ADVANCES IN INTELLIGENT SYSTEMS AND COMPUTING 201

Jagdish Chand Bansal
Pramod Kumar Singh
Kusum Deep
Millie Pant
Atulya K. Nagar (Eds.)

Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)

Springer

Advances in Intelligent Systems and Computing

Volume 201

For further volumes:
http://www.springer.com/series/11156

Jagdish Chand Bansal ·
Pramod Kumar Singh · Kusum Deep
Millie Pant · Atulya K. Nagar
Editors

Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)

Volume 1

Springer

Editors
Jagdish Chand Bansal
South Asian University
Chankya Puri, New Delhi
India

Pramod Kumar Singh
ABV-IIITM, Gwalior
Gwalior, Madhya Pradesh
India

Kusum Deep
Department of Mathematics
Indian Institute of Technology Roorkee
Roorkee
India

Millie Pant
Department of Applied Science
and Engineering
Indian Institute of Technology Roorkee
Roorkee
India

Atulya K. Nagar
Department of Mathematics
and Computer Science
Liverpool Hope University
Liverpool
UK

ISSN 2194-5357 ISSN 2194-5365 (electronic)
ISBN 978-81-322-1037-5 ISBN 978-81-322-1038-2 (eBook)
DOI 10.1007/978-81-322-1038-2
Springer New Delhi Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012954374

Printed on acid-free paper

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

Preface

Human beings have always been fascinated by nature and especially by biological diversity and their evolutionary process. This has resulted into inspirations drawn from natural or biological systems, and phenomenon, for problem solving and has seen an emergence of a new paradigm of computation known as Natural Computing with Bio-inspired Computing as its subset. The widely popular methods, e.g., evolutionary computation, swarm intelligence, artificial neural networks, artificial immune systems, are just some examples in the area. Such approaches are of much use when we need an imprecise, inaccurate but feasible solution in a reasonable time as many real-world problems are too complex to be dealt using traditional methods of finding exact solutions in a reasonable time. Therefore, bio-inspired approaches are gaining popularity as the size and complexity of the real-world problems require the development of methods which can give the solution within a reasonable amount of time rather than an ability to guarantee the exact solution. Bio-inspired Computing can provide such a rich tool-chest of approaches as it tends to be, just like its natural system counterpart, decentralized, adaptive and environmentally aware, and as a result have survivability, scalability and flexibility features necessary to deal with complex and intractable situations.

Bio-Inspired Computing: Theories and Applications (BIC-TA) is one of the flagship conferences on Bio-Computing bringing together the world's leading scientists from different branches of Natural Computing. Since 2006 the conferences have taken place at Wuhan (2006), Zhengzhou (2007), Adelaide (2008), Beijing (2009), Liverpool and Changsha (2010), Penang (2011). BIC-TA has attracted wide ranging interest amongst researchers with different backgrounds resulting in a seventh edition in 2012 at Gwalior. It is our privilege to have been part of this seventh edition of the BIC-TA series which is being hosted for the first time in India.

This volume in the AISC series contains papers presented at the Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012) held during December 14–16, 2012 at ABV-Indian Institute of Information Technology and Management Gwalior (ABV-IIITM Gwalior), Madhya Pradesh, India. The BIC-TA 2012 provides a unique forum to researchers and practitioners working in the ever growing area of bio-inspired computing methods and their applications to solve various real-world problems.

BIC-TA 2012 attracted attention of researchers from all over the globe and we received 188 papers related to various aspects of bio-inspired computing with umpteen applications, theories, and techniques. After a thorough peer-review process a total of 91 thought-provoking research papers are selected for publication in the Proceedings, which is in two volumes (Volume 1 and 2). This thus corresponds to an acceptance rate of 48% and is intended to maintain a high standard in the conference proceedings. We hope that the papers contained in this proceeding will serve the purpose of inspiring more and more researchers to work in the area of bio-inspired computing and its application.

The editors would like to express their sincere gratitude to the authors, plenary speakers, invited speakers, reviewers, and members of international advisory committee, programme committee and local organizing committee. It would not have been possible to come out with the high quality and standard of the conference as well as this edited Proceeding without their active

participation and whole hearted support. It would not be fair on our part if we forget to mention special thanks to the ABV – Indian Institute of Information Technology and Management Gwalior (ABV-IIITM Gwalior) and its Director Prof. S. G. Deshmukh for providing us all the possible help and support including excellent infrastructure of the Institute to make this conference a big success. We express our gratitude to the Department of Mathematics and Computer Science, Liverpool Hope University, Liverpool, UK headed by Prof. Atulya K. Nagar for providing us much valued and needed support and guidance. Finally, we would like to thank all the volunteers; their untiring efforts in meeting the deadlines and managerial skills in managing the resources effectively and efficiently which has ensured a smooth running of the conference.

It is envisaged that the BIC-TA conference series will continue to grow and include relevant future research and development challenges in this exciting field of Computing.

Jagdish Chand Bansal, South Asian University, New Delhi, India
Pramod Kumar Singh, ABV-IIITM, Gwalior, India
Kusum Deep, Indian Institute of Technology, Roorkee, India
Millie Pant, Indian Institute of Technology, Roorkee, India
Atulya K. Nagar, Liverpool Hope University, Liverpool, UK

Editors

Jagdish Chand Bansal
Pramod Kumar Singh
Kusum Deep
Millie Pant
Atulya K. Nagar

About Editors

Dr. Jagdish Chand Bansal is an Assistant Professor with the South Asian University New Delhi, India. Holding an excellent academic record, he is a budding researcher in the field of Swarm Intelligence at the International Level.

Dr. Pramod Kumar Singh is an Associate Professor with the ABV-Indian Institute of Information Technology and Management, Gwalior, India. He is an active researcher and has earned a reputation in the areas of Nature-Inspired Computing, Multi-/Many-Objective Optimization, and Data Mining.

Dr. Kusum Deep is a Professor with the Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, India. Over the last 25 years, her research is increasingly well-cited making her a central International figure in the area of Bio-Inspired Optimization Techniques, Genetic Algorithms and Particle Swarm Optimization.

Dr. Millie Pant is an Associate Professor with the Department of Applied Science and Engineering, Indian Institute of Technology, Roorkee, Roorkee, India. At this age, she has earned a remarkable International reputation in the area of Genetic Algorithms, Differential Algorithms and Swarm Intelligence.

Prof. Atulya K. Nagar is the Professor and Head of Department of Mathematics and Computer Science at Liverpool Hope University, Liverpool, UK. Prof. Nagar is an internationally recognized scholar working at the cutting edge of theoretical computer science, natural computing, applied mathematical analysis, operations research, and systems engineering and his work is underpinned by strong complexity-theoretic foundations.

Organizing Committees

BIC-TA 2012 was held at ABV- Indian Institute of Information Technology and Management Gwalior, India. Details of the various organizing committees are as follows:

Patron:	S. G. Deshmukh, ABV-IIITM Gwalior, India
General Chairs:	Atulya Nagar, Liverpool Hope Uuniversity Liverpool, UK Kusum Deep, IIT Roorkee, India
Conference Chairs:	Jagdish Chand Bansal, South Asian University New Delhi, India Pramod Kumar Singh, ABV-IIITM Gwalior, India
Program Committee Chairs:	Millie Pant, IIT Roorkee, India T. Robinson, MCC Chennai, India
Special Session Chair:	Millie Pant, IIT Roorkee, India
Publicity Chairs:	Manoj Thakur, IIT Mandi, India Kedar Nath Das, NIT Silchar, India
Best Paper Chair: (Technically Sponsored by KanGAL, IIT Kanpur, India)	Kalyanmoy Deb, IIT Kanpur, India
Conference Secretaries:	Harish Sharma, ABV-IIITM Gwalior, India Jay Prakash, ABV-IIITM Gwalior, India Shimpi Singh Jadon, ABV-IIITM Gwalior, India Kusum Kumari Bharti, ABV-IIITM Gwalior, India
Local Arrangement Committee: (ABV-IIITM Gwalior, India)	Jai Prakash Sharma Narendra Singh Tomar Alok Singh Jadon Rampal Singh Kushwaha Mahesh Dhakad Balkishan Gupta

International Advisory Committee:

Atulya K. Nagar, UK
Gheorghe Paun, Romania
Giancarlo Mauri, Italy
Guangzhao Cui, China
Hao Yan, USA
Jin Xu, China
Jiuyong Li, Australia
Joshua Knowles, UK
K G Subramanian, Malaysia
Kalyanmoy Deb, India
Kenli Li, China
Linqiang Pan, China
Mario J. Perez-Jimenez, Spain
Miki Hirabayashi, Japan
PierLuigi Frisco, UK
Robinson Thamburaj, India
Thom LaBean, USA
Yongli Mi, Hong Kong

Special Sessions:

Session 1: Computational Intelligence in Power and Energy Systems, Amit Jain, IIIT Hyderabad, India

Session 2: Bio-Inspired VLSI and Embedded System, Balwinder Raj, NIT Jalandhar, India

Session 3: Recommender System: Design Using Evolutionary & Natural Algorithms, Soumya Banerjee Birla Institute of Technology Mesra, India & Shengbo Guo, Xerox Research Centre Europe, France

Session 4: Image Analysis and Pattern Recognition, K. V. Arya, ABV-IIITM Gwalior, India

Session 5: Applications of Bio-inspired Techniques to Social Computing, Vaskar Raychoudhury, IIT Roorkee, India

Keynote Speakers:

Title: Spiking Neural P Systems
Speaker: Pan Linqiang

Title: Advancements in Memetic Computation
Speaker: Yew-Soon Ong

Title: Machine Intelligence, Generalized Rough Sets and Granular Mining: Concepts, Features ans Applications
Speaker: Sankar Kumar Pal

Title: Of Glowworms and Robots: A New Paradigm in Swarm Intelligence
Speaker: Debasish Ghose

Title: Advances in Immunological Computation
Speaker: Dipankar Dasgupta

Title: Selection of Machinery Health Monitoring Strategies using Soft Computing
Speaker: Ajit Kumar Verma

Title: Can Fuzzy logic Formalism via Computing with Words Bring Complex Environmental Issues into Focus?
Speaker: Ashok Deshpande

Technical Program Committee:

Abdulqader Mohsen, Malaysia
Abhishek Choubey, India
Adel Al-Jumaily, Australia
Aitor Rodriguez-Alsina, Spain
Akila Muthuramalingam, India
Alessandro Campi, Italy
Amit Dutta, India
Amit Jain, India
Amit Pandit, India
Amreek Singh, India
Anand Sharma, India
Andre Aquino, Brazil
Andre Carvalho, Brazil
Andrei Paun, USA
Andres Muñoz, Spain
Anil K Saini, India
Anil Parihar, India
Anjana Jain, India
Antonio J. Jara, Spain
Anupam Singh, India
Anuradha Fukane, India
Anurag Dixit, India
Apurva Shah, India
Aradhana Saxena, India
Arnab Nandi, India
Arshin Rezazadeh, Iran
Arun Khosla, India
Ashish Siwach, India
Ashraf Darwish, Egypt
Ashwani Kush, India
Atulya K. Nagar, UK
B.S. Bhattacharya, UK
Bahareh Asadi, Iran
Bala Krishna Maddali, India
Balaji Venkatraman, India
Balasubramanian Raman, India

Banani Basu, India
Bharanidharan Shanmugam, Malaysia
Carlos Coello Coello, Mexico
Carlos Fernandez-Llatas, Spain
Chang Wook Ahn, Korea
Chi Kin Chow, Hong Kong
Chu-Hsing Lin, Taiwan
Chun-Wei Lin, Taiwan
Ciprian Dobre, Romania
D.G. Thomas, India
Dakshina Ranjan Kisku, India
Dana Petcu, Romania
Dante Tapia, Spain
Deb Kalyanmoy, India
Debnath Bhattacharyya, India
Desmond Lobo, Thailand
Devshri Roy, India
Dipti Singh, India
Djerou Leila, Algeria
Asoke Nath, India
K K Shukla, India
Kavita Burse, India
Mrutyunjaya Panda, India
Shirshu Varma, India
Raveendranathan K.C., India
Shailendra Singh, India
Eduard Babulak, Canada
Eric Gregoire, France
Erkan Bostanci, UK
F N Arshad, UK
Farhad Nematy, Iran
Francesco Marcelloni, Italy
G.R.S. Murthy, India
Gauri S. Mittal, Canada
Ghanshyamsingh Thakur, India
Gheorghe Paun, Romania
Guoli Ji, China
Gurvinder Singh-Baicher, UK
Hasimah Hj. Mohamed, Malaysia
Hemant Mehta, India
Holger Morgenstern, Germany
Hongwei Mo, China
Hugo Proença, Portugal
Ivica Boticki, Croatia
Jaikaran Singh, India
Javier Bajo, Spain
Jer Lang Hong, Malaysia
Jitendra Kumar Rai, India
Joanna Kolodziej, Poland
Jose Pazos-Arias, Spain
Juan Mauricio, Brazil
K K Shukla, India
K V Arya, India
K.G. Subramanian, Malaysia
Kadian Davis, Jamaica
Kamal Kant, India
Kannammal Sampathkumar, India
Katheej Parveen, India
Kazumi Nakamatsu, Japan
Kedar Nath Das, India
Khaled Abdullah, India
Khelil Naceur, Algeria
Khushboo Hemnani, India
Kittipong Tripetch, Thailand
Kunal Patel, USA
Kusum Deep, India
Lalit Awasthi, India
Lam Thu Bui, Australia
Li-Pei Wong, Malaysia
Lin Gao, China
Linqiang Pan, China
M.Ayoub Khan, India
Madhusudan Singh, Korea
Manjaree Pandit, India
Manoj Saxena, India
Manoj Shukla, India
Marian Gheorghe, UK
Mario Koeppen, Japan
Martin Middendorf, Germany
Mehdi Bahrami, Iran
Mehul Raval, India
Michael Chen, China
Ming Chen, China
Mohammad A. Hoque, United States
Mohammad Reza Nouri Rad, Iran
Mohammed Abdulqadeer, India
Mohammed Rokibul Alam Kotwal, Bangladesh
Mohd Abdul Hameed, India
Monica Mehrotra, India
Monowar T, India
Mourad Abbas, Algeria
Mps Chawla, India
Muhammad Abulaish, Saudi Arabia
N.Ch.Sriman Narayana Iyengar, India
Nand Kishor, India
Narendra Chaudhari, India
Natarajamani S, India
Navneet Agrawal, India
Neha Deshpande, India
Nikolaos Thomaidis, Greece
Ninan Sajeeth Philip, India
O. P. Verma, India
P. G. Sapna, India
P. N. Suganthan, Singapore
Philip Moore, U.K
Pierluigi Frisco, UK
Ponnuthurai Suganthan, Singapore
Pramod Kumar Singh, India
Vidya Dhamdhere, India
Kishan Rao Kalitkar, India
Punam Bedi, India
Qiang Zhang, China

R. K. Singh, India
R. N. Yadav, India
R. K. Pateriya, India
Rahmat Budiarto, Malaysia
Rajeev Srivastava, India
Rajesh Sanghvi, India
Ram Ratan, India
Ramesh Babu, India
Ravi Sankar Vadali, India
Rawya Rizk, Egypt
Razib Hayat Khan, Norway
Reda Alhajj, Canada
Ronaldo Menezes, USA
S. M. Sameer, India
S. R. Thangiah, USA
Sami Habib, Kuwait
Samrat Sabat, India
Sanjeev Singh, India
Satvir Singh, India
Shan He, UK
Shanti Swarup, India
Shaojing Fu, China
Shashi Bhushan Kotwal, India
Shyam Lal, India
Siby Abraham, India
Smn Arosha Senanayake, Brunei Darussalam
Sonia Schulenburg, UK
Sotirios Ziavras, United States
Soumya Banerjee, India
Steven Gustafson, USA
Sudhir Warier, India
Sumithra Devi K A, India
Sung-Bae Cho, Korea
Sunil Kumar Jha, India
Suresh Jain, India
Surya Prakash, India
Susan George, Australia
Sushil Kulkarni, India
Swagatam Das, India
Thambi Durai, India
Thamburaj Robinson, India
Thang N. Bui, USA
Tom Hendtlass, Australia
Trilochan Panigrahi, India
Tsung-Che Chiang, Taiwan
Tzung-Pei Hong, Taiwan
Umesh Chandra Pati, India
Uzay Kaymak, Netherlands
V. Rajkumar Dare, India
Vassiliki Andronikou, Greece
Vinay Kumar Srivastava, India
Vinay Rishiwal, India
Vittorio Maniezzo, Italy
Vivek Tiwari, India
Wahidah Husain, Malaysia
Wei-Chiang Samuelson Hong, China
Weisen Guo, Japan
Wenjian Luo, China
Yigang He, China
Yogesh Trivedi, India
Yoseba Penya, Spain
Yoshihiko Ichikawa, Japan
Yuan Haibin, China
Yunong Zhang, China
Yuzhe Liu, US

Contents

Stochastic Algorithms for 3D Node Localization in Anisotropic Wireless Sensor Networks 1
Anil Kumar, Arun Khosla, Jasbir Singh Saini and Satvir Singh

An Evaluation of Classification Algorithms Using Mc Nemar's Test 15
Betul Bostanci and Erkan Bostanci

Permitting Features in *P* Systems Generating Picture Arrays 27
K.G. Subramanian, Ibrahim Venkat, Linqiang Pan and Atulya K. Nagar

An ACO Framework for Single Track Railway Scheduling Problem 39
Raghavendra G. S. and Prasanna Kumar N

Bio-Inspired Soft-Computational Framework for Speech and Image Application 53
Dipjyoti Sarma and Kandarpa Kumar Sarma

Leukocyte Classification in Skin Tissue Images 65
Mukesh Saraswat and K. V. Arya

Solving Application Oriented Graph Theoretical Problems with DNA Computing 75
Veronika Halász, László Hegedüs, István Hornyák and Benedek Nagy

Human Identification using Heartbeat Interval Features and ECG Morphology 87
Yogendra Narain Singh and Sanjay Kumar Singh

Improved Real-Time Discretize Network Intrusion Detection System 99
Heba F. Eid, Ahmad Taher Azar and Aboul Ella Hassanien

Identification and Impact Assessment of High Priority Field Failures in Passenger Vehicles using Evolutionary Optimization 111
Abhinav Gaur, Sunith Bandaru, Vineet Khare, Rahul Chougule and Kalyanmoy Deb

Automatic Agricultural Leaves Recognition System 123
Meenakshi, Durga Puja, Mukesh Saraswat and K. V. Arya

Non-Uniform Mapping in Binary-Coded Genetic Algorithms 133
Kalyanmoy Deb, Yashesh D. Dhebar and N. V. R. Pavan

Control Words of Transition P Systems 145
Ajeesh Ramanujan and Kamala Krithivasan

Iso-Array Splicing Grammar System 157
D. K. Sheena Christy, V. Masilamani and D. G. Thomas

GA based Dimension Reduction for enhancing performance of k-Means and Fuzzy k-Means: A Case Study for Categorization of Medical Dataset 169
Asha Gowda Karegowda, M. A. Jayaram, A. S. Manjunath, T. Vidya and Shama

A Computational Intelligence Based Approach to Telecom Customer Classification for Value Added Services 181
Abhay Bhadani, Ravi Shankar and D. Vijay Rao

An Efficient Approach on Rare Association Rule Mining 193
N. Hoque, B. Nath and D. K. Bhattacharyya

A Hybrid Multiobjective Particle Swarm Optimization Approach for Non-redundant Gene Marker Selection 205
Anirban Mukhopadhyay and Monalisa Mandal

Application of High Quality Amino Acid Indices to AMS 3.0: An Update Note 217
Indrajit Saha, Ujjwal Maulik and Dariusz Plewczynski

Constructive Solid Geometry Based Topology Optimization Using Evolutionary Algorithm 227
Faez Ahmed, Bishakh Bhattacharya and Kalyanmoy Deb

Array P Systems with Hybrid Teams 239
P. Helen Chandra and S. M. Saroja Theerdus Kalavathy

An Approach for the Ordering of Evaluation of Objectives in Multiobjective Optimization 251
Preeti Gupta, Sanghamitra Bandyopadhyay and Ujjwal Maulik

Extended Forma: Analysis and an Operator Exploiting it 263
Dharmani Bhaveshkumar C

Incorporating Great Deluge with Harmony Search for Global Optimization Problems 275
Mohammed Azmi Al-Betar, Osama Nasif Ahmad, Ahamad Tajudin Khader and Mohammed A. Awadallah

Boundary Handling Approaches in Particle Swarm Optimization 287
Nikhil Padhye, Kalyanmoy Deb and Pulkit Mittal

Diversity Measures in Artificial Bee Colony 299
Harish Sharma, Jagdish Chand Bansal and K. V. Arya

Digital Video Watermarking Using Scene Detection 315
Dinesh Tiwari, K. V. Arya and Mukesh Saraswat

Self Adaptive Acceleration Factor in Particle Swarm Optimization . . . 325
Shimpi Singh Jadon, Harish Sharma, Jagdish Chand Bansal and Ritu Tiwari

Applying Case Based Reasoning in Cuckoo Search for the Expedition of Groundwater Exploration 341
Daya Gupta, Bidisha Das and V. K. Panchal

Reversible OR Logic Gate Design Using DNA 355
Pradipta Roy, Debarati Dey, Swati Sinha and Debashis De

Performance Enhanced Hybrid Artificial Neural Network for Abnormal Retinal Image Classification 367
D. Jude Hemanth and J. Anitha

Algorithmic Tile Self-assembly Model for the Minimum Dominating Set Problem 379
Zhen Cheng, Jianhua Xiao and Yufang Huang

Semantic Sub-tree Crossover Operator for Postfix Genetic Programming 391
Vipul K. Dabhi and Sanjay Chaudhary

Exploration Enhanced Particle Swarm Optimization Using Guided Re-Initialization 403
Karan Kumar Budhraja, Ashutosh Singh, Gaurav Dubey and Arun Khosla

Using Firefly Algorithm to Solve Resource Constrained Project Scheduling Problem 417
Pejman Sanaei, Reza Akbari, Vahid Zeighami and Sheida Shams

Analysis of Cellular Automata and Genetic Algorithm based Test Pattern Generators for Built in Self Test 429
Balwinder Singh, Sukhleen Bindra Narang and Arun Khosla

Ant Colony-based System for Retinal Blood Vessels Segmentation . . . 441
Ahmed. H. Asad, Ahmad Taher Azar and Aboul Ella Hassaanien

AN Efficient Neural Network Based Background subtraction method 453
Naveen Kumar Rai, Shikha Chourasia and Amit Sethi

JustThink: Smart BCI Applications 461
Rabie A. Ramadan, Ahmed Ezzat AbdElGawad and Mohammed Alaa

Interpretability Issues in Evolutionary Multi-Objective Fuzzy Knowledge Base Systems 473
Praveen Kumar Shukla and Surya Prakash Tripathi

Hybrid Firefly Based Simultaneous Gene Selection and Cancer Classification Using Support Vector Machines and Random Forests 485
Atulji Srivastava, Saurabh Chakrabarti, Subrata Das, Shameek Ghosh and V. K. Jayaraman

Recognition of Online Handwritten Gurmukhi Strokes Using Support Vector Machine 495
Mayank Gupta, Nainsi Gupta and Rahul Agrawal

An Optimal Fuzzy Logic Controller Tuned with Artificial Immune System 507
S. N. Omkar, Nikhil Ramaswamy, R. Ananda, N. G. Venkatesh and J. Senthilnath

Comparisons of Different Feature Sets for Predicting Carbohydrate-Binding Proteins From Amino Acid Sequences Using Support Vector Machine 519
Suchandra Payal, Piyali Chatterjee, Subhadip Basu, Mahantapas Kundu and Mita Nasipuri

A PSO Based Smart Unit Commitment Strategy for Power Systems Including Solar Energy 531
Ravikanth Reddy Gaddam, Amit Jain and Lingamurthy Belede

A User-Oriented Content Based Recommender System Based on Reclusive Methods and Interactive Genetic Algorithm 543
Vibhor Kant and Kamal K. Bharadwaj

Author Index 555

Stochastic Algorithms for 3D Node Localization in Anisotropic Wireless Sensor Networks

Anil Kumar, Arun Khosla, Jasbir Singh Saini, and Satvir Singh

Abstract This paper proposes two range based 3D node localization algorithms using application of Hybrid Particle Swarm Optimization (HPSO) and Biogeography Based Optimization (BBO) for anisotropic Wireless Sensor Networks (WSNs). Target nodes and anchor nodes are randomly deployed with constraints over three layer boundaries. The anchor nodes are randomly distributed over top layer only and target nodes over middle and bottom layers. Radio irregularity factor, i.e., an anisotropic property of propagation media and an heterogenous property (different battery backup statuses) of devices are considered. PSO models provide fast but less mature convergence whereas the proposed HPSO algorithm provides fast and mature convergence. Biogeography is based upon the collective learning of geographical allotment of biological organisms. BBO has a new comprehensive energy based on the science of biogeography and apply migration operator to share selective information between different habitats, i.e., problem solutions. Due to size and complexity of WSN, localization problem is articulated as an NP-hard optimization problem . In this work, an error model in a highly noisy environment is depicted for estimation of optimal node location to minimize the location error using HPSO and BBO algorithms. The simulation results establish the strength of the proposed algorithms by equating the performance in terms of the number of target nodes localized with accuracy, and computation time. It has been observed that existing sensor networks localization algorithms are not significant to support the rescue operations

Anil Kumar
Panipat Institute of Engg. and Technoly, Panipat, Haryana, India e-mail: anil.rose@rediffmail.com

Arun Khosla
National Institute of Technology, Jalandhar, Punjab, India e-mail: khosla@nitj.ac.in

Jasbir Singh Saini
DCR Univ. of Sc. and Tech., Murthal, Sonepat, Haryana, India, e-mail: jssain@rediffmail.com

Satvir Singh
SBS State Technical Campus, Ferozpur, Punjab, India,e-mail: satvir15@gmail.com

J. C. Bansal et al. (eds.), *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, Advances in Intelligent Systems and Computing 201, DOI: 10.1007/978-81-322-1038-2_1, © Springer India 2013

involving human lives. Proposed algorithms are beneficial for rescue operations too to find out the accurate location of target nodes in highly noisy environment.

Key words: Wireless Sensor Networks, Biogeography Based Optimization, Hybrid Particle Swarm Optimization, Anisotropic Network

1 Introduction

The WSNs play an important role in our society, as they have become the archetype of pervasive technology. WSNs consist of an array of sensors, either of same or diverse types, interconnected by communication network. Central aims of the sensor networks admit ease of deployment, reliability, accuracy, flexibility, cost and effectiveness . Sensors perform routing function to create single or multi-hop wireless networking to convey data from one to other sensor nodes. The rapid deployment, self-organization and fault-tolerance characteristics of WSN make them promising for a number of military and civilian applications [1, 2, 3]. In most of the applications, main role of a WSN is to detect and report events which can be meaningfully ingested and reacted to only if the exact location of the event is known. The locations of sensor nodes are often needed when identifying where the collected information comes from. The determination of coordinates of the sensors is one of challenging problems and is referred to as the localization problem, in WSNs.

Localization techniques are employed to estimate the location of the sensor nodes where coordinates are not known in a network (termed as target nodes) using available a priori knowledge of positions of typically a few specific sensor nodes called anchors, based on inter-sensor parameters/measurements such as connectivity distance, Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AOA), etc. [4, 5].

WSN localization is a two-phase process, i.e., ranging and position estimation process. 2D localization assumptions are violated in underwater, atmospheric and space applications where height of the network can be significant and nodes are distributed over a three-dimensional (3D) space [6]. For example, underwater ad hoc sensor networks, which are 3 dimensional, have attracted a lot of attention recently [6, 7]. In underwater sensor networks, nodes may be deployed at different depths of an ocean and thus the network becomes three-dimensional. Better weather forecasting and climate monitoring can be done by deploying three-dimensional networks in the atmosphere.

This paper proposes the application of HPSO and BBO algorithms for range based 3D node localization in anisotropic WSNs. Both algorithms performed better in terms of number of nodes localized, localization accuracy and computation time. Nodes are randomly deployed with constraints over three layer boundaries. However, the anchor nodes are randomly distributed over top layer only and target nodes over lower layers beneath. Radio irregularity factor i.e. anistropic properties of prop-

agation media (background noise and environmental factors) and an heterogenous properties (different battery backups) of devices are considered.

The rest of the paper is organized as follows: Literature Survey on WSN Localization is presented in Section II. Section III ushers the readership into a gentle overview of PSO and BBO algorithms used for localization in this work. This is followed with implementation of above said algorithms in section IV. Section V presents simulation results and comparative study. Finally, section VI presents conclusions and makes a acoustic projection on potential future research paths.

2 Literature Survey

A detailed survey of the relevant literature is available in [8, 9, 10, 11]. An efficient localization system with Accurate Positioning System (APS) extends the GPS capabilities to non-GPS nodes in ad hoc networks as anchors flood their location information to all nodes in the networks proposed in [12]. Then each target node performs a triangulation to three or more anchors to find its position. Node localization accuracy is improved by measuring anchor distances from their neighbors by introducing a refinement phase [13]. The issue of error accumulation is addressed in [14] through Kalman filter based least square estimation in [15, 16] to simultaneously locate the position of all sensor nodes. Node localization problem is addressed using convex optimization based on semidefinite programming. The semidefinite programming approach is further extended to nonconvex inequality constraints [17]. In [18], the gradient search technique demonstrates the use of a data analysis technique called *multidimensional scaling* (MDS) in estimating the position of unknown nodes. The algorithm localizes an individual patch by first computing all pair wise shortest paths between sensors in the patch. Then it applies MDS to these distances to get an initial layout. Finally, an absolute map is obtained by using the known node positions. These techniques work well with few anchors and reasonably high connectivity.

Soft computing plays a crucial role in optimization problems. WSN is treated as multi-modal and multidimensional optimization problem and addressed through population based stochastic techniques. A few GA-based node localization algorithms are presented in [19, 20, 21], that estimate optimal node locations of all one-hop neighbors. Simulated Annealing Algorithm (SAA) and GA based two phase centralized localization scheme is presented in [22]. PSO-based algorithm is proposed in [23, 24] to minimize the localization error. In [25], two intelligent localization schemes for WSNs are introduced for range-free localization, which utilize received signal strength (RSS) from the anchor nodes. In the first scheme, the edge weight of each anchor node is separately calculated and combined to calculate the location of sensor nodes. Fuzzy Logic System (FLS) is used to model edge weights and further optimized by the GA. In the second scheme, the localization is approximated as a single problem where the entire sensors' locations from the anchor node signals are mapped by a Neural Network (NN) [26]. In [27] a two-objective evolu-

tionary algorithm which takes at the same time into account, both the localization accuracy and certain topological constraints induced by connectivity are considered during the evolutionary process, using metaheuristic approach, namely Simulated Annealing (SA), is proposed. An empirical study of the performance of several variants of the guiding functions and several metaheuristic are used to solve real Localization Distance (LD) problem presented in [23]. Each target node is localized under imprecise measurement of distances from three or more neighboring anchors/settled nodes. The methods proposed in this paper have following advantages:

1. There is better trade off between localization accuracy and fast convergence in highly noisy environments.
2. Energy efficiency of networks increased, due to minimum use of hardware (minimum number of anchor nodes)
3. Scalability for large scale deployment is possible.

3 Stochastic Algorithms (HPSO and BBO) for WSN Localization

Widespread acceptance of bio-inspired algorithms is credited to their correctness, and their fair computational load [23, 24, 26, 28, 29, 30, 31, 32, 33, 34, 35]. To get better and fast solution, an improved variant of the PSO, i.e., HPSO, and a recent optimization algorithm, i.e., BBO is applied for range-based distributive node localization in this paper. Ease of implementation and fast convergence are the qualities of *global best* PSO; however, it is likely to get trapped in local optima that leads to pre-matured convergence. The proposed HPSO and BBO algorithms provide matured convergence and better accuracy as compared to the PSO method proposed in [23, 24]. The following sections present an overview of HPSO and BBO.

3.1 Particle Swarm Optimization

The PSO method employs a number of practicable solutions within the search space, called a *Swarm of Particles* with random initial locations. The value of the objective function (which reflects error) corresponding to each particle location is evaluated. These particles move in the search space obeying rules inspired by bird flocking behavior [36, 37] to find new locations with better fitness. Each particle is induced to move towards the best position, the particle has come across so far (*pbest*) and the best position encountered by the entire swarm (*gbest*). To get an accurate solution, the whole swarm is subdivided into sub-swarms and the particle with the best fitness within the local swarm is termed as *lbest*. The *lbest* PSO model provides matured but slow convergence, whereas, in our proposed PSO variant named *HPSO*, the ith particle belonging to a sub-swarm feels collective attraction towards its past *pbest* location, P_i, the locally best location within the sub-swarm, P_l, and the overall best location P_g as explained below.

Consider a search space is d-dimensional and ith particle in the swarm can be represented as $X_i = [x_{i1}, x_{i2}, \ldots, x_{id}]$ and its velocity can be represented by another d-dimensional vector $V_i = [v_{i1}, v_{i2}, \ldots, v_{id}]$. Let the best position ever visited in the past by the ith particle be denoted by $P_i = [p_{i1}, p_{i2}, \ldots, p_{id}]$. Many a times, the whole swarm is subdivided into smaller groups and each group/sub-swarm has its own local best particle denoted $P_l = [p_{l1}, p_{l2}, \ldots, p_{ld}]$, and an overall best particle, denoted as $P_g = [p_{g1}, p_{g2}, \ldots, p_{gd}]$, where subscripts l and g are particle indices. The particle iterates in every unit time according to (1) and (2):

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) + c_3 r_3 (p_{ld} - x_{id}) \tag{1}$$

$$x_{id} = x_{id} + v_{id} \tag{2}$$

(The eq. (2) is dimensionally valid in unit time) The parameters w, c_1, c_2 and c_3 termed as inertia weight, cognitive, social and neighborhood learning parameters, respectively, and have a critical role in the convergence characteristics of HPSO. The particles randomize the attraction with uniform random numbers r_1, r_2, and r_3 in the range [0, 1]. The weight factor w should be neither too large, (which results in an early convergence), nor too small, (which, on the contrary, slows down the convergence process). A value of $w = 0.7$ and $c_1 = c_2 = c_3 = 1.494$ were recommended for fast convergence by Eberhart and Shi after experimental tests in [34].

3.2 Biogeography Based Optimization

Biogeography is the study of migration, speciation, and extinction of species, that has often been considered as a process which applies equilibrium in the number of species in habitats [35, 38, 39, 40]. A habitat is an ecological space that is inhabited by plant or animal species and which is geographically isolated from other habitats. Each habitat is classified by Habitat Suitability Index (HSI) that is termed as fitness in other EAs. The features that characterize the habitat are called Suitability Index Variables (SIVs). Habitats with high HSI have a large population, high emigration rate, μ, simply by virtue of the large number of species that migrate to other habitats. The immigration rate, λ, is low for these habitats as these are already saturated with species. Habitats with low HSI have high immigration, λ, and low emigration, μ, because of sparse population. The suitability index of habitats with low HSI is likely to improve with the influx of species from other habitats as it is a function of its biological diversity. However, if HSI does not increase and remains low, species in that habitat go extinct, and this leads to additional immigration. For the purpose of simplicity, it is safe to accept a linear relationship between a habitat HSI and its immigration and emigration rates and, further the rates are same for all habitats. The immigration and emigration rates depend upon the number of species in the habitats. The values of immigration and emigration rates are respectively given as:

$$\lambda = I\left(1 - \frac{k}{n}\right) \tag{3}$$

and

$$\mu = \frac{E}{n} \tag{4}$$

where I is the maximum possible immigration rate; E is the maximum possible emigration rate (I is not necessarily equal to E); k is the number of species of the kth individual and n is the number of species and S_{max} is maximum number of species in a habitat. For a pseudo-code of the algorithm, one may refer to [35].

4 HPSO and BBO Based Node Localization

The main objective in WSN localization is to find out the coordinates of maximum number of target nodes by using M anchor nodes with range-based distributed technique. To estimate coordinates of N target nodes, the process followed is as below:

1. N Target nodes are randomly deployed over middle layer and bottom layer and M anchor nodes are randomly deployed at the top layer. Each target node and anchor node has their transmission range R. Anchor nodes compute their location awareness and transmit their coordinates. The nodes, which get settled at the end of iteration, serve as pseudo anchors or as reference nodes during the next iteration and behave like anchors.
2. The node that falls within transmission range of four or more anchors is considered as localizable node.
3. Each localizable node measures its distance from each of its neighboring anchors. The distance measurements are corrupted with gaussian noise, n_i, due to environment consideration and due to DOI. A node estimates its distance from ith anchor as $\hat{d_i} = [d_i + n_i]$ where d_i is actual distance given by (5)

$$d_i = \sqrt{(x_i - x_{ai})^2 + (y_i - y_{ai})^2 + (z_i - z_{ai})^2} \tag{5}$$

   whereas (x_i, y_i, z_i) is the location of the target node and (x_{ai}, y_{ai}, z_{ai}) is the location of the ith anchor node in the neighborhood. The Gaussian assumption for range measurement is valid on practical experimental result of [41] and, therefore, localization results depend on the noise variance, σ_d^2, too.
4. HPSO and BBO-based two separate case studies are conducted, where each localizable target node runs HPSO and BBO algorithms seperately to localize itself. Both HPSO and BBO find the coordinate (x, y, z) that minimize the objective function that represents the error defined in (6).

$$f(x,y,z) = \frac{1}{M}\sum_{i=1}^{M}\left(\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} - \hat{d_i}\right) \tag{6}$$

where M $\geq$ 4 (3D location of a node needs minimum 4 anchors) is the number of anchors within transmission range, R, of the target node.

5. HPSO and BBO evolve the optimal location of target nodes, i.e., (x_i, y_i, z_i) by minimizing the error function (6).
6. After coordinates of all localizable nodes (say, N_L) are determined, the total localization error is computed as the mean of square of distances of computed node coordinates (x_i, y_i, z_i) and the actual node coordinates (X_i, Y_i, Z_i), for $i = 1, 2, \ldots, N_L$ determined for both cases of HPSO and BBO, as in (7):

$$E_l = \frac{1}{N_L} \sum_{i=1}^{L} (\sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2 + (z_i - Z_i)^2}) \tag{7}$$

7. Steps 2 to 6 are repeated until all target nodes get localized or no more nodes can be localized. The performance of the localization algorithm is based on E_l and N_{NL}, where $N_{NL} = [N - N_L]$ is the number of nodes that could not be localized. Lesser the value of N_{NL} and E_l, the better the performance is.

As the iterations progress, the number of localized nodes increases. This increases the number of references available for already localized nodes. A node that localizes using just four references in an iteration k may have more references in iteration $k+1$. This decreases the probability of the flip ambiguity. On the other hand, if a node has more references in iteration $k+1$ than in iteration k, the time required for localization increases, to reduce the time and energy consumption we considered the nearest four anchor nodes/ pseudo nodes to localize each of the unlocalized nodes. It has been observed from implementations of the above proposed algorithms that the maximum number of anchor nodes can be safely restricted to eight, with a view to minimize the hardware cost, increase the network scalability and increase energy efficiency of the network.

5 Simulation Results and Discussion

The WSN localization simulations are conducted using HPSO and BBO in MATLAB environment. 20 target nodes are deployed over middle layer and 20 target nodes are deployed over bottom layer (thus 40 target nodes) and 10 anchor nodes are randomly deployed at top layer of sensor field of 10×10 l units. Each anchor has a transmission range of $R = 4$ units. Other strategic settings are specific to HPSO and BBO algorithms as discussed below:

5.1 HPSO based node localization

In the proposed framework, each target node that can be localized runs HPSO algorithm to localize itself. HPSO parameters for node localization are fixed as:

1. Population size = 20
2. Max iteration = 100
3. Noise variance (σ_d^2) = 0.02, 0.06 and 0.08 (for three sets of simulation experiments).
4. DOI = 0.01
5. Distance between middle and each of top and bottom layers = 2, 2.5 and 3m (for three sets of simulation experiments).

To localize each node, HPSO runs thirty tials (each trials consisting of 100 iterations) with Gaussian noise. Average of total localization errors defined in (6) for all 30 trials are computed.

5.2 BBO based node localization

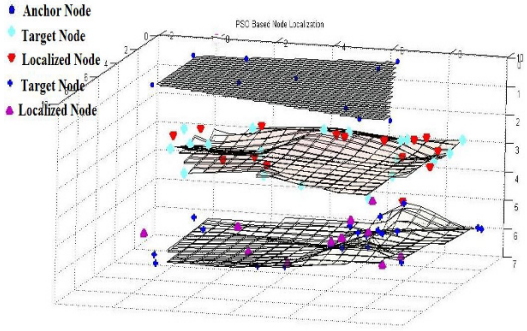
For each target node that can be localized, a BBO algorithm is run. BBO strategy parameters for node localization are taken as:

1. Population size = 20
2. Max iteration = 100
3. Probability of mutation = 0.05
4. Noise variance (σ_d^2) = 0.02, 0.06 and 0.08 (for three sets of simulation experiments).
5. DOI = 0.01
6. Distance between middle and each of top and bottom layers = 2, 2.5 and 3m (for three sets of simulation experiments).

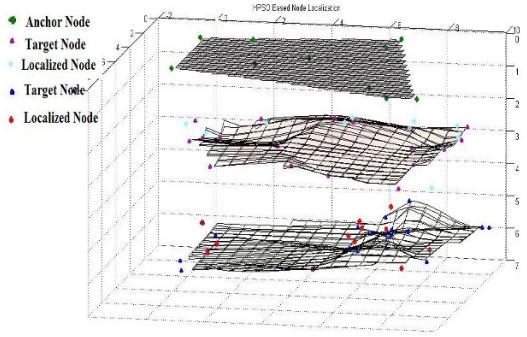
Thirty trials (each trials consisting of 100 iterations) experiment of BBO based localization is conducted with Gaussian noise. Average of total localization error defined as fitness function in (6) for all 30 trials is computed and minimized using BBO algorithm.

Both the proposed algorithms are stochastic; so, one can't expect the same solution in all trials even with identical deployment. This is the reason why the results of 30 trial runs are averaged. The initial deployment is random, so, the number of localizable nodes in each iteration is not expected to be the same, which makes the total computing time variable.

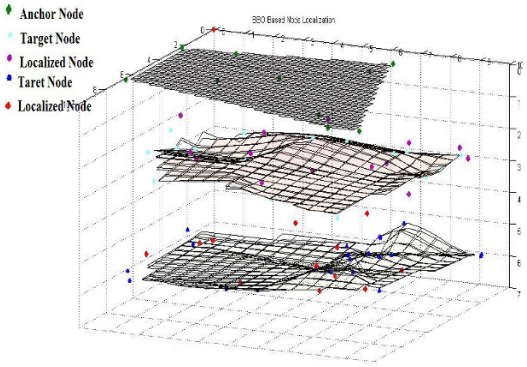
The actual nodes, anchors locations and coordinates estimated by PSO (implementation parameters are same as HPSO, except the division of swarm into sub swarm, i.e., *lbest* parameter), HPSO and BBO in a trial run are shown in Fig. 1.3. The distance between actual nodes and estimated nodes is shown in Fig. 2. It has been observed that performance of both the algorithms depends upon the gaussian noise level; lower the E_L more the localized nodes. It can be observed that PSO requires less memory and gives fast convergence but yields less accuracy. Proposed HPSO algorithm gives better accuracy and fast convergence. BBO gives better accuracy than HPSO, however, convergence is significantly slower than for HPSO. A choice between HPSO and BBO is dependent upon the trade off between accuracy



(a) PSO



(b) HPSO



(c) BBO

Fig. 1 Node localizations with different stochastic algorithms

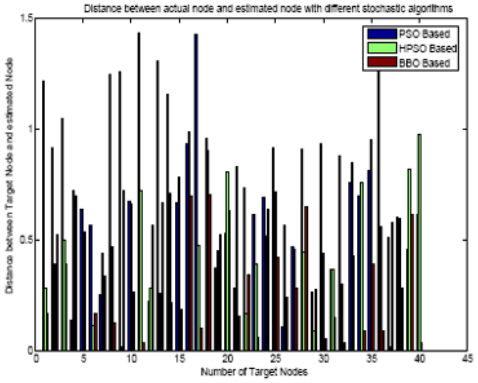


Fig. 2 Distance between actual Node and estimated Node with different stochastic algorithms

and fast convergence. Each point in the simulation results after 30 repetitions of experiments performed on 30 independent configurations.

Table 1 Simulation results of 30 trial runs for comparison of PSO, HPSO and BBO based WSN node location

	$\sigma_d^2 = 0.08$			$\sigma_d^2 = 0.06$			$\sigma_d^2 = 0.02$		
SAs	Mean of No. of un-localized node (N_{NL})	Mean Error (E_L)	Total Computing Time (sec)	Mean of No. of un-localized node (N_{NL})	Mean Error (E_L)	Total Computing Time(sec)	Mean of No. of un-localized node (N_{NL})	Mean Error (E_L)	Total Computing Time(sec)
PSO	1.0924	0. 03078	75.006	1.027	0.02018	56.721	0.862	0.00951	32.70
HPSO	1.0493	0.01594	**61.3870**	0.749	0.00911	**39.692**	0.437	0. 00219	**27.06**
BBO	**1.0379**	**0.01483**	80.8421	**0.482**	**0.00353**	73.154	**0.371**	**0.00109**	49.38

The Gaussian noise is a crucial parameter for distance measurements, which influences the localization accuracy. It is observed that accuracy decreases as noise increases (mean error E_L increases as σ_d increases). The dependence of mean error (E_L) on Gaussian noise variance (σ_d) is shown in Fig. 3. The localization algorithms discussed are iterative and the number of localized nodes increases with each iteration. The anchor node density improves the performance of localizability of the target node that can be observed in Fig. 4. It can be seen in Fig. 4 that with 4 anchor nodes (minimum 4 anchor nodes are required to get 3D coordinates of the target node), the percentage of localized target node increases sharply. It can also be noticed that as the number of anchor nodes increases, the percentage of localized nodes increases as shown in Fig. 4. The implemented algorithms provide clear insight into the cost trade-off between a WSN with all target nodes and anchor nodes equipped with GPS devices. As shown in Fig. 4, with 10 anchor nodes almost 100 percent

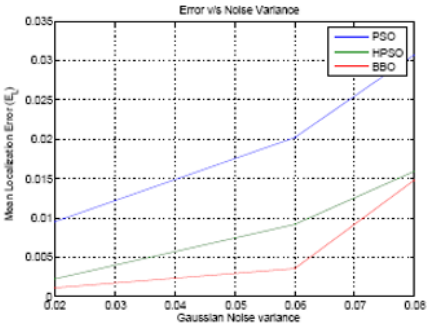


Fig. 3 Error Vs Noise variance (standard deviation)

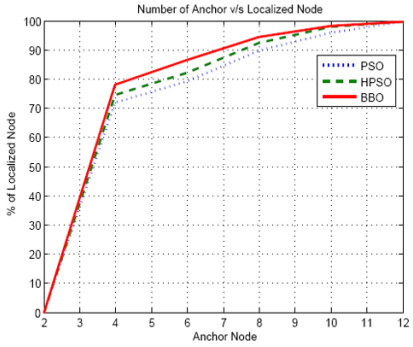


Fig. 4 Number of anchors Vs localized nodes

target nodes are localized. Simulation result (refer to Table 1) of Target Nodes = 40, transmission range = 4 unit, Gaussian noise = 0.02, field size = 10x10 units show that 10 GPS enabled anchor nodes can localize all the 40 target nodes. This results in about 75-80 percent saving in cost of GPS hardware. It is noticed that beyond a particular threshold, the effect of anchors becomes negligible as shown in Fig. 4.

5.3 Effect of Distance between Layers

The distance between the layers is varied keeping the radio range constant. The inter-layer spacing affects the success rate (in percentage) as shown in Table 2. It has been noticed from the Table 2 that HPSO and BBO based localization algorithms have better success rate as compared to PSO based localization algorithm.

Table 2 Effect on success rate due to distance between layers

Max. Radio Range	Distance Between Layers	Success Rate		
		PSO	HPSO	BBO
4	2.0	90.1	92.4	94.6
4	2.5	86.2	88.7	91.4
4	3.0	84.7	87.1	90.2
4	3.5	82.9	85.8	88.1
4	4.5	81.3	83.4	86.2

6 Conclusion and Future Scope

Stochastic range-based distributed node localization algorithms namely HPSO and BBO have been presented. The proposed algorithms have better accuracy and fast convergence in highly noisy (DOI = 0.01)environment. The HPSO-based localization algorithm determines the accurate coordinates quickly, whereas BBO-based localization algorithm finds the coordinate of the nodes more accurately. The choice between the two algorithms depends upon the trade-off between accuracy and the fast convergence. The proposed algorithms also reduce the number of transmissions to the base station (nearest 4 anchor nodes/pseudo nodes are selected to localize the target nodes), which helps the node to conserve more energy, so, the node can perform for longer periods. This paper, through extensive simulations, emphasizes that as iterations progress, more nodes get settled and require few anchors to find the coordinates of the target nodes. The proposed application is beneficial for the rescue operations to find out the accurate location of target nodes in highly noisy environment. Further, the proposed algorithms may be implemented for range-free localization and a comparison can be made for energy awareness. A hybrid stochastic algorithm may be proposed to achieve both more accuracy and faster convergence.

References

1. D. Estrin, D. Culler, K. Pister, G. Sukhatme, "connecting the physical world with pervasive networks", Pervasive Computing, IEEE 1 (2002) 59–69.
2. G. J. Pottie, W. J. Kaiser, "wireless integrated network sensors", Communications of the ACM 43 (5) (2000) 51–58.
3. I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications magazine 40 (8) (2002) 102–114.
4. L. Doherty, et al., Convex position estimation in wireless sensor networks, in: INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings., Vol. 3, 2001, pp. 1655–1663.
5. A. Pal, Localization algorithms in wireless sensor networks: Current approaches and future challenges, Network Protocols and Algorithms 2 (1) (2010) 45–73.
6. S. Alam, Z. Haas, Topology control and network lifetime in three-dimensional wireless sensor networks, Arxiv preprint cs/0609047.
7. I. Akyildiz, D. Pompili, T. Melodia, Underwater acoustic sensor networks: research challenges, Ad hoc networks 3 (3) (2005) 257–279.

8. I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer networks 38 (4) (2002) 393–422.
9. J. Wang, R. K. Ghosh, S. K. Das, A survey on sensor localization, Journal of Control Theory and Applications 8 (1) (2010) 2–11.
10. A. Boukerche, H. Oliveira, E. Nakamura, A. Loureiro, Localization systems for wireless sensor networks, wireless Communications, IEEE 14 (6) (2007) 6–12.
11. J. Hightower, G. Borriello, Location systems for ubiquitous computing, Computer 34 (8) (2001) 57–66.
12. D. Niculescu, B. Nath, Ad hoc positioning system (aps), in: Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE, Vol. 5, IEEE, 2001, pp. 2926–2931.
13. N. Bulusu, D. Estrin, L. Girod, J. Heidemann, Scalable coordination for wireless sensor networks: self-configuring localization systems, in: International Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, UK, 2001.
14. A. Savvides, H. Park, M. Srivastava, The bits and flops of the n-hop multilateration primitive for node localization problems, in: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, ACM, 2002, pp. 112–121.
15. M. Di Rocco, F. Pascucci, Sensor network localisation using distributed extended kalman filter, in: IEEE/ASME international conference on Advanced intelligent mechatronics, 2007, IEEE, 2007, pp. 1–6.
16. R. Kalman, A new approach to linear filtering and prediction problems, Journal of basic Engineering 82 (Series D) (1960) 35–45.
17. P. Biswas, T. Lian, T. Wang, Y. Ye, Semidefinite programming based algorithms for sensor network localization, ACM Transactions on Sensor Networks (TOSN) 2 (2) (2006) 188–220.
18. Y. Shang, W. Ruml, Improved mds-based localization, in: Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2004, Vol. 4, IEEE, 2004, pp. 2640–2651.
19. S. Yun, J. Lee, W. Chung, E. Kim, S. Kim, A soft computing approach to localization in wireless sensor networks, Expert Systems with Applications 36 (4) (2009) 7552–7561.
20. Q. Zhang, J. Wang, C. Jin, Q. Zeng, Localization algorithm for wireless sensor network based on genetic simulated annealing algorithm, in: 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08., IEEE, 2008, pp. 1–5.
21. Q. Zhang, J. Huang, J. Wang, C. Jin, J. Ye, W. Zhang, A new centralized localization algorithm for wireless sensor network, in: Third International Conference on Communications and Networking in China, 2008. ChinaCom 2008., IEEE, 2008, pp. 625–629.
22. Y. Li, J. Xing, Q. Yang, H. Shi, Localization research based on improved simulated annealing algorithm in wsn, in: 5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09, IEEE, 2009, pp. 1–4.
23. R. Kulkarni, G. Venayagamoorthy, M. Cheng, Bio-inspired node localization in wireless sensor networks, in: IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009., pp. 205–210.
24. A. Gopakumar, L. Jacob, Localization in wireless sensor networks using particle swarm optimization, in: IET International Conference on Wireless, Mobile and Multimedia Networks, 2008., IET, 2008, pp. 227–230.
25. R. Stoleru, J. A. Stankovic, Probability grid: A location estimation scheme for wireless sensor networks, in: First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004., IEEE, 2004, pp. 430–438.
26. P. Chuang, C. Wu, An effective pso-based node localization scheme for wireless sensor networks, in: Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2008. PDCAT 2008, IEEE, 2008, pp. 187–194.
27. G. Mao, B. Fidan, B. Anderson, Wireless sensor network localization techniques, Computer Networks 51 (10) (2007) 2529–2553.
28. Y. del Valle, G. Venayagamoorthy, S. Mohagheghi, J. Hernandez, R. Harley, Particle swarm optimization: basic concepts, variants and applications in power systems, IEEE Transactions on Evolutionary Computation 12 (2) (2008) 171–195.

29. R. Schaefer, H. Telega, Foundations of global genetic optimization, Springer Verlag, 2007.
30. K. Price, R. Storn, J. Lampinen, Differential evolution: a practical approach to global optimization, Springer-Verlag New York Inc, 2005.
31. Y. Chen, W. Peng, M. Jian, Particle swarm optimization with recombination and dynamic linkage discovery, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 37 (6) (2007) 1460–1470.
32. D. Kim, A. Abraham, J. Cho, A hybrid genetic algorithm and bacterial foraging approach for global optimization, Information Sciences 177 (18) (2007) 3918–3937.
33. J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings., IEEE International Conference on Neural Networks, 1995, Vol. 4, 1995, pp. 1942–1948.
34. Y. Shi, et al., Particle swarm optimization: developments, applications and resources, in: Proceedings of the 2001 Congress on Evolutionary Computation, 2001., Vol. 1, IEEE, 2001, pp. 81–86.
35. D. Simon, Biogeography-based optimization, Evolutionary Computation, IEEE Transactions on 12 (6) (2008) 702–713.
36. M. Noel, P. Joshi, T. Jannett, Improved maximum likelihood estimation of target position in wireless sensor networks using particle swarm optimization, in: Third IEEE International Conference on Information Technology: New Generations, 2006. ITNG 2006., 2006, pp. 274–279.
37. Y. Chen, V. Dubey, Ultrawideband source localization using a particle-swarm-optimized capon estimator, in: IEEE International Conference on Communications, 2005., Vol. 4, 2005, pp. 2825–2829.
38. A. Wallace, The Geographical Distribution of Animals, MA: Adamant Media Corporation, 2005.
39. C. Darwin, The Origin of Species, New York: Gramercy, 1859.
40. R. MacArthur, E. Wilson, The theory of island biogeography, Princeton Univ Press, 1967.
41. N. Patwari, J. Ash, S. Kyperountas, A. Hero III, R. Moses, N. Correal, Locating the nodes: cooperative localization in wireless sensor networks, Signal Processing Magazine, IEEE 22 (4) (2005) 54–69.

An Evaluation of Classification Algorithms Using Mc Nemar's Test

Betul Bostanci and Erkan Bostanci

Abstract Five classification algorithms namely J48, Naive Bayes, Multilayer Perceptron, IBK and Bayes Net are evaluated using Mc Nemar's test over datasets including both nominal and numeric attributes. It was found that Multilayer Perceptron performed better than the two other classification methods for both nominal and numerical datasets. Furthermore, it was observed that the results of our evaluation concur with Kappa statistic and Root Mean Squared Error, two well-known metrics used for evaluating machine learning algorithms.

Key words: Classifier Evaluation, Classification algorithms, Mc Nemar's test

1 INTRODUCTION

Evaluating the performance of machine learning methods is as crucial as the algorithm itself since this identifies the strengths and weaknesses of each learning algorithm. This paper investigates the usage of Mc Nemar's test as an evaluation method for machine learning methods.

Mc Nemar's test has been used in different studies in previous research. Dietterich [1] examined 5 different statistical tests including Mc Nemar's test to identify how these tests differ in assessing the performances of classification algorithms. A similar evaluation was performed on a large database by Bouckaert [2]. Demsar [3] has evaluated decision tree, naive bayes and k-nearest neighbours methods

Betul Bostanci
School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK
e-mail: bbosta@essex.ac.uk

Erkan Bostanci
School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK
e-mail: gebost@essex.ac.uk

J. C. Bansal et al. (eds.), *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, Advances in Intelligent Systems and Computing 201, DOI: 10.1007/978-81-322-1038-2_2,

using other non-parametric tests including ANOVA (ANalysis Of VAriance) [4] and Friedman test [5, 6].

Other studies have evaluated classifiers using this test over a large set but our method differs in that we use a different criterion that compares how the individual instances are classified and how this is reflected in the whole dataset.

Five different machine learning methods namely J48 (Decision Tree), Naive Bayes [7], Multilayer Perceptron [7] IBK [8] and Bayes Net [9] were used in the experiments. WEKA [10] was used to obtain the classification results of these algorithms. These classification methods are used to classify samples from different datasets. Later, the classification results are analyzed using a non-parametric test in order to identify how a pair of learning methods differ from each other and which of the two performs better.

The rest of the paper is structured as follows: Section 2 presents the nominal and numeric datasets used in the experiments. Section 3 introduces Mc Nemar's test which is the main evaluation method proposed in this study followed by Section 4 where the experimental design is presented. Section 5 presents Mc Nemar's test results and compares them with two conventional evaluation criteria. Finally, the paper is drawn to a conclusion in Section 6.

2 DATASETS

In order to perform a fair evaluation, a relatively large number of datasets obtained from UCI Machine Learning Repository [11] are used. The datasets are selected from the ones including nominal (Table 1) and numeric data (Table 2).

Table 1 Nominal Datasets

Dataset	Number of Instances	Number of Attributes	Number of Classes
Car	1728	7	4
Nursery	12960	9	5
Tic-Tac-Toe	958	10	2
Zoo	101	18	7

Table 2 Numeric Datasets

Dataset	Number of Instances	Number of Attributes	Number of Classes
Diabetes	768	9	2
Glass	214	10	7
Ionosphere	351	35	2
Iris	150	5	3
Segment-Challenge	1500	20	7
Waveform-5000	5000	41	3

3 Mc NEMAR'S TEST

Mc Nemar's test [12, 13] is a variant of χ^2 test and is a non-parametric test used to analyse matched pairs of data. According to Mc Nemar's test, two algorithms can have 4 possible outcomes arranged in a 2×2 contingency table [14] as shown in Table 3.

Table 3 Possible results of two algorithms [13]

	Algorithm A failed	Algorithm A succeeded
Algorithm B failed	N_{ff}	N_{sf}
Algorithm B succeeded	N_{fs}	N_{ss}

N_{ff} denotes the number of times (instances) when both algorithms failed and N_{ss} denotes success for both algorithms. These two cases do not give much information about the algorithms' performances as they do not indicate how their performances differ. However, the other two parameters (N_{fs} and N_{sf}) show cases where one of the algorithms failed and the other succeeded indicating the performance discrepancies.

In order to quantify these differences Mc Nemar's test employs z score (Equation 1).

$$z = \frac{(|N_{sf} - N_{fs}| - 1)}{\sqrt{N_{sf} + N_{fs}}} \tag{1}$$

z scores are interpreted as follows: When $z = 0$, the two algorithms are said to show similar performance. As this value diverges from 0 in positive direction, this indicates that their performance differs significantly. Furthermore, z scores can also be translated into confidence levels as shown in Table 4.

Table 4 Confidence levels corresponding to z scores for one-tailed and two-tailed predictions [13]

z score	One-tailed Prediction	Two-tailed Prediction
1.645	95%	90%
1.960	97.5%	95%
2.326	99%	98%
2.576	99.5%	99%

Following the table, it is worth mentioning that *One-tailed Prediction* is used to determine when one algorithm is better than the other where *Two-tailed Prediction* shows how much the two algorithms differ.

Mc Nemar's test is known to have a low *Type-I* error which occurs when an evaluation method detects a difference between two learning algorithms when there is no difference [1].

4 EVALUATION CRITERION

By adopting the Mc Nemar's test to evaluate classification algorithms, the following criterion is defined: An algorithm is regarded as "successful" if it can identify the class of an instance correctly. Conversely, it is regarded as "failed" when it performs an incorrect classification for an instance.

Using this criterion, the z scores are calculated using Mc Nemar's test for the five classification algorithms. All the algorithms were used with their default parameters as parameter tuning may favor one algorithm to produce better results.

The null hypothesis (H_0) for this experimental design suggests that different classifiers perform similarly whereas the alternative hypothesis (H_1) claims otherwise suggesting that at least one of the classifiers performs differently as shown in Equation 2.

$$\begin{aligned} &H_0 : C_1 = C_2 = C_3 = C_4 = C_5 \\ &H_1 : \exists C_i : C_i \neq C_j, (i,j) \in (1,2,3,4,5), i \neq j \end{aligned} \tag{2}$$

At the end of the experiment, the z scores will indicate whether we should accept H_0 and reject H_1 or vice versa. In order to calculate the z scores, the classification results of the three classifiers must be identified for each individual instance.

This operation is performed for all instances in the given datasets. In WEKA, there are two options to see whether an instance is correctly classified or not. The first option is the graphical one (shown in Figure 1 with the squares while crosses denote correct classifications). The second option to show the incorrect classifications is via the "Output predictions" option of the classifier which displays a "+" in the output next to the instance which has been incorrectly classified.

10-fold cross-validation is used in the evaluation which works as folllows: First the data is separated into 10 sets each having $n/10$ instances. Then, the training is performed using 9 of these sets and testing is performed on the remaining 1 set. This process is repeated 10 times to consider all of the subsets created and the final result for the accuracy is obtained by taking the average of these iterations.

The first option is quite useful to see the result graphically, however in order to calculate the number of correct and incorrect classifications by the classifiers, one needs to export these results into a spreadsheet (e.g. Excel). For this reason, the second method was used to calculate number of instances where the classifiers succeeded and failed. Using these figures, the z scores were calculated using Equation 1.

In order to decide which classifier performed better, N_{sf} and N_{fs} values for two classifiers are examined. For example, classifier A is said to perform better than classifier B if N_{sf} is larger than N_{fs} according to Table 3.

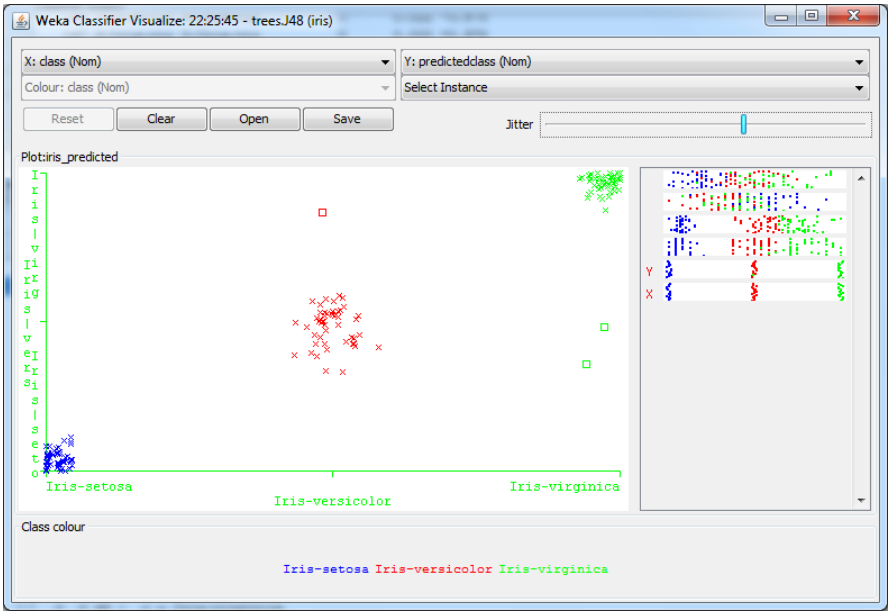


Fig. 1 Visualization of Classification Errors in WEKA

5 RESULTS

This section presents the results of the experiment. Results for the Mc Nemar's test will be given first and then these results will be compared with two other evaluation criteria namely Kappa statistic and Root Mean Squared Error (RMSE).

5.1 McNemar's Test Results

In Tables 5 and 6, the arrowheads ($\leftarrow, \uparrow$) denote which classifier performed better in the given datasets. z scores are given next to the arrowheads as a measure of how statistically significant the results are.

By looking at the Mc Nemar's test results for the nominal datasets (Table 5), one can deduce that Multilayer Perceptron has produced significantly better results than J48 and Naive Bayes classifiers (H_1 is accepted with a confidence level of more than 99.5%). J48 classifier performed better than the Naive Bayes for *Nursery* and *Tic-Tac-Toe* datasets. For the *Zoo* dataset, Naive Bayes performed better than J48 and equally to the Multilayer perceptron (H_0 is not rejected.). The performance differences between IBK and all other classifiers were not found to be statistically significant for the *Zoo* dataset but for the rest of the nominal datasets, there were significant differences. Bayes Net shows a poor performance overall except for the *Zoo*

Table 5 Mc Nemar's Test Results for Nominal Datasets

Car

	Naive Bayes	Multilayer Perceptron	IBK	Bayes Net
J48	0	↑ 10.63	↑ 1.62	← 6.93
Naive Bayes		↑ 10.63	↑ 1.62	← 6.93
Multilayer Perceptron			← 9.82	← 15.08
IBK				← 9.75

Nursery

	Naive Bayes	Multilayer Perceptron	IBK	Bayes Net
J48	← 24.66	↑ 17.32	↑ 34.89	← 24.64
Naive Bayes		↑ 34.89	↑ 31.68	0
Multilayer Perceptron			← 12.09	← 34.87
IBK				← 31.66

Tic-tac-toe

	Naive Bayes	Multilayer Perceptron	IBK	Bayes Net
J48	← 8.44	↑ 10.06	↑ 15.73	← 8.56
Naive Bayes		↑ 15.73	← 0.70	← 0.70
Multilayer Perceptron			← 15.90	← 15.90
IBK				0

Zoo

	Naive Bayes	Multilayer Perceptron	IBK	Bayes Net
J48	← 0.67	↑ 1.23	0	↑ 0.5
Naive Bayes		0	0	0
Multilayer Perceptron			0	0
IBK				0

dataset where it performed better than J48 although the result was not statistically significant.

Many differences in the classification performance are noticeable in the numeric dataset results (Table 6). For the *Glass* and *Segment-Challenge* datasets J48 has given better classification performance than Naive Bayes. For the former dataset, the Multilayer Perceptron performed equally with J48 and Naive Bayes produced a poorer classification result than these two. IBK and Bayes Net shows better performance over J48, Naive Bayes and Multilayer Perceptron, however there was no statistically significant performance difference between these two classification methods.

It is interesting to see that the first three (J48, Naive Bayes and Multilayer Perceptron) classifiers performed similarly on the *Ionosphere* dataset (H_0 is not rejected for all pairs.). Some differences can noticeable between these classifiers and Bayes Net however the results are not significant ($z = 0.75$ for Naive Bayes and Multilayer Perceptron) A similar result is also visible when the *Iris* dataset is consided since the values are quite close to zero. For the *Diabetes* dataset, Naive Bayes showed better performance over J48 yet the difference was not very significant for the latter (with a confidence level less than 95%) whereas Naive Bayes performs significantly better than J48 for the *Waveform-5000* dataset.

We can also see that the Multilayer Perceptron did not produce good results for the *Ionosphere* dataset where a relatively large number of attributes are present. This

Table 6 McNemar's Test Results for Numeric Datasets

Diabetes

	Naive Bayes	Multilayer Perceptron	IBK	Bayes Net
J48	↑ 1.61	↑ 0.96	← 0.56	↑ 0.26
Naive Bayes		← 0.56	← 3.40	← 1.29
Multilayer Perceptron			← 2.97	← 0.59
IBK				↑ 2.16

Glass

	Naive Bayes	Multilayer Perceptron	IBK	Bayes Net
J48	← 4.07	0	↑ 4.07	↑ 0.97
Naive Bayes		↑ 4.07	↑ 5.05	↑ 5.24
Multilayer Perceptron			↑ 0.95	↑ 0.97
IBK				0

Ionosphere

	Naive Bayes	Multilayer Perceptron	IBK	Bayes Net
J48	0	0	0	← 1.05
Naive Bayes		0	← 2.71	← 0.75
Multilayer Perceptron			← 2.71	← 0.75
IBK				↑ 1.37

Iris

	Naive Bayes	Multilayer Perceptron	IBK	Bayes Net
J48	0	↑ 0.41	↑ 0.5	← 1.51
Naive Bayes		↑ 0.5	0	← 1.51
Multilayer Perceptron			← 1.16	← 2.00
IBK				← 1.23

Segment

	Naive Bayes	Multilayer Perceptron	IBK	Bayes Net
J48	← 12.95	↑ 1.69	↑ 14.22	← 6.58
Naive Bayes		↑ 14.22	↑ 13.48	↑ 8.53
Multilayer Perceptron			← 0.86	← 7.92
IBK				← 7.05

Waveform-5000

	Naive Bayes	Multilayer Perceptron	IBK	Bayes Net
J48	↑ 6.90	↑ 12.40	← 1.84	↑ 6.71
Naive Bayes		↑ 5.78	← 8.77	← 0.54
Multilayer Perceptron			← 13.89	← 6.05
IBK				↑ 8.59

lower performance can be due to an underfitting problem as the default parameters were used without any parameter tuning.

5.2 Comparison with Other Evaluation Criteria

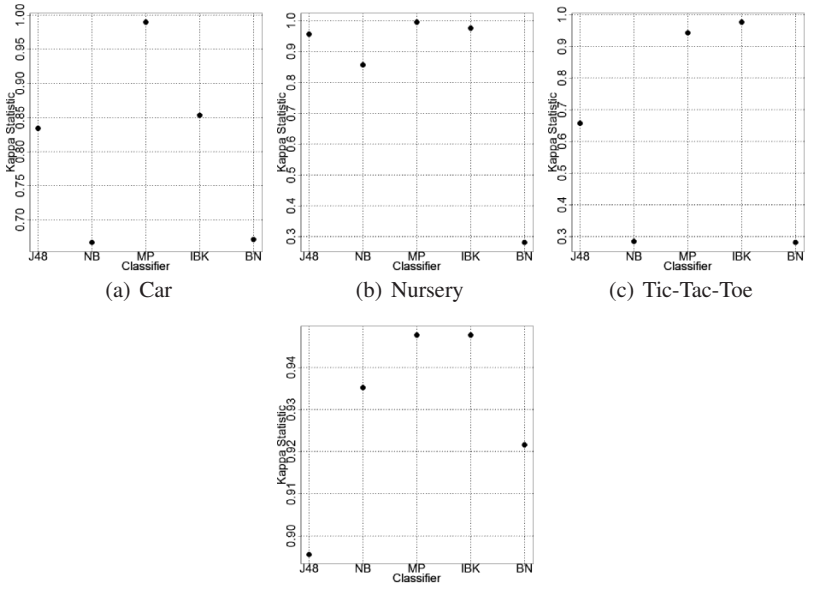
Mc Nemar's test result showed that there are significant discrepancies in the performances of the classifiers. Additional experiments were carried out to see how the

results for Mc Nemar's test conform with other evaluation criteria namely Kappa Statistic and Root Mean Squared Error.

5.2.1 Kappa Statistic

Kappa Statistic is a measure of the agreement between the predicted and the actual classifications in a dataset [15]. For this reason, we expect a higher value for a classifier which has more overlapping predictions and observations.

By looking at the nominal datasets in Figure 2, we see that Multilayer Perceptron has the highest value in 3 out of 4 datasets. J48 is better than Naive Bayes except for the *Zoo* dataset (Figure 2(d)). IBK shows good performance in all nominal datasets, although the poorest performance can be seen in the *Car* dataset.



(a) Car (b) Nursery (c) Tic-Tac-Toe

(d) Zoo

Fig. 2 Kappa Statistics for Nominal Datasets. NB: Naive Bayes, MP: Multilayer Perceptron, BN: Bayes Net

The ranking between J48 and Multilayer Perceptron changes significantly for the *Glass* and *Segment-Challenge* datasets for the numeric datasets in Figure 3. IBK has a good performance in these two datasets ($Kappa = 0.60$ and $Kappa = 0.95$ respectively). Naive Bayes produced good results only for the *Diabetes* dataset in this group. We can also say the Bayes Net achieves higher classification performance for the numeric datasets than the nominal datasets.

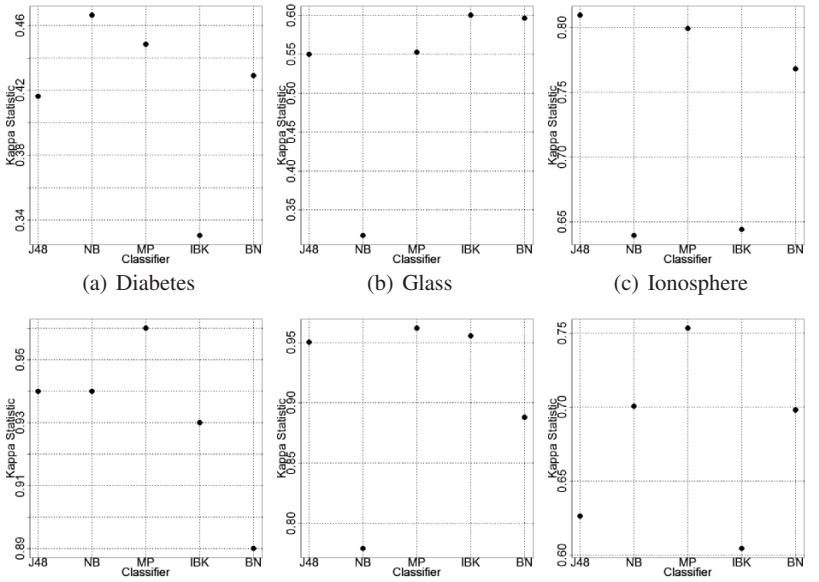


Fig. 3 Kappa Statistics for Numeric Datasets. NB: Naive Bayes, MP: Multilayer Perceptron, BN: Bayes Net

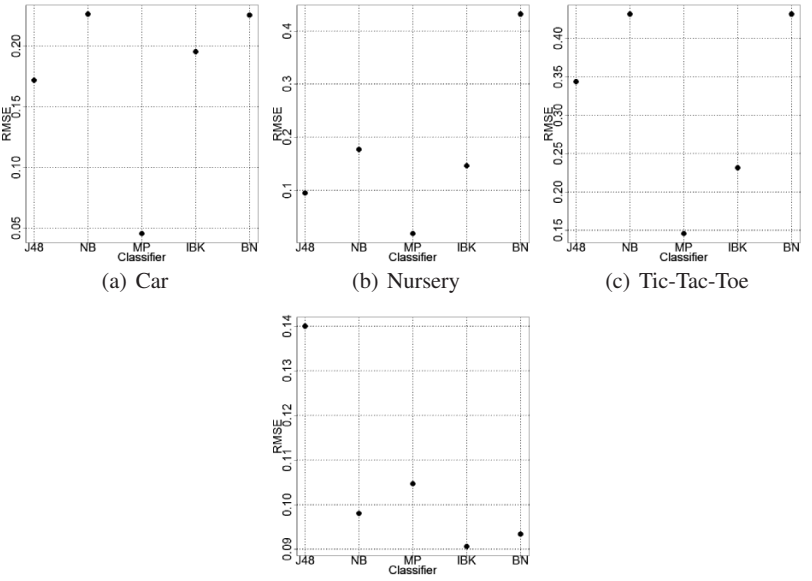
5.2.2 Root Mean Squared Error

Root Mean Squared Error (RMSE) [15] shows the error in the predicted and actual classes which the instances in a dataset belong to. RMSE should have lower values for more accurate classification results.

In nominal dataset results (Figure 4), Multilayer Perceptron had the lowest RMSE values for *Car*, *Nursery* and *Tic-Tac-Toe* datasets. J48 performed better than the Naive Bayes for the these datasets as well, while the ranking changed between them in the *Zoo* dataset shown in Figure 4(d). IBK shows the worst performance on the *Diabetes* and the best performance on the *Zoo* dataset. Bayes Net has poor performance in *Car* and *Tic-Tac-Toe* datasets.

A first look on the results in Figure 5 reveals that the Multilayer Perceptron results in lowest RMSE values for 4 out of 6 numeric datasets. Naive Bayes has a poor performance in *Glass*, *Ionosphere* and *Segment-Challenge* datasets. Naive Bayes showed the lowest performance in all datasets of the numeric dataset results except for the *Diabetes* dataset.

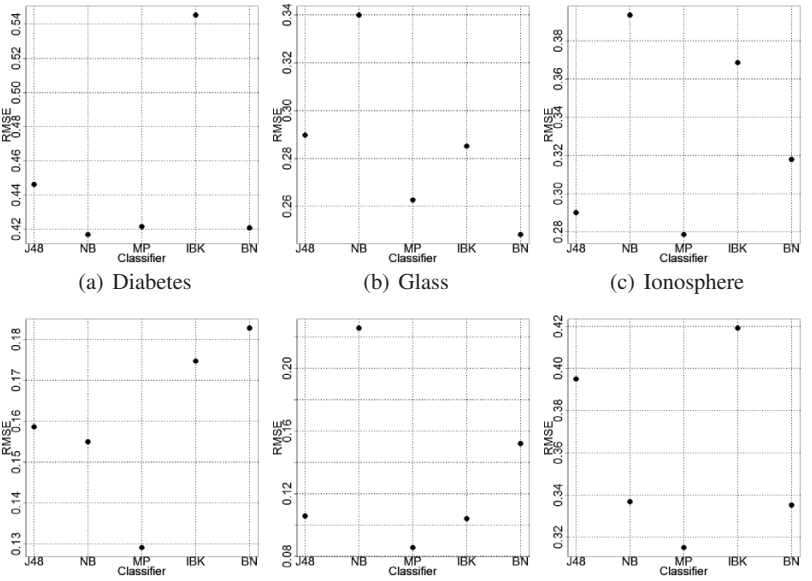
Table 7 show the mean results for all classifiers for the nominal and numeric datasets. Multilayer Perceptron has the highest values for Kappa statistic and lowest values for RMSE showing that the classification results using this classifier are accurate. IBK also shows a good classification performance for nominal and numeric data. Poor results are visible for Naive Bayes and Bayes Net.



(a) Car (b) Nursery (c) Tic-Tac-Toe

(d) Zoo

Fig. 4 RMSE for Nominal Datasets. NB: Naive Bayes, MP: Multilayer Perceptron, BN: Bayes Net



(a) Diabetes (b) Glass (c) Ionosphere

(d) Iris (e) Segment-Challenge (f) Waveform-5000

Fig. 5 RMSE for Numeric Datasets. NB: Naive Bayes, MP: Multilayer Perceptron, BN: Bayes Net

Table 7 Mean Kappa statistic and RMSE values for nominal and numeric datasets

	Nominal		Numeric	
	Kappa	RMSE	Kappa	RMSE
J48	0.82	0.19	0.72	0.36
Naive Bayes	0.69	0.23	0.64	0.31
Multilayer Perceptron	0.97	0.08	0.75	0.25
IBK	0.94	0.17	0.68	0.32
Bayes Net	0.54	0.30	0.71	0.28

From the three evaluation criteria (Mc Nemar's test, Kappa statistic and RMSE), Table 8 can be used to summarize the performance difference over the nominal and numeric datasets where + indicates performance grade. By looking at this summary table, it is evident that the Mc Nemar's test agrees with other evaluation criteria as an important result of the experiments. An exception can be seen for the comparison of Naive Bayes and J48 which is due to insignificance of the differences in Mc Nemar's test.

Table 8 Summary of the performances for nominal and numeric datasets

Mc Nemar's test					
	J48	Naive Bayes	Multilayer Perceptron	IBK	Bayes Net
Nominal	+++	++	+++++	++++	+
Numeric	+	++++	+++++	++	+++
Kappa statistic					
Nominal	+++	++	+++++	++++	+
Numeric	++++	+	+++++	++	+++
RMSE					
Nominal	+++	++	+++++	++++	+
Numeric	+	+++	+++++	++	++++

6 CONCLUSION

This study employed Mc Nemar's test in order to evaluate machine learning algorithms namely J48, Naive Bayes and Multilayer Perceptron, IBK and Bayes Net. By defining the success and failure criteria of Mc Nemar's test as correctly or incorrectly identifying the class of an instance in a dataset, the experiments presented the usage of a non-parametric test as a new method to evaluate classification algorithms.

The results showed that Multilayer Perceptron produced better results than the other methods for both nominal and numerical data. Bayes Net was placed in the lowest ranks for both types of data. Another interesting finding of the experiment is that the results of the Mc Nemar's test mostly conformed with Kappa statistic and RMSE as a justification of method's integrity.

The effect of parameter tuning is considered as future research. In this case, the classifiers will be tuned to achieve the optimal results and then the same tests can be applied to see whether there will be any changes in the rankings.

Acknowledgements The authors would like to thank Nadia Kanwal for discussions on Mc Nemar's test.

References

1. T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, pp. 1895–1923, 1998.
2. R. R. Bouckaert and E. Frank, "Evaluating the replicability of significance tests for comparing learning algorithms," in *Proceedings 8th Pacific-Asia Conference*, pp. 3–12, 2004.
3. J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, pp. 1–6, 2006.
4. D. A. Berry, "Logarithmic transformations in ANOVA," *Biometrics*, vol. 43, no. 2, pp. 439–456, 1987.
5. M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.
6. M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 1, no. 1, pp. 86–92, 1940.
7. P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson, 2006.
8. D. Aha and D. Kibler, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.
9. N. Friedman, D. Geiger, and Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, pp. 131–163, 1997.
10. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations*, vol. 11, 2009.
11. "UCI Machine Learning Repository." http://archive.ics.uci.edu/ml/, 2012.
12. Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, no. 12, pp. 153–157, 1947.
13. A. F. Clark and C. Clark, "Performance Characterization in Computer Vision: A Tutorial."
14. D. Liddell, "Practical tests of 2 2 contingency tables," *Journal of the Royal Statistical Society*, vol. 25, no. 4, pp. 295–304, 1976.
15. I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

Permitting features in *P* systems generating picture arrays

K.G. Subramanian, Ibrahim Venkat, Linqiang Pan and Atulya K. Nagar

Abstract In the area of membrane computing, the biologically inspired model known as *P* system has proved to be a rich framework for studying several types of problems. Picture array generation is one such problem for which different *P* systems have been constructed in the literature. Incorporating the feature of permitting symbols in the rules, array *P* systems are constructed here for generating picture languages consisting of picture arrays. The advantage of this approach is that there is a reduction in the number of membranes used in the construction, in comparison to the existing array *P* system model.

Key words: Membrane Computing, P System, Picture Arrays, Permitting Features

1 Introduction

The computability model known as *P* system, introduced by Păun [10] inspired by the structure and functioning of living cells, has turned out to be a versatile frame

K.G. Subramanian
School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia
e-mail: kgsmani1948@gmail.com

Ibrahim Venkat
School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia
e-mail: ibrahim@cs.usm.my

Linqiang Pan
Department of Control Science and Engineering,Huazhong University of Science and Technology, Wuhan 430074, Hubei, China
e-mail: lqpan@mail.hust.edu.cn

Atulya K. Nagar
Department of Mathematics and Computer Science,
Liverpool Hope University, Liverpool L16 9JD UK

J. C. Bansal et al. (eds.), *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, Advances in Intelligent Systems and Computing 201, DOI: 10.1007/978-81-322-1038-2_3,

work for studying computational problems in many different fields [11]. Picture grammar is one such area where different kinds of P systems for generating picture languages consisting of picture arrays, have been introduced and investigated. In [3], array P systems are introduced extending the string-objects P systems to array-objects P systems, thereby giving a link between picture grammars and P systems. Motivated by the study in [3], several variants of array P systems have been introduced (See for example, [1, 2, 16]).

On the other hand, regulating rewriting [6] in a grammar by permitting or forbidding the application of a rule based on the presence or absence of a set of symbols is known in formal language theory. Picture grammars that use this feature of permitting or forbidding symbols have also been introduced in [7, 8, 9].

Here we associate permitting symbols with rules in the regions of an array P system [3]. We call the resulting array P system as a permitting array P system and construct such a P system for generating picture languages consisting of picture arrays. The advantage of this approach is that the number of membranes used in the construction is reduced when compared to array P system [3]. The problem of generation of geometric figures such as squares, rectangles are of interest in the study of picture grammars (see for example [21, 20]). We consider in permitting array P system, the feature of $t-$communication in array P systems considered in [17] and this enables us to generate picture arrays representing solid squares with a reduced number of membranes in comparison to the $t-$communicating array P system given in [17] to generate such solid squares.

2 Preliminaries

For notions related to array grammars and array languages, we refer to [12, 13, 20], for notions on array P systems, we refer to [3, 16] and for notions of formal language theory to [14, 15].

Given an alphabet V, the set of all words over V, including the empty word λ, is denoted by V^* and $V^+ = V^* - \lambda$.

A picture array or simply an array in the two-dimensional plane consists of a finite number of labelled unit squares or pixels, with the labels belonging to an alphabet V and the unit squares not labelled with elements of V are considered to have the *blank symbol* $\# \notin V$. An array can be formally specified by listing the coordinates and the corresponding labels of the pixels. For example, for the T shaped array in Figure 1, this kind of specification is given as follows:

$$\{((0,5),a),\ ((1,5),a),\ ((2,5),a),\ ((3,5),a),\ ((4,5),a),\ ((5,5),a),$$
$$((6,5),a),\ ((7,5),a),\ ((8,5),a),\ ((9,5),a),\ ((10,5),a),$$

$$((5,0),a),\ ((5,1),a),\ ((5,2),a),\ ((5,3),a),\ ((5,4),a)\}$$

We note that only the relative positions of non-blank pixels in the array matter for us. The non-blank labels of the T shaped array are pictorially indicated in Figure 1.

$$\begin{array}{c} a\ a\ a\ a\ a\ a\ a\ a\ a\ a\ a \\ a \\ a \\ a \\ a \\ a \end{array}$$

Figure 1: T-shaped array with equal arms

We denote by V^{+2} the set of all two-dimensional non-empty finite arrays over V. The empty array is denoted by λ, and then the set of all arrays over V is $V^{*2} = V^{+2} \cup \{\lambda\}$. Any subset of V^{*2} is called an *array language*.

The array grammars [12, 13, 20] that involve array rewriting rules are extensions of string grammars [14, 15] to two dimensional picture arrays. We recall here the context-free and regular types of array rewriting grammars of the isometric variety which means that the rules preserve the geometric shape of the rewritten subarray.

An array grammar $G = (N, T, S, P, \#)$ where N, T are alphabets, $N \cap T = \phi$ and $S \in N$ is the start symbol. The elements of the finite set N are called nonterminals and those of T, terminals. P is a finite set of array rewriting rules of the form $r : \alpha \to \beta$ where α and β are arrays over $V \cup \#$ satisfying the following conditions:

1. the arrays α and β have identical shapes;
2. there is at least one element of N in α;
3. the symbols of T that occur in α are retained in their respective positions in β;
4. the application of the rule $r : \alpha \to \beta$ preserves the connectivity of the rewritten array.

For two arrays γ, δ over V and a rule r as above, we write $\gamma \Rightarrow_p \delta$ if δ can be obtained by replacing with β, a subarray of γ identical to α. The reflexive and transitive closure of the relation $\Rightarrow$ is denoted by $\Rightarrow^*$.

An array grammar is called:

1. *context-free*, if for all the rules $r : \alpha \to \beta$, the non$-\#$ symbols in α are not replaced by symbol $\#$ in β and for each rule $\alpha \to \beta$, α contains exactly one nonterminal with the remaining squares containing $\#$ and β contains no blank symbol $\#$;
2. *regular*, if the rules are of the following forms:

$$A\,\# \to a\,B,\ \#\,A \to B\,a,\ \begin{array}{c}\#\\A\end{array} \to \begin{array}{c}B\\a\end{array},\ \begin{array}{c}A\\\#\end{array} \to \begin{array}{c}a\\B\end{array},\ A \to B,\ A \to a,$$

where A, B are nonterminals and a is a terminal.

The array language generated by G is

$$L(G) = \{p \mid S \Rightarrow^* p \in T^{+2}\}.$$

Note that the start array is indeed $\{((0,0),S)\}$ and it is understood that this square labelled S is surrounded by #, denoting empty squares with no labels.

We denote by *AREG* and *ACF* respectively the families of array languages generated by array grammars with regular and context-free array rewriting rules.

We now recall the basic model of a rewriting array-objects P system introduced in [3].

An array P system (of degree $m \geq 1$) [3] is a construct

$$\Pi = (V, T, \#, \mu, F_1, \ldots, F_m, R_1, \ldots, R_m, i_o),$$

where: V is the alphabet of nonterminals and terminals, $T \subseteq V$ is the terminal alphabet, $\# \notin V$ is the blank symbol, μ is a membrane structure with m membranes labelled in a one-to-one way with $1, 2, \ldots, m$, $F_1, \ldots, F_m$ are finite sets of arrays over V associated with the m regions of μ, $R_1, \ldots, R_m$ are finite sets of array rewriting rules over V associated with the m regions of μ; the array-rewriting rules (context-free or regular) of the form $\mathscr{A} \rightarrow \mathscr{B}(tar)$ have attached targets *here, out, in* (in general, we omit mentioning *here*); finally, i_o is the label of an elementary membrane of μ which is the output membrane.

A computation in an array P system is defined in the same way as in a string rewriting P system [10] with the successful computations being the halting ones. Every array, from each region of the system, which can be rewritten by a rule associated with that region (membrane), should be rewritten; the rewriting is sequential at the level of arrays which means that one rule is applied ; the array obtained by rewriting is placed in the region indicated by the target associated with the rule used (*here* means that the array remains in the same region, *out* means that the array exits the current membrane and thus, if the rewriting was done in the skin membrane, then it exits the system; (arrays leaving the system are "lost" in the environment), and *in* means that the array is immediately sent to one of the directly lower membranes, nondeterministically chosen if several exist; if no internal membrane exists, then a rule with the target indication *in* cannot be used).

A computation is successful only if it stops and a configuration is reached where no rule can be applied to the existing arrays. The result of an halting computation consists of the arrays composed only of symbols from T placed in the output membrane with label i_o in the halting configuration. The set of all such arrays computed or generated by a system Π is denoted by $AL(\Pi)$. The families of all array lan-

guages $AL(\Pi)$ generated by systems Π as above, with at most m membranes, with CF and regular array-rewriting rules are respectively denoted by $EAP_m(CF)$ and $EAP_m(REG)$.

We illustrate with an example the computation in an array P system.

Example 1. An array P system generating T shaped arrays (Figure 1) over $\{a\}$ is as follows:

$$\Pi_1 = (\{A,B,C,B',C',a\},\{a\},\#,[_1[_2[_3\ [_4\ ]_4]_3]_2]_1,$$
$$\left\{\begin{matrix} AXB \\ C \end{matrix}\right\},\emptyset,\emptyset,\emptyset,R_1,R_2,R_3,R_4,4),$$

with

$$R_1 = \{\#\ A \to A\ a(in),\},$$
$$R_2 = \{B\ \# \to a\ B'(in),\ B'\ \to B\ (out)\},$$
$$R_3 = \{\begin{matrix} C \\ \# \end{matrix} \to \begin{matrix} a \\ C \end{matrix}(out), \begin{matrix} C \\ \# \end{matrix} \to \begin{matrix} a \\ C' \end{matrix}(in)\},$$
$$R_4 = \{A \to a, B' \to a, C' \to a\}.$$

A computation in Π_1 starts with the initial array $\begin{matrix} AXB \\ C \end{matrix}$ in region 1, with other regions having no initial array. An application of the rule $\#\ A \to A\ a(in)$ grows the horizontal arm one step on the left, after which the array is sent to region 2, due to the target indication *in* in the rule. In region 2, the rule $B\ \# \to a\ B'(in)$ alone can be applied which grows the horizontal arm one step on the right, after which the array is sent to region 3, due to the target indication *in* in the rule. If the rule $\begin{matrix} C \\ \# \end{matrix} \to \begin{matrix} a \\ C \end{matrix}(out)$ is applied in region 3, then the vertical arm grows one step down and the array is sent back to region 2 due to the target indication *out*. In region 2, the primed version of the nonterminal B is changed into B and the array is brought back to region 1 and the process can repeat. If in region 3, the rule applied is $\begin{matrix} C \\ \# \end{matrix} \to \begin{matrix} a \\ C' \end{matrix}(in)$ then the array is sent to the output region 4 wherein all the nonterminals are changed into the terminal a and the computation halts yielding a T shaped array over $\{a\}$ with equal arms which is collected in the language generated.

3 Permitting array P systems

We consider permitting CF (respy. regular)array rewriting rule, which is a context-free array rewriting rules with permitting symbols. We then define a permitting array P system that makes use of such permitting CF array rewriting rules in its regions.

If $\mathscr{B}$ is a subarray of $\mathscr{A}$, then we denote by $\mathscr{A} \setminus \mathscr{B}$, the array formed by the la-

belled squares of $\mathscr{A}$ that are not labelled squares of $\mathscr{B}$. We denote by $l(\mathscr{A})$, the set of all symbols in the labelled squares of the array $\mathscr{A}$. Note that in a CF array rewriting rule $\mathscr{A} \to \mathscr{B}$, $\mathscr{A}$ contains exactly one labelled square with a nonterminal symbol as label.

A permitting CF (respy. regular) array rewriting rule is of the form $(\mathscr{A} \to \mathscr{B}, per)$ where $\mathscr{A} \to \mathscr{B}$ is a context-free array rewriting rule and $per \subseteq N$ with N being the set of nonterminals of the array grammar. If $per = \phi$, then we omit mentioning it in the rule. For any two arrays $\mathscr{C}, \mathscr{D}$, and a permitting CF array rule $(\mathscr{A} \to \mathscr{B}, per)$, the array $\mathscr{D}$ is derived from $\mathscr{C}$ by replacing $\mathscr{A}$ in $\mathscr{C}$ by $\mathscr{B}$, provided $per \subseteq l(\mathscr{C} \setminus \mathscr{A})$.

We now introduce the notion of an array P system with permitting symbols associated with the rules in the regions.

A permitting array P system (of degree $m \geq 1$ ($pEAPS_m(CF)$), is a construct

$$\Pi = (V, T, \#, \mu, F_1, \ldots, F_m, R_1, \ldots, R_m, i_o),$$

where the components $V, T, \#, \mu, F_1, \ldots, F_m, i_o$ are as in an array P system and the rules in the sets $R_1, \ldots, R_m$ are permitting CF array rewriting rules of the form $(\mathscr{A} \to \mathscr{B}, per)$ where $\mathscr{A} \to \mathscr{B}$ is a context-free array rewriting rule and $per \subseteq V - T$ with $V - T$ being the set of nonterminals.

A computation in $pEAPS_m(CF)$) is also as in an array P system except that the application of a permitting CF array rewriting rule in any $R_i, 1 \leq i \leq m$, is regulated by the associated permitting symbols as described earlier in deriving an array from a given array. The successful computations are the halting ones. The result of a computation is the set of arrays collected in the output elementary membrane i_0 in the halting configuration.

The family of all array languages generated by systems Π as above, with at most m membranes, with permitting array rewriting rules of type regular or CF is respectively denoted by $pEAP_m(REG)$ or $pEAP_m(CF)$.

We illustrate computation in a permitting array P system with an example.

Example 2. A permitting array P system generating T shaped arrays (Figure 1) over $\{a\}$ is as follows:

$$\Pi_2 = (\{A, B, C, A', B', C', D, a\}, \{a\}, \#, [_1 [_2]_2]_1,$$

$$\left\{ \begin{matrix} AXB \\ C \end{matrix} \right\}, \emptyset, R_1, R_2, 2),$$

with

$$R_1 = \{(\# A \to A'\ a, \{B, C\}), (B\ \# \to a\ B', \{A', C\}), (\begin{matrix} C \\ \# \end{matrix} \to \begin{matrix} a \\ C' \end{matrix}, \{A', B'\}),$$

$$\begin{aligned}
&\big(A' \to A, \{B', C'\}\big), \big(B' \to B, \{A, C'\}\big), \big(C' \to C, \{A, B\}\big),\\
&\big(C' \to D(\{in\}), \{A, B\}\big)\}\\
R_2 = \{&A \to a, B \to a, D \to a\}
\end{aligned}$$

A computation in Π_2 starts with the initial array $\begin{matrix} A X B \\ C \end{matrix}$ in region 1, with region 2 having no initial array. The rule $\# \; A \to A' \; a$ alone is applicable as the permitting symbols B, C are present. The application of this rule grows one step on the left, the horizontal arm. The rule $B \; \# \to a \; B'$ can now be applied as the permitting symbols A', C are present in the array. The application of this rule grows one step on the right, the horizontal arm. Likewise the rule $\begin{matrix} C \\ \# \end{matrix} \to \begin{matrix} a \\ C' \end{matrix}$ can then be applied growing the vertical arm one step down. The primed versions of the nonterminals A, B, C are changed into their original versions A, B, C due to the application of the rules $A' \to A, B' \to B, C' \to C$ with the corresponding permitting symbols being present and the process can repeat. If the rule $C' \to D$ is applied instead of $C' \to C$, then the array is sent to the inner region 2, due to the target indication *in* In region 2, the nonterminals are changed into the terminal a with the computation coming to a halt yielding a T shaped array over $\{a\}$ with equal arms which is collected in the language generated.

Note that the array P systems in both the examples 1 and 2, generate the same picture language consisting of picture arrays representing T shaped figure. But the number of membranes used is only two in example 2 where permitting symbols are used in the rules whereas the number of membranes used in example 1 is four, where the feature of permitting symbols is absent. Although not entirely unexpected, this shows the power of permitting symbols in the rules in reducing the number of membranes.

Theorem 1.

1. $pEAP_m(\alpha) \subseteq pEAP_{m+1}(\alpha), \alpha \in \{REG, CF\}$
2. $pEAP_2(REG) - EAP_2(REG) \neq \emptyset$
3. $pEAP_2(REG) - AREG \neq \emptyset$

Proof. The statement 1 is immediate from the definition of the family $pEAP_m(\alpha), \alpha \in \{REG, CF\}$.

The statement 2 can be seen as follows: The picture language L consisting of T shaped arrays with equal arms is in the family $pEAP_2(REG)$, as seen in example 2 where a permitting array P system with two membranes generates L. But application of regular array rewriting rules just alternating between two membranes can not keep generating all three arms of equal length, namely the left horizontal arm, the right horizontal arm and the vertical arm of the T shaped array, together once the derivation reaches the 'junction' in the T shaped array. Hence without the feature of permitting symbols in the rules, any basic model array P system [3] with regular array rewriting rules will require at least three membranes to generate T.

The statement 3 is due to the fact that no regular array grammar by the nature of its rules can ensure that the arms are of equal length. In fact the regular array grammar rules cannot generate two arms together. □

The maximal mode or $t-$mode of derivation has been studied in a cooperating distributed grammar system [4] which was developed as a language-theoretic model of distributed complex systems. In [5] the $t-$ communication mode is brought into string rewriting P systems [10] thereby linking cooperating distributed string grammar systems and string-objects P systems. As a natural extension of the study in [5], Subramanian et al [17] incorporated this $t-$mode of communication into array P systems [3].

We now briefly recall a $t-$communicating array P system of type tin introduced in [17].

A $t-$communicating array P system of degree $m \geq 1$ and of type tin, $(tEAPS_m(tin, CF))$, is a construct

$$\Pi = (V, T, \#, \mu, F_1, \ldots, F_m, R_1, \ldots, R_m, i_o),$$

where the components $V, T, \#, \mu, F_1, \ldots, F_m, i_o$ are as in an array P system and the rules in the sets $R_1, \ldots, R_m$ are CF array rewriting rules of the form $\mathscr{A} \rightarrow \mathscr{B}$.

The computation is done in the usual way starting with the initial arrays (if any) in the regions. The arrays are communicated among the regions in the following manner: If an array-rewriting rule with target indication *out*, is applied to an array, then the resulting array is sent to its immediately direct upper region. If an array-rewriting rule has no target indication, then the array to which it is applied remains in the same region if it can be further rewritten there but if no rule can be applied to it in that region, then it is sent to the immediately direct inner region if one such region exists. In other words the $t-$mode or maximal derivation performed enforces the *in* target command. If the membrane is elementary, the rewritten array remains there. Note that the system does not have rules with target indication *in*. The result of a computation is the set of arrays over T collected in the output elementary membrane in the halting configuration.

The family of all array languages generated by a $t-$communicating array P system of type tin Π as above, with at most m membranes, with rules of type $\alpha \in \{REG, CF\}$ is denoted by $tEAP_m(tin, \alpha)$.

The problem of generation of picture arrays representing geometric figures such as solid squares over $\{a\}$ is a problem of interest in the area of picture grammars. It is known that the set S_s of all $n \times n$ $(n \geq 2)$ solid squares over a, can be generated [21] by a regular array grammar but the number of rules required is very large. In [17], a $t-$communicating array P system of type tin is given to generate it. Here we

endow the rules of a $t-$communicating array P system of type tin with permitting symbols and construct such a system to generate the set S_s of solid squares of $a's$. Such a solid square of $a's$ is shown in Figure 2.

$$\begin{array}{c} a\ a\ a\ a\ a\ a \\ a\ a\ a\ a\ a\ a \\ a\ a\ a\ a\ a\ a \\ a\ a\ a\ a\ a\ a \\ a\ a\ a\ a\ a\ a \\ a\ a\ a\ a\ a\ a \end{array}$$

Figure 2: A Solid square of $a's$

A permitting $t-$communicating array P system of type tin and of degree $m \geq 1$, is a $t-$communicating array P system of degree $m \geq 1$ and of type tin [17]except that the array-rewriting rules in the regions are of the form $(\mathscr{A} \to \mathscr{B}, per)$ where $\mathscr{A} \to \mathscr{B}$ is a context-free array rewriting rule and $per \subseteq V - T$ with $V - T$ being the set of nonterminals. In a computation in the system, application of the rules to arrays in the regions is done as in a permitting array P system and communication of arrays from one region to another is done as in the $t-$communicating array P system of type tin. As usual, a successful computation is a halting computation with the arrays collected in the output membrane constituting the language generated.
The family of picture array languages generated by permitting $t-$communicating array P systems of type tin is denoted by $ptEAP_m(tin,CF)$ or $ptEAP_m(tin,REG)$ depending on the array rewriting rules in the system being context-free or regular.

In [17], a $t-$communicating array P system of type tin is given to generate the set S_s of all $n \times n$ $(n \geq 2)$ solid squares over a and it is known that $S_s \in tEAP_4(tin,CF)$ [17] so that the number of membranes used is four and the number of rules in all the four membranes together is 12. Here we construct a $t-$communicating array P system of type tin with permitting symbols and regular array-rewriting rules to generate the set S_s of solid squares of $a's$, which requires only two membranes.

Theorem 2. $S_s \in ptEAP_2(tin,REG)$.

Proof. To prove the theorem, we construct a permitting $t-$communicating array P system $ptEAP_m(tin,CF)$

$$\Pi_3 = (\{A,B,A',B',C,D,C',D',X,Y,a\},\{a\},\#,[_1[_2]_2]_1,$$

$$\left\{\begin{array}{c} aA \\ BZ \end{array}\right\},\emptyset,\emptyset,R_1,R_2,2),$$

$$R_1 = \left\{(1)\ \left(A\,\# \to a\,A',\{B\}\right),(2)\ \left(\begin{array}{c} B \\ \# \end{array} \to \begin{array}{c} a \\ B' \end{array},\{A'\}\right),\ (3)\ \left(A' \to A,\{B'\}\right),\right.$$

$$(4)\ \left(B' \to B, \{A\}\right),\ (5)\ \left(B'\,\# \to a\,C, \{A'\}\right), (6)\ \left(C' \to C, \{D'\}\right),$$

$$(7)\ \left(D' \to D, \{C\}\right),\ (8)\ \left(\begin{matrix} A' \\ \# \end{matrix} \to \begin{matrix} a \\ D \end{matrix}, \{C\}\right), (9)\ \left(C\,\# \to a\,C', \{D\}\right),$$

$$(10)\ \left(\begin{matrix} D \\ \# \end{matrix} \to \begin{matrix} a \\ D' \end{matrix}, \{C'\}\right), (11)\ \left(C\,\# \to a\,X, \{D\}\right),\ (12)\ \left(\begin{matrix} D \\ \# \end{matrix} \to \begin{matrix} a \\ Y \end{matrix}, \{X\}\right),$$

$$(13)\ \left(Z\,\# \to a\,Z, \{X\}\right),\ (14)\ \left(\begin{matrix} Z \\ \# \end{matrix} \to \begin{matrix} a \\ Z \end{matrix}, \{X\}\right),$$

$$15)\ \left(\#\,Z \to Z\,a, \{X\}\right),\ (16)\ \left(\begin{matrix} \# \\ Z \end{matrix} \to \begin{matrix} Z \\ a \end{matrix}, \{X\}\right)\Bigg\}$$

$$R_2 = \{(17) X \to a,\ (18)\ D \to a,\ (19)\ Z \to a\}$$

The computation starts with the initial array in region 1. Rules (1) to (4) enable the top border and left border to grow equally, one step at a time until the rules(5) and (8) make the top border to turn down and the left border to turn right. The rules (6), (7), (9), (10) make the bottom border and right border to grow equally, one step at a time until the rule (11) is applied which makes in a correct computation the symbols D and E to meet. This makes the rule (12) not applicable which really is an indication of a correct computation. The remaining rules (13) to (16) enable filling up the interior in rows and columns, until no more rule is applicable. The application of the rules throughout the computation is guided by the permitting symbols. Due to type tin of the system, the array moves to region 2 where all the nonterminals are changed into the terminal a thus yielding a solid square of $a's$ in a halting computation. Note that any incorrect sequence of application of the rules will result in the symbol Y getting stuck in the array and thus not contributing anything to the language. □

Remark 1. We note that the $t-communicating$ array P system with context-free array rewriting rules generating the set of solid squares of $a's$ given in [17] involves four membranes whereas two membranes are enough when the system is endowed with the additional permitting feature, with regular array rewriting rules only.

4 Conclusion

We have considered here the features of permitting symbols in the rules and $t-$mode of communication in the regions of an array P system and examined the generative power of such a system. It is of interest to note that array P system with $t-$communication and permitting symbols and regular array rewriting rules generates solid squares of $a's$. It is possible to construct permitting $t-$communicating array P systems, as done for solid squares, to generate picture arrays representing other kinds of geometric figures such as hollow squares, solid and hollow rectangles and so on. Comparison with the array P systems considered in [19] can also be

made. Also the techniques used here can be applied to construct corresponding *P* systems for triangle-tiled pictures [18].

Acknowledgements The authors are grateful to the referees for their time spent and for their very useful comments. The first and the second authors gratefully acknowledge support from a FRGS grant No. 203/PKOMP/6711267 of the Ministry of Higher education, Malaysia.

References

1. Ceterchi, R., Gramatovici, R., Jonoska, N.: Tiling rectangular pictures with P systems. Lecture Notes in Comp. Sci. **2933**, 263-269 (2004).
2. Ceterchi, R., Gramatovici, R., Jonoska, N., Subramanian, K.G.:Tissue-like P Systems with Active Membranes for Picture Generation. Fundam. Inform. **56**, 311-328 (2003).
3. Ceterchi, R., Mutyam, M., Păun, Gh., Subramanian, K.G.: Array - rewriting P systems. Natural Computing **2**, 229-249 (2003).
4. Csuhaj-Varjú, E., Dassow, J., Kelemen, J., Păun, G.: Grammar Systems: A Grammatical Approach to Distribution and Cooperation. Gordon and Breach Science Publishers, Topics in Computer Mathematics 5, Yverdon 1994).
5. Csuhaj-Varjú, T., Vaszil, G., Păun, G.: Grammar systems versus membrane computing: The case of CD grammar systems. Fundamenta Informaticae **76**, 271-292 (2007).
6. Dassow, J., Păun, G.: Regulated Rewriting in Formal Language Theory. Springer-Verlag, Berlin (1989).
7. Ewert, S., van der Walt, A.: Random Context Picture Grammars. Publicationes Mathematicae Debrecen **54**, 763-786 (1999).
8. Ewert, S., van der Walt, A.: Generating Pictures using Random Permitting Context. Int. J. Pattern Recogn. Artificial Intell. **13** 339-355 (1999).
9. Ewert, S., van der Walt, A.: Generating Pictures using Random Forbidding Context. Int. J. Pattern Recogn. Artificial Intell. **12**, 939-950 (1998).
10. Păun, G.: Computing with membranes. Journal of Computer and System Sciences **61**, 108-143 (2000).
11. Păun, G., Rozenberg, G., Salomaa, A. (Eds.): The Oxford Handbook of Membrane Computing. Oxford University Press, Inc., New York, NY, USA (2010)
12. Rosenfeld, A.: Picture Languages - Formal Models for Picture Recognition. Academic Press, New York, (1979).
13. Rosenfeld, A., Siromoney, R.; Picture languages - a survey, Languages of design. **1**, 229 - 245 (1993).
14. Rozenberg, G., Salomaa, A. (Eds.): Handbook of Formal Languages. Vol. 1-3, Springer, Berlin (1997).
15. Salomaa, A.: Formal languages. Academic Press, London (1973).
16. Subramanian, K.G.: P systems and picture languages. Lecture Notes in Comp. Sci., **4664**, 99-109 (2007).
17. Subramanian, K.G., Ali, R.M., Nagar, A.K., Margenstern, M.: Array P systems and t-communication. Fundam. Inform. **91**, 145-159 (2009).
18. Subramanian, K.G., Geethalakshmi, M., Nagar, A.K., Lee, S.K.: Triangle-tiled picture languages, In "Progress in Combinatorial Image Analysis" (Eds. P. Wiederhold, R.P. Barneva), Research Publishing Services, 165-180 (2010).
19. Subramanian, K.G., Pan, L., Lee, S.K., Nagar, A.K.: A P system model with pure context-free rules for picture array generation. Math. Comp. Modelling **52** 1901-1909 (2010).
20. Wang, P.S.P. (ed.): Array grammars, Patterns and recognizers, Series in Computer Science, Vol. 18, World Scientific, (1989).

21. Yamamoto, Y., Morita, K., Sugata, K.: Context-sensitivity of two-dimensional regular array grammars, In “Array Grammars, Patterns and Recognizers” (P.S.-P. Wang, ed.), WSP Series in Computer Science, 18, World Scientific Publ., Singapore 17- 41 (1989).

An ACO framework for Single Track Railway Scheduling Problem

Raghavendra G. S. and Prasanna Kumar N

Abstract This work focus on application of ant algorithms to railway scheduling problem. The railway scheduling problem especially on a single track is considered to be NP hard problem with respect to number of conflicts in the schedule. The train scheduling is expected to satisfy several operational constraints, thus making the problem more complex. The ant algorithms have evolved as more suitable option to solve the NP hard problem. In this paper, we propose a mathematical model to schedule the trains that fits into ACO framework.The solution construction mechanism is inspired by orienteering problem. The proposed methodology has the capability to explore the complex search space and provides the optimal solution in reasonable amount of time. The proposed model is robust in nature and flexible enough to handle additional constraints without any modification to the model. The model assumes that set of trains will be scheduled in a zone, that covers several cities and they are optimized with respect to number of conflicts.

Keywords: Ant, Optimization, Railway, Schedules, Train.

1 Introduction

The train timetable generation is a tedious and time consuming task. Traditionally, timetable is generated manually by trial and error method based on experience and information. The advent of computer aided tools have helped the planner to come up with the effective timetable [1,2] and to access the effectiveness in terms of robustness in routing [3], revenue profitablity etc. The aim of the train scheduling problem is to come with the ideal timetable that satisfies several objectives. The objectives can be maximizing the number of passengers, minimizing the number of conflicts,

Raghavendra G. S.
BITS-Pilani K. K. Birla Goa Campus, Goa e-mail: gsr@bits-goa.ac.in

Prasanna Kumar N
BITS-Pilani K. K. Birla Goa Campus, Goa e-mail: prasannak@bits-goa.ac.in

J. C. Bansal et al. (eds.), *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, Advances in Intelligent Systems and Computing 201, DOI: 10.1007/978-81-322-1038-2_4,