends a b , it is replicated in such a way that one b' is sent to all neurons σ_j such that $(i, j) \in syn$. The rules of type $b^r \rightarrow \lambda$ are forgetting rules; r number of b 's are simply removed ("forgotten") when applying. Like in the case of spiking rules, the left hand side of a forgetting rule must "cover" the contents of the neuron, that is, $a^s \rightarrow \lambda$ is applied only if the neuron contains exactly s spikes.

There is an additional fact that a and \bar{a} cannot stay together, so annihilate each other. If a neuron has either objects a or objects \bar{a} , and further objects of either type (maybe both) arrive from other neurons, such that we end with a^r and \bar{a}^s inside, then immediately an annihilation rule $a \bar{a} \rightarrow \lambda$, which is implicit in each neuron, is applied in a maximal manner, so that either a^{r-s} or \bar{a}^{s-r}

remain for the next step, provided that $r \geq s$ or $s \geq r$, respectively. This mutual annihilation of spikes and anti-spikes takes no time and that annihilation rule has priority over spiking and forgetting rules, so the neurons always contains either only spikes or anti-spikes.

$lhs(v)$ and $rhs(v)$ gives the multiset of spikes/anti-spikes present in the left and right hand sides of rule v respectively. Like in [5], we avoid using rules $\bar{a}^x \rightarrow \bar{a}$, but not the other three types, corresponding to the pairs (a, a) , (a, \bar{a}) , (\bar{a}, a) . If $E=b^r$ then we will write it in the simplified form $b^r \rightarrow b^r$.

The standard SN P system works in a similar way but with only one type of object called *spike*(a) and so there exist no annihilation rules.

Definition 2.2 (Configuration). The configuration of the system is described $\mathcal{C} = \langle n_1, n_2, \dots, n_m \rangle$ where n_i is the multiset written in the form $n_i = a^x \bar{a}^y$, where x is the number of spikes and y is the number of anti-spikes present in neuron σ_i . Because a neuron always contains spikes or anti-spikes, either $n_i(a) = 0$ or $n_i(\bar{a}) = 0$.

Definition 2.3 (Vector rule). We define a vector rule V as a mapping with domain Π such that $V(i) = r_{ij}$, r_{ij} is a spiking or forgetting rule from R_i i.e $|V(i)| = 0$ or 1 where $1 \leq i \leq m$. If no rule is applicable from σ_i then $V(i) = r_{i0}$. If a vector rule V is enabled at a configuration $\mathcal{C} = \langle n_1, n_2, \dots, n_m \rangle$ then \mathcal{C} can evolve to $\mathcal{C}' = \langle n'_1, n'_2, \dots, n'_m \rangle$ (after applying annihilation rules in each neuron in exhaustive way), where

$$n'_i = n_i - lhs(V(i)) + \sum_{(j,i) \in syn} rhs(V(j))$$

Definition 2.4 (Transition). Using the vector rule, we pass from one configuration of the system to another configuration, such a step is called a transition. For two configurations \mathcal{C} and \mathcal{C}' of Π we denote by $\mathcal{C} \xrightarrow{V} \mathcal{C}'$, if there is a direct transition from \mathcal{C} to \mathcal{C}' in Π .

A computation of Π is a finite or infinite sequences of transitions starting from the initial configuration, and every configuration appearing in such a sequence is called reachable. Note that the transition of \mathcal{C} is non-deterministic in the sense that there may be different vector rules applicable to \mathcal{C} , as described above. A computation halts if it reaches a configuration where no rule can be used. There are various ways of using such a device [6].

Example 2.1. Consider the graphical representation of an SN P system with anti-spikes in Fig.1(a), the neurons are represented by nodes of a directed graph whose arrows represent the synapses; an arrow also exits from the output neuron, pointing to the environment; in each neuron we specify the rules and the spikes present in the initial configuration. It is formally denoted as

$$\begin{aligned} \Pi = & (\{a, \bar{a}\}, \sigma_1, \sigma_2, \sigma_3, \sigma_4, syn, 4), \text{ with} \\ \sigma_1 = & (a^3, \{a^3/a \rightarrow a, a^3 \rightarrow \bar{a}\}), \sigma_2 = (a, \{a \rightarrow a\}), \\ \sigma_3 = & (a, \{a \rightarrow a\}), \sigma_4 = (a, \{a \rightarrow \bar{a}, \bar{a} \rightarrow a\}), syn = \{(1, 2), (2, 1), (1,4), (4,1), \\ & (1,3), (3,1)\}. \end{aligned}$$

We have four neurons, with labels 1, 2, 3, 4; neuron 4 is the output neuron. Initially neuron 1 has three spikes with non-determinism between its first two

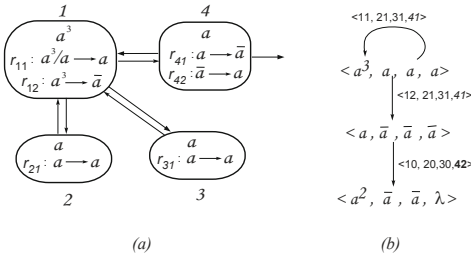


Fig. 1. SN PA system II

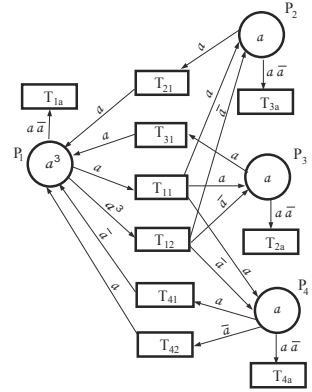


Fig. 2. CPL-net equivalent to II

rules and neurons 2, 3 and 4 have one spike each. The initial configuration of the system is $\langle a^3, a, a, a \rangle$.

The evolution of the system II can be analysed on a transition diagram as that from Fig.1(b) because the number of configurations reachable from the initial configuration is finite, we can place them in the nodes of a graph and between two nodes/configurations we draw an arrow if and only if a direct transition is possible between them. In the Fig.1(b), we have also indicated the rules used in each neuron with the following conventions; for each r_{ij} we have written only the subscript ij ; when a neuron $i=1, 2, 3, 4$ uses no rule, we have written $i0$.

The functioning can easily be followed on this diagram, so that we only briefly describe it. We start with spikes in all neurons. Neuron 1 can behave non-deterministically choosing one of the two rules. As long as neuron 1 uses the rule $a^3/a \rightarrow a$, the computation cycles in the initial configuration sending a spike to neurons 2, 3 and 4; neuron 4 uses its first rule and sends an anti-spike to environment and neuron 1. Neurons 2 and 3 use their rules and send a spike to neuron 1. So neuron 1 receives one anti-spike and two spikes (and two spikes are already present in it), after using annihilation rule, the neuron will have again three spikes. Neuron 2, 3 and 4 will have one spike each.

If neuron 1 uses its second rule $a^3 \rightarrow \bar{a}$, the three spikes are consumed and an anti-spike is sent to other three neurons. So neuron 1 will have one spike and neurons 2, 3 and 4 will have one anti-spike each, reaching the configuration $\langle a, \bar{a}, \bar{a}, \bar{a} \rangle$. In the next step neurons 1, 2 and 3 cannot fire and neuron 4 uses the rule $\bar{a} \rightarrow a$ sending a spike to environment and neuron 1, reaching the configuration $\langle a^2, \bar{a}, \bar{a}, \lambda \rangle$ and the system halts.

3 Colored Petri Nets

In coloured PN, color refers to the type of data associated with tokens. In other words, tokens can have arbitrary values determined by their type or color. Each

place has an associated color set, which constrains the number and color of tokens that may reside in the place or move along the arc from that place.

Definition 3.1 (*Coloured Petri net*). A coloured Petri net is represented by a tuple $\mathcal{N} \stackrel{df}{=} (\Sigma, P, T, C, A, W, \Gamma, G, M_0)$, where Σ , P and T are finite, non-empty set of colours, places and transitions respectively.

$C: P \rightarrow \Sigma$ is a color function that assigns set of colours to every element of P .

$A \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs which connect places with transitions and transitions with places.

W : Assigns a multi-set of coloured tokens from the domain of input places if $f \in (P \times T)$ and from domain of output places if $f \in (T \times P)$.

$G: T \rightarrow \{true, false\}$, the guard function maps each transition T_i to boolean expression, which specifies an additional constraint which must be fulfilled before the transition is enabled.

$M_0 = \{m_1, m_2, \dots, m_n\} \in P$, each m_i is a multi set of tokens initially associated with each place P_i and n is the number of places in the net \mathcal{N} .

Coloured Petri nets are drawn in a similar way as simple Petri nets with coloured tokens drawn as coloured dots (for tokens of different attributes) or as multi-set inside the places. The directed arcs connecting places to transitions and transitions to places may be labelled with an arc expression having the multi-set of tokens of different colours.

Definition 3.2 (*Marking*). M is a marking which assigns a finite semi-positive multi-set over $C(p)$ to every $p \in P$. M_0 is called the initial marking.

The state or marking of Petri net is changed by the occurrence of transition. Firing rules in the Petri net model are:

1. Transition T_j is enabled iff T_j satisfies the guard condition and its every input place has enough tokens of needed colours as specified in the input arc expression.
2. Upon firing the transition T_j removes number of tokens from each of its input places equal to the weight of the input arcs and deposits number of tokens into the output places equal to the weight of output arcs.

Concurrency is also a concept that Petri net systems represent in an extremely natural way. Two transitions are concurrent at a given marking if they can be fired at the same time i.e. simultaneously. This determines a new marking in a net, a new set of enabled transitions, and so on. An important concept in Petri nets is that of conflict. Conflict occurs between transitions that are enabled by the same marking, where the firing of one transition disables the other. A major feature of net is that they do not define in any way how and when a given conflict should be resolved, leading to non-determinism in its behaviour. This non-determinism is inherent in Petri nets.

Definition 3.3 (*Step*). A step is a multi-set U of transitions which are enabled at a marking M and we denote this by $M[U]$. The input and output of places of step U are given by $IN_{\mathcal{N}}U(p) = \sum_{t \in U} U(t).W(p, t)$ and $OUT_{\mathcal{N}}U(p) = \sum_{t \in U} U(t).W(t, p)$ for each $p \in P$

A step U which is enabled at a marking M can be executed leading to the marking $M' = M + OUT_{\mathcal{N}}U - IN_{\mathcal{N}}U$. We denote this by $M [U] M'$. A step U is a maximal step at a marking M if $M [U]$ and there is no transition t' such that $M [U + \{t'\}]$ and for every place $p \in P$, transition $t \in U$, t can only be executed if it satisfies the guard function.

A Petri net system \mathcal{N} with maximal concurrency is such that for each markings M and M' if there is a step U such that $M [U] M'$, then U is a maximal step.

A computation of a Petri net \mathcal{N} is a finite or infinite sequences of executions starting from the initial marking and every marking appearing in such a sequence is called reachable.

A major strength of Petri nets is their support for analysis of many properties and problems associated with concurrent systems such as reachability, boundedness and liveness. The reachability problem for Petri net is the problem of finding if a marking M_i is reachable from the initial marking M_0 . Formally a Petri net with a given marking is said to be in deadlock if and only if no transition is enabled in the marking. A Petri net where no deadlock can occur starting from a given marking is said to be live. Generally Petri nets are analysed using tools to study important behavioural properties of the system like reachability, liveness, boundedness etc. Before implementation, we introduce the most appropriate Petri net semantics.

Definition 4.1 (*Coloured Petri nets with Localities*). A coloured Petri net with localities (or CPL-net) is a tuple $\mathcal{NL} \stackrel{df}{=} (\Sigma, P, T, C, A, W, \Gamma, G, M_0, \psi, L)$, where $UND(\mathcal{NL}) \stackrel{df}{=} (\Sigma, P, T, C, A, W, \Gamma, G, M_0)$. $\psi : T \rightarrow N$ defines the priority function and $L : T \rightarrow N$ is a location mapping, for the transition set T . The markings and steps already defined for coloured Petri nets carry over to CPL-net.

Definition 4.2 (*Enabled step*). A step U is enabled at a marking M if $M \geq IN_{\mathcal{NL}}U$ and, for every place $p \in P$ and transition $t \in U$, $G(t) = true$. Moreover, an enabled U is: *lseq-gmax* (locally sequential globally maximal) enabled if $t \in U$ then there is no transition $u \in U$ such that $L(u) = L(t)$ and there is no transition l such that $U + l$ is enabled at M such there $L(t) \neq L(u)$ for any $u \in U$. The priority of all transitions in a step should be the same. So the priority of the step is same as priority transitions in the step. If two steps are enabled at a marking then the step with higher priority is executed first. In this paper we consider *prioritized locally sequential globally maximal* firing of CPL-nets.

Definition 4.3 (*Executed step and Computation*). We introduce two modes of execution of *lseq-gmax* enabled step U . In *minimal execution mode* every $t \in U$ is fired only once and is denoted as U_{min} where as in *exhaustive execution mode* every $t \in U$ is fired as many times as possible with the same binding for its controlled variables and we denote this as U_{exh} .

Let $m \in \{min, exh\}$ be a mode of execution. A *lseq-gmax* enabled step U at a marking M can be *m-executed* leading to the marking M' . We denote this by $M[U >_m M']$. A *computation* of a CPL-net \mathcal{NL} is a finite or infinite sequence

of executions starting from the initial marking, and every marking appearing in such a sequence is called reachable. The step semantics and the execution modes defined above corresponds to the way rules are fired in the SN PA systems. Using this correspondence we will give in the next section a faithful translation of SN PA systems into CPL-nets.

4 SN P System with Anti-spikes to Petri Net

In this section, we propose a formal method to translate SN PA systems into CPL-nets.

Definition 5.1 (*SN PA system to CPL-net*). Let $\Pi=(O, \sigma_1, \sigma_2, \sigma_3, \dots, \sigma_m, syn, i_0)$ be an SN P system with anti-spikes, then the corresponding CPL-net $\mathcal{N}\mathcal{L}_\Pi \stackrel{df}{=} (\Sigma, P, T, C, A, W, \Gamma, G, M_0, \psi, L)$, where

1. $\Sigma=O$
2. $P = \{P_1, P_2, \dots, P_m\}$ is the set of places. P_{i_0} is the output place.
3. $T = T_1 \cup T_2 \cup \dots \cup T_m$ where each T_i contains a distinct transition T_{ij} for every rule of $r_{ij} \in R_i$ and T_{ia} , a transition corresponding to the annihilation rule implicitly present in each neuron P_i .
 For every place $p = P_i \in P$ and every transition $t = T_{jk} \in T$ and $k \neq a$,
 $W(p, t) = lhs(r_{jk})$ if $i = j$, $W(t, p) = rhs(r_{jk})$ if $i \neq j$ and $(i, j) \in syn$,
 $\psi(t) = 1$, $L(t) = i$ and
 for every $p = P_i \in P$, $W(p, T_{ia}) = a\bar{a}$, $\psi(T_{ia}) = 2$, $L(T_{ia}) = i$.
4. For every place $P_i \in P$, its initial marking is $M_0(P_i) \stackrel{df}{=} n_i$.

To capture the very tight correspondence between the SN P system with anti-spikes Π and Petri nets $\mathcal{N}\mathcal{L}_\Pi$, we introduce a straight forward bijection between configurations of Π and markings of $\mathcal{N}\mathcal{L}_\Pi$, based on the correspondence between places and neurons.

Let $\mathcal{C} = \langle n_1, n_2, \dots, n_m \rangle$ be a configuration of an SN PA system Π . Then the corresponding marking $\phi(\mathcal{C})$ of $\mathcal{N}\mathcal{L}_\Pi$ is given by $\phi(\mathcal{C})(P_i) \stackrel{df}{=} n_i$ for every place P_i of $\mathcal{N}\mathcal{L}_\Pi$.

Similarly, for any vector rule $V = (r_{1j_1}, r_{2j_2}, \dots, r_{mj_m})$ of Π , we define a *lseq-gmax* enabled step $\xi(V)$ of transitions of $\mathcal{N}\mathcal{L}_\Pi$ such that $\xi(V)(T_{ij}) \stackrel{df}{=} r_{ij}$ for every $T_{ij} \in T$ and $j \neq a$. It is clear that ϕ is a bijection from the configurations of Π to the markings of $\mathcal{N}\mathcal{L}_\Pi$, and that ξ is a bijection from vector rules of Π to *lseq-gmax* enabled steps of $\mathcal{N}\mathcal{L}_\Pi$.

We now can formulate a fundamental property concerning the relationship between the dynamics of the SN PA system Π and that of the corresponding CPL-net: $\mathcal{C} \xrightarrow{V} \mathcal{C}'$ if and only if $\phi(\mathcal{C})[\xi(V) \triangleright_{min} [H \triangleright_{exh} \phi(\mathcal{C}')$.

where H is *lseq-gmax* step only containing $T_{ia} \in T$ for every $1 \leq i \leq m$.

Since the initial configuration of Π corresponds through ϕ to the initial marking of $\mathcal{N}\mathcal{L}_\Pi$, the above immediately implies that the computations of Π coincide with the locally sequential and globally maximal concurrency semantics of the CPL-net $\mathcal{N}\mathcal{L}_\Pi$.

The reader might by now have observed that the structure of neurons in Π is used in the definitions of the structure of the CPL-net \mathcal{NL}_{Π} (i.e., in the definitions of places, transitions and the weight function). Let \mathcal{C} be a configuration of Π and there is a vector rule V enabled at \mathcal{C} reaching a configuration \mathcal{C}' . As there is a mapping between configuration and markings, $\phi(\mathcal{C})$ is marking of CPL-net \mathcal{NL}_{Π} corresponding to the configuration \mathcal{C} of Π . There is a one-to-one mapping between the rules in the SN PA system and transitions in CPL-net. For locally sequential and globally maximal concurrency firing semantics of SN PA systems, the *prioritized locally sequential globally maximal steps* are defined CPL-nets. Two execution modes *minimal* and *exhaustive* are defined to encode the working spiking rules and annihilations rules in the SN PA system. So there exists a step $\xi(V >$ enabled at the marking $\phi(\mathcal{C})$. After the execution of the step in the minimal mode, and after firing the step H containing the transitions corresponding the the annihilation rules in exhaustive way, the system reaches the configuration $\phi(\mathcal{C}')$. We can prove only if part in the similar way. So the evolution of the CPL-net \mathcal{NL}_{Π} is same as the evolution of the SN PA system Π . The CPL-net \mathcal{NL}_{Π} corresponding to the SN PA system Π is given in the Fig.2.

5 Conclusion

In this paper we have proposed an approach to the performance modeling of the behaviour of SN P systems with anti-spikes through a class of Petri nets, called coloured Petri nets with localities. The annihilation rule implicitly present in a neuron is successfully encoded as highest priority sink transition and its exhaustive firing enables to represent working of annihilation rule. Based upon the introduction these features, the neural structure can be successfully encoded as a Petri nets model which permit the description the behavioural state based process run-time structure change of SN P system with anti-spikes.

References

1. Ionescu, M., Păun, G., Yokomori, T.: Spiking Neural P Systems. Fund. Infor. 71, 279–308 (2006)
2. Jensen, K.: A Brief Introduction to Coloured Petri Nets. In: Brinksma, E. (ed.) TACAS 1997. LNCS, vol. 1217, pp. 203–208. Springer, Heidelberg (1997)
3. Kleijn, J., Koutny, M.: A Petri net model for membrane systems with dynamic structure. Natural Computing 8(4), 781–796 (2009)
4. Kleijn, J., Koutny, M., Rozenberg, G.: Towards a Petri Net Semantics for Membrane Systems. In: Freund, R., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2005. LNCS, vol. 3850, pp. 292–309. Springer, Heidelberg (2006)
5. Linqiang, P., Păun, G.: Spiking Neural P Systems with Anti-Spikes. Int. J. of Computers, Communications and Control 4, 273–282 (2009)
6. Păun, G.: Spiking Neural P Systems Used as Acceptors and Transducers. In: Holub, J., Ždárek, J. (eds.) CIAA 2007. LNCS, vol. 4783, pp. 1–4. Springer, Heidelberg (2007)
7. Reisig, W., Rozenberg, G. (eds.): APN 1998. LNCS, vol. 1491, 1492. Springer, Heidelberg (1998)
8. Venkata Padmavati, M., Kamala, K., Deepak, G.: Modeling Spiking Neural P systems using Timed Petri nets. In: Int. Conf. on Nature and Biologically Inspired Computing (NaBIC 2009). IEEE Xplore (2009)

Learning in a Distributed Software Architecture for Large-Scale Neural Modeling

Jasmin Léveillé¹, Heather Ames², Benjamin Chandler², Anatoli Gorchetchnikov¹,
Ennio Mingolla², Sean Patrick¹, and Massimiliano Versace²

¹ Department of Cognitive and Neural Systems, Boston University, Boston, MA 02215

² Center of Excellence for Learning in Education, Science, and Technology, Boston University,
Boston, MA 02215

{jasminl, anatoli, versace, ennio}@cns.bu.edu,
{heather.m.ames, sean.patrick.619, bchandle}@gmail.com

Abstract. Progress on large-scale simulation of neural models depends in part on the availability of suitable hardware and software architectures. Heterogeneous hardware computing platforms are becoming increasingly popular as substrates for general-purpose simulation. On the other hand, recent work highlights that certain constraints on neural models must be imposed on neural and synaptic dynamics in order to take advantage of such systems. In this paper we focus on constraints related to learning in a simple visual system and those imposed by a new neural simulator for heterogeneous hardware systems, *CogExMachina* (Cog).

Keywords: Large-scale system, learning laws, neural networks, neural network software, heterogeneous computing.

1 Introduction

Building neural models capable of simulating complex behaviors requires simulating large-scale models linking the macro-scale of behavior to the micro-scale of individual neural events. In recent years a large number of simulation platforms have emerged that satisfy different modeling needs, ranging from simulators that include the low-level spiking behavior of individual neurons [1] to some that abstract away biological details about neurons but try to maintain some of their functions [2]. Although large-scale neural modeling has greatly benefited from the introduction of supercomputers [3], the potential of heterogeneous computing systems still remains largely unexplored. Heterogeneous computing offers many advantages for neural modeling, including the possibility to scale simulations at virtually no cost, and to employ a variety of hardware accelerators to optimize specific model components. On the other hand, heterogeneous computing severely constrains the design of general-purpose neural network modeling software. This paper addresses the issue of how to embed learning in such a system based on our experience with *CogExMachina* (Cog), a new neural simulator designed for heterogeneous computing systems [4].

We start in the next section with a brief description of the general modeling framework imposed by Cog. Section 3 examines in more detail the constraints that relate specifically to learning and presents simulation results of a model of the early visual system able to learn orientation preference and ocular dominance maps. We conclude with a description of our current efforts on generalizing the learning capabilities of the system.

2 Neural Modeling Framework

Single neurons in Cog are implemented based on the following neuron model [5]:

$$y = f(\mathbf{w}^T \mathbf{x}), \quad (1)$$

where \mathbf{x} and \mathbf{w} are the presynaptic input and associated synaptic weight vectors, respectively, f is a scalar-valued activation function, and y is the activation value of the postsynaptic neuron. The product $\mathbf{w}^T \mathbf{x}$ is referred to as the *partial inference*. Cog imposes virtually no restrictions on the choice of the activation function, thereby leaving the modeler free to determine the kind of computation performed by each neuron. However, external input is obtained via partial inference only. This restriction ensures that models implemented remain tractable and efficient by removing the need for sophisticated synchronization mechanisms to handle parallelism.

The segregation of computation into a set of partial inferences followed by an activation function is a critical bridge between biological and silicon computation built into Cog. A neural population can maintain relatively little state, but perform potentially highly nonlinear computations. The web of dendrites feeding that population has states stored in each synapse, but computes in a much more rigid manner.

This dual-natured computation maps extremely well to heterogeneous computers. Conventional general-purpose processors can only work on relatively small sets of data efficiently, but include highly robust strategies for handling irregularity and nonlinearity. Special-purpose accelerators like graphics processors are designed for an opposite set of constraints. They require dramatically higher memory bandwidth, but de-prioritize handling of irregular computation. The signal function component of computation is best mapped to a conventional processor, where the partial inference calculation maps efficiently to a graphics processor.

Beyond graphics processors, single-purpose hardware offers multiple additional orders of magnitude in power efficiency. Graphics processors are vastly more efficient than conventional processors for computations like partial inference, but data is still not as physically local as needed for power efficiency rivaling biology. A graphics card includes its own bank of memory with a very wide connection to the processor, but these two components are still physically separated. This separation means data must be shuttled from a memory chip, across a bus, and finally in to the processor. Efficiency would be dramatically higher if memory could be co-located directly on the processor.

Memristive crossbar memory is a viable contender for this unification of processing and memory. Memristor crossbars can be manufactured directly on top of a conventional chip, but with dramatically higher density than existing memory technologies. This means a massively multi-core chip designed to handle partial inference can localize storage of the weights directly on top of the processing core performing calculations – in particular, partial inference and learning - with those weights. Weight data need not move more than a few millimeters, resulting in a massive increase in energy efficiency. Cog’s design is in part motivated by such considerations of locality [4].

In summary, the modeling framework provided by Cog limits communication between neurons to partial inferences and tries to maintain computation related to synaptic weights local to the partial inference processors. The next section addresses the impact of these constraints on learning.

3 Learning

The above considerations lead to the question of how to efficiently introduce synaptic weight learning in a large-scale model instantiated on a distributed, heterogeneous network. One way to answer this question is to restrict learning to a single learning rule which can then be optimized in the same hardware that computes partial inference. From this point of view, one good candidate learning rule might be spike-timing-dependent-plasticity [6] since it appears to give the most complete account of biological synaptic modification mechanisms. Indeed, some large-scale models adopt this rule as their only learning mechanism [7]. However, in a system in which biological realism is only secondary to functionality, such a choice is no longer justified. Moreover, it is fundamentally impossible to determine in advance which learning law is most appropriate without knowing exactly which behavioural task the model needs to perform. Finally, even if the task is known, it is not always clear which learning law will perform best, and if so in what parameter range can it be expected to perform well. Thus, learning in Cog follows the approach of implementing only a generic form of the learning equation which can then be tailored for specific applications. Crucially, the use of a generic form allows for hardware acceleration on the same processors that also compute partial inferences.

3.1 Current General Form of Learning Laws

Cog currently supports learning laws for which weight changes can be implemented in the following general form [8]:

$$\Delta w_{ij} = \lambda s \left(\Delta w_{ij}^H + \Delta w_{ij}^C + \Delta w_{ij}^N \right), \quad (2)$$

where λ is a learning rate, s is a sign factor (-1 or +1), and Δw_{ij}^H , Δw_{ij}^C and Δw_{ij}^N are weight-change terms related to Hebbian, competitive and normalization operations respectively. Presynaptic and postsynaptic units are respectively denoted in Eq. 2 by

indexes i and j . This general form was shown in [8] to be able to encapsulate a number of learning rules performing independent component analysis and is implemented in Cog as the following sequence of three steps:

$$w_{ij}^1 \leftarrow w_{ij}^0 + h_j x_i , \tag{3}$$

$$\begin{aligned} x'_i &\leftarrow w_{ij}^1 h_j \\ w_{ij}^2 &\leftarrow w_{ij}^1 - h_j x'_i \end{aligned} , \tag{4}$$

$$w_{ij}^3 \leftarrow w_{ij}^2 - g_j w_{ij}^2 . \tag{5}$$

Eqs. 3, 4 and 5 implement the Hebbian, competitive and normalization steps, respectively. Quantities h_j and g_j incorporate the learning rate λ and sign s and are computed by the postsynaptic neuron y_j at each time step. Crucially, the forms of h_j and g_j are determined by the user so as to implement a particular learning rule.

Simulation flow in Cog can thus be described as a sequence of two operations performed at each time step. First, all partial inferences are computed and learning is performed based on Eqs. 2-5 with feedback terms h_j and g_j returned from postsynaptic neurons computed at the previous time step. Second, all activation functions are computed as well as feedback terms h_j and g_j , which are sent back to the partial inference processors to be used at the next time step.

3.2 Examples

The simplest example of a learning law that can be implemented in Eqs. 4-6 is Hebbian learning:

$$\dot{w}_{ij} = \lambda x_i y_j , \tag{6}$$

where continuous-time notation is used for simplicity. A sequential implementation of this learning rule is easily obtained by setting $h_j = \lambda y_j$ in Eq. 3 and skipping the remaining steps. A slightly more complicated learning rule is the *instar* law [9]:

$$\dot{w}_{ij} = y_j (\lambda x_i - \alpha w_{ij}(t)) , \tag{7}$$

where learning is gated by postsynaptic activity y_j and an additional decay rate, α , needs to be specified by the modeler. Another common law used in the literature on self-organization is the Hebbian rule with postsynaptic normalization [10, 11]:

$$\dot{w}_{ij} = \frac{w_{ij} + \alpha x_i y_j}{\sum_k (w_{kj} + \alpha x_k y_j)} - w_{ij} . \quad (8)$$

Eq.8 can be implemented in Cog as follows. First, the Hebbian step is computed as $h_j = \lambda y_j$. The competitive step is then skipped by setting $w_{ij}^2 = w_{ij}^1$, and the normalization step is computed as follows:

$$g_j = 1 - \frac{1}{\sum_k (w_{kj}^0 + \alpha x_k y_j)} = 1 - \frac{1}{1 + \alpha y_j \sum_k x_k} , \quad (9)$$

where the fact that all weights sum to 1 (due to postsynaptic normalization) is used to simplify the denominator. Still, Eq. 9 implies that postsynaptic unit y_j needs to have access to the unweighted sum of its presynaptic inputs, requiring additional data transfers between processors computing the signal functions. Nevertheless, this can be accomplished in Cog by allocating an additional synaptic projection consisting of constant unit weights (i.e. $w'_{ij} = 1$) connecting presynaptic inputs x_i to postsynaptic neurons y_j . This costly duplication of weights illustrates well that any learning rule that does not comply naturally with the general form in Eq. 2 cannot be efficiently implemented in a large-scale, distributed, heterogeneous system.

3.3 Simulations

Learning of orientation preference and ocular dominance maps was implemented in Cog as a preliminary test of the simulation framework. The model implemented here is based on the LISSOM architecture [11] which builds on a widely popular tradition of research starting with the model in [10]. Network topology consists of two retinas, each projecting to distinct populations of on-center off-surround and off-center on-surround LGN cells (Fig.1a). Projections to area V1 consist of bottom-up inputs from each LGN population as well as horizontal excitatory and inhibitory connections from within V1. Details about the model and training procedure can be found in [11]. The network was trained using the Hebbian rule with postsynaptic normalization (Eq. 8) which implements competition between synaptic weights and has been shown to be efficient at learning cortical maps.

Figs. 1b-d illustrate the effect of learning using randomly presented inputs on the distribution of ocular dominance across V1 cells, where 0 indicates a cell sensitive to the left eye and 1, a cell sensitive to the right eye. Before learning, all cells are clustered at the center of the x -axis, indicating no strong preference for any eye (Fig.1b). After a period of normal rearing, a large proportion of cells become more selectively tuned to a particular eye (Fig.1c). In contrast, if the network is trained with inputs to the right eye only, all cells become sensitive to the right eye (Fig. 1d).

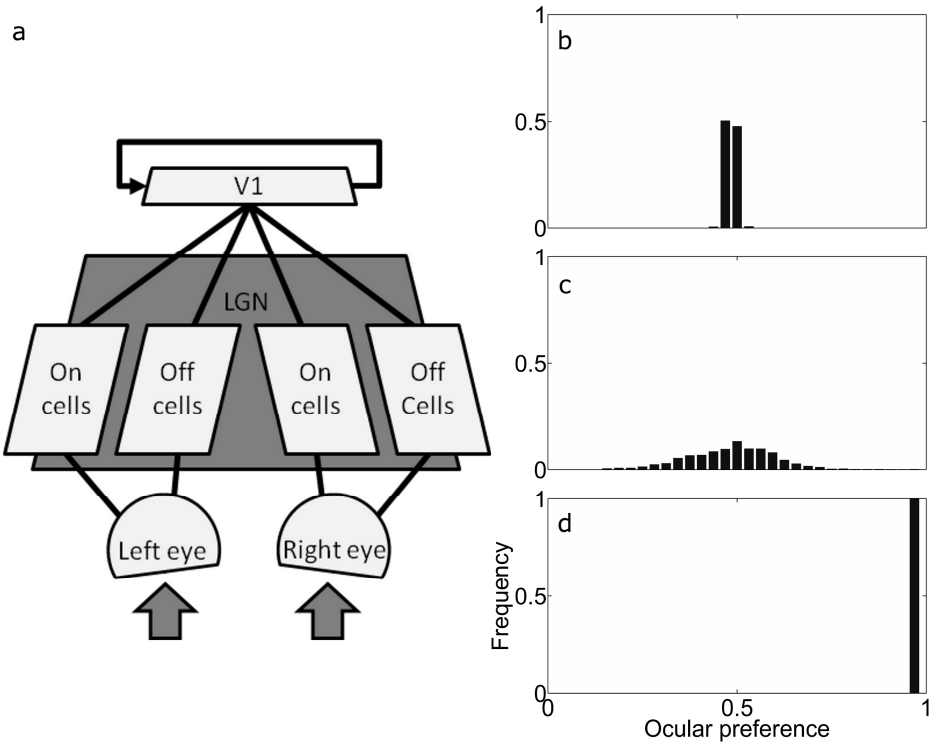


Fig. 1. a) LISSOM model architecture for orientation preference and ocular dominance map development. b) Distribution of ocular dominance before learning (0: left eye selectivity; 1: right eye selectivity). c) Distribution after a period of binocular training. d) Distribution after monocular training.

Figure 2 shows the topography of orientation maps and ocular dominance maps before learning (a and c, respectively) and after a period of learning (b and d, respectively).

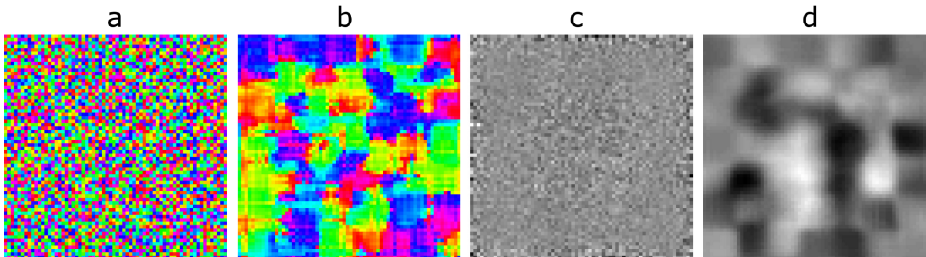


Fig. 2. Topography of learned feature maps. Panels a) and c) respectively display orientation and ocular dominance maps before learning. Panels b) and d) display corresponding maps after a period of learning. In a) and b), color codes for preferred orientation, and in c) and d) grayscale values indicate ocular dominance.

As a large-scale learning experiment in Cog, the LISSOM model can be criticized on the grounds that it does not lead to stable representations without concurrently decreasing the learning rate, and that it requires a duplication of weights to implement synaptic weight normalization (Eq. 9). Both issues remain important topics which we are addressing in our current modeling efforts. In particular, on the second issue, the instar rule (Eq. 7) can be shown to lead to normalization based on postsynaptic activity only, thereby escaping the duplication of weights, but requires that inputs be normalized to a fixed constant. Although this assumption does not hold in the LISSOM model, it remains to be shown whether strict normalization is necessary in general. Another candidate learning rule is the BCM law which also implements competition based only on postsynaptic quantities, and also displays stability [12].

3.4 Toward a Generalized Learning Law Equation

In order to allow for a more thorough study of learning in large-scale systems, the general form in Eq. 2 and its associated three-step procedure in Eqs. 3-6 must be generalized to encompass a wider class of learning rules. For example, the *outstar* learning rule [13]:

$$\dot{w}_{ij} = x_i (\lambda y_j - \alpha w_{ij}(t)), \quad (10)$$

cannot be directly mapped to the existing learning procedure due to the multiplication of weights w_{ij} by the inputs x_i . As in the case of Hebbian learning with postsynaptic normalization, Eq.10 can be implemented by a suitable modification of the network topology, but this would reduce the efficiency of the framework.

Our group recently introduced a new general form of learning law capable of handling several classes of learning rules, including Hebb rule derivatives, threshold-based rules, input reconstruction-based rules and trace-based rules [14]. Crucially, this generalization is achieved by inserting only one additional postsynaptic feedback term to the already existing two terms (h_j and g_j) of the current procedure, making it a suitable candidate for hardware acceleration as mentioned in Section 2.

4 Conclusions

In this paper, we investigated the kind of issues that are likely to be faced when implementing learning in a large-scale neural model instantiated on a distributed, heterogeneous network. In particular, we argue that learning calculations should be local to the partial inference processors, should minimize data transfer to and from signal function processors, and should be implementable in a common general form equation. Our current efforts are aimed at implementing such a general form, which will facilitate the study of learning in large-scale neural network.

References

1. Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J.M., Diesmann, M., Morrison, A., Goodman, P.H., Harris Jr., F.C., Zirpe, M., Natschlag, T., Pecevski, D., Ermentrout, B., Djurfeldt, M., Lansner, A., Rochel, O., Vieville, T., Muller, E., Davison, A.P., El Boustani, S., Destexhe, A.: Simulation of networks of spiking neurons: A review of tools and strategies. *J. Comp. Neurol.* 23, 349–398 (2007)
2. O'Reilly, R.C., Munakata, Y.: *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. MIT Press (2000)
3. Markram, H.: The blue brain project. *Nat. Rev. Neurosci.* 7, 153–160 (2006)
4. Snider, G.: Intelligent Machines built with Memristive Nanodevices. In: 12th IEEE International Workshop on Cellular Nanoscale Networks and their Applications, CNNA (2010)
5. Haykin, S.: *Neural networks: A comprehensive foundation*. Prentice-Hall (1999)
6. Levy, W.B., Steward, O.: Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience* 8, 791–797 (1983)
7. Izhikevich, E.: Large-scale model of the mammalian thalamocortical systems. *PNAS* 105, 3593–3598 (2008)
8. Hyvärinen, A., Oja, E.: Independent component analysis by general nonlinear Hebbian-like learning rules. *Signal Processing* 64, 301–313 (1998)
9. Grossberg, S.: Adaptive pattern classification and universal recoding: I Parallel development and coding of neural feature detectors. *Biol. Cybern.* 23, 121–134 (1976)
10. von der Marlsburg, C.: Self-organization of orientation-selective cells in the striate cortex. *Kybernetik* 15, 85–100 (1973)
11. Mikkulainen, R., Bednar, J.A., Choe, Y., Sirosh, J.: *Computational Maps in the visual cortex*. Springer (2005)
12. Bienenstock, E.L., Cooper, L., Munro, P.: Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neurosci.* 2, 31–48 (1982)
13. Grossberg, S.: Some nonlinear networks capable of learning a spatial pattern of arbitrary complexity. *PNAS* 59, 368–372 (1968)
14. Gorchetchnikov, A., Versace, M., Ames, H., Léveillé, J., Yazdanbakhsh, A., Chandler, B., Mingolla, E., Snider, G.: General form of learning algorithms for neuromorphic hardware implementation. In: *The International Computational Neuroscience Meeting (CNS)*, San Antonio, TX (July 2010)

Bio-inspired Robotics Hands: A Work in Progress

Ebrahim Mattar¹ and Khaled Al Mutib²

¹ University of Bahrain, P. O. Box 13184, Kingdom of Bahrain

² King Saud University, P. O. Box 51178, Kingdom of Saudi Arabia
ebrgallaf@eng.uob.bh, mutib@ccis.ksu.edu.sa

Abstract. Towards dexterous manipulation and human-like grasping capabilities, a new phase of robotics hands have been developed recently with biomimetically oriented functionalities. This manuscript is an attempt to survey in details biomimetic based dexterous robotics hands. In particular, the article focuses on a number of developments that have been taking place over the last two decades, in recent development related to this biomimetic field of research.

Keywords: Biomimetic, Dexterous Manipulation, Biomimetic Robotics.

1 Introduction

Over the past two decades, large number of dexterous robotics hands have been developed that explicitly emulate human like hand shape and movements. However, it was clear that biological functionalities could not be fully emulated due to lack of the right technologies. Over past few years, engineering such biomimetic intelligent creatures, such as robots, was hindered by physical and technological constraints and limitations. Literatures in this vital field have been focused since early 1990, when the Utah and JPL hand researchers published their research work in progress in 1984 (refer to Fig. 1., [1,2]). Biomimetic robot hands, as indicated by Yoseph B. [3], have been introduced lately over the last decade. This is due to a number of potential advantages over purely mechanical robot hands, as known here as the classical hand. Within this article, we shall survey a number of potential research frameworks that have been focused lately towards Biomimetic robot hands design and implementation.

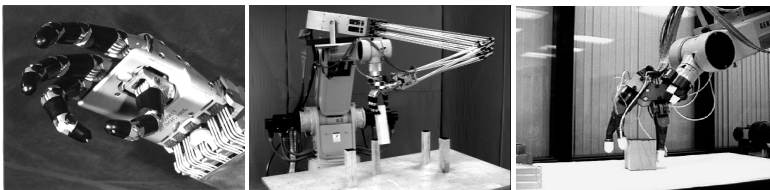


Fig. 1. The Utah and JPL hands, early attempts to emulate human grasps, [1,2]

2 Bio-inspired Dexterous Robotics Hands

2.1 Hand Biomimetic Compliance Control Structures of Human Finger

In their paper, Byoung K. et. al. [4], for an object grasped by a robot hand to work in compliance control domain, they analyzed necessary condition for successful stiffness modulation in the operational space. They proposed a new compliance control method for robot hands which consist of two steps. RIFDS (Resolved Inter-Finger Decoupling Solver) is to decompose the desired compliance characteristic specified in the operational space into the compliance characteristic in the fingertip space without inter-finger coupling, and RIJDS (Resolved Inter-Joint Decoupling Solver) is to decompose the compliance characteristic in the fingertip space into the compliance characteristic in the joint space without inter-joint coupling. It was found, a finger structure should be biomimetic in the sense that either kinematic redundancy or force redundancy are required to implement the proposed compliance control scheme.

2.2 Control of A Multi-finger Prosthetic Hand

In [5], Craelius W. et. al. have developed a control of multi-finger prosthetic hand. The prosthetic hand is controlled by extrinsic flexor muscles and tendons of the metacarpal-phalangeal joints. The hand uses Tendon-Activated Pneumatic control and has provided most subjects, including amputees and those with congenital limb absence, control of multiple fingers of the hand. This is illustrated in Fig. 2.

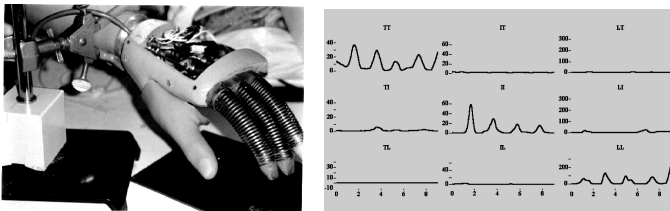


Fig. 2. (a) Linear actuators provided movement of 3 independent fingers. (b) Signals derived from TAP sensors, 9-sec. period of repetitive finger flexions, [5].

2.3 A Biomimetic Controller for a Multifinger Prosthesis

In [6], Abboudi R. et. al. have presented a novel controller for a multifinger hand prosthesis, and developed it and tested to measure its accuracy and performance in transducing volitional signals for individual "phantom" fingers. Pneumatic sensors were fabricated from open-cell polymeric foam, and were interposed between the prosthetic socket and superficial extrinsic tendons associated with individual finger flexion. Test subjects were prompted to step individual fingers or combinations thereof to perform either taps or grasps. Sensor outputs were processed by a computer that controlled motions of individual fingers on a mechanical prosthesis. Trials on three upper-limb amputees illustrated that after brief training sessions, the TAP controller was effective at producing voluntary flexions of individual fingers and grasping motions.

2.4 Tendon and Muscle Force Requirements for Humanlike

In [7], Pollard N. And Richards G. have stated that adapting human examples to a robot manipulator is a complex problem, however, in part due to differences between human and robot hands. Force transmission mechanisms in robot fingers are generally symmetric about flexion / extension axes, but in human fingers they are focused toward flexion, Fig. 3. The reserach describes how a tendon driven robot finger can be optimized for force transmission capability equivalent to the human index finger. They have shown that two distinct tendon arrangements that are similar to those that have been used in robot hands can achieve the same range of forces as the human finger with minimal additional cost in total muscle force requirements.

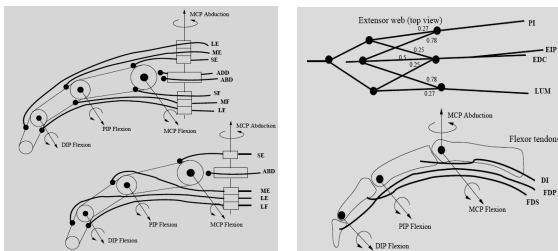


Fig. 3. (a) : 2N and N+1 tendon fingers. (b) : Flexor and extensor tendons of the index, [7].

2.5 Principal Components Analysis Based Prosthetic Hand

In [8], Matrone G. et. al. presented prosthetic hands and controlled by means of non-invasive interfaces based on electromyography (EMG). Driving a multi degrees of freedom (DoF) hand for achieving hand dexterity implies to selectively modulate many different EMG signals in order to make each joint move independently, and this could require significant cognitive effort to the user. A principal components analysis based algorithm is used to drive a 16 DoFs underactuated prosthetic hand prototype with a two dimensional control input, in order to perform the three prehensile forms mostly used in Activities of Daily Living (ADLs). Principal components set has been derived directly from the artificial hand by collecting its sensory data while performing 50 grasps, and subsequently used for control. This is shown in Fig. (8).



Fig. 4. Power, precision and lateral grasp, as in [8]

2.6 Biomimetic Tactile Sensor for Control of Grip

In [9], Wettels N. et. al. , are developing a novel, robust tactile sensor array that mimics the human fingertip and its distributed set of touch receptors. The mechanical components are similar to a fingertip, with a rigid foundation surrounded by a weakly conductive fluid contained within an elastomeric skin, Fig. 5. It uses the deformable properties of the finger pad as part of the transduction process. Multiple electrodes are mounted on the surface of the rigid core and connected to impedance measuring circuitry within the core. External forces deform the fluid path around the electrodes, resulting in a distributed prototype of impedance changes containing information about those forces and the objects that applied them.

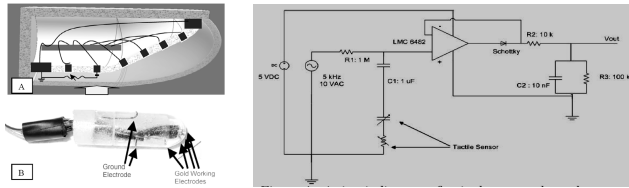


Fig. 5. eft : a) Drawing of biomimetic tactile sensor. b) Sensor prototype core with “skin” removed, Right : A circuit diagram for single sensor channel, [9].

2.7 A Shape Memory Alloy-Based Tendon-Driven Actuation

According to Bundhoo V. et. al. [10], a new biomimetic tendon-driven actuation system for prosthetic and wearable robotic hand applications is presented. It is based on the combination of compliant tendon cables and one-way shape memory alloy (SMA) wires that form a set of agonist–antagonist artificial muscle pairs for the required flexion/extension or abduction/adduction of the finger joints, Fig. 6. The performance of the proposed actuation system is demonstrated using a 4 degree-of-freedom (three active and one passive) artificial finger test-bed, also developed based on a biomimetic design approach. A microcontroller-based pulse-width-modulated proportional-derivation (PWM-PD) feedback controller and a minimum jerk trajectory feedforward controller are implemented and tested in an ad hoc fashion to evaluate the performance of the finger system in emulating natural joint motions.

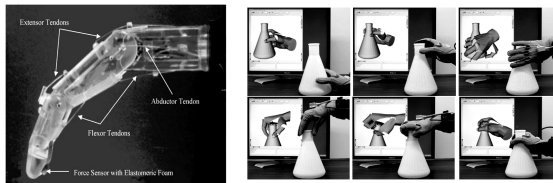


Fig. 6. Left : An artificial finger, [10]. Right : Examples of interactive grasp planning using input provided by an operator, [11].

2.8 Biomimetic Grasp Planning for Cortical Control of a Robotic Hand

In [11], Ciocarlie M. et. al., and in their paper, they outlined a grasp planning system designed to augment the cortical control of a prosthetic arm and hand. A key aspect of this task is the presence of on-line user input, Fig. 6., which will ultimately be obtained by identifying and extracting the relevant signals from brain activity. The grasping system can combine partial or noisy user input and autonomous planning to enable the robot to perform stable grasping tasks. They used principal component analysis applied to the observed kinematics of physiologic grasping to reduce the dimensionality of hand posture space and simplify the planning task for on-line use. The planner then accepts control input in this reduced-dimensionality space, and uses it as a seed for a hand posture optimization algorithm based on simulated annealing.

2.9 Grip Control Using Biomimetic Tactile Sensing Systems

Wettels N. et. al. in [12], have presented a proof-of-concept for controlling the grasp of an anthropomorphic mechatronic prosthetic hand by using a biomimetic tactile sensor, Bayesian inference and simple algorithms for estimation and control. The sensor takes advantage of its compliant mechanics to provide a tri-axial force sensing end-effector for grasp control, Fig.7. By calculating normal and shear forces at the fingertip, the prosthetic hand is able to maintain perturbed objects within the force cone to prevent slip. A Kalman filter is used as a noise-robust method to calculate tangential forces. Biologically-inspired algorithms and heuristics are presented that can be implemented on-line to support rapid, reflexive adjustments of grip.

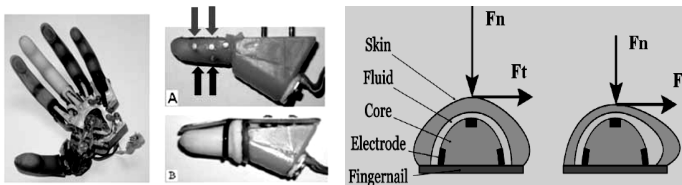


Fig. 7. Left: Otto Bock M2 Hand. Tac prototype sensor array Right: Effect of normal and tangential forces on different electrodes, [12].

2.10 Development of Bio-mimetic Robot Hand Using Parallel Mechanisms

In [13] Lee S. et. al., describe a development of bio-mimetic robot hands and its control scheme. Each robot hand has four under-actuated fingers, which are driven by two linear actuators coupled, Fig. 8. Each fingertip can reach toward objects by curved surface workspace in 3D-space. The robot hand was designed considering the dexterity and the size suited for human tools and has tactile sensors equipped on the fingertips of each finger. The robot hand has 4 fingers with totally nine DOFs including two linear actuators and linkage knuckles. In the former part of this paper, the design of the robot hand is presented. And in the latter part of the paper, computational simulations are described. The simulations show the performance of the robot hand to manipulate tools of various shapes.

2.13 Design and Control of a Shape Memory Alloy Based Dexterous Hand

In [16], Price A. et. al., modern externally powered upper-body prostheses are conventionally actuated by electric servomotors. Although these motors achieve reasonable kinematic performance, they are voluminous and heavy. Deterring factors such as these lead to a substantial proportion of upper extremity amputees avoiding the use of their prostheses. It was found, it is apparent that there exists a need for functional prosthetic devices that are compact and lightweight. The realization of such a device requires an alternative actuation technology, and biological inspiration suggests that tendon based systems are advantageous. Shape memory alloys are a type of smart material that exhibit an actuation mechanism resembling the biological equivalent.

2.14 Bio-inspired Grasp Control In A Robotic Hand (Massive Sensorial Input)

In [17] Ascari L. et. al., found that no robotic tools able to perform an advanced control of the grasp as, for instance, the human hand does, have been demonstrated to date. In their paper a bio-inspired approach to tactile data processing has been followed in order to design and test a hardware–software robotic architecture that works on the parallel processing of a large amount of tactile sensing signals. The working principle of the architecture bases on the cellular nonlinear/neural network (CNN) paradigm, while using both hand shape and spatial–temporal features obtained from an array of microfabricated force sensors, in order to control the sensory-motor coordination of the robotic system. Prototypical grasping tasks were selected to measure the system performances applied to a computer-interfaced robotic hand.

2.15 Adaptive Grasping by Multi Fingered Hand

In [18] Takahashi T. et. al., proposed a new robust force and position control method for property-unknown objects grasping. The proposed control method is capable of selecting the force control or position control, and smooth and quick switching according to the amount of the external force. The proposed method was applied to adaptive grasping by three-fingered hand which has 12 DOF, and the experimental results revealed that the smooth collision process and the stable grasping is realized even if the precise surface position, the mass and the stiffness are unknown. This is shown in Fig. 9.

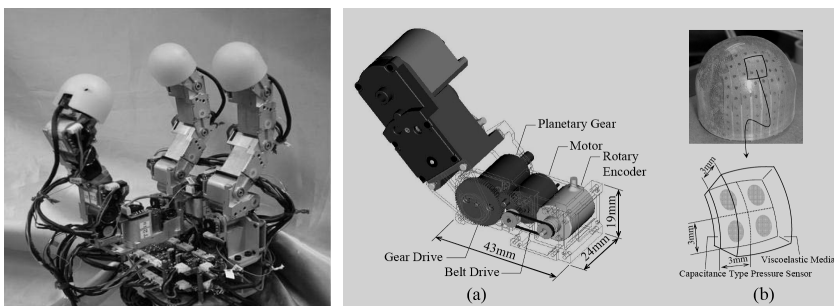


Fig. 9. (a) Joint Drive Unit (b) Tactile sensor, [18]

3 Conclusion

This article is a part of work in progress that is prepared to be looking into and surveying a number of research efforts towards building biomimetic based dexterous robotics hands. The study has indicated that, there are tremendous number of efforts towards building dexterous robotics multi-fingered hand with biomimetic based ideas and initiatives. In addition, it was shown that research is even moving towards muscles type hand fingers, which means moving totally from the concept of motor driven fingers movements and grasping. In terms of bio-inspired robotics hand design and control, technology is promising us with vital solutions in this directions.

References

1. Jacobsen, S.C., Wood, J.E., Knutti, D.F., Biggers, B.: The UTAH/M.I.T. Dexterous Hand: Work in Progress. *International Journal of Robotics Research* 3, 21–50 (1984)
2. Matthew, M., Salisbury, K.: *Robot Hands and the Mechanics of Manipulation*. MIT Press, USA (1985)
3. Yoseph, B.: Biologically Inspired Intelligent Robotics, Paper 5051-5002. In: *Proceedings of the SPIE Smart Structures Conference*, San Diego, CA (2003)
4. Byoung, K., Byung, Y., Hong, O., Sang, H., Yeh, S.: Biomimetic Compliance Control of Robot Hand by Considering Structures of Human Finger. In: *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3879–3886 (2000)
5. Craelius, W., Abboudi, R., Newby, N.: A Biomimetic Controller for a Multifinger Prosthesis. *IEEE Trans. Rehabil. Engineering* 7(2), 121–129 (1999)
6. Abboudi, R., Glass, C., Newby, N., Flint, J., Craelius, W.: A Biomimetic Controller for a Multifinger Prosthesis. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, pp. 3417–3422 (2009)
7. Pollard, N., Richards, G.: Tendon Arrangement and Muscle Force Requirements for Humanlike Force Capabilities in a Robotic Finger. In: *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, ICRA 2002, Washington, DC, USA (2002)
8. Matrone, G., Cipriani, C., Secco, E., Magenes, G., Carrozza, G.: Principal Components Analysis Based Control of a Multi-dof Underactuated Prosthetic Hand. *Journal of NeuroEngineering and Rehabilitation* 7(16) (2010)
9. Wettels, N., Popovic, D., Santos, V., Loeb, G.: Biomimetic Tactile Sensor for Control of Grip. In: *IEEE 10th International Conf. on Rehabilitation Robotics* (2007)
10. Bundhoo, V., Haslam, E., Birch, B., Park, E.: A Shape Memory Alloy-Based Tendon-Driven Actuation System for Biomimetic Artificial Fingers, part I: Design and Evaluation. *Journal of Robotica*, 1–16 (2008)
11. Ciocarlie, M., Clanton, S., Spalding, M., Allen, P.: Biomimetic Grasp Planning for Cortical Control of a Robotic Hand. In: *Proc IEEE/RJS Intl Conf. on Intelligent Robots and Systems*, pp. 2271–2276 (2008)
12. Wettels, N., Parnandi, A., Moon, J., Loeb, G., And Sukhatme, G.: Grip Control Using Biomimetic Tactile Sensing Systems, *Mechatronics*. *IEEE/ASME Transactions* 14(6), 718–723 (2009)

13. Lee, S., Noh, S., Lee, Y., Park, J.: Development of Bio-mimetic Robot Hand Using Parallel Mechanisms. In: Proceedings of the IEEE International Conf. on Robotics and Biomimetics, Guilin, China, December 19-23 (2009)
14. Petroff, N.: Biomimetic Sensing For Robotics Manipulation. A Dissertation Submitted to the Graduate School of the University of Notre Dame in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, Indiana (2006)
15. Edinc, B., Ascarib, L., Beccaia, L., Roccellaa, S., Cabibihana, J., Carrozza, M.: Bio-Inspired Sensorization of a Biomechatronic Robot Hand for the Graspan - Lift Task. *Brain Research Bulletin* 75(6), 785–795 (2008)
16. Price, A., Jnifene, A., Naguib, H.: Design and Control of a Shape Memory Alloy Based Dexterous Robot Hand. *Journal of Smart Materials and Structures* 16(4) (2007)
17. Ascari, L., Bertocchi, U., Corradi, P., Laschi, C., Dario, P.: Bio-inspired Grasp Control in a Robotic Hand with Massive Sensorial Input. *Journal of Biological Cybernetics* 100(2) (2009)
18. Takahashi, T., Tsuboi, T., Kishida, T., Kawanami, Y., Shimizu, S., Iribe, M., Fukushima, T., Fujita, M.: Adaptive Grasping by Multi Fingered Hand with Tactile Sensor, Based on Robust Force and Position Control. In: 2008 IEEE International Conference Robotics and Automation, Pasadena, CA, USA (2008)

Bio-inspired Routing and Wavelength Assignment Algorithm for Optical Mesh Networks

Daniel Padilla, Edward Guillen, and Carlos Ramos

Nueva Granada Military University
GISSIC Investigation Group, Bogotá D.C., Colombia
{gissic,edward.guillen}@unimilitar.edu.co
<http://gissic.umng.edu.co/>

Abstract. Optical networks have been widely implemented and they are usually used as the backbone of high-capacity telecommunication infrastructures. However, the nowadays traffic demand over these networks has increased in a way that congestion over the architecture is considerable. Due to the high costs of implementing new optical links, the routing and wavelength assignment process become an important point of analysis. This paper presents a time-efficient routing and wavelength assignment algorithm for optical mesh networks by applying bio-inspired calculation methods. The algorithm's main goal is to reduce computational costs and time during lightpath selection and wavelength assignment.

Keywords: Bio-Inspiration, Genetic Algorithms, Optical Mesh Networks, Routing, Wavelength Assignment.

1 Introduction

Optical networks work over a wavelength-division multiplexing –*DWM*– technology which allows a single optical channel to carry multiple signals and enable bidirectional communication links by multiplexing carrier signals into different wavelengths of laser light [1].

An optical network is usually composed by a mesh of interconnected routing nodes. For each new request between source and destination nodes it must be given a route and a wavelength. This optical network's challenge is known as the routing and wavelength assignment –*WRA*– problem [2,3]. The main goal of a *WRA* algorithm is to maximize the number of established connections within the optical mesh network [4].

2 A Useful Representation of Optical Mesh Networks

Optical mesh networks can be easily represented by using matrices. For a given network, a matrix is built in order to represent the presence or absence of

direct links between nodes. The links $-L-$ matrix is then given by the conditions represented in (1).

$$L_{(i,j)} = \begin{cases} 0 & \text{if } n_i \text{ and } n_j \text{ are the same node} \\ 1 & \text{if there is direct link between } n_i \text{ and } n_j \\ 1000 & \text{if there is no direct link between } n_i \text{ and } n_j \end{cases} \quad (1)$$

Then, $L = 1$ means that two different nodes are directly communicated and $L = 1000$ represents the absence of direct link between nodes. The final representation of a n -nodes optical network is finally given by (2).

$$L = \begin{bmatrix} 0 & L_{(1,2)} & L_{(1,3)} & \dots & \dots & \dots & L_{(1,n)} \\ L_{(2,1)} & 0 & L_{(2,3)} & \dots & \dots & \dots & L_{(2,n)} \\ L_{(3,1)} & L_{(3,2)} & 0 & \dots & \dots & \dots & L_{(3,n)} \\ \dots & \dots & \dots & 0 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ L_{(n,1)} & L_{(n,2)} & L_{(n,3)} & \dots & \dots & \dots & 0 \end{bmatrix} \quad (2)$$

3 Routing and Wavelength Assignment Algorithm

The *RWA* problem is approached by dividing the algorithm into two parts. The first one solves the routing issues and the second part deals with the wavelength assignment process.

3.1 Routing Algorithm

Let us assume there is an optical network with n routing nodes and a communication request between a source node $-n_s-$ and a destination node $-n_d-$ is received.

In order to find the best path $-BP-$ throughout the network, the routing algorithm defines an initial matrix, which we will refer to as *BP* matrix. The *BP* matrix is given by a $m \times n$ array, where m represents the number of initial possible solutions to the problem.

The matrix's content is set under the criteria given by (3).

$$BP_{(i,j)} = \begin{cases} n_s & \text{if } j = 1 \\ \text{random}(1 \sim n) & \text{if } 1 < j < n \\ n_d & \text{if } j = n \end{cases} \quad (3)$$

Every single row, in fact, represents a possible path between n_s and n_d and this is why the first and last columns of the matrix are fixed and respectively

identify the source and destination nodes. The remaining data on the BP matrix is then filled with random values between 1 and n . The final representation of BP matrix is shown in (4).

$$BP = \begin{bmatrix} n_s & r(1,2) & r(1,3) & \dots & r(1,n-1) & n_d \\ n_s & r(2,2) & r(2,3) & \dots & r(2,n-1) & n_d \\ n_s & r(3,2) & r(3,3) & \dots & r(3,n-1) & n_d \\ n_s & \cdot & \cdot & \dots & \cdot & n_d \\ n_s & \cdot & \cdot & \dots & \cdot & n_d \\ n_s & \cdot & \cdot & \dots & \cdot & n_d \\ n_s & r(m,2) & r(m,3) & \dots & r(m,n-1) & n_d \end{bmatrix} \tag{4}$$

Once the initial matrix is defined, the algorithm compute the path cost $-PC$ over every single proposed solution.

The PC mainly depends on the network links matrix represented by (2) and is given by the total sum of the cost for each hop between nodes, as shown in equation (5).

$$PC_i = \sum_{k=1}^{n-1} L_{(BP_{(i,k)}, BP_{(i,k+1)})} \tag{5}$$

The algorithm’s main goal is to minimize the PC and indeed three genetic operators are applied over the BP matrix. **Selection:** the $m/2$ paths that present the lowest costs are selected to be part of a new generation of best paths. **Crossover:** $m/2$ new paths are obtained by crossing the previously selected paths at a random point within its second and $(n-1)$ -th column. **Mutation:** the nodes within the paths matrix have a given probability of changing.

After the process, a new BP matrix has been created and its average PC is lower than the previous one’s. The whole process can be iterated as many times as required in order to get closer to the best path solution.

3.2 Wavelength Assignment Algorithm

The second part of the RWA problem is solved by applying the well-known First Fit algorithm [5]. First Fit is one of the most common methods for wavelength assignment and consists on simply choosing the available wavelength with the lowest index. It has to be here considered that the assigned wavelength must be defined for the entire lightpath.

4 Conclusion

The bio-inspired routing and wavelength assignment algorithm represent a time-efficient solution for the optical mesh network’s RWA problem. The goal of maximizing the number of possible connections within an optical telecommunication architecture is finally achieved by minimizing the time taken during the genetic algorithms-based calculation of best paths and wavelength assignments.

References

1. Ramaswami, R., Sivarajan, K.N.: *Optical Networks, a Practical Perspective*. Academic Press A. Harcourt Science and Technology Compan., USA (1998)
2. Zang, H., Jue, J.P., Mukherjee, B.: A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks. *Opt. Net. Mag.* 1(1), 47–60 (2000)
3. Hu, J.Q., Leida, B.: Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks. In: *Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1(1), p. 501 (March 2004)
4. Barry, R.A.: *Wavelength Routing for All Optical Network*. Massachusetts Institute of Technology (1993)
5. Sun, X., Li, Y., Lambadaris, I., Zhao, Y.: Performance Analysis of First-Fit Wavelength Assignment Algorithm in Optical Networks. In: *Proceedings of the Seventh International Conference on Telecommunications*, vol. 2(1), pp. 403–409 (June 2003)
6. Souza, G.: *Using genetic Algorithms in constrained routing and wavelength assignment*. Optinet, State University of Campinas
7. Zang, H., Jue, J.P.: *Dynamic Lightpath Establishment in Wavelength-Routed WDM Networks*. *IEEE Communications Magazine* (September 2001)
8. Kuri, J., Puech, N., Gagnaire, M., Dotaro, E., Douville, R.: Routing and Wavelength Assignment of Scheduled Lightpath Demands. *IEEE Journal on Selected Areas in Communications* 21(8) (October 2003)
9. Ozdaglar, A.E., Bertsekas, D.P.: Routing and Wavelength Assignment in Optical Networks. *LIDS Report*, p. 2535 (December 2001)
10. Nayak, T.K., Sivarajan, K.N.: Routing and Dimensioning in Optical Networks Under Traffic Growth Models: An Asymptotic Approach. *IEEE Journal on Selected Areas in Communications* 21(8) (October 2003)
11. Ramaswami, R., Sivarajam, K.N.: Design of Logical Topologies for Wavelength-Routed Optical Networks. *IEEE Journal on Selected Areas in Communications* 14(5) (June 1996)

Artificial Immune Systems – AIS as Security Network Solution

Edward Guillen¹ and Rafael Paez²

¹Military University “Nueva Granada”, Bogota, Colombia
edward.guillen@unimilitar.edu.co

²Javeriana University, Bogota, Colombia
paez-r@javeriana.edu.co

Abstract. Network security attacks have a dynamical performance and their rate of change is commonly faster than protection technologies. The defense mechanisms usually act with reactive non proactive solutions in a non-self learning procedure. However, bio-inspired methods such as AIS could give a new dynamical method to defend entire data network from malicious attacks. This short paper briefly analyzes the possible use of AIS in secure network architectures to be implemented with future research projects.

Keywords: AIS, BIS, Network Security.

1 Introduction

The Artificial Immune Systems –AIS began in the early 90s as a branch of the Computational Intelligence –CI [1], and their main goal is to develop computational methods inspired by Biological Immune Systems –BIS in order to solve computational problems [2]. The natural characteristics of BIS are their ability of pattern matching by the recognition of foreign cells that are mixed with the cells that belong to the body [3]. Other important features of BIS includes: feature extraction, memory, learning, and distributed nature [2]. When applying these concepts to computational problems, it is possible to find interesting solutions with the analogies found on BIS, for example: computer security applications, machine learning, detection in time series, anomaly detection, and chemical spectrum recognition. [4], [5], [6], [7], [8], [9]. We want to focus the AIS solutions into security applications by using analogies between BIS and the behavior of malicious attacks over telecommunication networks, in order to propose investigation projects to solve the challenge of protecting information.

2 Security Threats and BIS

Malicious software has a similar behavior than its biological counterpart. Trojan horses get into computer network by simulating a trustable behavior and turns into

malicious application inside the network. Worms can act by a mechanism of self-replication affecting the protection system and creating backdoors for the entrance of more dangerous attacks. Rootkits help malware to remain hidden. A bug is an error into the normal function of a system. A Botnet acts in a highly distributed system with a well synchronized performance in order to achieve a massive attack.

In the BIS, the central lymphoid organs generate immune cells and include bone marrow and the thymus [2]. There are a wide range of immune cells with different task in order to act in an immune response. Examples of these cells are stem cell, Lymphocytes, B and T cell progenitor, Natural killer cell, Neutrophil, Eosinophil and so on. New network security architecture should have a structure that could produce digital cell detectors for an entire network and not just a passive detection.

It is possible to map pathogens and antigens with malware and network attacks, immune cells with detectors, proteins with strings, immunological response with elimination strategies [4].

3 Conclusion

With AIS, it is possible to propose an interactive security scheme, not only at the final equipment but also in a Bio-inspired complete network defense architecture to be integrated with classical and new security appliances.

References

1. Dasgupta, D.: Advances in Artificial Immune Systems. IEEE Computational Intelligence Magazine (November 2006)
2. Dasgupta, D., Nino, L.F.: Immunological Computation, Theory and Applications. CRC Press, Boca Raton (2009)
3. Engelbrecht, A.: Computational Intelligence: An Introduction, 2nd edn. John Wiley & Sons Ltd., England (2007)
4. Harmer, P., Williams, P., Gunsch, G., Lamont, G.: An Artificial Immune System Architecture for Computer Security Applications. IEEE Transactions on Evolutionary Computation 6(3), 252–280 (2002)
5. Zhou, X.: Evolutionary Algorithm and its Application in Artificial Immune System In: The Second International Symposium on Intelligent Information Technology Application (2008)
6. Dasgupta, D., Forrest, S.: Novelty Detection in Time Series Data using Ideas from Immunology. In: ISCA 5th International Conference on Intelligent Systems, Reno, Nevada, June 19-21 (1996)
7. Dasgupta, D.: Using Immunological Principles in Anomaly Detection. In: The Artificial Neural Networks in Engineering (ANNIE 1996), St. Louis, USA, Nov. 10–13 (1996)
8. Cao, Y.D., Dasgupta, D.: An Immunogenetic Approach in Chemical Spectrum Recognition. In: Ghosh, Tsutsui (eds.) Advances in Evolutionary Computing, ch. 36. Springer-Verlag, Inc. (January 2003)
9. Timmis, J., Neal, M., Knight, T.: AINE: Machine Learning Inspired by the Immune System. IEEE Transactions on Evolutionary Computation (June 2002)

WebSeA: A Secure Framework for Multi-site Knowledge Representation in Software Engineering

Muhammad Ilyas¹, Ahmad Ali², and Josef Kueng¹

¹ FAW Institute, University of Linz, Altenberger Str. 69, A-4040 Linz, Austria

²Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan
{milyas, jkueng}@faw.uni-linz.ac.at, engr.ahmadali@yahoo.com

Abstract. Multi-site software engineering is one of the most extensively used mean of sharing and communicating information of “software projects” to remotely located stake holders. It involves different domains and large number of users. This requires different security measures, to interact and protect relevant data sources. That is why, the issue of securing the data from unauthorized access is very critical. This research work elaborates a secure framework, named WebSeA, to counter the security measures of multi-site software engineering. WebSeA application and WebSeA services are also developed for the practical implementation of proposed WebSeA framework.

Keywords: Multi-site software engineering, WebSeA, Security measures.

1 Introduction

Multi-site software engineering is an approach dealing with software projects that are carried out by multiple teams over multiple sites where team members use different tools, methods and platforms. Exchange of semantics among team members can be complicated if software engineering principles and discipline are not understood and followed exactly. They could use a particular text as their personal guide, and when they share, their own terminology and knowledge-base could be inconsistent in the perception of software engineering theories and practices. As a result, multiple practical issues arise that need to be explored.

Handling the shortcomings linked with remote communication is a question in multi-site software engineering. A common or unique communication language is necessary for knowledge sharing. This enables efficient ways of reaching an agreement of understanding which is of useful to remote team members in a multi-site software engineering. Successful secure communication and coordination among multiple sites is very important for worldwide software development. Due to rapid increase of business at global level, software engineering projects demonstrates importance while dealing with multi-site project development as well as multi-lingual project management. But as here internet is used as a communication medium, there must be preventive measure in terms of security of such communication. Users with mal-intentions may alter such data and create in-consistencies in mutually agreed

project information. To handle such potential vulnerabilities, we have given an idea of WebSeA, a framework for a secure and authentic solution for web based applications. Moreover, we have developed WebSeA application and its services, and the practical implementation of WebSeA framework.

Section 2 gives some background information. Section 3 presents the WebSeA framework, WebSeA application and WebSeA Service, where as Section 5 focuses on conclusion and future work.

2 Background

Sidharth et. al. [1] has worked on web based security issues and presented an integrated application and protocol-based framework. IAPF approach based framework provides protection against vulnerabilities in the UDDI/ WSDL protocol, and prevention against DoS/ DDoS based attacks.

Singh [2] has addressed security requirements of grid services and presented a layered grid security model which deals with web services security specifications. This is a five layered model which provides advanced security features like dynamic trust establishment, privacy enforcement, authorization, secure logging, management of certificates, audit trails and key distribution etc. Provision of these features is based on policies like WS-Policy, WS-Trust, WS-Federation and WS-Secure Conversation specifications. The Security Application Layer provides security services are provided by security function so that other domains can also access these features.

Similarly Zhang [3] has presented an integrated security framework for web services. They have focused on different threats to web services like loss of confidentiality, principal spoofing, falsified messages, forged claims and services denial etc. In their model, they have given the concept of usage of Role-based access control, attribute-based access control and risk-adaptive access control respectively.

3 WebSeA Framework

WebSeA framework focuses on techniques to safeguard the web repository to ensure its consistency i.e. use of spam control pictures, alpha-numeric passwords, user identification and authentication, password aging and password social engineering, cryptography in security, personal identity verification, electronic authentication and user authentication on web. Figure 1 represents the WebSeA framework.

This framework supports three types of users. First type is internet users, who just view or browse available web resources. These users do not require registration to view the available web resources. Second type of users requires registration through WebSeA framework to view and participate for the improvement of web repository. Once registered, the user can raise issues which after categorization and filtrations are directed to respective think tanks for further evaluation. However, to prevent unauthorized and computer-based DDOS attacks, we propose in WebSeA framework that each raised issue must accompany spam control image data.

Third type of users is most privileged being software team members working on software projects on multiple sites. Baseline privileges of these users are similar to that of the second type of users discussed above. However, these users may also request for the updating, deletion and append rights on the web repository.

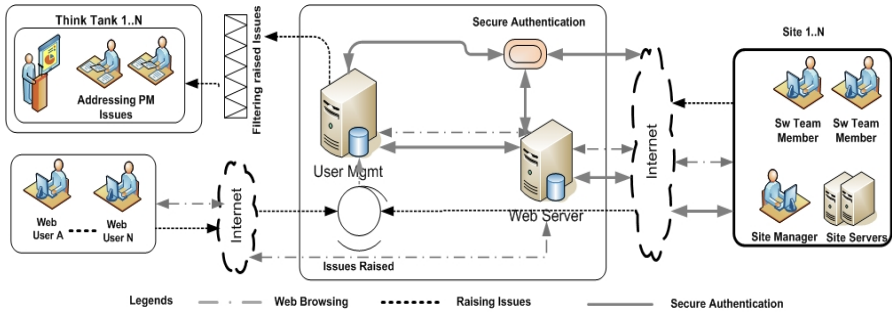


Fig. 1. WebSeA Framework

Here, figure 2 shows the proposed Authentication process of WebSeA framework. The secure authentication process is explained below:

- S-1 Privileged signed-in Member Requests for Admin Rights followed by a spam controller
- S-2, S-3 Generate Dynamic UID and PWD, Encrypt UID, PWD and Session Duration using User’s Public Credentials and email to members registered e-mail.
- S-4 Member Get and decrypt UID, PWD and Session Duration and again sign in with new parameters.
- S-5 Server Create a session with admin rights and Connects to Web Repository in editable mode.
- S-6 User is connected to Web Repository Server in editable mode.

Here second and third step are the key innovation in WebSeA framework. For the implementation of WebSeA framework, we have developed an application. WebSeA Application is a web base application, implementing all the processes discussed in WebSeA framework, however, WebSeA Web Service is the backbone of WebSeA Application. Privileged users request for editing rights, these requests are listened and entertained by WebSeA Web Service.

Any user can access web repository by accessing WebSeA Website, however, only registered users can raise issues. These issues are categorized and after getting filtered sent to respective category think tanks for be resolved.

The privileged users, like software project teams can also request for appending, editing and deleting web repository data. However, this access is granted by sending encrypted access key to the user’s email box. These users have decrypting application on their remote sites. Figure 3 shows the screen shot of decrypting application.

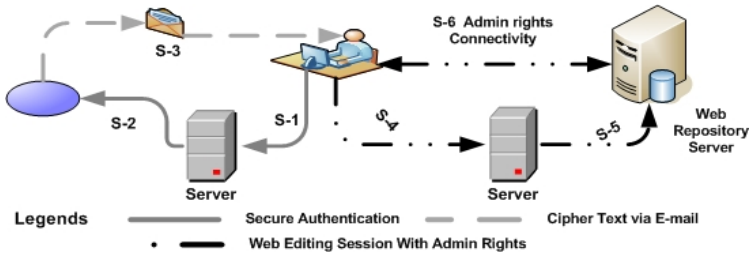


Fig. 2. WebSeA Secure Authentication Process



Fig. 3. WebSeA Decryption Application

Users after receiving encrypted key via email use decrypting application and their pass key can get dynamically generated username and password, however, age of this username and password is also very limited, generally 10 to 20 minutes (or as decided by the organization). This username and password will no more be valid once that session gets expired. Moreover, user must also have to pass through the same spam image control check each time he/she requests for editing rights.

4 Conclusion and Future Work

WebSeA framework has been introduced as a mean of secure web based knowledge representation. Usage of spam control images, dynamic generation of usernames and passwords for a limited time, usage of non-dictionary words as usernames and passwords, sending of encrypted key via email and decrypting application are the features used by remote site users under the supervision of site manger. User needs encrypted key and passkey to execute the required action. All the above mentioned features have been implemented in WebSeA framework against authentication vulnerabilities. Currently, WebSeA framework is based on symmetric encryption methodology. In future, we plan to implement the same by using Asymmetric encryption methodology.

References

1. Sidharth, N., Jigang, L.: A Framework for Enhancing Web Services Security. In: 31st Annual International, COMPSAC 2007, July 24-27, vol. 1, pp. 23–30 (2007)
2. Singh, S., Bawa, S.: A Framework for Handling Security Problems in Grid Environment using Web Services Security Specifications. In: SKG 2006, pp. 68–68 (2006)
3. Wenjun, Z.: Integrated Security Framework for Secure Web Services. In: 2010 Third International Symposium on IITSI 2010, April 2-4, pp. 178–183 (2010)

SRA: A Salmon-Like Approach to MANET Routing

Filomena de Santis and Daniele Mastrangelo

Dipartimento di Informatica e Applicazioni, University of Salerno
Via Ponte don Melillo, I-84084, Fisciano (SA), Italy
fds@unisa.it, danielemastrangelo@tiscali.it

Abstract. Wireless mobile ad-hoc networks are characterized by the lack of physical connections. Due to the mobility of nodes, interferences, multipath propagations and path losses, they do not exhibit a fixed topology; hence, dynamic routing protocols are required. In recent years, new approaches inspired by nature have been tried: among them, particular interest has been raised by ants and bees colonies. The characteristics inherited by the collective behaviors of social insects empower algorithms with features such autonomy, self-organization, adaptivity, robustness, and scalability. Here, we propose a salmon-based approach, that, although different since salmons do not show evidence of social behaviors, suggests interesting cues to solve the routing problem when observing salmons in their way from the birth river to the sea, and back at the spawning time.

Keywords: Wireless communications, mobile ad-hoc networks, routing algorithms, bio-inspired paradigms of computation.

1 Introduction

Advances in wireless communication technology have strongly encouraged the use of low-cost and powerful wireless transceivers in mobile applications. As compared with wired networks, mobile networks exhibit unique features: recurrent network topology changes, link capacity fluctuations, critical bounds to their performances.

Mobile networks can be classified into infrastructure networks and mobile ad hoc networks [1]. In an infrastructure mobile network, mobile nodes communicate through wired access points that work in the node transmission range and create the backbone of the network. In a mobile ad hoc network (MANET) nodes are self-organized without any infrastructure support: they move arbitrarily causing the network to experience quick and random topology changes, some of them do not communicate directly with each other, every node has to act as a router, too.

The design of MANET routing protocols is a challenging task. *Proactive* routing, *reactive* routing and *hybrid* routing, [2] are the most popular classes of MANET routing protocols. In a *proactive* routing protocol nodes continuously evaluate routes towards all reachable nodes and maintain consistent, up-to-date routing information even though network topology changes occur. In a *reactive* routing protocol, routing paths are searched only when needed by means of a route discovery operation established between the source and destination node. *Hybrid* routing protocols

combine the merits of both proactive and reactive protocols and overcome their shortcomings.

Many routing protocols for MANETs, have been considered in the literature for a number of different network scenarios. Among them, algorithms inspired by the behavior of some classes of insects have shown interesting performances. More specifically, the ability of ant colonies to discover shortest paths between their nest and sources of food has been put to work in the general optimization framework of the Ant Colony Optimization (ACO), and the communication and recruitment strategies adopted for effective foraging within a beehive have inspired the development of some novel algorithms for routing problems, [3].

In the remainder of this report, Sections 2 and 3, respectively, introduce the basic salmon behaviors and a new routing algorithm for MANET inspired by it. Even though different from the underlying philosophy of self organized systems intrinsic with social insects, this new approach suggests interesting cues to solve MANET routing.

2 Homing Mechanisms in Salmon

In the framework of nature inspired algorithms, we have examined the salmon behavior. Their life history is dominated by their strong tendency to home to their natal site for reproduction. Typically, salmon spawn in streams or lakes and, after a variable period of freshwater residence (0–3 years), the offspring migrate to the ocean. Salmon remain in the ocean until they mature and then return to their natal site to spawn. Homing migrations are often thousands of kilometers from home river, and in-river migrations back to their natal site may be as long as the ocean migration. These diverse marine and freshwater habitats provide distinct sets of orientation clues and pose distinct challenges for orientation. Despite these challenges, homing is generally precise. Although little experimental evidence exists regarding orientation mechanisms in the ocean, the final freshwater phase of the homing migration is governed primarily by olfactory discrimination of the homestream water. The olfactory imprinting hypothesis for salmon homing was first proposed by Hasler and Wisby (1951) based on behavioral experiments demonstrating that fish can discriminate between the waters of different streams on the basis of odors. This hypothesis has several components: (1) streams differ in chemical characteristics that are stable over time; (2) salmon can distinguish these differences; and (3) salmon learn the chemical characteristics of their natal stream prior to or during their seaward migration, remember (even after 4 years) them without reinforcement during ocean residence, and respond to them as adults,[4].

Foraging ants in a colony converge on the shortest paths connecting their nest to a food source by means of a catalyst, the pheromone. While moving, ants lay pheromone on the ground attracting, in this way, other ants on the same path, that is to say they use a form of learning and control based on indirect communication which locally modify the environment and react to these modifications leading to a phase of global coordination (stigmergy).

Thus, the common point between the ant and salmon behaviors is the recognition of a trail on the route they passed through some time before. The main differences rely on two aspects: first, the salmon learning is individual, second the learning object is used quite immediately by ants and after a long time by salmons.

3 Salmon Routing Algorithm

In this section we propose the Salmon Routing Algorithm (SRA), based on the behavior of salmons. The network topology is determined by the stream morphology of the areas interested by the salmon migration; thus, we consider a star topology where all nodes connect to a unique destination (the sea). Such an hypothesis of unique destination exactly reflects the migration conditions. SRA is constituted by two phases: *Descent*, associated to the salmon path from its birth river to the sea, and *Ascent*: associated to the salmon path from the sea to its birth river. *Descent* and *Ascent* are similar to *forward* and *backward* phases of ACO with the substantial difference that *forward* and *backward* are consecutively executed, whereas a *Ascent* is executed a long time after a *Descent* according to the real behaviour of a salmon. In the network context, this implies that the routing information stored in *Descent* will not be any more consistent for *Ascent* because the high node mobility will have completely modified the network topology. To overcome this mess, SRA uses different data in the two phases: routing information in *Descent*, and *pheromone tables* in *Ascent*. Network topology changes and nodes mobility are accounted by river modifications due to landslides, floods, earthquakes, etc. Such cases, where in nature salmons cannot return to their birth rivers, are associated in the network framework to a link failure state, where the *salmon packet* cannot reach the source node.

In the next subsection we analyze two variants of the SRA algorithm: P-SRA (ProactiveSRA), and R-SRA(Reactive SRA), which, respectively, in *Descent* use AODV and DSR paradigms. The *Ascent* is the same for both proactive and reactive versions.

3.1 P-SRA

A proactive algorithm maintains routing information for each node of the network through the periodic update of *routing tables* ensuring stable and reliable routing information. In the P-SRA, as well as in ACO, we use two tables:

Routing table: destination, next hop, distance of the destination, sequence number

Pheromone table: *pheromone values*, that is to say the identification items, *id*, of the salmon packets traversing a node in the descent phase.

P-SRA executes the *Descent* and *Ascent* as follows:

- *Descent*: the *salmon packet* leaves from the source node *s* and arrives at the destination node *sea* by using the information in the *routing tables* of each node it passes through. Along the path, for each node P-SRA builds/updates the *pheromone table* by means of the *id* of the *salmon packet*.

- *Ascent*: the *salmon packet* ascends from the destination node *sea* to the source node *s* by using the information stored in the *pheromone tables*. More specifically, P-SRA looks for an entry correspondent with the *id* of its *salmon packet* in order to choose the correct sub-network containing its birth river. If there is a link failure in the path, the salmon packet will be eliminated after a fixed TTL.

3.2 R-SRA

A reactive algorithm does not maintain routing information for each node of the network, but looks for and obtains a path only on demand ensuring a scarce overhead for routing information.

R-SRA executes the *Descent* and *Ascent* as follows:

- *Descent*: two types of packets, *SalmonRouteRequest* and *SalmonRouteReply*, are flooded through the network. The first is used when a path for a determinate destination is required; the second is a reply and contains a possible path known by the destination node itself or another node of the network. Every *SalmonRouteRequest* contains a *sequence number* of updating useful to prevent cycles and multiple transmissions of the same message.
- *Ascent*: it is the same as P-RSA.

3.3 Performance Evaluation

We are actually testing SRA on a real MANET constituted by 12 node positioned along the perimeter of our campus. The mobility is simulated by means of random changes in the topology of network, and the SRA performances are compared to AODV, DSR, DSDV, ACO algorithms and BeeAdHoc [5]. Although in an early stage, experimental results seem to be encouraging because of the following considerations.

SRA drastically decreases the overhead for storing the routing information. More precisely, major advantages are achieved by P-SRA *Ascent* since the routing of packets is exclusively based on the *pheromone tables* stored in *Descent*: there is no need of *routing tables*. A same result holds for R-SRA *Ascent* where the use of pheromone tables allows to avoid the flooding of *SalmonRouteRequest* and *SalmonRouteReply* packets. *Descent* shows similar performances to AODV for P-SRA and to DSR for R-SRA, respectively.

References

1. IETF Manet charter,
<http://www.ietf.org/html.charters/manet-charter.html>
2. Royer, E., Toh, C.K.: A Review of Current Routing Protocols for ad hoc Mobile Wireless Networks. IEEE Personal Communications 6, 46–55 (1999)

3. Bonabeau, E., Dorigo, M., Teraulaz, G.: *Swarm Intelligence: from Natural to Artificial Systems*. Oxford University Press, NY (1999)
4. Dittman, A.H., Quinn, T.P.: Homing in Pacific Salmon: Mechanisms and Ecological Basis. *Journal of Experimental Biology* 199, 83–91 (1996)
5. Ad-hoc protocols implementations, <http://www.wikipedia.org/wiki/>

Author Index

- Akaishi, Jin 155
Ali, Ahmad 682
Al-Mutib, Khalid 607
Ames, Heather 659
Antunes, Mário J. 1
Arita, Takaya 143
Attarzadeh, Iman 334
- Balasubramaniam, Sasitharan 565
Barth, Erhardt 469, 581, 601
Bhatia, Neha 402
Bizdoaca, Nicu 554
Boccardo, Davidson R. 637
Boonma, Pruet 32
Botvich, Dmitri 565
Bouecke, Jan D. 483
Bush, Benjamin James 155
- Calcavecchia, Nicolò Maria 48
Cao, Yu 303
Caprarescu, Bogdan Alexandru 48
Carmo, Luiz F.R.C. 637
Carroll, Raymond 565
Carvalho, Marco 22
Chandler, Benjamin 659
Chandra, Satish 402
Chaurasia, Anurag 402
Chen, Xiujian 155
Correia, Manuel E. 1
Cox, David 505
- De Jong, Kenneth A. 213
de Santis, Filomena 687
Di Nitto, Elisabetta 48
Dionne, Shelley D. 155
Donnelly, William 565
Dorr, Michael 469, 581, 601
Dressler, Falko 128
D'Souza, Rio G.L. 229, 440
Dubois, Daniel J. 48
Ducommun, Bernard 40
Duthen, Yves 40
- Eckford, Andrew W. 16
Eichelberger, Christopher N. 452
Eltabach, Mario 554
- Farsad, Nariman 16
Florescu, Mihaela 554
Furusawa, Chikara 238
- Gallege, Lahiru S. 310
Gao, Kehan 288
Garg, Deepak 651
Gerla, Mario 128
Getoor, Lise 387
Godfrey, W. Wilfred 533
Gorchetchnikov, Anatoli 659
Goudarzi, Alireza 163
Gross, Thilo 521
Gruska, Damas P. 92
Guillen, Edward 676, 680
Gulbahce, Natali 163
Gupta, Alka 155
- Hadzikadic, Mirsad 452
Haller, Kenneth J. 56
Hamdan, Hani 554, 607
Hamdan, Mohamad 607
Han, Il Song 621
Han, Woo Joon 621
Hao, Chanyu 155
Haraguchi, Tokuko 30
Haspel, Nurit 343
Head, Hadassah J. 155
Hiyama, Satoshi 16
Hogrefe, Dieter 375
- Ijichi, Kota 238
Ilyas, Muhammad 682
Ivanescu, Mircea 554
- Kabir, K. Habibul 264
Kamath, Uday 213
Kandasamy, A. 229, 440
Khalid, M. 425
Khoshgoftaar, Taghi M. 288
Khoushab, Feisal 56
Kobayashi, Shouhei 30
Koch, Roman 128
Kornprobst, Pierre 483
Krithivasan, Kamala 425, 651

- Kueng, Josef 682
 Kundra, Rohan 402
 Kuppusamy, Lakshmanan 425
 Kurokawa, Haruhisa 475
- Lamkin, Tom 22
 Lee, Chonho 189
 Léveillé, Jasmin 659
 Li, Ming 303
 Licamele, Louis 387
 Liu, Jian-Qin 563, 587, 629
 Lobjois, Valérie 40
 Luga, Hervé 40
- Machado, Raphael C.S. 637
 Mahendran, Anand 425
 Masson, Guillaume S. 483
 Mastrangelo, Daniele 687
 Mattar, Ebrahim 607, 667
 Memarmoshrefi, Parisa 375
 Metta, Venkata Padmavati 651
 Mingolla, Ennio 659
 Molloy, Kevin 249
 Moreira, João 205
 Moritani, Yuki 16
 Mukhopadhyay, Snehasis 310
 Murata, Masayuki 276
 Mutib, Khaled Al 667
- Nair, Shivashankar B. 533
 Nakano, Tadashi 30, 100, 563
 Napolitano, Amri 288
 Nascimento, Tiago M. 637
 Neumann, Heiko 483
 Nishimura, Haruhiko 629
- Ohsaki, Hiroyuki 59, 102, 111
 Okaie, Yutaka 100
 Olson, Brian 249
 Ow, Siew Hock 334
- Pacheco, Jorge M. 178, 205, 523
 Padilla, Daniel 676
 Paez, Rafael 680
 Pascalie, Jonathan 40
 Patrick, Sean 659
 Perez, Carlos 22
 Phatak, Sucheta 310
 Philip, Swapna 303
 Pinheiro, Flávio L. 523
- Pinto, Nicolas 505
 Poleksic, Aleksandar 363
 Pomarjanschi, Laura 581
 Prado, Charles B. 637
- Raje, Rajeev R. 310
 Rajendran, Chandrasekharan 413
 Rajendran, Suchithra 413
 Ramos, Carlos 676
 Rasche, Christoph 581
- Santos, Francisco C. 178, 205, 523
 Santos, Marta D. 178
 Sasabe, Masahiro 264
 Sawai, Hidefumi 59, 82, 102, 111
 Sayama, Hiroki 155, 497
 Seibel, Roman 375
 Sekaran, K. Chandra 229, 440
 Serban, Andra 155
 Shastri, Rashmi 310
 Shaw, Leah B. 519
 Shehu, Amarda 213, 249
 Shimizu, Hiroshi 238
 Suzuki, Hideaki 59, 82, 102, 111
 Suzuki, Junichi 32, 189
 Suzuki, Reiji 143
- Takine, Tetsuya 264
 Teng, Rui 587
 Teuscher, Christof 163
 Tlapale, Émilien 483
 Tomita, Kohji 475
 Traulsen, Arne 521
 Tunc, Ilker 519
- Vasilakos, Athanasios V. 189, 542
 Versace, Massimiliano 659
 Vig, Eleonora 469, 601
- Wakamiya, Naoki 276
 Wei, Guiyi 542
 Wei, Hung-Yu 542
- Xu, Lei 74
- Yamabhai, Montarop 56
 Yamamoto, Hiroshi 276
 Yamanishi, Teruya 629
 Yammarino, Francis J. 155
 Yi, Dongyun 357

Yoon, Jong P. 325
Yoon, Joyce 325
Yoshida, Mikio 82

Zhang, Bing 587
Zhang, Xue 357

Zhao, Chengli 357
Zhu, Bingchun 32
Zhu, Ping 542
Ziegler, Hans 413
Zschaler, Gerd 521